ISSN : 2527-5836
e-ISSN : 2528-0074

Vol. 11 No. 1, January 2026

JISKa

Jurnal Informatika Sunan Kalijaga

Jurusan Teknik Informatika
Fakultas Sains dan Teknologi
UIN Sunan Kalijaga Yogyakarta

JISKa Editorial Team

January 2026 Edition

Reviewers

Agung Dewandaru, Institut Teknologi Bandung, Indonesia

Agus Mulyanto, UIN Sunan Kalijaga Yogyakarta, Indonesia

Ahmad Fathan Hidayatullah, Universitas Islam Indonesia Yogyakarta, Indonesia

Alam Rahmatulloh, Universitas Siliwangi Tasikmalaya, Indonesia

Anggi Rizky Windra Putri, Universitas Aisyiyah Yogyakarta, Indonesia

Ardiansyah Musa Efendi, Singapore Chipset Algorithm Design Lab, Huawei, Singapore

Bambang Sugiantoro, UIN Sunan Kalijaga Yogyakarta, Indonesia

Enny Itje Sela, Universitas Teknologi Yogyakarta, Indonesia

Ganjar Alfian, Universitas Gadjah Mada, Indonesia

Mandahadi Kusuma, UIN Sunan Kalijaga Yogyakarta, Indonesia

Maria Ulfah Siregar, UIN Sunan Kalijaga Yogyakarta, Indonesia

Millati Pratiwi, Pusan National University, South Korea

Muhammad Dzulfikar Fauzi, Telkom University Surabaya, Indonesia

Muhammad Habibi, Universitas Jenderal Achmad Yani Yogyakarta, Indonesia

Muhammad Rifqi Maarif, Universitas Jenderal Achmad Yani Yogyakarta, Indonesia

Mohd. Fikri Azli bin Abdullah, Multimedia University, Malaysia

M. Alex Syaekhoni, Who's Good, South Korea

Niki Min Hidayati Robbi, Universitas Gadjah Mada, Indonesia

Norma Latif Fitriyani, Sejong University Seoul, South Korea

Okfalisa, UIN Sultan Syarif Kasim Riau, Indonesia

Oman Somantri, Politeknik Negeri Cilacap, Indonesia

Puguh Jayadi, Universitas PGRI Madiun, Indonesia

Puji Winar Cahyo, Universitas Jenderal Achmad Yani Yogyakarta, Indonesia

Qorry Aina Fitroh, UIN KH. Abdurrahman Wahid Pekalongan, Indonesia

Ridho Surya Kusuma, Universitas Siber Muhammadiyah, Yogyakarta, Indonesia

Rischan Mafrur, Macquarie University, Sydney, Australia

Shofwatul Uyun, UIN Sunan Kalijaga Yogyakarta, Indonesia

Sumarsono, UIN Sunan Kalijaga, Indonesia

Sunu Wibirama, Universitas Gadjah Mada, Indonesia

Tundo, Sekolah Tinggi Ilmu Komputer Cipta Karya Informatika (STIKOM CKI), Indonesia

Windra Swastika, Universitas Ma Chung, Indonesia

Yudistira Dwi Wardhana Asnar, Institut Teknologi Bandung, Indonesia

ISSN : 2527-5836

e-ISSN: 2528-0074

JISKa (Jurnal Informatika Sunan Kalijaga)

Vol. 11, No. 1, JANUARY 2026

TABLE OF CONTENT

JISKA (Jurnal Informatika Sunan Kalijaga)
Vol. 11, No. 1, JANUARI, 2026, Pp. 1 – 12
ISSN: 2527 – 5836 (print) | 2528 – 0074 (online)

Deteksi Diabetes Mellitus dengan Menggunakan Teknik Ensemble XGBoost dan LightGBM

Naufal Adhi Pratama [(1)], Danang Wahyu Utomo [(2)*]
Departemen Teknik Informatika, Universitas Dian Nuswantoro, Semarang, Indonesia.
e-mail : 111202113590@mhs.dinus.ac.id, danang.wu@dsn.dinus.ac.id.
* Penulis korespondensi.
Artikel ini diajukan 28 Desember 2024, direvisi 30 April 2025, diterima 15 Mei 2025, dan dipublikasikan 25 Januari 2026.

Abstract

Diabetes mellitus is a metabolic disease characterized by elevated blood sugar levels due to impaired insulin secretion, insulin action, or both. The disease has a major impact on public health and contributes to high morbidity and mortality rates in many countries. Prevention and early detection are essential to reduce the adverse effects of this disease. This study aims to analyze and apply machine learning algorithms in detecting diabetes mellitus, focusing on the use of XGBoost and LightGBM algorithms. The dataset used in this study includes various features related to diabetes risk factors, such as age, gender, body mass index (BMI), hypertension, smoking history, and HbA1c and blood glucose levels. Preprocessing was performed to clean and balance the data using the SMOTE-Tomek technique. Next, the model was built and evaluated using the K-Fold cross-validation method to measure the accuracy and stability of the model. The results showed that the XGBoost model achieved 97.31% accuracy, while the LightGBM model produced 97.26% accuracy. Combining the two models through blending techniques resulted in an accuracy of 97.51%, indicating that the combination of models can improve prediction performance. This study shows the great potential of machine learning algorithms, especially XGBoost and LightGBM, in detecting diabetes mellitus accurately and efficiently. Hopefully, the results of this study can contribute to the development of decision support systems for more effective early diagnosis of diabetes.

Keywords: Diabetes Mellitus, Machine Learning, XGBoost, LightGBM, Early Detection

Abstrak

Diabetes mellitus adalah penyakit metabolik yang ditandai dengan peningkatan kadar gula darah akibat gangguan sekresi insulin, kerja insulin, atau keduanya. Penyakit ini memiliki dampak besar terhadap kesehatan masyarakat dan berkontribusi pada tingginya angka morbiditas dan mortalitas di banyak negara. Pencegahan dan deteksi dini sangat penting untuk mengurangi dampak buruk dari penyakit ini. Penelitian ini bertujuan untuk menganalisis dan menerapkan algoritma machine learning dalam mendeteksi diabetes mellitus, dengan fokus pada penggunaan algoritma XGBoost dan LightGBM. Dataset yang digunakan dalam penelitian ini mencakup berbagai fitur terkait faktor risiko diabetes, seperti usia, jenis kelamin, indeks massa tubuh (BMI), hipertensi, riwayat merokok, serta kadar HbA1c dan glukosa darah. Proses preprocessing dilakukan untuk membersihkan dan menyeimbangkan data menggunakan teknik SMOTE-Tomek. Selanjutnya, model dibangun dan dievaluasi dengan menggunakan metode K-Fold cross-validation untuk mengukur akurasi dan kestabilan model. Hasil penelitian menunjukkan bahwa model XGBoost mencapai akurasi 97.31%, sementara model LightGBM menghasilkan akurasi 97.26%. Penggabungan kedua model melalui teknik blending menghasilkan akurasi 97.51%, yang menunjukkan bahwa kombinasi model dapat meningkatkan performa prediksi. Penelitian ini menunjukkan potensi besar dari algoritma machine learning, khususnya XGBoost dan LightGBM, dalam mendeteksi diabetes mellitus secara akurat dan efisien. Diharapkan, hasil penelitian ini dapat berkontribusi pada pengembangan sistem pendukung keputusan untuk diagnosis dini diabetes yang lebih efektif.

Kata Kunci: Diabetes Mellitus, Pembelajaran Mesin, XGBoost, LightGBM, Deteksi Dini

1. PENDAHULUAN

Diabetes mellitus, menurut definisi World Health Organization (WHO), adalah penyakit degeneratif kronis yang disebabkan oleh produksi insulin yang tidak mencukupi di pankreas atau ketidakmampuan tubuh untuk secara efektif menggunakan insulin yang diproduksi (Tanwar & Bhatia, 2024). *Hyperglycemia* (peningkatan kadar glukosa darah) menjadi indikator utama dari penyakit ini. Insulin sendiri adalah hormon yang berfungsi mengatur kadar gula darah. *Diabetes mellitus* merupakan salah satu penyakit dengan pertumbuhan tertinggi di dunia. Diperkirakan sekitar 537 juta orang dewasa di seluruh dunia, yang berusia antara 20 hingga 79 tahun, menderita diabetes, yang setara dengan 10,5% dari seluruh populasi pada rentang usia tersebut. Pada tahun 2030, jumlah penderita diabetes diperkirakan akan mencapai 643 juta orang secara global, dan meningkat menjadi 783 juta pada tahun 2045 (Mujumdar & Vaidehi, 2019). Menurut edisi ke-10 International Diabetes Federation (IDF), insiden *diabetes* di negara-negara Asia Tenggara (SEA) telah meningkat selama lebih dari 20 tahun, dan perkiraan saat ini telah melampaui seluruh proyeksi sebelumnya (Ogurtsova et al., 2017).

Menurut artikel yang diterbitkan oleh Kementerian Kesehatan Republik Indonesia, prevalensi diabetes di Indonesia mencapai 11,7% pada tahun 2023 (Rif'at et al., 2023). Ini menunjukkan peningkatan dibandingkan dengan tahun-tahun sebelumnya, di mana prevalensi diabetes tercatat sebesar 8,5% pada tahun 2018. Oleh karena itu, pemanfaatan teknologi dalam manajemen diabetes menjadi sangat penting. Dengan penerapan teknologi digital, kita dapat memprediksi dan mengidentifikasi kekurangan dalam perawatan pasien melalui penggunaan Machine Learning.

Pendekatan *Machine Learning* banyak digunakan dalam studi epidemiologi klinis terkait diabetes, karena keunggulannya dalam memprediksi dan mengklasifikasikan karakteristik pasien dengan mengenali pola dalam kumpulan data (Mengcan et al., 2021). Machine learning sendiri menggunakan berbagai jenis model algoritma, yang umumnya terbagi menjadi tiga kategori, yaitu *supervised learning*, *unsupervised learning*, dan *reinforcement learning*. Masing-masing kategori memiliki karakteristik dan pendekatan yang berbeda dalam proses pembelajaran. Banyaknya penelitian yang dilakukan menunjukkan tingginya minat dalam mengembangkan metode serta merancang model machine learning yang mampu memprediksi dan mengklasifikasikan karakteristik pasien *diabetes mellitus* secara akurat.

Penelitian oleh Butt et al. (2021) menunjukkan bahwa model *Multilayer Perceptron* (MLP) mampu mengungguli pengklasifikasi lainnya dengan akurasi 86,08%. Selain itu, model *Long Short-Term Memory* (LSTM) meningkatkan performa prediksi secara signifikan dengan akurasi mencapai 87,26%. Penelitian tersebut juga melakukan analisis komparatif dengan teknik lain yang sudah ada, dan menunjukkan bahwa pendekatan yang digunakan memiliki kemampuan adaptasi yang tinggi dalam berbagai aplikasi di bidang kesehatan.

Penelitian yang dilakukan oleh Chang et al. (2023) menunjukkan bahwa model klasifikasi Naive Bayes mampu memberikan hasil yang lebih baik dibandingkan dengan model Random Forest dan Decision Tree (J48) dalam hal akurasi prediksi. Pada subset data yang hanya melibatkan tiga faktor, performa model Naive Bayes tetap kompetitif dan sebanding dengan hasil yang diperoleh oleh model Random Forest pada dataset lengkap. Model Naive Bayes mencapai akurasi sebesar 79,13%, sedangkan model Random Forest memperoleh 79,57%, keduanya menjadi akurasi tertinggi yang dicapai dalam percobaan tersebut.

Penelitian oleh Saxena et al. (2022) melakukan penelitian dengan membandingkan performa *Multilayer Perceptron*, *Decision Tree*, *K-Nearest Neighbour*, dan *Random Forest*, dengan penerapan beberapa teknik pemilihan fitur untuk mendeteksi *diabetes* pada tahap awal. Proses *preprocessing* dilakukan terhadap data mentah, termasuk penghilangan *outlier* dan imputasi nilai yang hilang berdasarkan rata-rata, serta optimasi *hyperparameter*. Eksperimen dilakukan pada dataset PIMA India menggunakan Weka 3.9. Akurasi yang dicapai oleh masing-masing model adalah sebagai berikut: *Multilayer Perceptron* sebesar 77,60%, *Decision Tree* sebesar 76,07%,

K-Nearest Neighbour sebesar 78,58%, dan *Random Forest* sebesar 79,8%, yang merupakan performa terbaik di antara model yang dibandingkan.

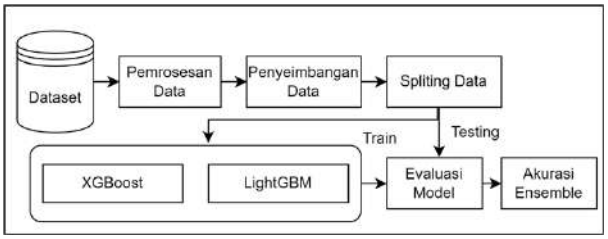
Selanjutnya penelitian oleh Lai et al. (2019) membangun model prediktif menggunakan teknik *Logistic Regression* dan *Gradient Boosting Machine* (GBM). Evaluasi performa dilakukan dengan menggunakan *receiver operating characteristic curve* (ROC). Penyesuaian ambang batas dan penerapan metode *class weighting* digunakan untuk meningkatkan sensitivitas, yaitu proporsi pasien *diabetes mellitus* yang terdeteksi secara tepat oleh model. Hasilnya, *Area Under the ROC Curve* (AUC) untuk model GBM mencapai 84,7% dengan sensitivitas 71,6%, sedangkan model *Logistic Regression* menghasilkan AUC sebesar 84,0% dengan sensitivitas 73,4%. Kedua model tersebut menunjukkan performa yang lebih baik dibandingkan dengan model *Decision Tree* dan *Random Forest*.

Berdasarkan berbagai studi tersebut, dapat disimpulkan bahwa akurasi dari model *Machine Learning* sangat bergantung pada pemilihan arsitektur algoritma yang tepat, terutama dalam mengidentifikasi kondisi medis seperti *diabetes mellitus*. Salah satu kemajuan terbesar dalam pengembangan *Machine Learning* adalah pendekatan *Ensemble Learning*. *Machine Learning* sendiri merupakan proses yang digunakan untuk menganalisis dan mengekstraksi informasi dari data dalam jumlah besar guna menemukan pengetahuan yang berguna.

Penelitian ini berfokus pada penerapan algoritma *Extreme Gradient Boosting* (XGBoost) dan *Light Gradient Boosting Machine* (LightGBM). Dataset yang digunakan mencakup berbagai fitur yang berkaitan dengan faktor risiko *diabetes*. Melalui proses *preprocessing*, termasuk penanganan data tidak seimbang menggunakan teknik *SMOTE-Tomek*, model dikembangkan dan dievaluasi dengan menggunakan metode *K-Fold Cross-Validation*. Oleh karena itu, penelitian ini bertujuan untuk mengembangkan dan mengevaluasi model *Machine Learning* yang mampu memprediksi risiko *diabetes mellitus* dengan akurasi tinggi, serta mengidentifikasi faktor-faktor risiko utama yang berkontribusi terhadap kondisi tersebut melalui algoritma XGBoost dan LightGBM.

2. METODE PENELITIAN

Alur metode penelitian yang digunakan digambarkan pada Gambar 1. Penelitian dimulai dengan pengumpulan *dataset* yang berisi berbagai *fitur* terkait faktor risiko diabetes mellitus. Langkah pertama adalah *data preprocessing*, yang mencakup pembersihan data dan penanganan nilai yang hilang. Selanjutnya, dilakukan penyimbangan data menggunakan teknik SMOTE-Tomek untuk mengatasi ketidakseimbangan kelas dalam *dataset*. Setelah itu, *dataset* dibagi menjadi dua bagian: data pelatihan (*training*) dan data pengujian (*testing*). Tahap berikutnya adalah klasifikasi menggunakan dua algoritma *machine learning*, yaitu XGBoost dan LightGBM, untuk memprediksi risiko diabetes. Evaluasi kinerja model dilakukan menggunakan metrik seperti akurasi. Untuk meningkatkan performa prediksi, dilakukan *ensemble learning* dengan menggabungkan hasil dari kedua model.



Gambar 1 Metode Penelitian

2.1 Dataset

Dataset yang digunakan dalam penelitian ini terdiri dari data medis dan demografis sekitar 100.000 pasien, mencakup status diabetes positif maupun negatif, dan tersedia secara publik di

Kaggle. Dataset ini berisi berbagai fitur klinis dan gaya hidup yang relevan untuk prediksi diabetes, seperti jenis kelamin, usia, riwayat hipertensi, penyakit jantung, kebiasaan merokok, indeks massa tubuh (BMI), kadar HbA1c, serta kadar glukosa darah. Berdasarkan studi sebelumnya oleh Alam et al. (2014), fitur-fitur ini terbukti memiliki korelasi yang signifikan terhadap risiko diabetes dan sering digunakan sebagai indikator utama dalam model prediksi medis berbasis machine learning.

Tabel 1 Struktur Dataset

	Gender	Age	Hypertension	Heart_disease	Smoking_history	Bmi	hbA1 Level	Blood_Glucose_Level	Diabetes
Count	961	961	961	961	961	961	961	961	961
Mean	0.41	41.7	0.07	0.04	2.23	27.	5.53	138	0.08
Std	0.49	22.4	0.26	0.19	1.87	6.7	1.07	40.9	0.28
Min	0.00	0.00	0.00	0.00	0.00	10	3.50	80.0	0.00
Max	2.00	80.0	1.00	1.00	5.00	95	9.00	300	1.00

Pada Tabel 1 terdapat gambaran lengkap mengenai faktor-faktor risiko yang berkaitan dengan diabetes mellitus, termasuk data medis dan demografis seperti usia, indeks massa tubuh, tekanan darah, kadar glukosa, dan riwayat keluarga. Informasi ini mencerminkan keragaman atribut yang relevan dalam konteks diagnosis dan klasifikasi diabetes, sehingga menyediakan landasan yang kuat untuk proses pelatihan model pembelajaran mesin dalam studi ini.

2.2 Pemrosesan Data

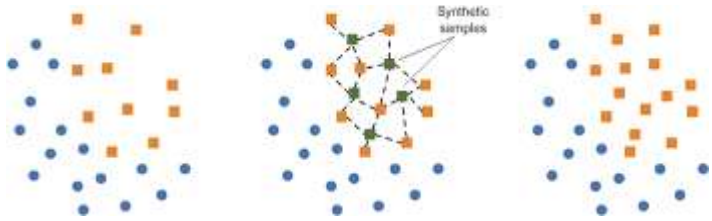
Pemrosesan data adalah langkah penting dalam mempersiapkan data sebelum digunakan untuk membangun model pembelajaran mesin (Galicia-garcia et al., 2020). Proses ini melibatkan serangkaian tahapan untuk membersihkan, mengubah, dan memformat data agar sesuai dengan kebutuhan model (Kharis & Zili, 2022). Langkah pertama adalah penanganan data yang hilang, dimana nilai yang hilang pada dataset diatasi dengan cara mengganti, menghapus, atau memperkirakan nilai yang hilang berdasarkan data yang ada. Selanjutnya, dilakukan normalisasi atau standardisasi untuk memastikan bahwa fitur-fitur numerik berada dalam skala yang sama, sehingga model tidak terpengaruh oleh perbedaan skala antar fitur. Kemudian, penanganan data tidak seimbang dilakukan, terutama jika dataset memiliki distribusi kelas yang tidak merata, menggunakan teknik seperti SMOTE atau undersampling untuk menyeimbangkan jumlah data antar kelas. Setelah itu, dilakukan encoding variabel kategorikal untuk mengubah data kategorikal menjadi format yang dapat diproses oleh model, seperti one-hot encoding atau label encoding. Langkah terakhir adalah pembagian dataset menjadi data pelatihan (training) dan data pengujian (testing) untuk memastikan bahwa model dapat dievaluasi secara efektif. Semua tahapan preprocessing ini bertujuan untuk meningkatkan kualitas data dan memaksimalkan kinerja model dalam melakukan prediksi.

2.3 Smote-Tomeks

Dalam penelitian ini, SMOTE (Synthetic Minority Over-sampling Technique) dan Tomek Links digunakan sebagai metode untuk mengatasi permasalahan ketidakseimbangan kelas dalam dataset (Fareed et al., 2022). SMOTE berfungsi dengan menambahkan sampel sintetis pada kelas minoritas untuk meningkatkan representasi data, sehingga model pembelajaran tidak bias terhadap kelas mayoritas. Sementara itu, Tomek Links digunakan untuk melakukan undersampling dengan menghapus pasangan data yang saling berdekatan namun berasal dari kelas berbeda, guna memperjelas batas keputusan antar kelas dan meningkatkan performa model secara keseluruhan (Muljono et al., 2024).

Gambar 2 menunjukkan teknik Smote-Tomeks bekerja dengan mengambil sampel acak dari kelas minoritas, mencari tetangga terdekatnya, kemudian menghasilkan data sintetis dengan mengurangkan sampel dari tetangga terdekat dan mengalikannya dengan bilangan acak. Sebaliknya, *Tomek Links* adalah strategi undersampling yang mengurangi jumlah sampel di kelas

mayoritas. Metode ini menemukan pasangan sampel dari kelas yang berbeda yang merupakan tetangga terdekat satu sama lain (Wang et al., 2019). Pasangan sampel ini, yang dikenal sebagai *Tomek Links*, kemudian dieliminasi dari kelas mayoritas.



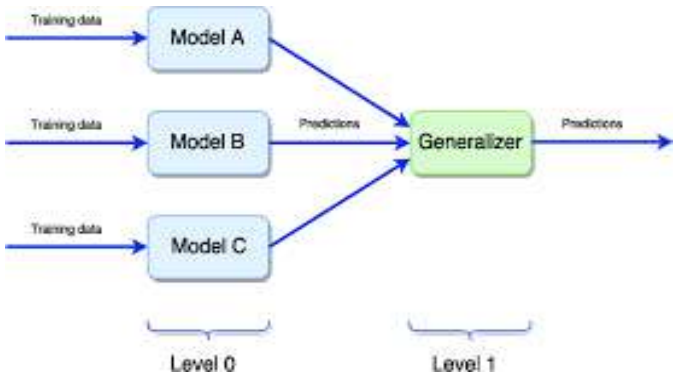
Gambar 2 Smote-Tomeks

2.4 Pembagian Data

Langkah berikutnya adalah membagi data menjadi subset pelatihan dan pengujian menggunakan metode stratifikasi label dengan rasio 80:20 setelah menyelesaikan prapemrosesan data dan penanganan ketidak seimbangan data. Mempersiapkan data untuk model pembelajaran mesin adalah langkah krusial. Train-Test Split memisahkan data yang telah diproses menjadi dua subset: data pelatihan (train) dan data pengujian (test) (Kahloot & Ekler, 2021). Dengan rasio 80:20, 80% dari data digunakan untuk melatih model, sementara 20% sisanya digunakan untuk menguji efektivitas model. Metode stratifikasi label memastikan bahwa distribusi kelas dalam dataset asli tetap terjaga di kedua subset, sehingga data pengujian secara akurat mewakili semua kelas.

2.5 Teknik Boosting

Boosting adalah model *ensemble* yang dibentuk dari beberapa model dasar yang secara berurutan model dasar tersebut dilatih dan digabungkan dalam prediksi. Algoritma boosting membangun model secara bertahap dengan mengoptimalkan suatu fungsi kerugian (Manconi et al., 2022). Pada Gambar 3 dijelaskan bahwa metode boosting menggunakan data masukan untuk melatih model boosting lemah, kesalahan pada klasifikasi, dan melatih model tersebut dengan set yang diterapkan pada klasifikasi sebelumnya peneliti lain menyatakan bahwa metode *boosting* ini berfokus pada pengurangan bias daripada variasi dengan meningkatkan boosting awal dasar yang dimiliki bias tinggi. Secara dalam analisis data prediktif dan meningkatkan kinerja model pada data kompleks atau sulit diklasifikasikan dan dapat bekerja baik dengan berbagai algoritma.

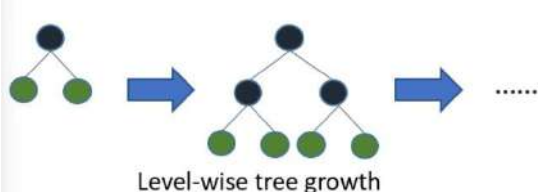


Gambar 3 Teknik Boosting

2.6 Teknik XGBoost

XGBoost merupakan implementasi dari *Gradient Boosting* untuk meningkatkan performa kinerja dan stabilitas, seperti yang ditunjukkan pada Gambar 4. Pada kasus klasifikasi dan regresi,

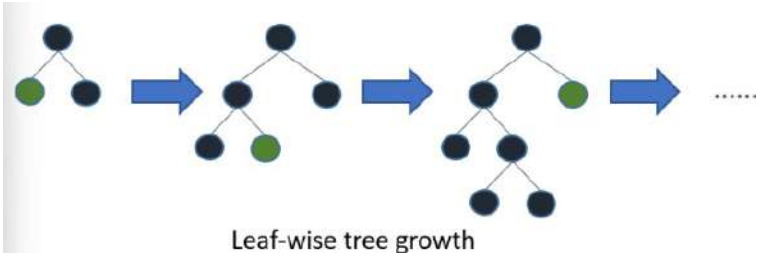
algoritma *XGBoost* tepat diusulkan karena mengacu pada pohon keputusan terbaik. Menurut Jafarzadeh et al menyatakan bahwa algoritma *XGBoost* adalah algoritma terbaik jika dibandingkan dengan algoritma ML lainnya (Sepbriant & Utomo, 2024). *XGBoost* menggunakan pohon keputusan sebagai model dasar dan menerapkan teknik boosting untuk meningkatkan kinerja model secara bertahap. Keunggulan utama *XGBoost* termasuk regularisasi untuk mencegah *overfitting*, *parallel processing* untuk mempercepat pelatihan model, dan kemampuan untuk menangani nilai yang hilang secara otomatis. Metode ini telah terbukti sangat efektif dalam berbagai kompetisi dan aplikasi machine learning, memberikan hasil prediksi yang akurat dan efisien (Azmi & Baliga, 2020).



Gambar 4 Teknik XGBoost

2.7 Teknik LightGBM

Teknik *LightGBM* dirancang untuk meningkatkan efisiensi model dan mengurangi penggunaan memori, seperti yang ditunjukkan pada Gambar 5. Dibandingkan dengan metode lainnya, *LightGBM* mengimplementasikan dua teknik inovatif, yaitu *Gradient-based One Side Sampling* (GOSS) dan *Exclusive Feature Bundling* (EFB) (Sari et al., 2023). GOSS berfokus pada sampling data yang lebih relevan, mengurangi jumlah data yang perlu diproses, sementara EFB menggabungkan fitur eksklusif untuk mengurangi dimensi dan mempercepat proses pelatihan (Machado et al., 2019). Teknik-teknik ini memungkinkan *LightGBM* untuk bekerja lebih cepat dan lebih efisien, terutama dalam menangani dataset besar, sekaligus menjaga akurasi yang tinggi. *LightGBM* sangat cocok digunakan dalam berbagai aplikasi *machine learning*, terutama yang membutuhkan kecepatan dan skalabilitas.



Gambar 5 Teknik LightGBM

2.8 Matrix Evaluasi

Matrix Evaluasi adalah sebuah alat dalam pembelajaran mesin dan statistik yang digunakan untuk menilai kinerja algoritma klasifikasi (Zhang et al., 2021). Ini adalah matriks persegi yang sering digunakan untuk merangkum hasil dari. suatu masalah klasifikasi. Matriks kebingungan memberikan rincian terperinci tentang prediksi yang benar dan salah yang dibuat oleh model klasifikasi. Alat ini sangat berguna untuk masalah klasifikasi biner (dua kelas), namun juga dapat digunakan untuk klasifikasi multi-kelas (Thohari et al., 2024). Dalam matriks kebingungan, terdapat empat istilah yang menggambarkan hasil proses klasifikasi. Seperti yang terlihat pada Tabel II, istilah-istilah tersebut adalah True Positive (TP), False Positive (FP), True Negative (TN), dan False Negative (FN). TP dan TN menunjukkan hasil klasifikasi yang benar, sedangkan FP dan FN menunjukkan hasil klasifikasi yang salah. Rumus-rumus yang digunakan dalam metrik evaluasi ditunjukkan pada Tabel 2.

Tabel 2 Matrix Evaluasi

Precision	Sensitivity	F1-Score	Accuracy
$\frac{TP}{TP+FP}$	$\frac{TP}{TP+FN}$	$\frac{Precision \times Sensitivity}{Precision + Sensitivity}$	$\frac{TP+TN}{TP+TN+FP+FN}$

Akurasi digunakan untuk menghitung proporsi total prediksi yang benar terhadap keseluruhan data, sehingga memberikan gambaran umum tentang seberapa tepat model dalam mengklasifikasikan semua kelas. Sementara itu, sensitivitas atau recall digunakan untuk mengevaluasi sejauh mana model mampu mengenali kelas positif dengan benar, yang sangat penting ketika fokus utama adalah meminimalkan kesalahan dalam mendeteksi kasus positif. Di sisi lain, F1-score digunakan untuk menyeimbangkan antara presisi dan sensitivitas, dan menjadi metrik yang sangat berguna dalam situasi di mana distribusi kelas tidak seimbang, karena mempertimbangkan baik kesalahan positif maupun kesalahan negatif.

2.9 Ensemble Learning

Ensemble learning adalah pendekatan dalam pembelajaran mesin yang menggabungkan beberapa *weak learners* untuk membentuk satu model yang lebih kuat dengan kinerja prediktif yang lebih baik. Sebuah *weak learner* merupakan model yang performanya hanya sedikit lebih baik dibandingkan dengan tebakan acak (Gomes et al., 2018). Tujuan utama dalam perancangan *ensemble* adalah memastikan bahwa setiap anggota dalam *ensemble* memiliki karakteristik kesalahan klasifikasi yang berbeda. Jika anggota-anggota *ensemble* cenderung melakukan kesalahan pada *instance* yang tidak saling tumpang tindih, maka kombinasi prediksi dapat saling melengkapi dan menghasilkan kinerja yang lebih tinggi dibandingkan model individual. Dalam *ensemble learning*, istilah seperti *kombinasi* atau *voting* digunakan untuk menggambarkan mekanisme penggabungan prediksi dari berbagai anggota untuk memperoleh prediksi akhir. Arsitektur *ensemble* merujuk pada susunan *klasifikator* di dalam sistem *ensemble*, sementara metode *voting* menentukan bagaimana hasil prediksi dari setiap anggota digunakan dalam proses pengambilan keputusan akhir (Kumar et al., 2022). Meskipun peningkatan keberagaman dalam *ensemble* tidak selalu menjamin perbaikan kinerja, pendekatan ini tetap menjadi fokus utama dalam penelitian pembelajaran mesin. Berbagai taksonomi dan metode baru terus dikembangkan untuk meningkatkan akurasi dan efektivitas model, baik dalam konteks pembelajaran *batch* maupun pada aliran data (*data streams*).

3. HASIL DAN PEMBAHASAN

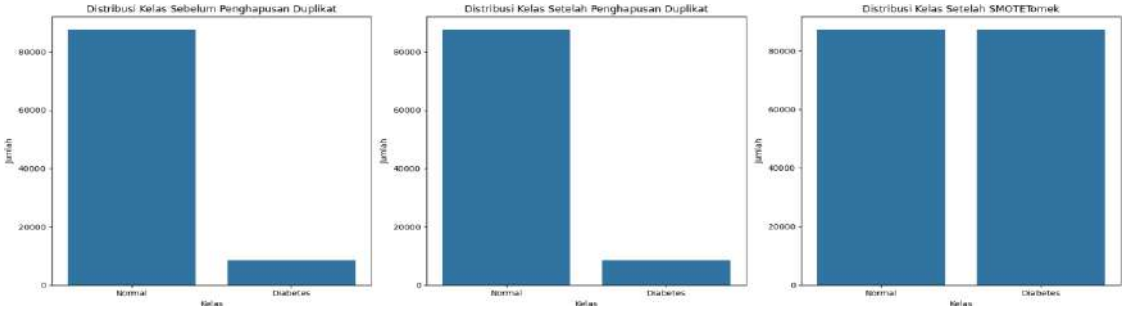
Dala penelitian ini, tahap pertama yang dilakukan adalah preprocessing dataset, yang dimulai dengan menghapus data duplikat. Tabel 3 menunjukkan bahwa pada tahap awal, dataset yang digunakan mengandung sejumlah data duplikat. Adanya duplikasi dalam dataset dapat memberikan pengaruh yang signifikan terhadap hasil analisis, karena data yang sama diulang-ulang dapat memberikan bobot yang tidak proporsional pada model, yang akhirnya menurunkan akurasi model. Oleh karena itu, langkah pertama yang diambil adalah menghapus data duplikat. Berdasarkan hasil yang didapat, dataset yang memiliki 100.000 entri awal, setelah penghapusan data duplikat, berkurang menjadi 96.146 entri. Hal ini memastikan bahwa model yang dibangun menggunakan data yang bersih dan lebih representatif. Selanjutnya dolakukan tranformasi data. Pada tahap transformating data kolom kategorikal smoking history dalam data frame diubah menjadi format numerik, sehingga data dapat digunakan dalam analisis lebih lanjut. Pada tabel menunjukkan data smoking history telah diubah menjadi format numerik.

Tabel 3 Penghapusan Data Duplikat

Jumlah Data	Data Duplikat	Setelah Penghapusan Data
100000	3854	96146

Dataset yang digunakan dalam penelitian ini awalnya berjumlah 100,000 entri, yang terdiri dari dua kelas, yaitu *Normal* dan *Diabetes*. Dari keseluruhan data, sebanyak 91,500 entri

dikategorikan sebagai *Normal* dan 8,500 entri lainnya diklasifikasikan sebagai *Diabetes*. Ketidakseimbangan kelas ini dapat menyebabkan model yang dibangun lebih cenderung untuk mengklasifikasikan data ke kelas mayoritas (*Normal*), sehingga berpotensi menurunkan performa model dalam mendeteksi kelas minoritas (*Diabetes*). Untuk mengatasi hal ini, langkah pertama yang dilakukan adalah menghapus data duplikat. Pada tahap ini, data duplikat diidentifikasi dan dihapus, menghasilkan 96,146 entri data yang unik. Penghapusan data duplikat sangat penting untuk memastikan bahwa setiap entri dalam dataset hanya mewakili informasi yang valid, yang akan mencegah distorsi dalam analisis dan model pembelajaran mesin.



Gambar 6 Preprocessing Dataset

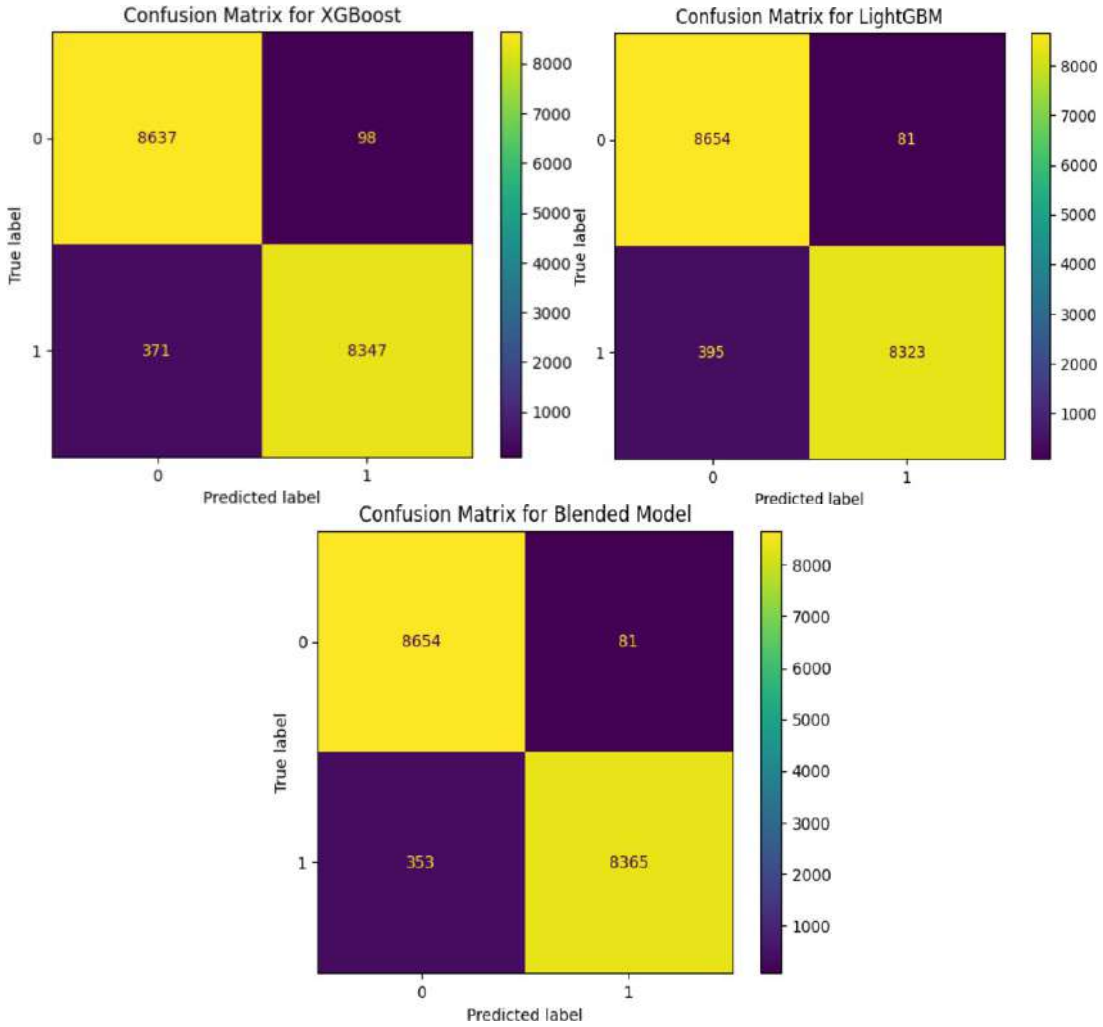
Setelah data duplikat berhasil dihapus, langkah selanjutnya adalah penyeimbangan dataset, karena terdapat ketidakseimbangan antara jumlah data untuk kelas *Normal* dan *Diabetes*. Ketidakseimbangan kelas ini dapat mempengaruhi akurasi model, karena model cenderung lebih banyak mengklasifikasikan data ke kelas yang lebih banyak jumlahnya. Untuk mengatasi masalah ini, diterapkan teknik SMOTETomek, yang menggabungkan dua pendekatan: *Synthetic Minority Over-sampling Technique* (SMOTE) dan *Tomek links*. SMOTE menghasilkan sampel sintetis dari kelas minoritas (*Diabetes*), sementara teknik Tomek links berfungsi untuk menghapus data yang tumpang tindih atau saling mendekati antara kelas mayoritas dan kelas minoritas, yang dapat mengurangi potensi kesalahan dalam klasifikasi.

Hasil dari penerapan SMOTETomek adalah terbentuknya distribusi kelas yang lebih seimbang, dengan masing-masing kelas (*Normal* dan *Diabetes*) memiliki 87,269 entri data. Dengan distribusi data yang seimbang, model yang dibangun diharapkan dapat memprediksi kedua kelas dengan lebih akurat, serta mengurangi bias yang mungkin terjadi terhadap kelas mayoritas. Langkah-langkah ini bertujuan untuk memaksimalkan kinerja model pembelajaran mesin dalam mendeteksi diabetes secara efektif dan tepat. Setelah proses persiapan data selesai, pelatihan model dilakukan menggunakan tiga pendekatan: XGBoost, LightGBM, dan Blended Model (gabungan XGBoost dan LightGBM). Evaluasi kinerja dilakukan menggunakan metrik akurasi, *precision*, *sensitivity* (recall), dan *F1- score*.

Gambar 7 dan Tabel 4 menunjukkan hasil evaluasi dari masing-masing model. Model XGBoost menunjukkan performa yang cukup baik, dengan akurasi sebesar 97,31%, *precision* sebesar 98,84%, *sensitivity* sebesar 95,74%, dan *F1-score* sebesar 97,27%. Berdasarkan *confusion matrix*, model ini mengklasifikasikan dengan benar 8.637 data sebagai Normal dan 8.347 sebagai Diabetes, sementara terdapat 98 kesalahan klasifikasi untuk data Normal (*false positive*) dan 371 kesalahan untuk data Diabetes (*false negative*).

Model LightGBM menunjukkan hasil yang sedikit berbeda, dengan akurasi sebesar 97,26%, *precision* sebesar 99,08%, *sensitivity* sebesar 95,46%, dan *F1-score* sebesar 97,23%. Model ini menghasilkan 81 *false positive* dan 395 *false negative*, serta mengklasifikasikan 8.654 data sebagai Normal dan 8.323 sebagai Diabetes. Meskipun tingkat akurasi sedikit lebih rendah dibandingkan XGBoost, nilai *precision*-nya lebih tinggi, menunjukkan kemampuan yang baik dalam menghindari kesalahan klasifikasi untuk kelas minoritas.

Model terbaik diperoleh dari pendekatan *ensemble* melalui Blended Model yang menggabungkan XGBoost dan LightGBM. Model ini mencatat akurasi tertinggi sebesar 97,51%, dengan *precision* sebesar 98,95%, *sensitivity* sebesar 96,00%, dan *F1-score* sebesar 97,46%. Berdasarkan hasil *confusion matrix*, model ini berhasil mengklasifikasikan 8.654 data sebagai Normal dan 8.365 sebagai Diabetes, dengan hanya 81 *false positive* dan 353 *false negative*.



Gambar 7 Hasil Evaluasi

Tabel 4 Evaluated Models

Model	Class	Precision	Recall	F1-Score	Accuracy
XGBoost	Predicted	0.9884	0.9574	0.9727	0.9731
LightGBM	Predicted	0.9908	0.9546	0.9723	0.9726
Ensemble	Predicted	0.9895	0.9600	0.9746	0.9751

4. KESIMPULAN

Penelitian ini bertujuan untuk mengembangkan dan mengevaluasi model machine learning dalam mendeteksi diabetes mellitus menggunakan algoritma XGBoost dan LightGBM, dua model pembelajaran mesin yang sering digunakan untuk klasifikasi data medis. Diabetes mellitus adalah penyakit metabolik yang berdampak besar terhadap kesehatan masyarakat, dan deteksi dini sangat penting untuk mengurangi dampak buruknya. Melalui penggunaan dataset yang mencakup berbagai fitur terkait faktor risiko diabetes, penelitian ini menguji efektivitas algoritma

XGBoost dan LightGBM dalam mendeteksi dan mengklasifikasikan data diabetes dan normal. Hasil eksperimen menunjukkan bahwa kedua model ini memiliki kinerja yang sangat baik dalam hal akurasi prediksi. Model XGBoost mencatatkan akurasi sebesar 97,31%, sedangkan LightGBM menghasilkan akurasi 97,26%.

Penerapan teknik SMOTE-Tomek untuk menangani ketidakseimbangan data menjadi salah satu faktor penting dalam meningkatkan kinerja model. Dalam dataset asli, terdapat ketidakseimbangan antara jumlah data pasien yang terdiagnosis diabetes dan yang tidak, yang berpotensi menyebabkan model kurang sensitif dalam mendeteksi kelas minoritas (diabetes). Dengan menerapkan SMOTE-Tomek, data yang tidak seimbang dapat diubah menjadi lebih seimbang, sehingga model dapat belajar lebih baik dalam membedakan antara kedua kelas. Proses preprocessing ini terbukti meningkatkan kemampuan model dalam mengenali pola-pola penting yang membedakan data diabetes dan normal, serta menghasilkan data pelatihan yang lebih representatif.

Selanjutnya, penggabungan kedua model ini melalui teknik ensemble atau blending menghasilkan peningkatan kinerja yang signifikan, dengan akurasi mencapai 97,51%. Hasil ini menunjukkan bahwa kombinasi dari kedua algoritma dapat meningkatkan kemampuan prediksi model dibandingkan jika hanya menggunakan satu model tunggal. Teknik ensemble memanfaatkan kelebihan masing-masing model, dengan mengurangi kelemahan yang ada, sehingga menghasilkan keputusan yang lebih baik. Penilaian model dilakukan menggunakan metode K-Fold cross-validation, yang memberikan hasil yang lebih stabil dan menghindari bias evaluasi yang bisa timbul jika hanya menggunakan satu data split. Dengan pendekatan ini, penelitian ini berhasil menghasilkan model yang dapat memberikan prediksi diabetes dengan tingkat keakuratan yang sangat tinggi.

Dari hasil penelitian ini, dapat disimpulkan bahwa penerapan algoritma XGBoost dan LightGBM, yang dipadukan dengan teknik SMOTE-Tomek dan ensemble, memberikan hasil yang sangat memuaskan dalam mendeteksi diabetes mellitus. Kombinasi model ini terbukti tidak hanya mampu meningkatkan akurasi, tetapi juga mengurangi kesalahan klasifikasi, seperti False Positives dan False Negatives. Oleh karena itu, model ini memiliki potensi besar untuk diimplementasikan dalam sistem pendukung keputusan medis yang dapat membantu dalam diagnosis dini diabetes mellitus secara lebih akurat dan efisien. Selain itu, penelitian ini juga memberikan wawasan baru mengenai pentingnya teknik preprocessing dan ensemble dalam meningkatkan kinerja model pembelajaran mesin, khususnya dalam konteks data medis yang sering kali tidak seimbang. Sebagai arah pengembangan penelitian selanjutnya, disarankan untuk mengeksplorasi pendekatan optimasi parameter (hyperparameter tuning) secara lebih mendalam serta penerapan metode feature selection berbasis algoritma evolusioner guna meningkatkan efisiensi dan kinerja model. Selain itu, penggunaan algoritma deep learning seperti deep neural networks atau transformer-based models dapat dipertimbangkan untuk mengakomodasi kompleksitas hubungan antar fitur yang tidak linier. Di samping itu, penelitian lanjutan sebaiknya tidak hanya terbatas pada tugas klasifikasi statis, tetapi juga mencakup prediksi longitudinal terkait progresi diabetes dan efektivitas intervensi medis. Pendekatan explainable AI (XAI) juga direkomendasikan agar hasil model dapat diterjemahkan secara interpretatif, sehingga mendukung transparansi dan kepercayaan dalam implementasi klinis, untuk lebih meningkatkan performa dalam deteksi dan prediksi diabetes mellitus.

DAFTAR PUSTAKA

Alam, U., Asghar, O., Azmi, S., & Malik, R. A. (2014). General Aspects of Diabetes Mellitus. In *Handbook of Clinical Neurology* (Vol. 4, pp. 211–222). https://doi.org/10.1016/B978-0-444-53480-4.00015-1

Azmi, S. S., & Baliga, S. (2020). An Overview of Boosting Decision Tree Algorithms Utilizing AdaBoost and XGBoost Boosting Strategies. *International Research Journal of Engineering and Technology*, *7*(5), 6867–6870. https://www.irjet.net/archives/V7/i5/IRJET-V7I51293.pdf

Butt, U. M., Letchmunan, S., Ali, M., Hassan, F. H., Baqir, A., & Sherazi, H. H. R. (2021). Machine Learning Based Diabetes Classification and Prediction for Healthcare Applications. *Journal of Healthcare Engineering*, *2021*, 1–17. https://doi.org/10.1155/2021/9930985

Chang, V., Bailey, J., Xu, Q. A., & Sun, Z. (2023). Pima Indians Diabetes Mellitus Classification Based on Machine Learning (ML) Algorithms. *Neural Computing and Applications*, *35*(22), 16157–16173. https://doi.org/10.1007/s00521-022-07049-z

Fareed, M. M. S., Zikria, S., Ahmed, G., Mui-Zzud-Din, Mahmood, S., Aslam, M., Jillani, S. F., Moustafa, A., & Asad, M. (2022). ADD-Net: An Effective Deep Learning Model for Early Detection of Alzheimer Disease in MRI Scans. *IEEE Access*, *10*, 96930–96951. https://doi.org/10.1109/ACCESS.2022.3204395

Galicia-garcia, U., Benito-vicente, A., Jebari, S., & Larrea-sebal, A. (2020). Costus Ignus: Insulin Plant and It's Preparations as Remedial Approach for Diabetes Mellitus. *International Journal of Molecular Sciences*, 1–34. https://doi.org/10.13040/IJPSR.0975-8232.13(4).1551-58

Gomes, H. M., Barddal, J. P., Enembreck, F., & Bifet, A. (2018). A Survey on Ensemble Learning for Data Stream Classification. *ACM Computing Surveys*, *50*(2), 1–36. https://doi.org/10.1145/3054925

Kahloot, K. M., & Ekler, P. (2021). Algorithmic Splitting: A Method for Dataset Preparation. *IEEE Access*, *9*, 125229–125237. https://doi.org/10.1109/ACCESS.2021.3110745

Kharis, S. A. A., & Zili, A. H. A. (2022). Learning Analytics dan Educational Data Mining pada Data Pendidikan. *Jurnal Riset Pembelajaran Matematika Sekolah*, *6*(1), 12–20. https://doi.org/10.21009/jrpms.061.02

Kumar, M., Singhal, S., Shekhar, S., Sharma, B., & Srivastava, G. (2022). Optimized Stacking Ensemble Learning Model for Breast Cancer Detection and Classification Using Machine Learning. *Sustainability*, *14*(21), Article ID: 13998. https://doi.org/10.3390/su142113998

Lai, H., Huang, H., Keshavjee, K., Guergachi, A., & Gao, X. (2019). Predictive Models for Diabetes Mellitus Using Machine Learning Techniques. *BMC Endocrine Disorders*, *19*(1), Article ID: 101. https://doi.org/10.1186/s12902-019-0436-6

Machado, M. R., Karray, S., & de Sousa, I. T. (2019). LightGBM: an Effective Decision Tree Gradient Boosting Method to Predict Customer Loyalty in the Finance Industry. *2019 14th International Conference on Computer Science & Education (ICCSE)*, 1111–1116. https://doi.org/10.1109/ICCSE.2019.8845529

Manconi, A., Armano, G., Gnocchi, M., & Milanesi, L. (2022). A Soft-Voting Ensemble Classifier for Detecting Patients Affected by COVID-19. *Applied Sciences*, *12*(15), Article ID: 7554. https://doi.org/10.3390/app12157554

Mengcan, M., Xiaofang, C., & Yongfang, X. (2021). Constrained Voting Extreme Learning Machine and Its Application. *Journal of Systems Engineering and Electronics*, *32*(1), 209–219. https://doi.org/10.23919/JSEE.2021.000018

Mujumdar, A., & Vaidehi, V. (2019). Diabetes Prediction Using Machine Learning Algorithms. *Procedia Computer Science*, *165*, 292–299. https://doi.org/10.1016/j.procs.2020.01.047

Muljono, Wulandari, S. A., Azies, H. Al, Naufal, M., Prasetyanto, W. A., & Zahra, F. A. (2024). Breaking Boundaries in Diagnosis: Non-Invasive Anemia Detection Empowered by AI. *IEEE Access*, *12*(2023), 9292–9307. https://doi.org/10.1109/ACCESS.2024.3353788

Ogurtsova, K., da Rocha Fernandes, J. D., Huang, Y., Linnenkamp, U., Guariguata, L., Cho, N. H., Cavan, D., Shaw, J. E., & Makaroff, L. E. (2017). IDF Diabetes Atlas: Global Estimates for the Prevalence of Diabetes for 2015 and 2040. *Diabetes Research and Clinical Practice*, *128*, 40–50. https://doi.org/10.1016/j.diabres.2017.03.024

Rif'at, I. D., Hasneli N, Y., & Indriati, G. (2023). Gambaran Komplikasi Diabetes Melitus pada Penderita Diabetes Melitus. *Jurnal Keperawatan Profesional*, *11*(1), 52–69. https://doi.org/10.33650/jkp.v11i1.5540

Sari, L., Romadloni, A., Lityaningrum, R., & Hastuti, H. D. (2023). Implementation of LightGBM and Random Forest in Potential Customer Classification. *TIERS Information Technology Journal*, *4*(1), 43–55. https://doi.org/10.38043/tiers.v4i1.4355

Saxena, R., Sharma, S. K., Gupta, M., & Sampada, G. C. (2022). A Novel Approach for Feature Selection and Classification of Diabetes Mellitus: Machine Learning Methods.

Computational Intelligence and Neuroscience, *2022*(2), 1–11. https://doi.org/10.1155/2022/3820360

Sepbriant, G. D., & Utomo, D. W. (2024). Ensemble Learning pada Kategorisasi Produk E-Commerce Menggunakan Teknik Boosting. *JISKA (Jurnal Informatika Sunan Kalijaga)*, *9*(2), 123–133. https://doi.org/10.14421/jiska.2024.9.2.123-133

Tanwar, A., & Bhatia, P. K. (2024). A Review on Diabetes Prediction Using Machine Learning Techniques. In *Lecture Notes in Electrical Engineering* (Vol. 1185, Number 09, pp. 513–524). https://doi.org/10.1007/978-981-97-1682-1_41

Thohari, A. N. A., Karima, A., Santoso, K., & Rahmawati, R. (2024). Crack Detection in Building Through Deep Learning Feature Extraction and Machine Learning Approch. *Journal of Applied Informatics and Computing*, *8*(1), 1–6. https://doi.org/10.30871/jaic.v8i1.7431

Wang, Z., Wu, C., Zheng, K., Niu, X., & Wang, X. (2019). SMOTETomek-Based Resampling for Personality Recognition. *IEEE Access*, *7*, 129678–129689. https://doi.org/10.1109/ACCESS.2019.2940061

Zhang, H., Liu, C., Zhang, Z., Xing, Y., Liu, X., Dong, R., He, Y., Xia, L., & Liu, F. (2021). Recurrence Plot-Based Approach for Cardiac Arrhythmia Classification Using Inception-ResNet-v2. *Frontiers in Physiology*, *12*, 1–13. https://doi.org/10.3389/fphys.2021.648950

JISKA (Jurnal Informatika Sunan Kalijaga)
Vol. 11, No. 1, JANUARI, 2026, Pp. 13 – 31
ISSN: 2527 – 5836 (print) | 2528 – 0074 (online)

Evaluasi Keamanan OTP Firebase pada Aplikasi Android: Perbandingan SAST dan IAST dalam Identifikasi Kerentanan

I Gede Surya Rahayuda [(1)*], Ni Putu Linda Santiari [(2)]
[1] Departemen Informatika, Universitas Udayana, Bali, Indonesia
[2] Departemen Sistem Informasi, ITB STIKOM Bali, Bali, Indonesia
e-mail : igedesuryarahayuda@unud.ac.id, linda_santiari@stikom-bali.ac.id.
* Penulis korespondensi.
Artikel ini diajukan 28 Desember 2024, direvisi 24 Maret 2025, diterima 25 Maret 2025, dan dipublikasikan 25 Januari 2025.

Abstract

Application security is crucial for protecting user data from cyber threats, particularly in Android applications that utilize One-Time Password (OTP)-based authentication. This study evaluates the security of Firebase OTP via email using a combination of Static Application Security Testing (SAST) with Mobile Security Framework (MobSF) and Interactive Application Security Testing (IAST) with AppSweep. The results show that the combination of SAST and IAST is superior to single testing methods due to its wider detection coverage. SAST detects vulnerabilities in static code, while IAST identifies exploits in runtime. The testing showed significant improvements, with high-severity vulnerabilities decreasing from 3 cases in OTP-1 to zero in OTP-5, and the security score increasing from 43 (B) to 78 (A) in MobSF. Meanwhile, the number of vulnerabilities in AppSweep decreased from 14 to 9, with all high-severity vulnerabilities resolved. However, this study still has limitations, such as limited dataset coverage and potential bias from the testing tool. For further improvement, additional research can integrate artificial intelligence to automate vulnerability detection, as well as explore biometric-based authentication to enhance system security even further.

Keywords: Firebase OTP, MobS, AppSweep, SAST, IAST

Abstrak

Keamanan aplikasi sangat penting untuk melindungi data pengguna dari ancaman siber, terutama dalam aplikasi Android yang menggunakan autentikasi berbasis One-Time Password (OTP). Penelitian ini mengevaluasi keamanan OTP Firebase melalui email menggunakan kombinasi *Static Application Security Testing* (SAST) dengan Mobile Security Framework (MobSF) dan *Interactive Application Security Testing* (IAST) dengan AppSweep. Hasil penelitian menunjukkan bahwa kombinasi SAST dan IAST lebih unggul dibandingkan metode pengujian tunggal karena cakupan deteksi yang lebih luas. SAST mendeteksi kelemahan dalam kode statis, sementara IAST mengidentifikasi eksploitasi dalam runtime. Pengujian menunjukkan perbaikan signifikan, di mana kerentanan tingkat tinggi berkurang dari 3 kasus pada OTP-1 menjadi nol pada OTP-5, dengan skor keamanan meningkat dari 43 (B) menjadi 78 (A) dalam MobSF. Sementara itu, jumlah kerentanan dalam AppSweep menurun dari 14 menjadi 9, dengan semua kerentanan tingkat tinggi terselesaikan. Namun, penelitian ini masih memiliki keterbatasan, seperti cakupan dataset yang terbatas dan potensi bias dari alat pengujian. Untuk perbaikan lebih lanjut, penelitian selanjutnya dapat mengintegrasikan kecerdasan buatan untuk otomatisasi deteksi kerentanan serta mengeksplorasi autentikasi berbasis biometrik guna meningkatkan keamanan sistem lebih lanjut.

Kata Kunci: OTP Firebase, MobSF, AppSweep, SAST, IAST

1. PENDAHULUAN

Keamanan aplikasi telah menjadi tantangan krusial dalam pengembangan perangkat lunak, terutama untuk aplikasi yang menangani data sensitif pengguna (Moon et al., 2023). Ancaman seperti pencurian data, akses tidak sah, dan manipulasi sistem terus meningkat di era digital yang semakin terhubung ini. Oleh karena itu, mekanisme autentikasi yang kuat sangat diperlukan untuk melindungi privasi pengguna. Salah satu metode yang banyak digunakan adalah OTP (One

Time Password), yang berfungsi untuk memverifikasi identitas pengguna dalam transaksi penting atau proses login.

Meskipun OTP dapat diimplementasikan melalui berbagai saluran, seperti SMS atau aplikasi pihak ketiga, OTP berbasis email yang menggunakan Firebase menawarkan kemudahan integrasi dan skalabilitas yang menarik untuk aplikasi Android (Asih & Hasibuan, 2023). Namun, implementasi OTP ini tidak bebas dari kerentanannya, baik pada sisi kode aplikasi, komunikasi jaringan, maupun konfigurasi sistem yang kurang optimal. Meskipun Firebase menyediakan platform yang kuat, efektivitas OTP sangat bergantung pada cara implementasi dan pengelolaan kode oleh pengembang.

Keamanan produk perangkat lunak secara umum juga menjadi perhatian penting, mencakup berbagai aspek seperti kerahasiaan, integritas, dan ketersediaan (Stanciu, 2023). Dalam penelitian ini, evaluasi terhadap OTP Firebase dilakukan dengan mempertimbangkan perbedaan mendasar dibandingkan metode OTP lainnya, seperti OTP berbasis SMS dan OTP melalui aplikasi pihak ketiga. Salah satu alasan utama adalah aspek keamanan, di mana OTP Firebase menawarkan integrasi yang lebih erat dengan ekosistem Google, tetapi masih memiliki potensi celah keamanan yang perlu dieksplorasi lebih dalam.

Studi ini menyoroti kelemahan dan keunggulan masing-masing metode melalui analisis keamanan yang lebih komprehensif, termasuk potensi serangan man-in-the-middle, phishing, dan eksploitasi API. Beberapa penelitian sebelumnya telah mengevaluasi keamanan OTP dalam berbagai konteks, seperti penggunaan algoritma Speck untuk enkripsi OTP guna mencegah serangan sniffing dan intercept nirkabel (Taqwim et al., 2021), serta optimasi autentikasi OTP menggunakan algoritma Random Forest untuk mendeteksi dan mencegah penipuan (Lubis & Riswanto, 2024). Selain itu, pendekatan OTP berbasis Time-Based One-Time Password (TOTP) juga telah diterapkan untuk meningkatkan perlindungan data digital (Wibawa et al., 2024).

Mengacu pada studi-studi tersebut, penelitian ini mengadopsi kombinasi metode *Static Application Security Testing* (SAST) dan *Interactive Application Security Testing* (IAST), karena keduanya saling melengkapi dalam mendeteksi celah keamanan secara lebih menyeluruh, dibandingkan hanya mengandalkan salah satu metode. Dengan pendekatan tersebut, penelitian ini memberikan kontribusi signifikan dalam menilai keamanan OTP Firebase secara lebih mendalam dibandingkan studi sebelumnya. Penelitian ini difokuskan pada implementasi OTP berbasis email dengan Firebase dalam aplikasi Android menggunakan Kotlin sebagai bahasa pemrograman. Kotlin menawarkan berbagai fitur yang meningkatkan keamanan aplikasi, seperti penanganan null yang eksplisit dan dukungan untuk coroutine. Meskipun demikian, implementasi OTP berbasis Firebase masih menghadapi beberapa masalah kerentanannya, yang dapat mempengaruhi keamanan aplikasi secara keseluruhan. Oleh karena itu, pengujian keamanan dengan pendekatan yang tepat menjadi sangat penting untuk memastikan bahwa aplikasi aman digunakan oleh pengguna.

Dalam konteks keamanan aplikasi seluler, berbagai standar dan teknik telah dikembangkan untuk memastikan keamanan dan privasi pengguna. OWASP (Open Worldwide Application Security Project) telah menetapkan standar industri seperti MASVS (Mobile Application Security Verification Standard) (Schleier, Holguera, Mueller, Willemsen, et al., 2024) dan MASTG (Mobile Application Security Testing Guide) (Schleier, Holguera, Mueller, & Willemsen, 2024), yang memberikan panduan komprehensif untuk pengujian dan pengembangan aplikasi seluler yang aman. Selain itu, penelitian tentang aplikasi seluler berbasis cloud (Chimuco et al., 2023) menyoroti pentingnya pemahaman taksonomi serangan, mekanisme keamanan, dan spesifikasi pengujian keamanan untuk ekosistem ini.

Metode autentikasi yang aman juga menjadi fokus utama dalam penelitian. Studi tentang sistem autentikasi perbankan online (Danthy et al., 2024) mengusulkan penggunaan Time-Bound Password untuk meningkatkan keamanan transaksi keuangan. Namun, penelitian lain (Ikram et al., 2025) mengungkapkan risiko keamanan dan privasi yang terkait dengan aplikasi autentikator

OTP seluler, menyoroti perlunya jaminan keamanan dan privasi yang lebih baik dalam aplikasi tersebut.

Deteksi dan pencegahan malware pada perangkat seluler Android juga merupakan area penelitian yang penting (Keteku et al., 2024). Berbagai teknik dan alat telah dikembangkan untuk mengidentifikasi dan mencegah malware, termasuk analisis statis dan dinamis. Penelitian tentang alat SAST (Static Application Security Testing) (Li et al., 2023; Smyčka, 2024) menunjukkan pentingnya evaluasi dan penerapan alat SAST yang efektif untuk mengidentifikasi kerentanan dalam kode sumber. Selain itu, platform analisis statis (Sonnekalb et al., 2023) dapat digunakan untuk memonitor tren keamanan dalam repositori dan mengidentifikasi potensi kerentanan.

Analisis statis ini penting karena kerentanan dapat muncul dari berbagai faktor, termasuk kelemahan pada kode sumber (Stanciu, 2023). Meskipun telah dilakukan penelitian bertahun-tahun, deteksi otomatis kerentanan dalam aplikasi seluler tetap menjadi tantangan. Alat SAST dapat mengidentifikasi kelemahan pada kode sumber atau kode yang dikompilasi tanpa menjalankan aplikasi, namun memiliki keterbatasan seperti deteksi kerentanan yang terbatas, kurangnya pembaruan, dan ketahanan terbatas terhadap teknik pengaburan (Pagano et al., 2023).

Pengujian keamanan aplikasi seluler juga melibatkan pendekatan dinamis (Sutter et al., 2024), yang memungkinkan analisis perilaku aplikasi saat dijalankan. Tinjauan literatur sistematis tentang analisis keamanan dinamis di Android (Sutter et al., 2024) menyoroti pentingnya alat instrumentasi dan pemantauan jaringan dalam mengidentifikasi kerentanan. Selain itu, penelitian tentang pengujian keamanan otomatis untuk aplikasi seluler (Nutalapati, 2023) menekankan peran toko aplikasi dalam memastikan keamanan dan kualitas aplikasi yang diinstal oleh pengguna.

Dalam beberapa tahun terakhir, penelitian tentang keamanan aplikasi seluler juga mencakup studi tentang operasi tersembunyi dan pencurian informasi pribadi (Lim et al., 2024). Studi ini menyoroti potensi penyalahgunaan fitur aksesibilitas Android dan perlunya langkah-langkah keamanan yang komprehensif untuk melawan eksploitasi tersebut. Selain itu, penelitian tentang kemajuan dalam pengujian keamanan (Pargaonkar, 2023) mencakup berbagai metodologi dan tren yang muncul, seperti integrasi DevSecOps dan pengujian keamanan berkelanjutan.

Terakhir, penelitian tentang evaluasi dan penerapan alat SAST (Smyčka, 2024) memberikan panduan sistematis untuk menerapkan alat SAST dalam praktik pengembangan perangkat lunak. Selain itu, penelitian tentang platform analisis statis (Sonnekalb et al., 2023) menawarkan solusi untuk memantau peringatan keamanan di seluruh riwayat versi repositori. Selain metode SAST, penelitian ini juga akan menggunakan metode IAST (Pargaonkar, 2023). Alat yang digunakan untuk SAST adalah Mobile-Security-Framework (MobSF), sementara untuk IAST menggunakan AppSweep. Dari berbagai alat yang tersedia untuk SAST, DAST, dan IAST, beberapa di antaranya meliputi eslint, mypy, trivy, trufflehog, pylint, super-linter, checkov, checkstyle, moose, semgrep, pmd, pypi-auto-scanner, error-prone, kube-bench, molecule, ansible-lint, git-secrets, clair, biome, kics, black, shellcheck, ruff, gitleaks, infer, SonarQube, CodeQL, Contrast, Horusec, Insider, SBwFSB, Toller, Ares, Q-Testing, Chimp, Monkey, ACVTool, TimeMachine, WCTester, Sapienz, Stoat, dan AndroTest (Li et al., 2023; Pargaonkar, 2023; Schleier, Holguera, Mueller, Willemsen, et al., 2024; Smyčka, 2024).

Pemilihan MobSF untuk SAST didasarkan pada keunggulannya sebagai alat open-source yang secara khusus dirancang untuk aplikasi mobile dan telah terbukti efektif serta terpercaya. Sementara itu, pemilihan AppSweep untuk IAST dipilih karena masih sangat sedikit sumber yang menyebutkan penggunaan alat ini untuk IAST, serta metode ini belum banyak diterapkan, terutama dalam konteks aplikasi mobile dan pemantauan kerentanannya secara real-time. Selain itu, AppSweep juga memiliki versi gratis yang mempermudah penerapannya dalam penelitian ini.

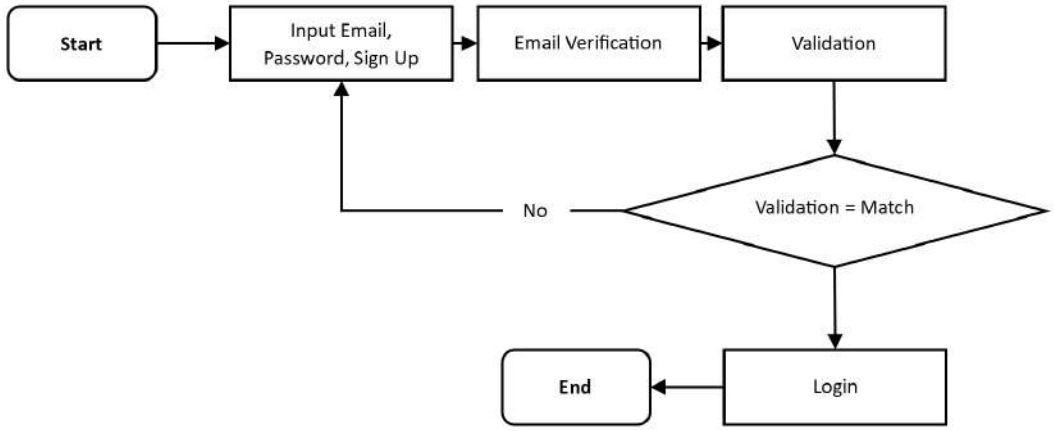
2. METODE PENELITIAN

Penelitian ini mengevaluasi keamanan implementasi OTP Firebase berbasis email pada aplikasi Android dengan menggunakan metode pengujian keamanan Static Application Security Testing (SAST) dan Interactive Application Security Testing (IAST). Pendekatan ini mencakup tiga aspek utama: implementasi OTP Firebase, pengujian SAST menggunakan Mobile Security Framework (MobSF), dan pengujian IAST menggunakan AppSweep.

2.1 OTP Firebase

Implementasi OTP berbasis email menggunakan Firebase Authentication memanfaatkan library Firebase untuk pengelolaan autentikasi. Firebase Authentication menyediakan antarmuka yang memungkinkan pengembang mengintegrasikan berbagai metode autentikasi, termasuk OTP, secara sederhana dan aman. Library dan modul yang digunakan adalah Firebase Authentication SDK yang digunakan untuk pengiriman dan validasi OTP dan Email Integration untuk mengirimkan kode OTP melalui email ke pengguna menggunakan layanan Firebase (Ikram et al., 2025; Lim et al., 2024).

Gambar 1 menunjukkan alur proses OTP, yang terdiri dari tiga tahap utama. Pertama, pada tahap permintaan OTP, aplikasi memanggil API Firebase untuk menghasilkan kode OTP dan mengirimkannya ke alamat email pengguna. Kedua, pada tahap validasi OTP, pengguna memasukkan kode OTP yang diterima, kemudian aplikasi melakukan verifikasi dengan Firebase untuk memastikan keabsahan kode tersebut. Terakhir, pada tahap manajemen token, setelah OTP berhasil diverifikasi, Firebase menghasilkan token autentikasi yang digunakan untuk menginisiasi sesi login pengguna.



Gambar 1 Alur OTP

2.2 SAST (Static Application Security Testing)

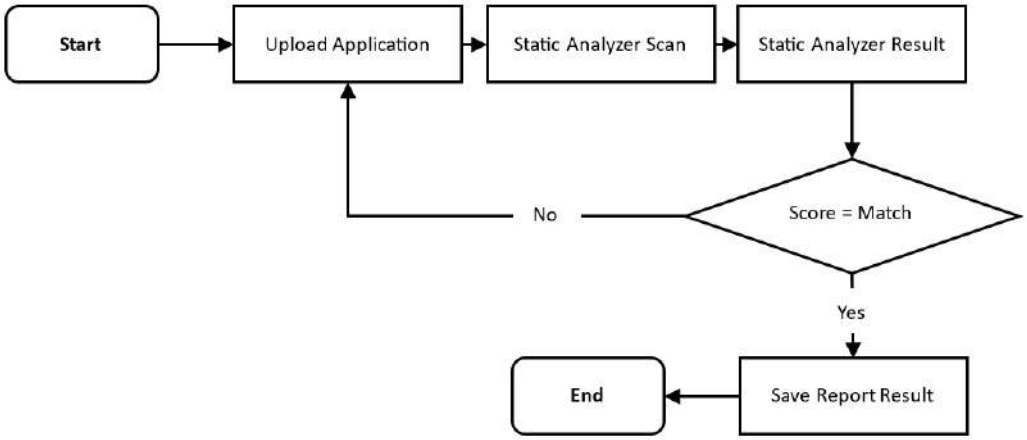
Static Application Security Testing (SAST) adalah metode pengujian keamanan yang menganalisis kode sumber aplikasi tanpa perlu menjalankan aplikasi tersebut. Pengujian ini biasanya dilakukan pada tahap awal pengembangan untuk mendeteksi kerentanan seperti penggunaan API yang tidak aman, *hardcoded credentials*, atau algoritma enkripsi yang lemah. Beberapa jenis SAST antara lain: Code Review Tools, yang mengidentifikasi masalah berdasarkan aturan tertentu seperti yang dilakukan oleh SonarQube; Binary Analysis Tools, yang menganalisis file biner seperti APK untuk menemukan kerentanan; serta Framework Analysis Tools, yang fokus pada analisis kerangka kerja yang digunakan, seperti Android SDK atau Firebase.

Dalam penelitian ini, Mobile Security Framework (MobSF) dipilih sebagai alat utama untuk pengujian SAST karena beberapa alasan. Pertama, MobSF mendukung analisis baik untuk file

APK (Android) maupun IPA (iOS), sehingga mampu menganalisis kode sumber dan juga file biner aplikasi. Kedua, MobSF memiliki fitur deteksi konfigurasi Firebase yang lemah, termasuk API Key yang terekspos atau akses database yang tidak aman. Ketiga, alat ini dapat diintegrasikan dengan proses CI/CD, memungkinkan otomatisasi pengujian keamanan dalam pipeline pengembangan. Keempat, sebagai solusi open-source yang andal, MobSF menjadi alternatif yang kuat dibandingkan alat berbayar seperti Checkmarx, Veracode, atau SonarQube (Nutalapati, 2023; Smyčka, 2024).

Gambar 2 memperlihatkan alur pengujian dengan MobSF, yang terdiri dari tiga tahap utama. Pertama, file APK diunggah ke antarmuka MobSF untuk dianalisis. Kedua, dilakukan analisis statis untuk menghasilkan laporan keamanan. Ketiga, hasil laporan dianalisis untuk mengidentifikasi kelemahan dan rekomendasi perbaikan yang diberikan oleh MobSF.

Sebagai alat pengujian keamanan aplikasi seluler, MobSF menawarkan berbagai fitur, termasuk Static Analysis yang menganalisis APK tanpa menjalankannya, Dynamic Analysis untuk menguji aplikasi pada emulator, dan Code Analysis untuk mendeteksi kelemahan dalam kode sumber aplikasi. Selain itu, MobSF juga melakukan pemindaian terhadap potensi kerentanan, seperti *hardcoded credentials* (misalnya API key atau password), komunikasi jaringan yang tidak aman (HTTP tanpa SSL), penggunaan *library* pihak ketiga yang rentan, serta kebijakan izin aplikasi yang terlalu luas. Kategori kerentanan yang dievaluasi oleh MobSF dapat dilihat pada Tabel 1, yang mengklasifikasikan tingkat keparahan kerentanan mulai dari *High*, *Medium*, hingga *Low*.



Gambar 2 Alur SAST dan MobSF

Tabel 1 Kategori Kerentanan MobSF

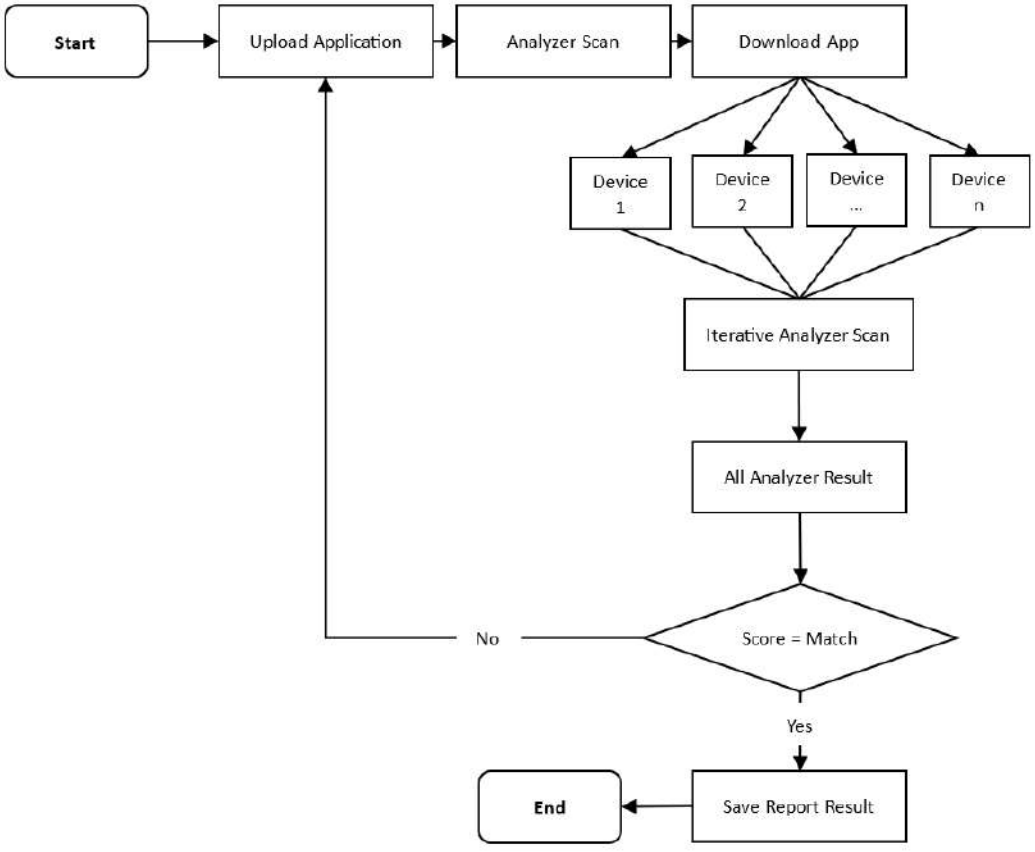
Tingkat Keparahan	Deskripsi
High	Kerentanan dengan tingkat risiko sangat tinggi yang dapat membahayakan aplikasi dan pengguna, sehingga perlu segera diperbaiki. Semakin tinggi tingkat risikonya, semakin besar potensi dampaknya.
Warning	Potensi risiko keamanan yang dapat menimbulkan masalah, namun tidak sekrusial tingkat High. Tetap perlu diperbaiki untuk mencegah dampak lebih lanjut. Semakin tinggi tingkat risikonya, semakin besar potensi dampaknya.
Info	Masalah minor yang tidak secara langsung mengancam keamanan, tetapi dapat memengaruhi kualitas aplikasi atau pengalaman pengguna. Semakin tinggi tingkat risikonya, semakin besar potensi dampaknya.
Secure	Hasil pengujian menunjukkan bahwa aplikasi memiliki perlindungan yang memadai dan tidak ditemukan kerentanan keamanan. Semakin tinggi nilainya, semakin baik tingkat keamanannya.

2.3 IAST (Interactive Application Security Testing)

Interactive Application Security Testing (IAST) adalah metode pengujian keamanan yang menggabungkan elemen dari pengujian statis (SAST) dan dinamis (DAST). Pengujian ini dilakukan saat aplikasi berjalan untuk memberikan wawasan mengenai bagaimana aplikasi merespons serangan keamanan secara langsung. Dalam penelitian ini, AppSweep dipilih sebagai alat utama karena keunggulannya dalam mendeteksi kerentanan aplikasi Android secara interaktif dan kontekstual. Alur proses pengujian IAST menggunakan AppSweep digambarkan pada Gambar 3.

AppSweep memiliki beberapa keunggulan dibandingkan alat IAST lainnya. Pertama, alat ini memiliki integrasi yang kuat dengan standar keamanan OWASP Mobile Top 10, memastikan bahwa pengujian berfokus pada kerentanan paling kritis dalam aplikasi seluler. Kedua, AppSweep melakukan analisis berbasis IAST yang memungkinkan deteksi kerentanan secara runtime, memberikan hasil yang lebih akurat dibandingkan metode statis (SAST) yang hanya menganalisis kode sumber tanpa menjalankannya (Li et al., 2023; Pargaonkar, 2023). Selain itu, AppSweep menyediakan automated security testing yang dapat dengan mudah diintegrasikan ke dalam alur kerja pengembangan (CI/CD pipeline) dan menghasilkan laporan yang komprehensif serta mudah dipahami.

AppSweep juga menjadi pilihan yang ekonomis karena menyediakan versi gratis, memungkinkan pengembang melakukan pengujian tanpa biaya tambahan. Dibandingkan dengan alat lain seperti Veracode IAST dan Contrast Security, AppSweep lebih fokus pada aplikasi mobile (Android), sedangkan dua alat lainnya umumnya digunakan untuk aplikasi web dan backend. Dengan demikian, AppSweep menjadi solusi yang fleksibel dan hemat biaya untuk mengamankan aplikasi Android, terutama pada tahap awal dan menengah pengembangan.



Gambar 3 Alur IAST Menggunakan Appsweep

Alur pengujian dengan AppSweep dijelaskan sebagai berikut:
- Unggah file APK ke AppSweep melalui antarmuka web atau plugin Android Studio.
- Instal dan jalankan aplikasi pada emulator atau perangkat nyata.
- Analisis laporan keamanan berdasarkan data yang dikumpulkan selama pengujian.

Fitur utama AppSweep meliputi pengujian pada emulator atau perangkat nyata, laporan berbasis OWASP Mobile Top 10, serta analisis izin aplikasi, penggunaan API, dan respons terhadap potensi serangan. Jenis-jenis kerentanan yang dianalisis AppSweep dirangkum pada Tabel 2 Kategori Kerentanan AppSweep, yang mencakup isu seperti data leakage, penyimpanan data tidak aman, ketidakamanan komunikasi jaringan, dan akses tidak sah ke API atau database.

Tabel 2 Kategori Kerentanan AppSweep

Tingkat Keparahan	Deskripsi
High	Masalah kritikal yang dapat membahayakan aplikasi dan data pengguna. Harus segera diperbaiki untuk mencegah potensi eksploitasi. Semakin tinggi tingkat risikonya, semakin berbahaya.
Medium	Kerentanan yang meningkatkan risiko keamanan aplikasi, tetapi tidak sebesar tingkat High. Perlu diperbaiki untuk mengurangi kemungkinan ancaman. Semakin tinggi tingkat risikonya, semakin berbahaya.
Low	Masalah dengan dampak kecil yang tidak langsung membahayakan, tetapi dapat memengaruhi kualitas atau kinerja aplikasi. Sebaiknya tetap diperbaiki untuk meningkatkan keamanan secara keseluruhan. Semakin tinggi tingkat risikonya, semakin berbahaya.
Issues	Jumlah total kerentanan yang terdeteksi, mencakup kategori High, Medium, dan Low.
OWASP	Klasifikasi masalah keamanan berdasarkan standar OWASP MASVS dan OWASP MASTG (Schleier, Holguera, Mueller, & Willemsen, 2024; Schleier, Holguera, Mueller, Willemsen, et al., 2024).
Libraries	Daftar kerentanan yang ditemukan dalam pustaka (libraries) yang digunakan dalam aplikasi. Merupakan penjabaran lebih detail dari issues.

Eksperimen dilakukan menggunakan dua emulator pada dua perangkat berbeda dengan spesifikasi sebagai berikut:

Perangkat 1 (PC - Pengembangan Kode):
- Prosesor Intel Core i3 generasi ke-10, RAM 8GB
- Sistem Operasi: Windows 11
- Android Emulator: Pixel 3a API 30 (Android 11)

Perangkat 2 (Laptop - Pengujian Keamanan):
- Prosesor Intel Core i3 generasi ke-12, RAM 8GB
- Sistem Operasi: Windows 11
- Android Emulator: Pixel 3a API 30 (Android 11)
- Pengujian dilakukan dari lokasi geografis berbeda menggunakan AppSweep

Dalam analisis keamanan aplikasi, terdapat kemungkinan terjadinya **false positives** (kerentanan terdeteksi padahal tidak ada) dan **false negatives** (kerentanan tidak terdeteksi). Untuk meminimalkan hal ini, dilakukan beberapa langkah:
- **Cross-validation:** hasil dari MobSF (SAST) dan AppSweep (IAST) dibandingkan untuk mendeteksi ketidaksesuaian hasil.
- **Manual review:** verifikasi manual terhadap hasil deteksi untuk memastikan akurasi laporan.
- **Re-run testing:** pengujian diulang beberapa kali dengan versi APK yang berbeda untuk memeriksa konsistensi hasil.

Perbandingan antara MobSF dan AppSweep dalam mendeteksi kerentanan dirangkum pada Tabel 3, yang menunjukkan bahwa kedua alat memiliki karakteristik pelengkap — MobSF unggul dalam mendeteksi kelemahan pada konfigurasi dan kode sumber, sedangkan AppSweep lebih efektif dalam mendeteksi kerentanan yang muncul selama runtime.

Tabel 3 Perbandingan MobSF dan AppSweep dalam Deteksi Kerentanan

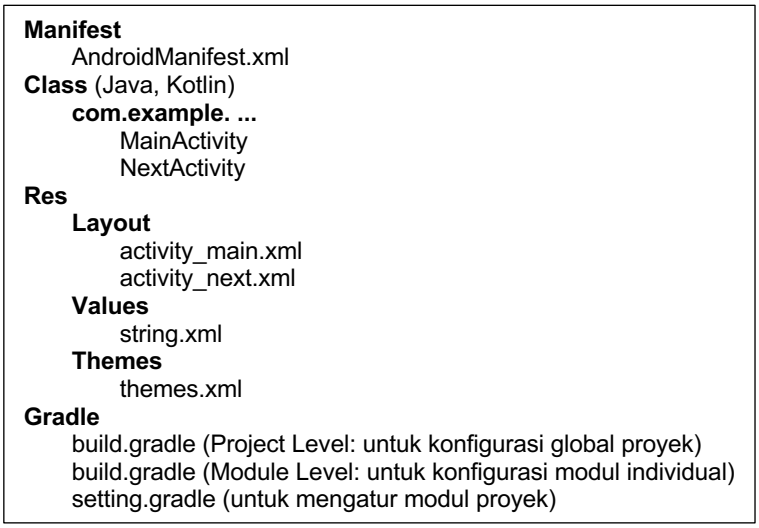
Jenis Kerentanan	MobSF (SAST)	AppSweep (IAST)
Hardcoded Credentials	Ya	Tidak
Insecure API Usage	Ya	Ya
Improper Cryptography	Ya	Tidak
Data Leakage	Tidak	Ya
Insecure Storage	Tidak	Ya
Insecure Communication	Ya	Ya
Runtime Exploitation	Tidak	Ya

3. HASIL DAN PEMBAHASAN

Pada bagian ini, akan dibahas secara rinci hasil implementasi dan pengujian keamanan dari aplikasi OTP Firebase berbasis email. Pembahasan mencakup tiga aspek utama, yaitu: (1) implementasi fitur OTP menggunakan Firebase Authentication, (2) pengujian keamanan aplikasi secara statis (SAST) menggunakan Mobile Security Framework (MobSF), dan (3) pengujian keamanan secara interaktif (IAST) menggunakan AppSweep. Setiap bagian dijelaskan secara sistematis dengan menampilkan potongan kode program, gambar tangkapan layar dari lingkungan pengembangan dan hasil pengujian, serta laporan analisis keamanan yang dihasilkan oleh masing-masing alat. Melalui pendekatan ini, diharapkan dapat memberikan pemahaman menyeluruh mengenai bagaimana setiap metode pengujian bekerja, jenis kerentanan yang terdeteksi, serta rekomendasi perbaikan yang dihasilkan dari proses analisis.

3.1 OTP Firebase

Pengujian implementasi OTP Firebase dilakukan dengan menggunakan Firebase Authentication untuk mengirimkan OTP melalui email dan memverifikasi kode yang diterima. Proses ini memerlukan pemahaman tentang bagaimana Firebase menangani autentikasi pengguna dan bagaimana kode implementasi dapat berfungsi dalam aplikasi Android. Untuk mengimplementasikan OTP menggunakan Firebase, struktur file aplikasi Android, yang ditunjukkan pada Gambar 4, terdiri dari beberapa komponen yang berfungsi untuk mengirimkan dan memverifikasi OTP.



Gambar 4 Struktur File Aplikasi Android

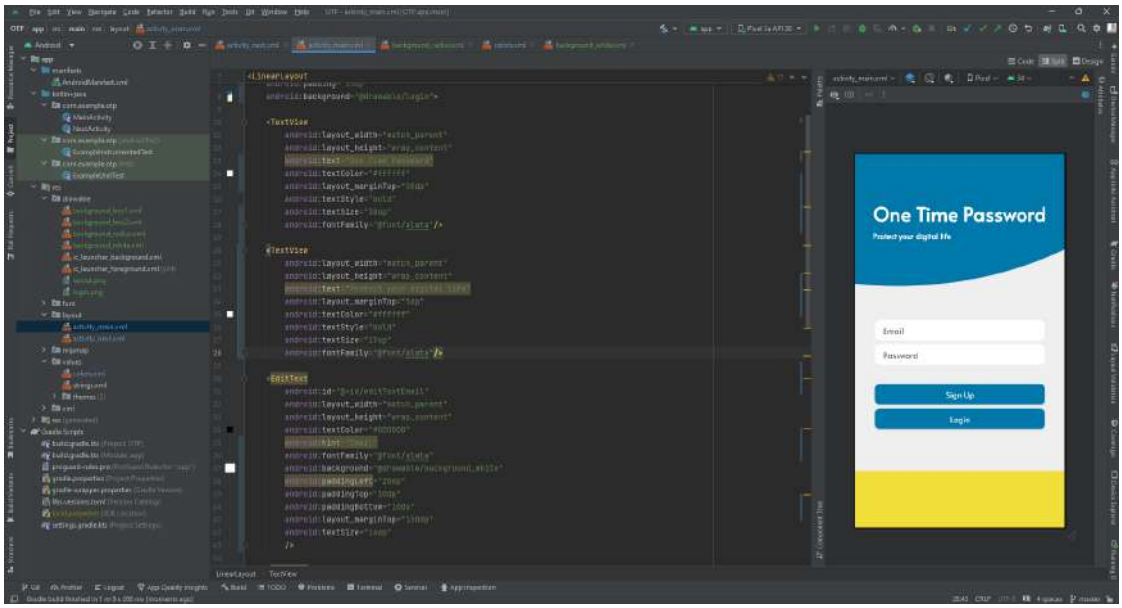
JISKA (Jurnal Informatika Sunan Kalijaga)
Vol. 11, No. 1, JANUARI, 2026: 13 – 31

Pada Gambar 5 menunjukkan potongan kode untuk mengirim dan memverifikasi OTP melalui email. Kode Tersebut merupakan bagian dari aplikasi Android yang dituliskan menggunakan Kotlin dan dapat dilihat pada repositori berikut: https://github.com/rahayuda/OTP-Evaluation.git. Pada potongan kode tersebut menggunakan dua fungsi utama, yaitu sendOTP dan verifyOTP. Fungsi sendOTP berfungsi untuk mengirimkan OTP ke alamat email pengguna melalui layanan FirebaseAuth, sedangakan fungsi verifyOTP digunakan untuk memverifikasi OTP yang dimasukkan oleh pengguna untuk proses autentikasi. Gambar 6 menunjukkan tampilan dashboard pada Android Studio yang digunakan sebagai lingkungan pengembangan aplikasi ini.

```
// Mengirim OTP melalui email
fun sendOTP(email: String) {
    FirebaseAuth.getInstance().sendSignInLinkToEmail(email, actionCodeSettings)
        .addOnCompleteListener { task ->
            if (task.isSuccessful) {
                println("OTP dikirim ke $email")
            } else {
                println("Gagal mengirim OTP: ${task.exception?.message}")
            }
        }
}

// Memverifikasi OTP
fun verifyOTP(email: String, otp: String) {
    FirebaseAuth.getInstance().signInWithEmailLink(email, otp)
        .addOnCompleteListener { task ->
            if (task.isSuccessful) {
                println("Login berhasil!")
            } else {
                println("Validasi OTP gagal: ${task.exception?.message}")
            }
        }
}
```

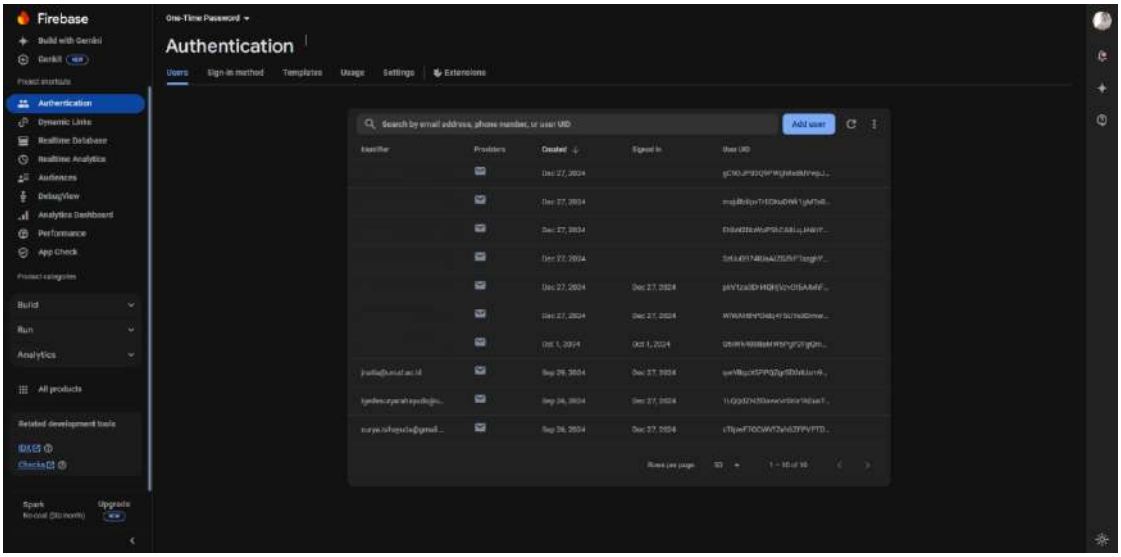
Gambar 5 Kode Pemrograman Pengiriman OTP



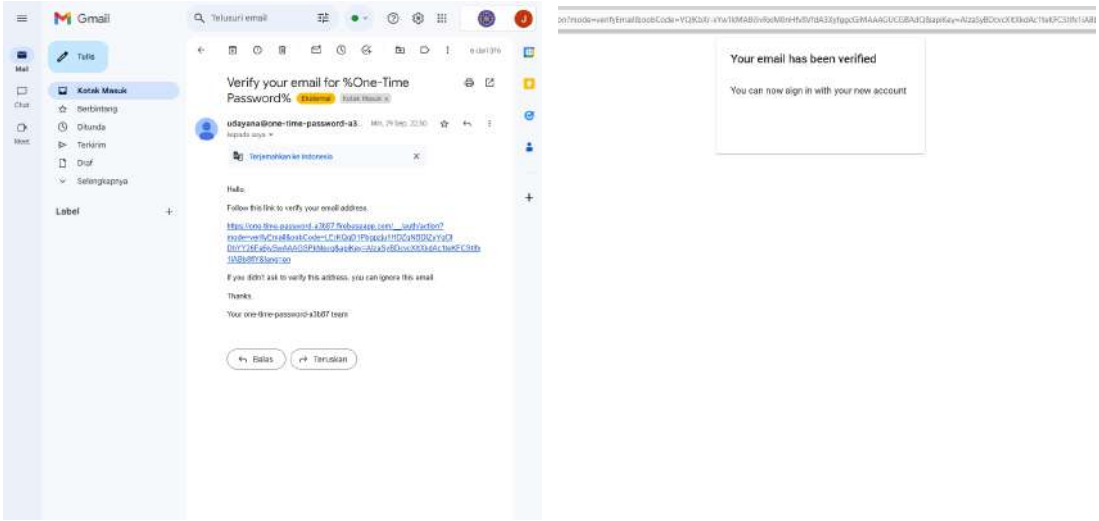
Gambar 6 Android Studio Dashboard

Untuk memverifikasi implementasi OTP, dilakukan konfigurasi di halaman Firebase Authentication. Firebase menyediakan antarmuka yang memungkinkan pengembang melihat status autentikasi pengguna, termasuk pengguna yang telah berhasil melakukan verifikasi email. Gambar 7 menunjukkan tangkapan layar dari halaman Firebase Authentication, yang menampilkan daftar pengguna yang terdaftar dan telah terverifikasi.

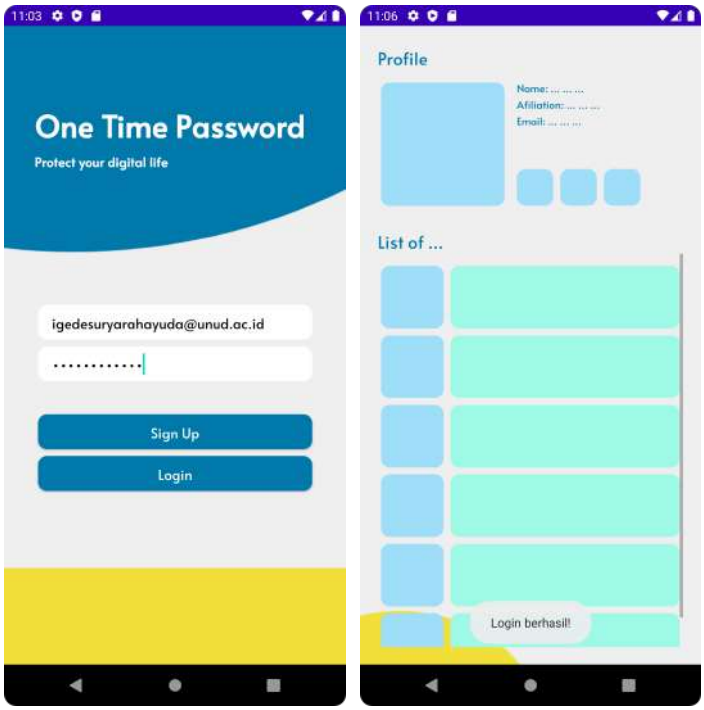
Jika akun pengguna belum terdaftar, pengguna tidak dapat melakukan login. Proses pendaftaran dilakukan dengan memasukkan email yang valid, setelah itu Firebase mengirimkan email verifikasi kepada pengguna. Pengguna kemudian harus memverifikasi email tersebut untuk menyelesaikan pendaftaran. Gambar 8 menampilkan tangkapan layar email verifikasi yang dikirim serta status verifikasi yang berhasil dilakukan. Setelah terdaftar dan berhasil memverifikasi email, pengguna dapat melakukan login dan diarahkan ke halaman profil. Sebaliknya, jika akun belum terdaftar atau email belum diverifikasi, proses login akan gagal. Gambar 9 menunjukkan tangkapan layer halaman login dan halaman profil pengguna setelah berhasil login.



Gambar 7 Firebase Authentication



Gambar 8 Verifikasi Email



Gambar 9 Halaman Login dan Profil

3.2 SAST (Static Application Security Testing)

Analisis Keamanan Statis (SAST) dilakukan menggunakan Mobile Security Framework (MobSF) untuk menganalisis file APK aplikasi secara statis. MobSF memindai aplikasi sebelum dijalankan untuk mendeteksi kerentanan yang dapat dieksploitasi oleh pihak yang tidak bertanggung jawab. Pengujian ini sangat penting untuk mengidentifikasi potensi masalah keamanan pada tahap awal pengembangan, sehingga dapat diatasi sebelum aplikasi dirilis ke publik. Tabel 4 menunjukkan hasil pemindaian dari lima percobaan yang dilakukan pada aplikasi OTP, dengan berbagai tingkat kerentanannya yang terdeteksi oleh MobSF.

Tabel 4 Evaluasi OTP App dengan MobSF

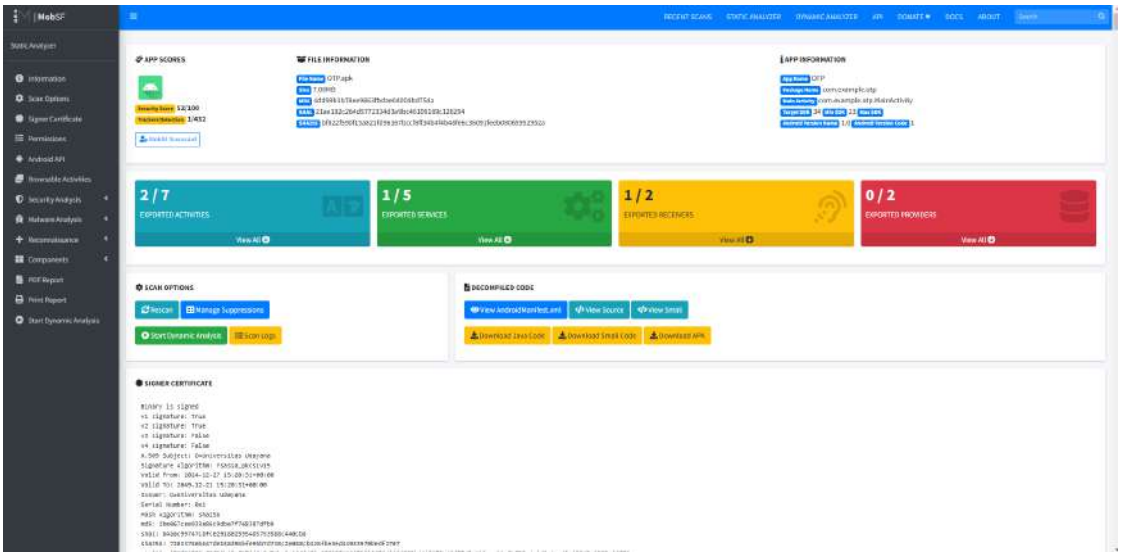
Evaluation	High	Warning	Info	Secure	Score	Grade	Report
OTP-1	3	8	1	1	43	B	OTP-1.pdf
OTP-2	1	8	1	1	52	B	OTP-2.pdf
OTP-3	0	6	1	1	60	A	OTP-3.pdf
OTP-4	0	3	1	1	67	A	OTP-4.pdf
OTP-5	0	3	1	2	78	A	OTP-5.pdf

Hasil pengujian menunjukkan adanya variasi kerentanan pada aplikasi OTP yang bervariasi, mulai dari tingkat tinggi (High) hingga kategori yang lebih ringan. Pada percobaan awal (OTP-1), ditemukan tiga temuan kategori "High", yang menunjukkan bahwa aplikasi masih sangat rentan pada tahap awal pengujian. Namun, setelah dilakukan rangkaian perbaikan, jumlah temuan kerentanan menurun secara signifikan pada percobaan berikutnya. Percobaan OTP-5 bahkan mencapai skor tertinggi yaitu 78 dengan peringkat "A", seperti ditunjukkan pada Gambar 10 (MobSF Score pada OTP-2) dan Gambar 11 (MobSF Score pada OTP-5).

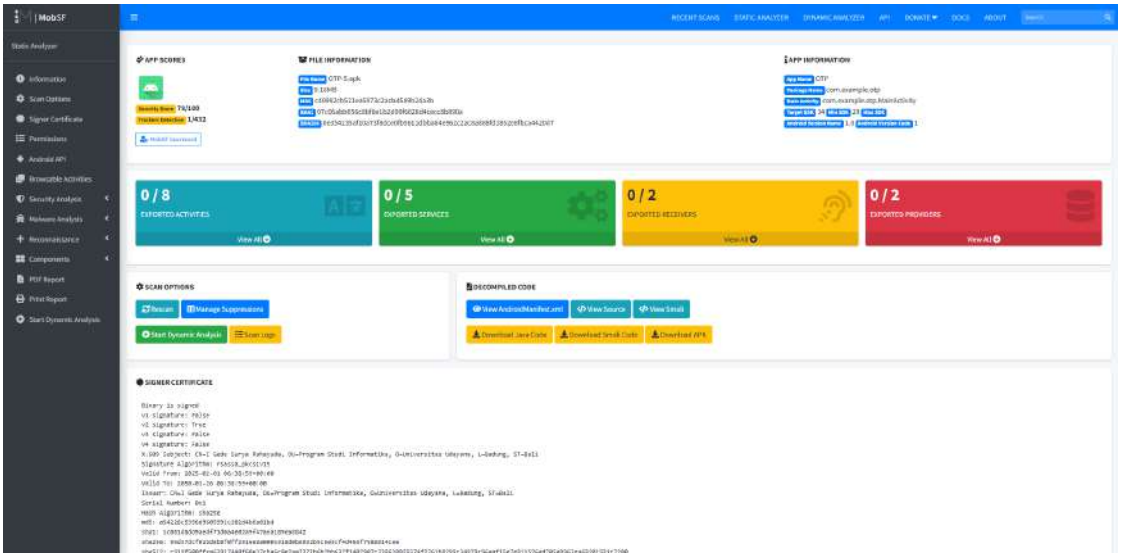
Kerentanan tingkat tinggi yang terdeteksi pada pengujian pertama dan kedua (OTP-1 dan OTP-2) berpotensi mengekspos data pengguna atau menyebabkan kerusakan fungsional pada aplikasi. Oleh karena itu, temuan ini harus segera diperbaiki agar menghindari potensi eksploitasi dan dampak serius terhadap keamanan sistem. Selain itu, MobSF juga mendeteksi sejumlah

peringatan dan informasi yang terdeteksi. Namun, temuan ini tidak terlalu kritikal dan dapat diabaikan karena tidak memberikan dampak signifikan terhadap keamanan aplikasi. Fokus utama dalam perbaikan seharusnya adalah menghilangkan masalah dengan kerentanan kategori "High" yang berpotensi lebih membahayakan.

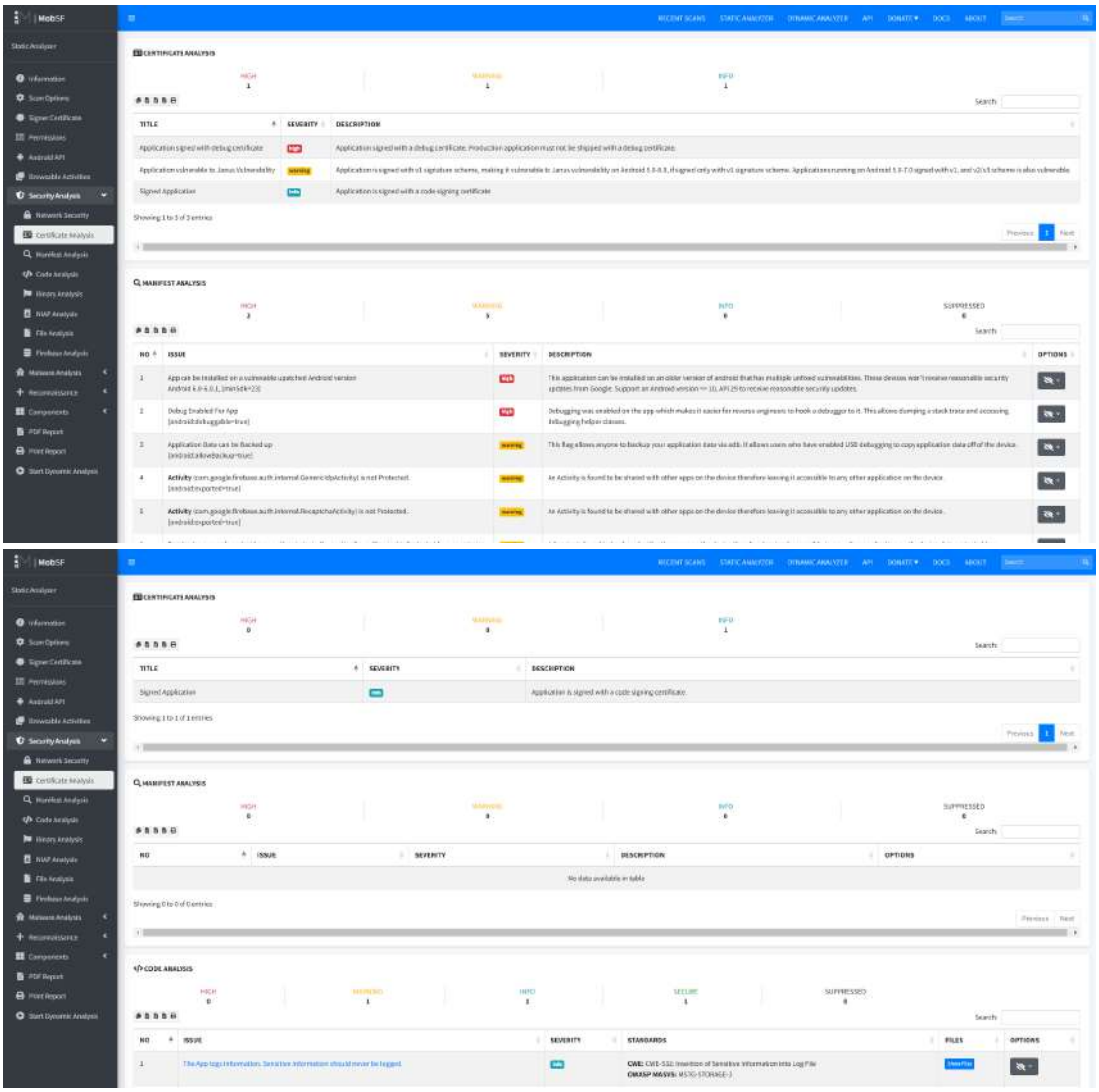
Secara keseluruhan, hasil pengujian ini menunjukkan bahwa meskipun aplikasi OTP telah mengalami peningkatan dari waktu ke waktu, potensi kerentanannya tetap perlu menjadi perhatian utama, terutama karena aplikasi ini menangani data sensitif seperti kode OTP. Kerentanan tingkat tinggi, seperti penggunaan pengkodean yang kurang aman atau manajemen kunci yang tidak terlindungi dengan baik, dapat membuka celah bagi serangan yang mengancam integritas dan keamanan aplikasi. Implikasi praktis dari temuan ini adalah pentingnya penerapan keamanan secara menyeluruh dalam tahap pengembangan serta pelaksanaan pemindaian keamanan secara berkala setelah aplikasi dirilis, dengan fokus utama pada penghapusan kerentanan kategori "High".



Gambar 10 Mobile Security Framework (MobSF) Score pada OTP-2



Gambar 11 Mobile Security Framework (MobSF) Score pada OTP-5



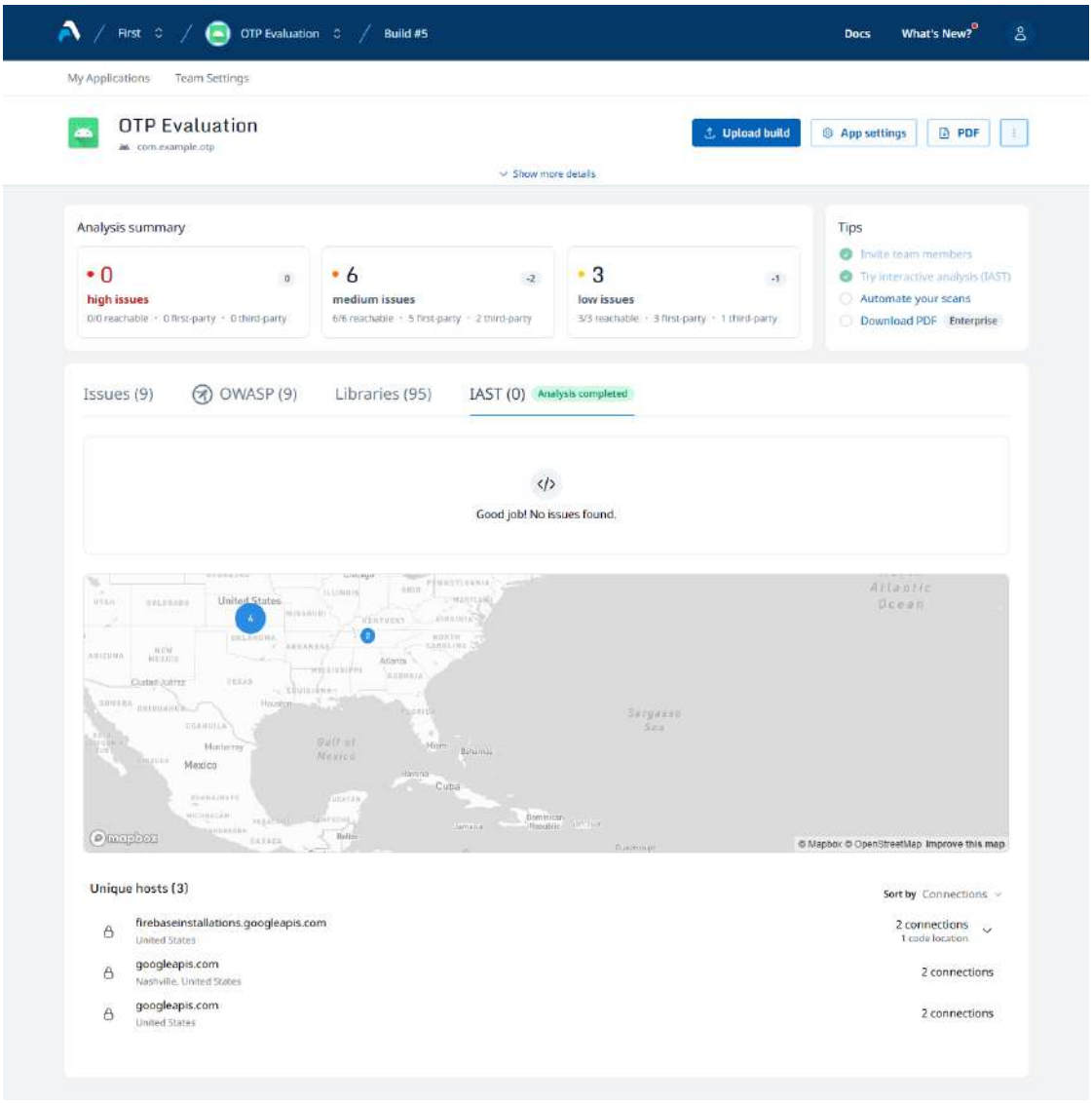
Gambar 12 Perbandingan High Issues pada OTP-1 dan OTP-5 MobSF

3.3 IAST (Interactive Application Security Testing)

Pengujian Interactive Application Security Testing (IAST) dilakukan menggunakan AppSweep untuk mengevaluasi keamanan aplikasi saat dijalankan di perangkat atau emulator. AppSweep membantu memeriksa berbagai masalah keamanan seperti kebocoran data, kesalahan konfigurasi izin, dan penggunaan API yang tidak aman, dengan mengacu pada standar OWASP Mobile Top 10. Hasil pengujian ditampilkan dalam berbagai halaman seperti Issues, OWASP, Libraries, dan laporan hasil pemindaian IAST. Gambar 13 memperlihatkan tampilan halaman IAST pada percobaan OTP-5, sedangkan Tabel 5 menyajikan hasil evaluasi keseluruhan dari lima percobaan aplikasi OTP yang telah dilakukan.

Tabel 5 Evaluasi OTP App dengan AppSweep

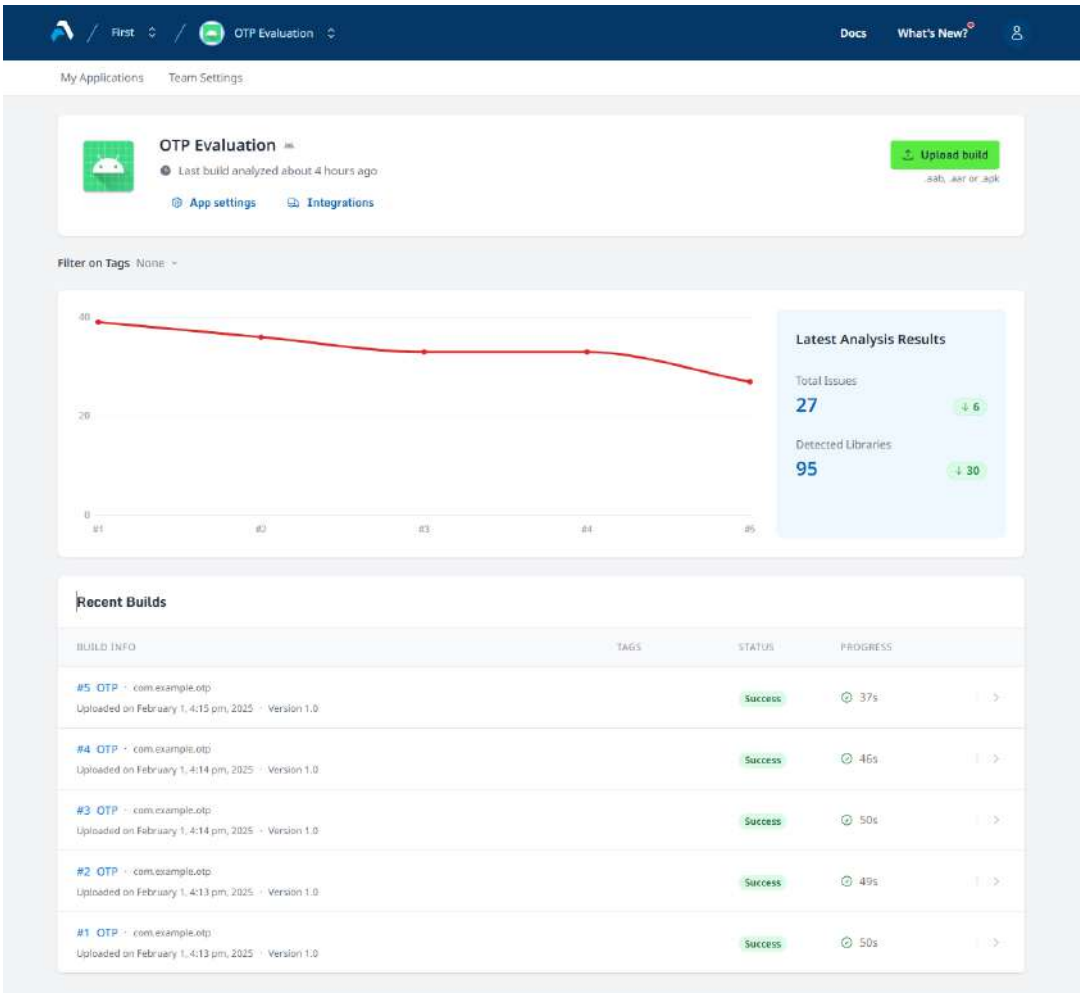
Evaluation	High	Medium	Low	Issues	OWASP	Libraries	Issues	Grade
OTP-1	2	8	4	14	14	125	0	Good
OTP-2	1	8	4	13	13	126	0	Good
OTP-3	0	8	4	12	12	125	0	Good
OTP-4	0	8	4	12	12	125	0	Good
OTP-5	0	6	3	9	9	95	0	Good



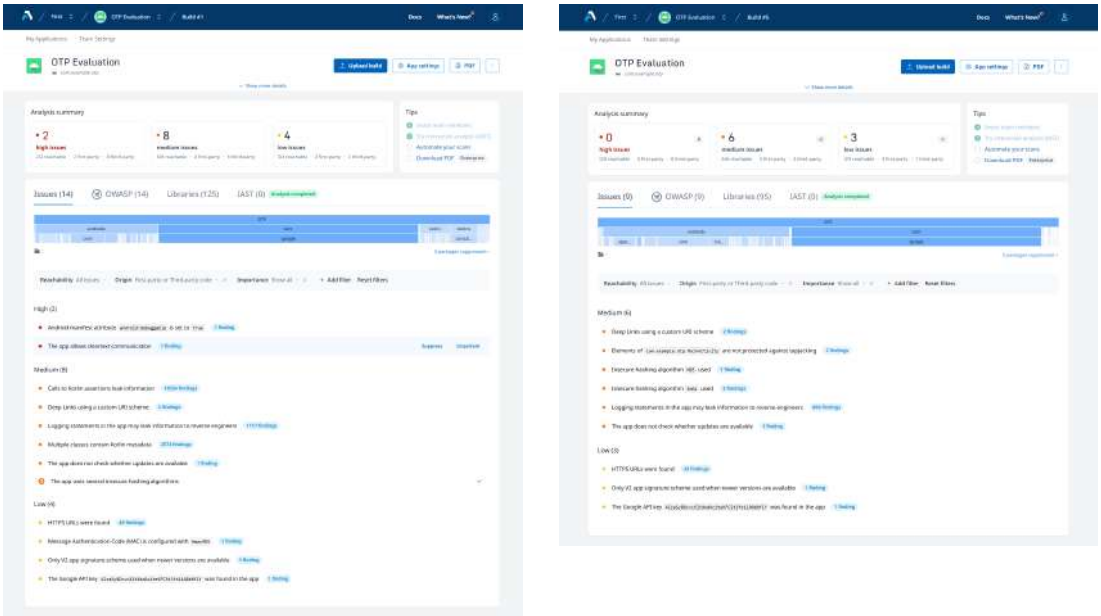
Gambar 13 Halaman IAST pada OTP-5

Berdasarkan hasil pengujian, aplikasi OTP secara keseluruhan memperoleh peringkat "Good" dalam hal kerentanannya. Tidak ada temuan yang masuk dalam kategori "Critical", yang menunjukkan bahwa aplikasi memiliki tingkat keamanan yang relatif baik. Namun, pada percobaan pertama dan kedua (OTP-1 dan OTP-2), terdapat temuan dalam kategori High yang mencakup potensi kebocoran data dan masalah izin yang signifikan. Kerentanan ini dapat memberikan akses yang tidak sah ke data sensitif atau memberikan hak akses yang salah kepada pengguna atau aplikasi pihak ketiga.

Seiring berjalannya pengujian, hasil menunjukkan peningkatan yang signifikan pada percobaan OTP-3 hingga OTP-5, di mana tidak ada lagi temuan dalam kategori High. Hal ini menunjukkan bahwa langkah-langkah perbaikan yang dilakukan berhasil mengatasi kerentanannya yang lebih serius. Secara kuantitatif, hasil pengujian menunjukkan bahwa jumlah kerentanannya berkurang secara signifikan dari 14 masalah pada OTP-1 menjadi 9 masalah pada OTP-5, seperti yang ditunjukkan pada Gambar 14. Hal ini menandakan kemajuan yang jelas dalam mengurangi jumlah dan tingkat kerentanannya. Sementara itu, Gambar 15 menggambarkan perbandingan jumlah High Issues antara OTP-1 dan OTP-5, yang menunjukkan penurunan signifikan setelah penerapan perbaikan.



Gambar 14 Grafik Evaluasi OTP pada AppSweep dalam 5 Kali Percobaan



Gambar 15 Perbandingan High Issues pada OTP-1 dan OTP-5 AppSweep

Meskipun sebagian besar kerentanannya terdeteksi pada kategori Medium dan Low, masalah ini tetap perlu diperhatikan. Dalam konteks aplikasi OTP, bahkan yang dianggap rendah bisa dimanfaatkan oleh peretas untuk mengeksploitasi celah keamanan, terutama yang berkaitan dengan kebocoran data atau pengaturan izin aplikasi. Misalnya, kebocoran data dapat menyebabkan informasi pribadi pengguna terekspos, sementara masalah izin dapat memberi akses yang tidak sah kepada pihak ketiga. Oleh karena itu, pengembang harus memastikan semua jenis kerentanannya, baik yang tinggi maupun rendah, ditangani dengan serius guna menjaga integritas dan keamanan aplikasi. Menghilangkan kerentanannya di kategori High dan memastikan bahwa IAST Issues telah diatasi sepenuhnya akan sangat meningkatkan keamanan aplikasi secara keseluruhan.

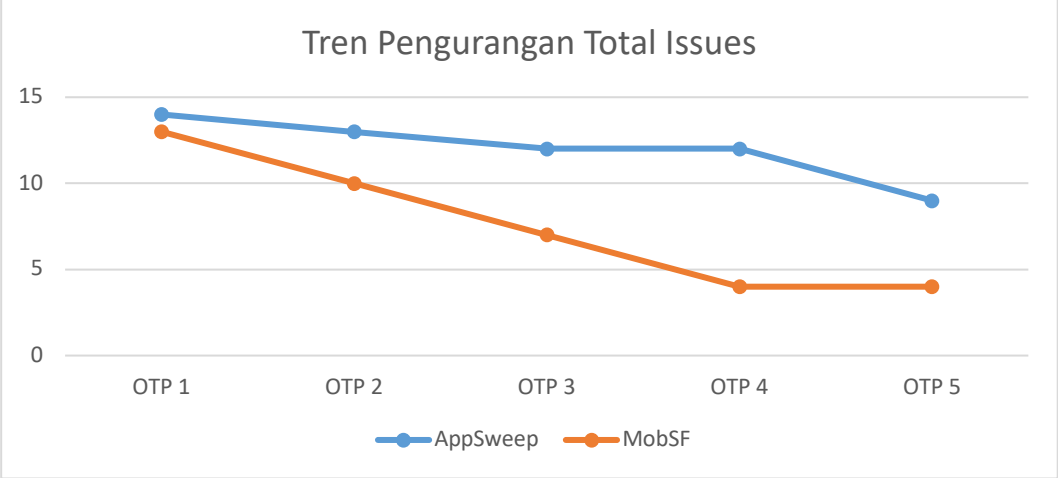
3.4 Perbandingan SAST dan IAST

Analisis Perbandingan SAST dan IAST menunjukkan perbedaan fokus dan keunggulan masing-masing metode. SAST cocok digunakan untuk deteksi dini selama proses pengembangan dan review kode, sementara IAST lebih efektif untuk mendeteksi eksploitasi di dunia nyata, meskipun lebih kompleks dan memerlukan runtime testing. Dalam konteks aplikasi mobile, kombinasi SAST dan IAST dapat meningkatkan keamanan secara menyeluruh dengan mengidentifikasi kelemahan sejak tahap awal pengembangan hingga pengujian di lingkungan runtime. Tabel 6 menyajikan perbandingan ringkas antara kedua metode ini.

Tabel 6 Perbandingan SAST dan IAST

Aspek	SAST (Static Application Security Testing)	IAST (Interactive Application Security Testing)
Metode Analisis	Analisis kode sumber, bytecode, atau biner tanpa menjalankan aplikasi.	Menguji aplikasi saat berjalan, memantau input, output, dan eksekusi runtime.
Cakupan Keamanan	Menemukan kerentanan dalam kode sebelum aplikasi dijalankan.	Mendeteksi kelemahan keamanan yang hanya muncul dalam kondisi runtime.
Waktu Deteksi	Dapat dilakukan lebih awal dalam pengembangan.	Mengidentifikasi kerentanan saat aplikasi berjalan, cocok untuk tahap pengujian akhir.
Akurasi	Cenderung menghasilkan false positives karena tidak mempertimbangkan kondisi runtime.	Lebih akurat dalam menemukan eksploitasi nyata karena berbasis eksekusi.
Kompleksitas	Relatif mudah diterapkan, tidak memerlukan eksekusi aplikasi.	Memerlukan lingkungan runtime yang mendukung pengujian interaktif.
Kecepatan	Lebih cepat karena tidak memerlukan eksekusi aplikasi.	Lebih lambat karena harus berjalan bersamaan dengan aplikasi.

Hasil evaluasi juga divisualisasikan dalam *line chart* yang menunjukkan tren penurunan jumlah isu keamanan dari OTP-1 hingga OTP-5, seperti terlihat pada Gambar 16. AppSweep secara konsisten mendeteksi lebih banyak isu dibandingkan MobSF, tetapi keduanya menunjukkan pola penurunan yang serupa, mencerminkan perbaikan keamanan pada setiap versi OTP. MobSF mengalami penurunan lebih tajam, dari 13 isu pada OTP-1 menjadi hanya 4 pada OTP-5, sementara AppSweep berkurang lebih bertahap, dari 14 menjadi 9. Hal ini menunjukkan bahwa MobSF cenderung lebih sensitif dalam mendeteksi perbaikan keamanan, sementara AppSweep mempertahankan standar deteksi yang lebih ketat. Kombinasi kedua metode ini dapat memberikan analisis yang lebih komprehensif, dengan MobSF mengidentifikasi celah yang dapat segera diperbaiki, sementara AppSweep memastikan evaluasi keamanan tetap menyeluruh hingga iterasi terakhir.



Gambar 16 Tren Pengurangan Total Issues

Hasil penelitian ini dapat diadaptasi dalam berbagai konteks yang memerlukan tingkat keamanan tinggi, seperti sistem perbankan dan e-commerce. Dalam perbankan digital, kombinasi SAST dan IAST dapat digunakan untuk mendeteksi dan mencegah eksploitasi keamanan pada aplikasi mobile banking, terutama dalam melindungi transaksi pengguna dan mencegah ancaman seperti man-in-the-middle attacks atau injeksi kode berbahaya. SAST dapat memastikan keamanan kode sejak tahap pengembangan, sementara IAST membantu mengidentifikasi kelemahan yang hanya muncul saat aplikasi berjalan, seperti vulnerable API calls atau autentikasi yang tidak aman. Dalam e-commerce, metode ini dapat diterapkan untuk menjaga keamanan transaksi pelanggan, terutama dalam sistem pembayaran online yang sering menjadi target serangan seperti phishing atau carding. Dengan menerapkan pendekatan serupa yang digunakan dalam penelitian ini, sistem perbankan dan e-commerce dapat meningkatkan ketahanan terhadap serangan siber serta memastikan pengalaman pengguna yang aman dan terpercaya.

4. KESIMPULAN

Penelitian ini berhasil mengimplementasikan OTP Firebase berbasis email pada aplikasi Android dengan pengujian keamanan menggunakan SAST (MobSF) dan IAST (AppSweep), yang efektif dalam mendeteksi kerentanan. Hasil pengujian menunjukkan perbaikan signifikan, di mana temuan kerentanannya berkurang dari 3 temuan tingkat tinggi pada percobaan pertama (OTP-1) menjadi 0 pada percobaan kelima (OTP-5), dengan skor aplikasi meningkat dari 43 (B) menjadi 78 (A) di MobSF. Begitu juga pada AppSweep, meskipun kerentanannya pada OTP-1 dan OTP-2 ditemukan pada kategori High, tidak ada temuan serupa pada OTP-3 hingga OTP-5, dengan jumlah total masalah menurun dari 14 menjadi 9. Keterbatasan penelitian ini mencakup cakupan dataset yang terbatas dan potensi bias alat pengujian, yang dapat diperbaiki dengan menggunakan dataset lebih besar dan metode tambahan seperti machine learning pada penelitian selanjutnya. Penelitian lanjutan juga akan mengadopsi model DevSecOps dengan IAST di tahap pengembangan (Dev) dan RASP di tahap operasi (Ops) untuk meningkatkan keamanan aplikasi lebih lanjut. Saat ini penelitian sudah mengintegrasikan RASP, namun masih pada level dasar dengan ProGuard. Metode machine learning mungkin akan diterapkan pada bagian Operasi (Ops), seperti pada IPS (Intrusion Prevention Systems), IDS (Intrusion Detection Systems), dan RASP (Runtime Application Self-Protection). Implikasi lebih luas dari penelitian ini adalah memberikan rekomendasi bagi pengembang aplikasi mobile untuk mengadopsi pendekatan pengujian keamanan yang lebih proaktif, dengan memanfaatkan kombinasi SAST dan IAST dalam siklus pengembangan perangkat lunak. Selain itu, penelitian masa depan dapat mengeksplorasi integrasi kecerdasan buatan (Artificial Intelligence) untuk otomatisasi deteksi kerentanan OTP serta menganalisis keamanan pada OTP berbasis biometrik. Dengan pendekatan ini, diharapkan sistem autentikasi berbasis OTP dapat semakin aman dan lebih adaptif terhadap berbagai ancaman siber di masa depan.

DAFTAR PUSTAKA

Asih, M. S., & Hasibuan, A. Z. (2023). Pengamanan Kunci Pintu Brankas Menggunakan Kriptografi One Time Pad (OTP) Berbasis Android. *Explorer*, *3*(2), 58–68. https://doi.org/10.47065/explorer.v3i2.738

Chimuco, F. T., Sequeiros, J. B. F., Lopes, C. G., Simões, T. M. C., Freire, M. M., & Inácio, P. R. M. (2023). Secure Cloud-Based Mobile Apps: Attack Taxonomy, Requirements, Mechanisms, Tests and Automation. *International Journal of Information Security*, *22*(4), 833–867. https://doi.org/10.1007/s10207-023-00669-z

Danthy, R., Pratham Pai, K., & Rao, V. (2024). Secure Online Banking Authentication System Using Time Bound Password. *2024 IEEE International Conference on Computing, Power and Communication Technologies (IC2PCT)*, 130–135. https://doi.org/10.1109/IC2PCT60090.2024.10486295

Ikram, M., Sentana, I. W. B., Asghar, H., Kaafar, M. A., & Kepkowski, M. (2025). More Than Just a Random Number Generator! Unveiling the Security and Privacy Risks of Mobile OTP Authenticator Apps. In M. Barhamgi, H. Wang, & X. Wang (Eds.), *Web Information Systems Engineering – WISE 2024* (pp. 177–192). Springer Nature Singapore. https://doi.org/10.1007/978-981-96-0576-7_14

Keteku, J., Dameh, G. O., Mante, S. A., Mensah, T. K., Amartey, S. L., & Diekuu, J.-B. (2024). Detection and Prevention of Malware in Android Mobile Devices: A Literature Review. *International Journal of Intelligence Science*, *14*(04), 71–93. https://doi.org/10.4236/ijis.2024.144005

Li, K., Chen, S., Fan, L., Feng, R., Liu, H., Liu, C., Liu, Y., & Chen, Y. (2023). Comparison and Evaluation on Static Application Security Testing (SAST) Tools for Java. *Proceedings of the 31st ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, 921–933. https://doi.org/10.1145/3611643.3616262

Lim, J. G. Q., Kwok, Z. Y., Soon, I., Yong, J. X., Song Yuhao, S., Binte Rosley, S. H., & Balachandran, V. (2024). A-COPILOT: Android Covert Operation for Private Information Lifting and OTP Theft: A Study on How Malware Masquerading as Legitimate Applications Compromise Security and Privacy. *Proceedings of the Fourteenth ACM Conference on Data and Application Security and Privacy*, 155–157. https://doi.org/10.1145/3626232.3658638

Lubis, D. J., & Riswanto, A. N. (2024). Implementasi Algoritma Random Forest untuk Optimasi Keamanan Autentikasi One-Time Password (OTP). *Informatech: Jurnal Ilmiah Informatika dan Komputer*, *1*(1), 23–29. https://doi.org/10.69533/eyp7ag46

Moon, I. T., Mimi, A., & Rahman Mridha, Md. M. (2023). Cryptographic Analysis: Popular Social Media Applications and Mitigations of Vulnerabilities. *2023 26th International Conference on Computer and Information Technology (ICCIT)*, 1–6. https://doi.org/10.1109/ICCIT60459.2023.10441090

Nutalapati, V. (2023). Automated Security Testing for Mobile Apps: Tools, Techniques, and Best Practices. *International Engineering & Applied Sciences (IRJEAS)*, *11*(1), 26. https://doi.org/10.55083/irjeas.2023.v11i01004

Pagano, F., Romdhana, A., Caputo, D., Verderame, L., & Merlo, A. (2023). SEBASTiAn: A Static and Extensible Black-Box Application Security Testing Tool for iOS and Android Applications. *SoftwareX*, *23*, Article ID: 101448. https://doi.org/10.1016/j.softx.2023.101448

Pargaonkar, S. (2023). Advancements in Security Testing: A Comprehensive Review of Methodologies and Emerging Trends in Software Quality Engineering. *International Journal of Science and Research (IJSR)*, *12*(9), 61–66. https://doi.org/10.21275/SR23829090815

Schleier, S., Holguera, C., Mueller, B., & Willemsen, J. (2024). *OWASP Mobile Application Security Testing Guide (MASTG)*. The OWASP Foundation. https://mas.owasp.org/MASTG/

Schleier, S., Holguera, C., Mueller, B., Willemsen, J., & Beckers, J. (2024). *OWASP Mobile Application Security Verification Standard v2.1.0*. The OWASP Foundation. https://mas.owasp.org/MASVS/

Smyčka, M. (2024). *Evaluation and Application of SAST Tools* [Masarykova Univerzita]. https://is.muni.cz/th/j4bxg/smycka_thesis.pdf

Sonnekalb, T., Knaust, C.-T., Gruner, B., Brust, C.-A., Heinze, T. S., Kurnatowski, L. von, Schreiber, A., & Mäder, P. (2023). A Static Analysis Platform for Investigating Security Trends in Repositories. *2023 IEEE/ACM 1st International Workshop on Software Vulnerability (SVM)*, 1–5. https://doi.org/10.1109/SVM59160.2023.00005

Stanciu, A.-M. (2023). Theoretical Study of Security for a Software Product. In *Lecture Notes in Networks and Systems* (Vol. 578, pp. 233–242). https://doi.org/10.1007/978-981-19-7660-5_20

Sutter, T., Kehrer, T., Rennhard, M., Tellenbach, B., & Klein, J. (2024). Dynamic Security Analysis on Android: A Systematic Literature Review. *IEEE Access*, *12*, 57261–57287. https://doi.org/10.1109/ACCESS.2024.3390612

Taqwim, M. A., Kusyanti, A., & Siregar, R. A. (2021). Implementasi Algoritme Speck untuk Enkripsi One-Time Password pada Two-Factor Authentication. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, *5*(7), 3103–3111. https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/9488

Wibawa, S., Suryanto, S., & Ningsih, R. (2024). Perlindungan Data Digital Dengan Time-Based One-Time Password (TOTP). *INSANtek*, *5*(1), 30–36. https://doi.org/10.31294/insantek.v5i1.3495

JISKA (Jurnal Informatika Sunan Kalijaga)
Vol. 11, No. 1, JANUARI, 2026, Pp. 32 – 43
ISSN: 2527 – 5836 (print) | 2528 – 0074 (online)

Komparasi *Distance Measure* pada K-Means dalam Klasterisasi Peserta KB Aktif

Mochammad Anshori [(1)*], Afifah Vera Ferencia Fitria Ningrum [(2)], Risqy Siwi Pradini [(3)]
Departemen Informatika, Institut Teknologi, Sains, dan Kesehatan RS. Dr. Soepraoen Kesdam V/BRW, Malang, Indonesia
e-mail : {moanshori,risqypradini}@itsk-soepraoen.ac.id, afifahveraferencia@gmail.com.
* Penulis korespondensi.
Artikel ini diajukan 3 Februari 2025, direvisi 12 November 2025, diterima 15 November 2025, dan dipublikasikan 25 Januari 2026.

Abstract

The rapid population growth in Indonesia poses significant challenges to public welfare, economic stability, and sustainable development. The Family Planning program aims to regulate population growth through various contraceptive methods; however, participation rates often differ across regions. Understanding these variations is crucial for designing targeted interventions. This study investigates how different distance measures in the K-Means clustering algorithm affect the segmentation quality of KB participants in Kalirejo Village, Lawang District. Eight distance metrics—Euclidean, Manhattan, Minkowski, Chebyshev, Mahalanobis, Bray-Curtis, Canberra, and Cosine—were compared using standardized data from the local BKKBN office (January–September). Cluster validity was evaluated using the Silhouette Coefficient across k=2–10. Results show that the Manhattan distance with k=2 achieved the best clustering quality (SC = 0.7191), effectively distinguishing participant groups by contraceptive method preference. The study highlights the importance of selecting suitable distance measures to improve data-driven policy and decision-making in family planning management.

Keywords: K-Means, Clustering, Silhouette Coefficient, Distance Measure, Manhattan

Abstrak

Pertumbuhan penduduk yang pesat di Indonesia menimbulkan tantangan besar terhadap kesejahteraan masyarakat, stabilitas ekonomi, dan pembangunan berkelanjutan. Program Keluarga Berencana (KB) bertujuan mengendalikan laju pertumbuhan penduduk melalui berbagai metode kontrasepsi, namun tingkat partisipasi masyarakat sering kali berbeda antarwilayah. Pemahaman terhadap variasi tersebut sangat penting untuk merancang intervensi yang lebih tepat sasaran. Penelitian ini mengkaji pengaruh penggunaan berbagai ukuran jarak (*distance measure*) pada algoritma K-Means terhadap kualitas segmentasi peserta KB di Desa Kalirejo, Kecamatan Lawang. Delapan jenis ukuran jarak—Euclidean, Manhattan, Minkowski, Chebyshev, Mahalanobis, Bray-Curtis, Canberra, dan Cosine—dibandingkan menggunakan data terstandarisasi dari BKKBN setempat (Januari–September). Validitas klaster dievaluasi dengan Silhouette Coefficient pada rentang k=2–10. Hasil menunjukkan bahwa ukuran jarak Manhattan dengan k=2 menghasilkan kualitas klaster terbaik (SC = 0,7191), yang secara efektif membedakan kelompok peserta berdasarkan preferensi metode kontrasepsi. Penelitian ini menegaskan pentingnya pemilihan ukuran jarak yang tepat untuk meningkatkan kualitas analisis berbasis data serta mendukung pengambilan keputusan kebijakan dalam pengelolaan program KB.

Kata Kunci: K-Means, *Clustering*, Silhouette Coefficient, *Distance Measure*, Manhattan

1. PENDAHULUAN

Salah satu program pemerintah yang bertujuan untuk mengontrol pertumbuhan penduduk dan meningkatkan kesejahteraan masyarakat adalah Keluarga Berencana (KB). Program ini mengatur jarak kelahiran untuk mengimbangi jumlah penduduk dan ketersediaan sumber daya. (Munawar et al., 2024; Sumarsih, 2023). Dengan mengendalikan angka kelahiran, Program KB diharapkan dapat menghentikan peningkatan populasi yang berlebihan dan mencegah dampak negatif yang dapat muncul akibat lonjakan populasi yang tidak terkendali. Selain itu, program KB

meningkatkan kualitas hidup keluarga dan individu dengan memberikan akses ke perawatan kesehatan reproduksi berkualitas tinggi (Tifannii et al., 2020).

Pertumbuhan penduduk yang tidak terkendali dalam konteks sosial dan ekonomi dapat menyebabkan berbagai masalah, seperti peningkatan kriminalitas, kesenjangan kesejahteraan dan makin terbatasnya lapangan kerja (Maharani & Yotenka, 2024). Oleh karena itu pemerintah mencanangkan program KB sebagai langkah strategis untuk pengelolaan kuantitas penduduk. Sebagai lembaga yang bertanggung jawab atas pelaksanaan program keluarga berencana di Indonesia, Badan Kependudukan dan Keluarga Berencana Nasional (BKKBN) berusaha untuk meningkatkan kinerja program melalui penggunaan berbagai teknik berbasis data dan teknologi (Lino et al., 2021). Untuk mendapatkan pemahaman tentang pola partisipasi dan efektivitas program di berbagai wilayah, peserta KB diklasifikasikan atau dibagi menjadi kelompok.

Desa Kalirejo di Kecamatan Lawang merupakan salah satu wilayah yang menerapkan program KB yang diinisiasi oleh Puskesmas setempat. Puskesmas ini berperan sebagai Klinik Keluarga Berencana (KKB) yang menyediakan layanan kontrasepsi bagi masyarakat setempat. Meskipun program ini telah berjalan, masih terdapat kecenderungan tingkat kelahiran yang tinggi di beberapa bagian desa. Hal ini menunjukkan adanya kebutuhan untuk memahami distribusi dan pola partisipasi peserta KB guna meningkatkan efektivitas penyuluhan dan layanan KB yang diberikan. Klasterisasi peserta KB aktif adalah langkah penting dalam mengoptimalkan program ini. Ini memungkinkan untuk menjangkau kelompok sasaran dengan lebih akurat.

Sebelum ini, penelitian klasterisasi peserta KB aktif di Desa Kalirejo telah dilakukan dengan menggunakan metode K-Means dengan Euclidean distance sebagai ukuran jarak. Hasil penelitian tersebut menunjukkan bahwa *silhouette coefficient* yang diperoleh sebesar 0,447 dengan jumlah klaster optimal sebanyak dua (Ningrum et al., 2025). Nilai *silhouette coefficient* ini berada dalam rentang -1 hingga 1, di mana semakin mendekati angka 1 menunjukkan bahwa klaster yang terbentuk semakin baik. Di sisi lain, hasil masih belum ideal, jadi diperlukan penelitian lebih lanjut untuk meningkatkan keakuratan dan keefektifan model klasterisasi.

Dalam metode K-Means, pemilihan ukuran jarak atau *distance measure* memiliki peran penting dalam menentukan kualitas klaster yang terbentuk. Oleh karena itu, diperlukan eksplorasi lebih lanjut terhadap berbagai *distance measure* yang dapat digunakan selain Euclidean distance. Beberapa *distance measure* yang umum digunakan dalam klasterisasi mencakup *Manhattan, Minkowski, Chebyshev, Mahalanobis, Bray-Curtis, Canberra dan Cosine distance* (Argiento et al., 2025; Idrus et al., 2022; Jollyta et al., 2023; Kumala & Rahning Putri, 2023; Pangestu & Fitriani, 2022; Shapcott, 2024; Telsiz Kayaoğlu & Eroğlu, 2024; Wurdianarto et al., 2014). Masing-masing metode memiliki karakteristik yang berbeda dan dapat memberikan hasil yang lebih baik pada kondisi data tertentu.

Penelitian terdahulu menunjukkan bahwa variasi *distance measure* dapat memberikan dampak yang signifikan terhadap hasil klasterisasi. Studi sebelumnya membandingkan perhitungan *euclidean*, *manhattan, dan cosine distance* dalam pengelompokan data dan menemukan bahwa *manhattan distance* memiliki performa yang lebih baik pada data dengan distribusi yang tidak merata (Pangestu & Fitriani, 2022). Riset lainnya telah melakukan perbandingan antara *chebyshev* dengan *manhattan* dan menghasilkan *chebyshev* menghasilkan klaster yang lebih sempurna (Solikhun et al., 2025).

Meskipun banyak penelitian telah mengeksplorasi perbandingan *distance measure* dalam klasterisasi, masih terdapat kesenjangan penelitian dalam penerapan metode ini pada klasterisasi peserta KB. Kebanyakan penelitian sebelumnya lebih fokus pada aplikasi *distance measure* dalam bidang kesehatan secara umum atau data non-klinis, Untuk meningkatkan efektivitas klasterisasi peserta KB aktif di Desa Kalirejo, penelitian ini bertujuan untuk mempelajari dan membandingkan berbagai ukuran jarak dalam metode K-Means.

Penelitian ini bertujuan untuk mengelompokkan peserta KB aktif di Desa Kalirejo dengan menerapkan metode K-Means dan berbagai ukuran jarak (*distance measure*) guna menentukan metode mana yang menghasilkan performa terbaik. Studi ini memberikan kontribusi dalam memperluas pemahaman mengenai penggunaan ukuran jarak dalam proses pengelompokan data, sekaligus mendukung pengambilan keputusan yang berbasis data dalam program KB. Diharapkan, temuan penelitian ini dapat memberikan masukan berharga dalam perencanaan dan pelaksanaan program KB yang lebih terarah, serta meningkatkan efektivitas layanan KB di tingkat desa.

2. METODE PENELITIAN

Penelitian ini melibatkan serangkaian langkah yang penting untuk mempersiapkan dan merencanakan studi secara menyeluruh. Penelitian ini dirancang untuk mengatasi permasalahan yang diidentifikasi dalam studi. Secara umum, terdapat lima tahapan utama yang dilakukan dalam penelitian ini, seperti yang diilustrasikan pada Gambar 1. Tahapan-tahapan tersebut meliputi pengumpulan data, praproses data, dan pembuatan model klasterisasi menggunakan metode K-Means dengan berbagai macam *distance measure* dan terakhir adalah evaluasi dengan menggunakan s*ilhouette coefficient*.

Gambar 1 Metode Penelitian

2.1 Pengumpulan Data

Penelitian ini memanfaatkan data sekunder yang bersumber dari Badan Kependudukan dan Keluarga Berencana Nasional (BKKBN) di Desa Kalirejo, Kecamatan Lawang. Data yang dikumpulkan meliputi informasi tentang lokasi tempat tinggal peserta KB (RT dan RW) serta jenis alat kontrasepsi yang digunakan, seperti IUD, MOW, MOP, implant, suntik, pil, dan kondom (Maulana, 2020). Data ini dikumpulkan dalam rentang waktu Januari hingga September. Untuk menjaga privasi individu, penelitian ini tidak mengikutsertakan data personal warga. Selanjutnya data dilakukan proses pembersihan.

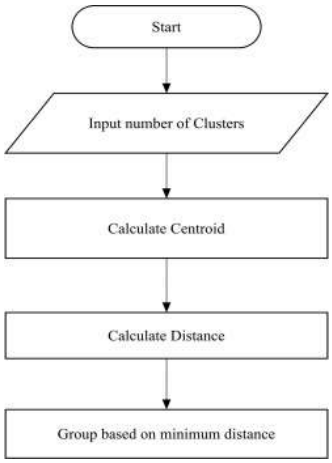
2.2 Praproses Data

Algoritme K-Means sangat sensitif terhadap skala data (Shapcott, 2024), sehingga tahap praproses dilakukan untuk memastikan bahwa data memiliki distribusi yang lebih seragam. Salah satu teknik yang digunakan adalah standardisasi data dengan metode z-score normalization, yang dapat mengurangi *skewness* dalam distribusi data dan membuat setiap atribut memiliki rata-rata nol serta varians satu (Ghosh, 2022). Proses ini bertujuan untuk menghindari bias akibat perbedaan skala antar atribut. Formula dari standardisasi ditunjukkan pada Pers (1), di mana x adalah data awal, μ adalah rerata dari atribut, σ adalah standar deviasi dan x' adalah data baru hasil standardisasi z-score.

$$x' = \frac{x - \mu}{\sigma} \quad (1)$$

2.3 Klasterisasi K-Means

Klasterisasi merupakan teknik yang digunakan untuk membagi data ke dalam beberapa kelompok berdasarkan karakteristik tertentu (Andriyani et al., 2024) dan termasuk metode yang prominent pada bidang ilmu komputer (Wala et al., 2024). Algoritma K-Means bekerja dengan mengelompokkan data ke dalam sejumlah klaster, di mana data dalam satu klaster memiliki kesamaan (homogenitas) dalam hal karakteristik (J. A. Muttaqin et al., 2023; W. W. W. Muttaqin et al., 2023). Metode ini dikenal efisien karena kecepatannya dalam memproses data dan kemampuannya untuk menangani data dalam skala besar. Namun, metode ini memiliki keterbatasan dalam hal akurasi ketika menghadapi noise atau data yang terisolasi (Hutagalung, 2022; Hutagalung & Sonata, 2021; Pratistha & Kristianto, 2024).



Gambar 2 Flowchart Algoritma K-Means

Cara kerja pengelompokan K-Means cukup sederhana. Gambar 2 menunjukkan mekanisme K-Means. Mengikuti gambar tersebut, pertama-tama harus menentukan k klaster. Kemudian menghitung centroid dengan inisialisasi acak. Langkah ketiga adalah menghitung jarak antara titik data ke centroid menggunakan ukuran jarak yang dipilih, jarak *euclidean* umumnya digunakan. Langkah terakhir adalah mengelompokkan klaster berdasarkan angka rerata terdekat (Rawal et al., 2021). Proses ini akan berulang hingga tidak ada pembaruan yang ditemukan untuk centroid.

2.4 *Distance Measure*

Berbagai macam pengukuran jarak yang akan digunakan dalam percobaan ini adalah *Euclidean distance, Manhattan distance, Minkowski distance, Chebyshev distance, Mahalanobis distance, Bray-Curtis distance, Canberra distance,* dan *Cosine distance*. Euclidean distance merupakan jarak yang paling umum digunakan dan paling sederhana antara titik, terutama untuk variabel kontinu. Rumus perhitungan Euclidean distance mengacu pada Pers (2). Canberra distance adalah fungsi metrik yang sering digunakan untuk distribusi di sekitar daerah asal, di mana perhitungannya dilakukan dengan membagi perbedaan absolut antara variabel dari dua objek dengan jumlah nilai variabel absolut tersebut. Rumus Canberra distance ditampilkan pada Pers (3). Manhattan distance, yang juga disebut sebagai *City-Block distance*, adalah pengukuran absolut yang dihasilkan berdasarkan jumlah selisih antara dua titik data, dengan rumus yang ditunjukkan pada Pers (4).

$$d_{euclidean}(i,j) = \sqrt{\sum_{k=1}(x_{ik} - x_{jk})^2} \quad (2)$$

$$d_{canberra}(i,j) = \sum_{k=1} \frac{|x_{ik} - x_{jk}|}{|x_{ik}| + |x_{jk}|} \quad (3)$$

$$d_{manhattan}(i,j) = \sum_{k=1} |x_{ik} - x_{jk}| \quad (4)$$

Selanjutnya, Cosine distance digunakan untuk menghitung jarak kosinus antara dua data point dan biasa dikenal dengan *cosine similarity* dengan formula yang ditunjukkan pada Pers (5). Chebyshev distance, atau jarak metrik maksimum, menghitung perbedaan absolut maksimum antara koordinat dua titik di sepanjang dimensi apa pun. Metode ini tangguh terhadap outlier dan dapat menangani data dengan skala yang bervariasi, sebagaimana ditunjukkan pada Pers (6). Bray-Curtis distance pada dasarnya menghitung perbedaan antara dua sampel berdasarkan komposisi spesiesnya, dengan nilai berkisar dari 0 (benar-benar mirip) hingga 1 (benar-benar berbeda). Formula Bray-Curtis ditunjukkan pada Pers (7).

$$d_{cosine}(i,j) = 1 - \frac{\sum_{k=1}(x_{ik} - x_{jk})}{\sqrt{\sum_{k=1} {x_{ik}}^2}\sqrt{\sum_{k=1} {x_{jk}}^2}} \quad (5)$$

$$d_{chebyshev}(i,j) = \max_k |x_{ik} - x_{jk}| \quad (6)$$

$$d_{bray-curtis}(i,j) = \frac{\sum_{k=1} |x_{ik} - x_{jk}|}{\sum_{k=1}(x_{ik} + x_{jk})} \quad (7)$$

Selain itu, Minkowski distance adalah pengukuran jarak dengan generalisasi pengukuran dari Euclidean dan Manhattan. Formula dari Minkowski distance ditunjukkan pada Pers (8). Terakhir, Mahalanobis distance adalah pengukuran yang digunakan untuk mengukur jarak antara data point dengan distribusinya dengan memanfaatkan *covariance* data, membuatnya cocok untuk karakteristik data dengan varians yang berbeda. Formula dari Mahalanobis distance ditunjukkan pada Pers (9).

$$d_{minkowski}(i,j) = (\sum_{k=1}(x_{ik} - x_{jk})^q)^{1/q} \quad (8)$$

$$d_{mahalanobis}(i,j) = \sqrt{(x_{ik} - x_{jk})^T Cov^{-1}(x_{ik} - x_{jk})} \quad (9)$$

2.5 Evaluasi

Eksperimen yang akan dilakukan adalah dengan mencoba berbagai macam *distance measure* yang telah dibahas sebelumnya. Pengujian juga dilakukan untuk mendapatkan klaster terbaik berdasarkan parameter *k* dari K-Means. Pengujian parameter *k* yang akan dilakukan dengan rentang nilai antara 2 hingga 10. Tahap berikutnya untuk mengetahui seberapa bagus klaster dengan dilakukan evaluasi model menggunakan *silhoutte coefficient* (SC). Koefisien siluet rata-rata dihitung menggunakan jarak intra-klaster dan jarak terdekat ke klaster untuk setiap titik data (Shahapure & Nicholas, 2020).

$$\mathrm{S} = \frac{(b - a)}{\max(a,b)} \quad (10)$$

Merujuk pada Pers (10) adalah formula dari SC, di mana maksimum (a, b) adalah nilai maksimum antara a dan b, sedangkan b adalah rata-rata jarak antara sampel dan semua sampel dalam klaster yang sama. Dalam analisis penelitian ini, SC menunjukkan pengelompokan data; SC dengan nilai mendekati +1 menunjukkan bahwa titik data berada di tingkat tinggi, sedangkan SC

dengan nilai mendekati 0 menunjukkan bahwa titik data mungkin berada di klaster lain. Interpretasi nilai SC adalah sebagai berikut (Kumala & Rahning Putri, 2023):

- 0.7 0,7 ≤ SC ≤ 1: Struktur klaster yang kuat.
- 0,5 ≤ SC < 0,7: Struktur klaster yang sedang.
- 0,25 ≤ SC < 0,5: Struktur klaster yang lemah.
- SC < 0,25: Tidak ada struktur klaster yang jelas.

Setelah implementasi model dan evaluasi klasterisasi, dilakukan analisis terhadap hasil untuk menentukan metode *distance measure* yang memberikan hasil terbaik. Hasil diperbandingkan berdasarkan nilai SC untuk masing-masing *distance measure* dan jumlah klaster k yang bervariasi antara 2 hingga 10. Visualisasi klaster menggunakan grafik *silhouette plot* dan grafik sebaran klaster dibuat untuk memahami distribusi data dalam setiap metode. Dengan metodologi yang sistematis ini, penelitian ini bertujuan untuk mengidentifikasi *distance measure* yang paling sesuai dalam klasterisasi peserta KB aktif, sehingga dapat memberikan rekomendasi untuk penerapan yang lebih akurat dalam program KB berbasis data.

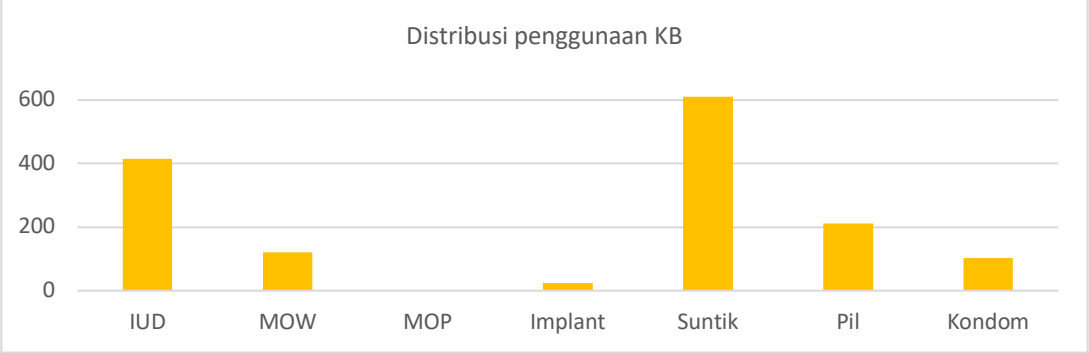
3. HASIL DAN PEMBAHASAN

Tujuan dari penelitian ini adalah untuk mengevaluasi seberapa efektif berbagai ukuran jarak metode K-Means dalam klasterisasi peserta KB aktif di Desa Kalirejo. Hasil analisis disajikan dalam beberapa langkah. Ini termasuk penjelasan tentang dataset, hasil eksperimen klasterisasi, pengujian model menggunakan SC, dan analisis tambahan tentang pilihan jumlah klaster dan metrik jarak yang paling cocok. Dataset yang digunakan terdiri dari data peserta KB di desa Kalirejo Lawang, yang diuraikan secara rinci dalam Tabel 1.

Tabel 1 Rentang Data

Fitur	Data range	Data Type
RT	1 – 8	Numerik
RW	1 – 16	Numerik
IUD	0 – 39	Numerik
MOW	0 – 7	Numerik
MOP	0 – 1	Numerik
Implant	0 – 3	Numerik
Suntik	0 – 86	Numerik
PIL	0 – 41	Numerik
Kondom	0 – 14	Numerik

Nama fitur dan rentang nilainya tercantum dalam Tabel 1. Terdapat sembilan fitur yang diketahui: RT, RW, IUD, MOW, MOP, Implant, Suntik, PIL, dan Kondom. Rentang data untuk fitur RT adalah 1–8, sedangkan RW adalah 1–16. Pengguna alat kontrasepsi IUD berkisar antara 0 dan 39, sementara MOW berkisar antara 0 dan 7, dan MOP berkisar antara 0 dan 1. Pengguna alat kontrasepsi implant berkisar antara 0 dan 3, sedangkan pengguna suntik berkisar antara 0 dan 86. Pengguna pil berkisar antara 0 dan 41, dan pengguna kondom berkisar antara 0 dan 14. Untuk memahami seberapa besar variasi penggunaan alat kontrasepsi di setiap RT/RW, rentang ini memberikan gambaran tentang penyebaran penggunaan alat kontrasepsi di Desa Kalirejo Lawang.



Gambar 3 Pengguna Alat Kontrasepsi Desa Kalirejo Lawang

Di Desa Kalirejo Lawang, pengguna alat kontrasepsi bervariasi di RT dan RW, seperti yang ditunjukkan pada Gambar 3. Metode kontrasepsi yang paling umum digunakan (460 peserta) adalah suntikan diikuti oleh IUD (349 peserta) dan pil (171 peserta). Sementara itu, metode MOP memiliki jumlah pengguna yang paling sedikit (1 peserta), menunjukkan tingkat adopsi yang rendah untuk metode ini di kalangan peserta KB.

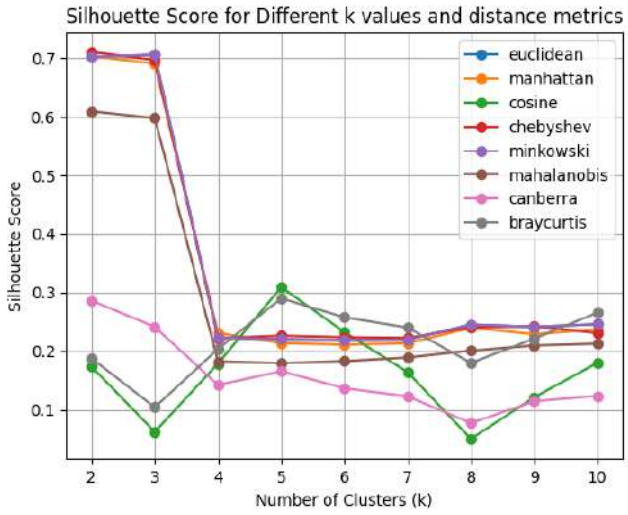
Selanjutnya eksperimen klasterisasi dengan K-Means diterapkan sesuai dengan skenario pengujian pada metodologi sebelumnya. Eksperimen klasterisasi dilakukan dengan menerapkan metode K-Means menggunakan berbagai *distance measure*, dengan jumlah klaster k yang bervariasi dari 2 hingga 10. Evaluasi kualitas klasterisasi dilakukan menggunakan SC. Tabel 2 menyajikan hasil eksperimen untuk masing-masing *distance measure*.

Tabel 2 Hasil Eksperimen Pengujian dengan Evaluasi SC

Distance	2	3	4	5	6	7	8	9	10
euclidean	**0,6990**	0,2321	0,2173	0,2172	0,2163	0,2353	0,2255	0,2415	0,2405
manhattan	**0,7191**	0,2407	0,2264	0,2270	0,2255	0,2208	0,2202	0,2488	0,2289
cosine	0,1716	**0,2166**	0,1826	0,0926	0,0387	0,1711	0,0803	0,1444	0,1843
chebyshev	**0,6821**	0,2328	0,2145	0,2085	0,2105	0,2453	0,2190	0,2202	0,2290
minkowski	**0,6990**	0,2321	0,2173	0,2172	0,2163	0,2353	0,2255	0,2415	0,2405
mahalanobis	**0,5887**	0,1913	0,1784	0,1810	0,1879	0,1949	0,2022	0,1884	0,1927
canberra	**0,2664**	0,1490	0,1457	0,1212	0,0981	0,1063	0,0989	0,1131	0,1163
braycurtis	0,1938	0,2529	0,2078	0,1748	0,1499	0,2357	0,2079	0,2532	**0,2746**

Berdasarkan Tabel 2 yang disajikan, eksperimen ini bertujuan untuk mengevaluasi performa berbagai ukuran jarak dalam algoritma K-Means dengan menggunakan *silhouette coefficient* (SC) sebagai metrik evaluasi. SC mengukur seberapa baik suatu klaster terpisah dan seberapa kohesif klaster tersebut. Semakin tinggi nilainya, semakin baik hasil klasterisasi. Dari hasil yang diperoleh, jarak Manhattan memiliki nilai SC tertinggi pada k=2 sebesar 0,7191, mengindikasikan bahwa dengan kecilnya jumlah klaster, Manhattan memberikan pemisahan terbaik. Namun, seiring bertambahnya jumlah klaster (k), performa Manhattan menurun, dan untuk k=10, nilai SC menurun menjadi 0,2289. Sebaliknya, jarak Bray-Curtis memiliki performa terbaik pada k=10 dengan nilai SC sebesar 0,2746, menunjukkan bahwa jarak ini lebih efektif untuk jumlah klaster yang lebih besar.

Sementara itu, jarak Euclidean dan Minkowski memiliki pola nilai yang sama, karena Minkowski adalah generalisasi dari Euclidean. Keduanya memiliki performa cukup baik pada k=2 (0,6990), tetapi menurun seiring bertambahnya k. Ukuran jarak Cosine memiliki performa paling buruk, dengan nilai SC yang rendah di hampir semua nilai k, terutama pada k=5 di mana SC turun drastis ke 0,0387. Hal ini menunjukkan bahwa *cosine similarity* tidak cocok untuk pengelompokan berbasis jarak dalam konteks ini. Selain itu, jarak Canberra memiliki nilai tinggi di k=2 (0,2664), tetapi turun signifikan ketika k bertambah, menunjukkan bahwa Canberra kurang stabil dalam menangani klaster yang lebih kompleks.



Gambar 4 Grafik Hasil Silhoutte Score untuk Setiap Klaster dan *Distance Measure*

Grafik SC terhadap berbagai nilai k dan metrik jarak dalam metode K-Means ditunjukkan pada Gambar 4. Grafik tersebut menunjukkan bahwa pemilihan jumlah klaster dan metrik jarak berperan krusial dalam menentukan kualitas klasterisasi. Pada k=2, terlihat bahwa metrik Euclidean, Manhattan, Chebyshev, dan Minkowski memiliki skor tertinggi, mendekati 0,7, yang mengindikasikan bahwa data terbagi dengan baik ke dalam dua klaster. Metrik Mahalanobis juga menunjukkan performa cukup baik pada k=2, meskipun sedikit lebih rendah dibandingkan metrik lainnya. Namun, ketika nilai k meningkat menjadi 3, terjadi penurunan drastis pada hampir semua metrik, terutama pada Cosine yang memiliki skor terendah di antara semua metode. Penurunan ini menunjukkan bahwa dengan lebih banyak klaster, pemisahan antar kelompok menjadi kurang jelas, sehingga menurunkan koherensi internal klaster.

Setelah k=4, skor Silhouette cenderung stabil dengan fluktuasi kecil, tetapi tetap jauh lebih rendah dibandingkan nilai awal pada k=2. Metrik Cosine mengalami sedikit peningkatan pada k=5, meskipun secara keseluruhan tetap lebih rendah dibandingkan metrik lainnya. Sementara itu, metrik Canberra secara konsisten menunjukkan skor yang lebih rendah dibandingkan metrik lainnya, mengindikasikan bahwa metode ini kurang efektif dalam menentukan pemisahan klaster pada dataset ini. Metrik Bray-Curtis menunjukkan fluktuasi yang lebih tinggi dibandingkan metrik lainnya, dengan sedikit peningkatan pada nilai k yang lebih besar, tetapi tetap dalam rentang yang relatif rendah.

Secara keseluruhan, hasil ini menunjukkan bahwa jumlah klaster yang optimal berdasarkan Silhouette Score adalah k=2, karena setelah nilai ini, kualitas klasterisasi menurun signifikan. Selain itu, metrik jarak seperti Euclidean, Manhattan, Chebyshev, dan Minkowski lebih cocok digunakan untuk dataset ini dibandingkan Cosine, Canberra, dan Bray-Curtis. Oleh karena itu, dalam pemilihan parameter untuk K-Means, perlu dilakukan analisis mendalam terhadap sifat data agar dapat menentukan kombinasi jumlah klaster dan metrik jarak yang optimal untuk hasil klasterisasi yang lebih tepat.

Tabel 3 Hasil Pengujian Terbaik Tiap Klaster dan *Distance Measure*

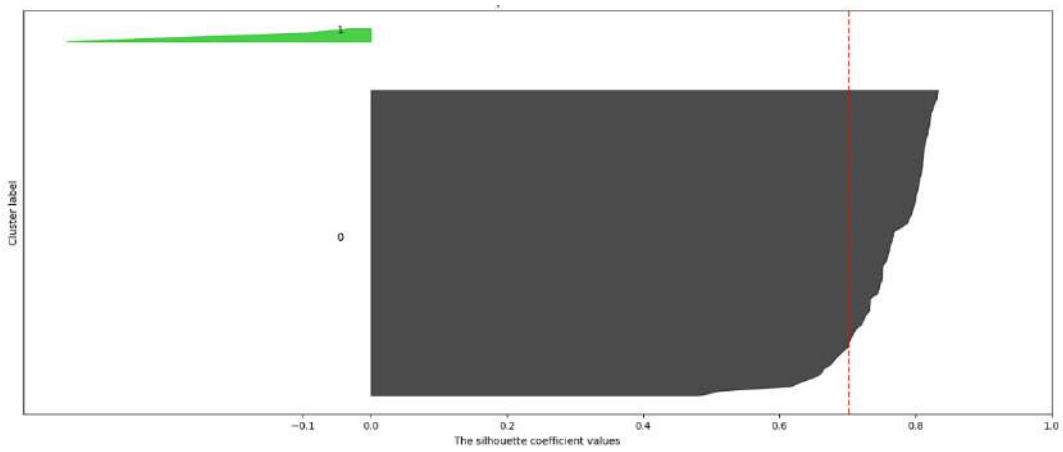
Distance Measure	Silhouette Coefficient (SC)
euclidean (k = 2)	0,6990
manhattan (k = 2)	**0,7191**
cosine (k = 3)	0,2166
chebyshev (k = 2)	0,6821
minkowski (k = 2)	0,6990

mahalanobis (k = 2)	0,5887
canberra (k = 2)	0,2664
braycurtis (k = 10)	0,2746

Berdasarkan Tabel 3, hasil klasterisasi K-Means dengan berbagai metrik jarak dan jumlah klaster k terbaik, dapat dilakukan analisis terhadap efektivitas masing-masing pendekatan dalam menghasilkan klaster yang lebih terstruktur. SC digunakan sebagai indikator kualitas klaster, dengan nilai yang lebih tinggi menunjukkan bahwa objek dalam satu klaster lebih mirip satu sama lain dibandingkan dengan objek di klaster lain. Dari tabel, terlihat bahwa metrik jarak Manhattan (k=2) memiliki nilai SC tertinggi, yaitu 0,7191, menunjukkan bahwa pendekatan ini menghasilkan klaster yang paling baik dibandingkan dengan metrik lainnya. Hal ini mengindikasikan bahwa dalam ruang vektor yang digunakan, penggunaan jarak Manhattan lebih efektif dalam mengelompokkan data dengan batasan yang lebih jelas dibandingkan metrik lainnya seperti Euclidean (0,6990) atau Minkowski (0,6990), yang memiliki karakteristik perhitungan jarak yang serupa.

Sebaliknya, metrik jarak Cosine (k=3) menghasilkan nilai SC terendah, yaitu 0,2166, yang menunjukkan bahwa metode ini kurang efektif dalam mengelompokkan data dalam konteks yang digunakan. Hal ini bisa terjadi karena jarak Cosine lebih berfokus pada sudut antara vektor daripada jauh jarak, sehingga mungkin kurang sesuai untuk struktur data yang berbasis perbedaan absolut antar titik. Jarak Chebyshev (k=2) juga menunjukkan performa yang cukup baik dengan SC 0,6821, yang berarti pendekatan ini dapat menangkap pola tertentu dalam data dengan cukup efektif. Namun, metrik lain seperti Mahalanobis (k=2) menghasilkan nilai yang lebih rendah (0,5887), yang mungkin disebabkan oleh sensitivitasnya terhadap korelasi antar variabel dan distribusi data yang tidak sesuai dengan asumsi Mahalanobis. Dua metrik lainnya, Canberra (k=2) dan Bray-Curtis (k=10), menunjukkan hasil yang lebih rendah, masing-masing dengan nilai 0,2664 dan 0,2746. Hal ini menunjukkan bahwa pendekatan berbasis perbandingan relatif atau distribusi antar nilai mungkin kurang optimal dalam menangkap pola klasterisasi dalam dataset yang digunakan.

Secara keseluruhan, analisis ini menunjukkan bahwa pemilihan metrik jarak yang tepat dalam K-Means sangat berpengaruh terhadap kualitas klasterisasi. Berdasarkan evaluasi yang dilakukan, jarak Manhattan dengan k=2 terbukti sebagai pilihan terbaik dalam konteks ini. Temuan ini menunjukkan bahwa data dalam dataset tersebut lebih sesuai untuk diukur dengan perbedaan absolut dibandingkan dengan pendekatan berbasis sudut atau distribusi relatif.

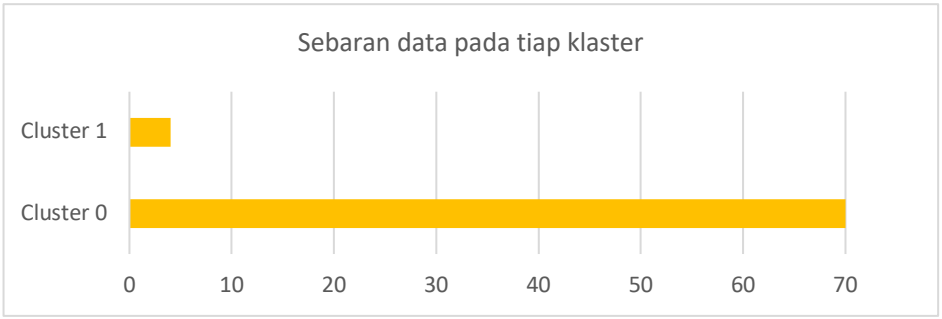


Gambar 5 Hasil Silhoutte Plot

Gambar 5 menunjukkan grafik *sillhouette plot* untuk menampilkan hasil klasterisasi menggunakan K-Means dengan k=2 dan metrik jarak Manhattan, yang sebelumnya diketahui memberikan nilai SC tertinggi (0,7191) dibandingkan metrik lainnya. Grafik ini membantu dalam memahami

bagaimana setiap sampel dalam klaster terdistribusi berdasarkan koefisien silhouette, yang mencerminkan seberapa baik objek berada dalam klasternya dibandingkan dengan klaster lain. Dari grafik, dapat diamati bahwa terdapat dua klaster utama, yaitu klaster 0 (hitam) dan klaster 1 (hijau). Klaster 0 mendominasi, dengan sebagian besar sampelnya memiliki nilai SC yang bervariasi, namun tetap berada dalam rentang yang menunjukkan kualitas klasterisasi yang cukup baik. Sementara itu, klaster 1 memiliki jumlah sampel yang jauh lebih sedikit tetapi memiliki nilai silhouette yang tinggi dan stabil, menandakan bahwa objek dalam klaster ini lebih jelas terpisah dari klaster lain.

Garis merah putus-putus yang ditampilkan pada sekitar 0,7191 menunjukkan rata-rata SC, yang relatif tinggi. Hal ini mengindikasikan bahwa mayoritas titik data memiliki pemisahan yang baik antara klaster, dengan hanya sebagian kecil sampel yang memiliki nilai SC negatif atau mendekati nol. Beberapa sampel dalam klaster 0 memiliki nilai SC yang mendekati nol atau negatif, yang berarti ada kemungkinan objek-objek tersebut berada di batas antara klaster dan memiliki kemiripan dengan klaster lainnya. Dari perspektif analisis klasterisasi, hasil ini mengonfirmasi bahwa jarak Manhattan dengan k=2 merupakan pilihan yang optimal dalam mendefinisikan struktur klaster, karena mayoritas sampel memiliki nilai SC yang positif dan cukup tinggi. Klasterisasi ini juga lebih stabil dibandingkan beberapa metrik lainnya yang sebelumnya memiliki nilai silhouette yang jauh lebih rendah. Namun, distribusi yang sangat tidak seimbang antara klaster 0 dan klaster 1 bisa menjadi indikasi bahwa data memiliki struktur yang secara alami condong ke satu kelompok besar dengan satu kelompok kecil yang lebih terisolasi.



Gambar 6 Grafik sebaran data pada klaster k = 2 dengan *manhattan distance*

Grafik yang ditampilkan pada Gambar 6 menunjukkan sebaran jumlah sampel dalam masing-masing klaster berdasarkan hasil klasterisasi K-Means dengan k=2 menggunakan metrik jarak Manhattan. Dari visualisasi ini, terlihat bahwa klaster 0 mendominasi dengan jumlah sampel yang jauh lebih besar dibandingkan klaster 1, yang hanya memiliki sedikit anggota. Ketimpangan ini mengindikasikan bahwa data yang digunakan memiliki struktur yang cenderung mengelompok ke dalam satu klaster utama, sementara klaster lainnya berisi sejumlah kecil sampel yang lebih terisolasi. Jika dikaitkan dengan silhouette plot sebelumnya, pola ini semakin memperkuat temuan bahwa klaster 1 memiliki nilai SC yang tinggi dan stabil, yang berarti objek dalam klaster ini memiliki jarak yang jelas terhadap klaster lain. Sebaliknya, klaster 0 berisi mayoritas sampel, tetapi memiliki variasi dalam nilai SC, dengan beberapa objek kemungkinan berada di batas antara klaster atau memiliki kedekatan dengan klaster lainnya. Namun, karena rata-rata SC masih tinggi (0,7191), bisa disimpulkan bahwa meskipun ada ketimpangan jumlah sampel antar klaster, pemisahan antar klaster cukup jelas dan valid. Hasil klasterisasi menunjukkan bahwa Klaster 0 didominasi peserta dengan metode suntik dan pil, sedangkan Klaster 1 lebih banyak peserta dengan metode IUD. Implikasi ini penting bagi pengambil kebijakan, karena klasterisasi dapat membantu menentukan strategi penyuluhan dan distribusi alat kontrasepsi yang lebih tepat sasaran.

4. KESIMPULAN

Penelitian ini menegaskan bahwa pemilihan ukuran jarak (*distance measure*) memiliki pengaruh signifikan terhadap kualitas hasil klasterisasi data peserta program Keluarga Berencana (KB) di

Desa Kalirejo. Berdasarkan hasil eksperimen menggunakan delapan jenis ukuran jarak, ditemukan bahwa algoritma K-Means dengan Manhattan distance pada k=2 memberikan hasil terbaik dengan Silhouette Coefficient (SC) sebesar 0,7191, menandakan struktur klaster yang sangat kuat dan terpisah dengan baik. Sebaliknya, ukuran jarak Cosine dan Canberra menunjukkan kinerja terendah dengan nilai SC masing-masing sebesar 0,2166 dan 0,2664, yang menandakan bahwa kedua ukuran tersebut kurang cocok digunakan pada dataset dengan karakteristik seperti ini. Temuan ini memperkuat bukti empiris bahwa pemilihan metrik jarak yang sesuai harus mempertimbangkan distribusi dan skala data. Hasil penelitian memberikan kontribusi metodologis dalam penerapan K-Means untuk data demografis, khususnya dalam konteks kebijakan dan perencanaan program KB berbasis data. Implikasi praktisnya adalah pemerintah daerah dan BKKBN dapat memanfaatkan pendekatan klasterisasi dengan ukuran jarak optimal untuk mengidentifikasi pola penggunaan kontrasepsi dan merancang strategi intervensi yang lebih efektif. Penelitian lanjutan disarankan untuk mengeksplorasi algoritma klasterisasi lain seperti K-Medoids atau DBSCAN guna membandingkan ketahanan hasil terhadap noise dan bentuk data yang kompleks.

DAFTAR PUSTAKA

Andriyani, W., Anshori, M., Normawati, D., Pradini, R. S., Zaenudin, M., Harisuddin, M. I., Haris, M. S., Astuty Sitinjak A. A., & Kusuma, W. T. (2024). *Matematika pada Kecerdasan Buatan*. Tohar Media.

Argiento, R., Filippi-Mazzola, E., & Paci, L. (2025). Model-Based Clustering of Categorical Data Based on the Hamming Distance. *Journal of the American Statistical Association*, *120*(550), 1178–1188. https://doi.org/10.1080/01621459.2024.2402568

Ghosh, A. (2022). Prediction of Diabetes Using Random Forest and XGBoost Classifiers. *International Journal of Computer Science Engineering and Information Technology Research (IJCSEITR)*, *12*(1), 19–28.

Hutagalung, J. (2022). Pemetaan Siswa Kelas Unggulan Menggunakan Algoritma K-Means Clustering. *JATISI (Jurnal Teknik Informatika dan Sistem Informasi)*, *9*(1), 606–620. https://doi.org/10.35957/jatisi.v9i1.1516

Hutagalung, J., & Sonata, F. (2021). Penerapan Metode K-Means untuk Menganalisis Minat Nasabah. *Jurnal Media Informatika Budidarma*, *5*(3), 1187–1194. https://doi.org/10.30865/mib.v5i3.3113

Idrus, A., Tarihoran, N., Supriatna, U., Tohir, A., Suwarni, S., & Rahim, R. (2022). Distance Analysis Measuring for Clustering Using K-Means and Davies Bouldin Index Algorithm. *TEM Journal*, *11*(4), 1871–1876. https://doi.org/10.18421/TEM114-55

Jollyta, D., Prihandoko, P., Priyanto, D., Hajjah, A., & Nora Marlim, Y. (2023). Comparison of Distance Measurements Based on k-Numbers and Its Influence to Clustering. *MATRIK: Jurnal Manajemen, Teknik Informatika dan Rekayasa Komputer*, *23*(1), 93–102. https://doi.org/10.30812/matrik.v23i1.3078

Kumala, A. A. S. P. A., & Rahning Putri, L. A. A. (2023). Measure Comparison Distance on K-Means Clustering for Grouping Music on Mood. *JELIKU (Jurnal Elektronik Ilmu Komputer Udayana)*, *11*(4), 663–674. https://doi.org/10.24843/JLK.2023.v11.i04.p03

Lino, M., Jedo, A., & Adam, C. V. (2021). Identifikasi Faktor-Faktor yang Mempengaruhi Pengambilan Keputusan Pasangan Usia Subur dalam Mengikuti Program KB (Studi Kasus di Desa Leraboleng Kecamatan Titehena Kabupaten Flores Timur). *Jurnal Administrasi dan Demokrasi*, *1*(2), 101–123. https://doi.org/10.35508/jad.v1i2.5599

Maharani, S., & Yotenka, R. (2024). Pengelompokan Kecamatan di Daerah Istimewa Yogyakarta Berdasarkan Jumlah Pengguna Alat Kontrasepsi Tahun 2022 dengan K-Medoids Cluster. *Emerging Statistics and Data Science Journal*, *2*(2), 222–237. https://doi.org/10.20885/esds.vol2.iss.2.art16

Maulana, P. F. (2020). Upaya Dinas Kesehatan dan Keluarga Berencana dalam Pelaksanaan Kebijakan Keluarga Berencana di Kelurahan Bontang Lestari Kota Bontang. *EJournal Ilmu Pemerintahan*, *8*(3), 741–754. https://ejournal.ip.fisip-unmul.ac.id/site/wp-content/uploads/2021/01/Pungki

Munawar, F., Utami, A. S. D., & Manurung, S. B. T. (2024). Klasterisasi Daerah Peserta KB Aktif di Kabupaten Asahan Menggunakan Metode K-Means. *J-Com (Journal of Computer)*, *4*(1), 58–67. https://doi.org/10.33330/j-com.v4i1.3047

Muttaqin, J. A., Harlina, S., S, W., Hakim, L., Anshori, M., Ambarwari, A., Kaunang, F. J., Sandag, G. A., Harizahayu, M. G. F., Ruslau, M. F. V, Prasetio, A., Nasir, K. R., & SIregar, M. N. H. (2023). *Data Science dan Pembelajaran Mesin*. Yayasan Kita Menulis.

Muttaqin, W. W. W., Munsarif, M., Mandias, G. F., Pungus, S. R., Widarman, A., Hapsari, W. K., Hardiyanti, S. A., Fatkhudin, A., Pasnur, B. E. F., Anshori, M. S., & Saputra, N. (2023). *Pengenalan Data Mining*. Yayasan Kita Menulis.

Ningrum, A. V. F. F., Anshori, M., & Pradini, R. S. (2025). Klasterisasi Peserta KB Aktif di Desa Kalirejo Lawang Menggunakan Metode K-Means. *Jurnal Indonesia: Manajemen Informatika dan Komunikasi*, *6*(1), 729–741. https://doi.org/10.35870/jimik.v6i1.1273

Pangestu, M. S., & Fitriani, M. A. (2022). Perbandingan Perhitungan Jarak Euclidean Distance, Manhattan Distance, dan Cosine Similarity dalam Pengelompokan Data Bibit Padi Menggunakan Algoritma K-Means. *Sainteks*, *19*(2), 141–155. https://doi.org/10.30595/sainteks.v19i2.14495

Pratistha, R. N., & Kristianto, B. (2024). Implementasi Algoritma K-Means dalam Klasterisasi Kasus Stunting pada Balita di Desa Randudongkal. *Jurnal Indonesia: Manajemen Informatika dan Komunikasi*, *5*(2), 1193–1205. https://doi.org/10.35870/jimik.v5i2.634

Rawal, K., Parthvi, A., Choubey, D. K., & Shukla, V. (2021). Prediction of Leukemia by Classification and Clustering Techniques. In P. Kumar, Y. Kumar, & M. A. Tawhid (Eds.), *Machine Learning, Big Data, and IoT for Medical Informatics* (pp. 275–295). Elsevier. https://doi.org/10.1016/B978-0-12-821777-1.00003-3

Shahapure, K. R., & Nicholas, C. (2020). Cluster Quality Analysis Using Silhouette Score. *2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA)*, 747–748. https://doi.org/10.1109/DSAA49011.2020.00096

Shapcott, Z. (2024). *An Investigation into Distance Measures in Cluster Analysis*. http://arxiv.org/abs/2404.13664

Solikhun, S., Siregar, M. R., Pujiastuti, L., Wahyudi, M., & Kurniawan, D. (2025). Comparison of Manhattan and Chebyshev Distance Metrics in Quantum-Based K-Medoids Clustering. *SISTEMASI*, *14*(4), 1562–1572. https://doi.org/10.32520/stmsi.v14i4.5193

Sumarsih, S. (2023). Hubungan Karakteristik Ibu Nifas Terhadap Pemilihan Metode Kontrasepsi Pascasalin di Puskesmas Selopampang Kabupaten Temanggung. *Sinar: Jurnal Kebidanan*, *5*(1), 1–14. https://doi.org/10.30651/sinar.v5i1.17321

Telsiz Kayaoğlu, G. İ., & Eroğlu, M. (2024). Farklı Uzaklık Fonksiyonlarının Spektral Kümeleme Algoritmasının Performansına Etkisi. *Deu Muhendislik Fakultesi Fen ve Muhendislik*, *26*(77), 237–241. https://doi.org/10.21205/deufmd.2024267706

Tifannii, W. F., Mayasari, M., & Maulana, R. (2020). Implementasi Program Keluarga Berencana (KB) Dalam Upaya Menekan Pertumbuhan Penduduk di Kelurahan Sumur Batu Kecamatan Bantar Gebang Kota Bekasi. *Jurnal Imiah Ilmu Administrasi*, *7*(3), 525–540. https://doi.org/10.25157/dinamika.v7i3.4348

Wala, J., Herman, H., & Umar, R. (2024). Implementasi K-Means Clustering pada Pengelompokan Pasien Penyakit Jantung. *JISKA (Jurnal Informatika Sunan Kalijaga)*, *9*(3), 205–216. https://doi.org/10.14421/jiska.2024.9.3.205-216

Wurdianarto, R. S., Novianto, S., & Rosyidah, U. (2014). Perbandingan Euclidean Distance dengan Canberra Distance pada Face Recognition. *Techno.COM*, *13*(1), 31–37. https://doi.org/10.33633/tc.v13i1.539

JISKA (Jurnal Informatika Sunan Kalijaga)
Vol. 11, No. 1, JANUARI, 2026, Pp. 44 – 55
ISSN: 2527 – 5836 (print) | 2528 – 0074 (online)

Sistem Deep-Learning Yolov8 untuk Deteksi Penggunaan APD Secara *Real-Time*

Nelson Mandela Rande Langi [(1)*], Arif Fadllullah [(2)]
Departemen Teknik Komputer, Universitas Borneo Tarakan, Tarakan, Indonesia
e-mail : usernelson90@gmail.com, arif.fadl@borneo.ac.id.
* Penulis korespondensi.
Artikel ini diajukan 28 Februari 2025, direvisi 17 September 2025, diterima 23 September 2025, dan dipublikasikan 25 Januari 2026.

Abstract

Although workplace safety regulations in construction are clear, many workers are still reluctant to use Personal Protective Equipment (PPE) due to a lack of awareness, work pressure, and limited facilities. As a result, the risk of serious accidents increases. Conventional approaches such as verbal warnings or CCTV monitoring are considered less effective for early detection and prevention of violations. This study proposes an automatic detection system for PPE usage in construction areas using YOLOv8. The model was trained on a secondary dataset of 3,569 images for 100 epochs, with a 60% training, 20% validation, and 20% test split. Testing on 90 real-time frames showed good performance in detecting 8 PPE classes, with an average precision of 0.935, recall of 0.806, and F1-measure of 0.862. The results indicate that the system can classify PPE usage with high accuracy. However, a recall below 1 suggests that some objects, particularly "not wearing glasses" and "not wearing shoes," failed to be detected. The F1-measure of 0.862 reflects a good balance between precision and recall.

Keywords: Personal Protective Equipment, Computer Vision, Deep Learning, YOLOv8, Object Detection

Abstrak

Meskipun aturan keselamatan kerja di konstruksi sudah jelas, banyak pekerja masih enggan menggunakan APD (Alat Pelindung Diri) karena kurangnya kesadaran, tekanan kerja, dan minimnya fasilitas. Akibatnya, risiko kecelakaan serius meningkat. Pendekatan konvensional seperti teguran lisan atau pemantauan CCTV dinilai kurang efektif untuk deteksi dini dan pencegahan pelanggaran. Penelitian ini mengusulkan sistem deteksi otomatis penggunaan APD di area konstruksi menggunakan YOLOv8. Model dilatih dengan 3.569 dataset sekunder selama 100 epoch, dengan pembagian data 60% training, 20% validasi, dan 20% testing. Pengujian pada 90 frame *real-time* menunjukkan kinerja baik dalam mendeteksi 8 kelas APD, dengan nilai masing-masing rata-rata precision 0,935, recall 0,806, dan F1-measure 0,862. Hasil menunjukkan sistem mampu mengklasifikasikan penggunaan APD dengan akurasi baik. Namun, recall di bawah 1 mengindikasikan beberapa objek, terutama "tidak pakai kacamata" dan "tidak pakai sepatu," gagal terdeteksi. Nilai F1-measure 0,862 mencerminkan keseimbangan precision dan recall yang baik.

Kata Kunci: Alat Pelindung Diri, Computer Vision, Deep Learning, YOLOv8, Deteksi Objek

1. PENDAHULUAN

Penggunaan APD merupakan salah satu implementasi sistem manajemen keselamatan dan kesehatan kerja. menurut Peraturan Menteri Tenaga Kerja dan Transmigrasi Republik Indonesia Nomor PER.08/MEN/VII/2010 Tentang Alat Pelindung Diri tahun 2010 adalah suatu alat yang biasa digunakan untuk melindungi seseorang atau pekerja dari potensi bahaya di lingkungan kerja. Meskipun ada peraturan dan pedoman yang mengatur peraturan K3 di area konstruksi, masih saja sering terjadi pelanggaran yang mengakibatkan risiko yang tidak perlu bagi para pekerja. Pelanggaran ini dapat disebabkan oleh berbagai faktor, termasuk tekanan waktu, kurangnya pengawasan, dan kurangnya konsekuensi yang tegas terhadap pelanggaran. Pada tahun 2019, sektor industri pengolahan mencatat jumlah hari kerja hilang tertinggi, yaitu 87.599 hari, dengan 10.872 kasus pekerja sementara tidak dapat bekerja. Dalam dunia konstruksi, 29%

kecelakaan kerja disebabkan oleh kejatuhan benda, 26% akibat tergelincir atau terpukul. Hingga tahun 2020, jumlah kecelakaan kerja meningkat hingga 177.000 kasus (Alfidyani et al., 2020). Khusus untuk di wilayah provinsi Kaltara (Kalimantan Utara) sendiri, berdasarkan catatan dari Badan Pusat Statistika (BPS) Kaltara, tahun 2022 ditemukan angka kecelakaan kerja sebanyak 670 kasus dan pada tahun 2023 meningkat sebanyak 685 kasus. Artinya terdapat penambahan 15 kasus kecelakaan kerja yang terjadi di Kaltara dari tahun 2022 sampai dengan 2023 (Julianti & Wibawa, 2024). Meskipun ada peraturan dan pedoman yang mengatur K3 dan penggunaan APD di area kontruksi, masih seringkali terjadi pelanggaran yang mengakibatkan risiko yang fatal bagi para pekerja. Pelanggaran ini dapat disebabkan oleh berbagai faktor, termasuk tekanan waktu, kurangnya pengawasan, dan kurangnya konsekuensi yang tegas terhadap pelanggaran K3 (Wulandari, 2023).

Penelitian serupa mengenai pendeteksi APD secara otomatis sudah pernah dilakukan sebelumnya, seperti penelitian dengan judul "*Deep learning-Based Automatic Safety Helmet Detection System for Construction Safety*" yang diteliti oleh Hayat & Morgado-Dias (Hayat & Morgado-Dias, 2022) yang menggunakan Kumpulan data *benchmark* yang berisi 5.000 gambar helm digunakan dalam penelitian ini, yang dibagi lagi dengan perbandingan 60:20:20 (%) untuk pelatihan, pengujian, dan validasi, masing-masing. Hasil eksperimen menunjukkan bahwa arsitektur YOLO tercapai rata-rata precision (mAP) terbaik sebesar 92,44%, sehingga menunjukkan hasil yang sangat baik dalam pendeteksian helm pengaman bahkan dalam kondisi cahaya redup. Akan tetapi penelitian tersebut masih menggunakan satu jenis dataset APD yaitu helm safety dan masih belum mencakup ke semua jenis APD yang digunakan pada area kontruksi seperti rompi, sepatu, kacamata, dan helm safety. Mengacu pada permasalahan yang telah diidentifikasi, penelitian ini mengusulkan sebuah sistem deteksi *real-time* menggunakan teknik deep learning, khususnya YOLOv8, untuk mendeteksi penggunaan APD secara multi-objek pada area konstruksi. Sistem ini mengintegrasikan kamera sensor dan algoritma YOLOv8 untuk secara akurat mengidentifikasi pekerja yang menggunakan atau tidak menggunakan APD.

2. METODE PENELITIAN

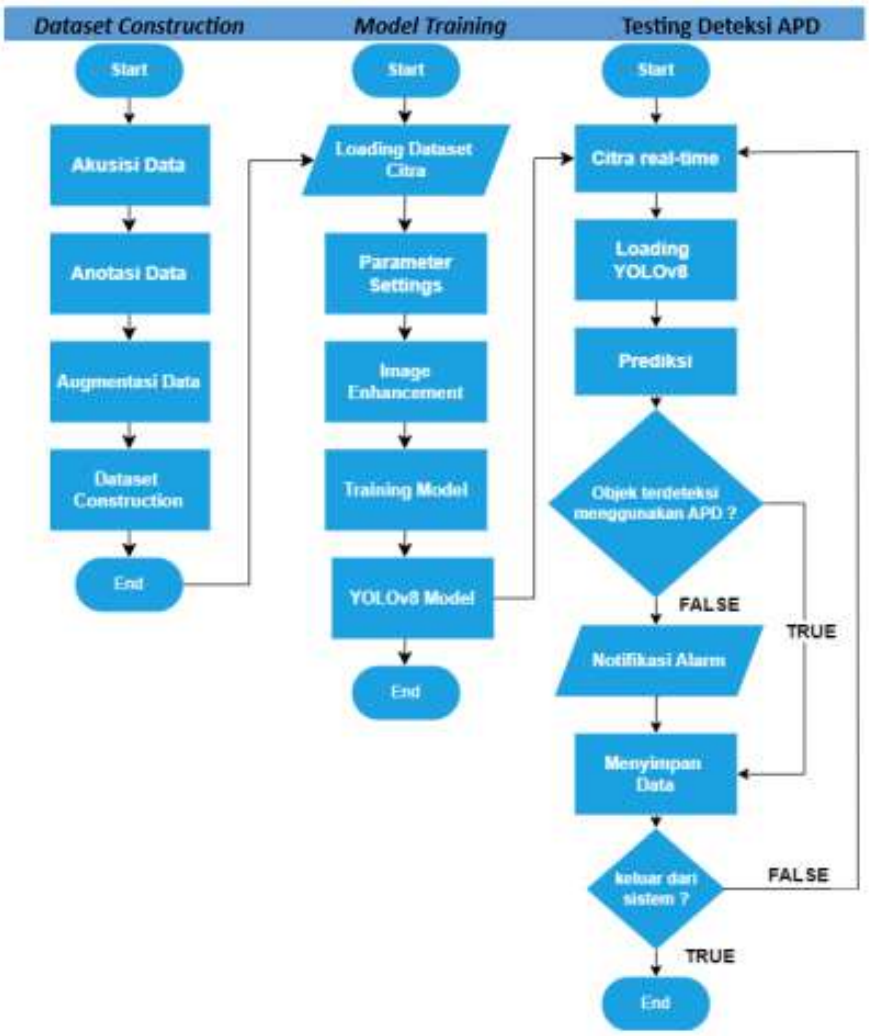
Penelitian ini mengoptimalkan pengembangan teknologi *Computer Vision* yang mengkombinasikan penggunaan kamera sensor,sistem ini juga menggunakan Deep-learning YOLOv8 (*You Only Look Once*) sebagai algoritma pemrosesan deteksi multi objek manusia yang menggunakan/tidak menggunakan APD. Kelebihan utama *Deep-learning* YOLO adalah kecepatan deteksi yang tinggi tanpa mengorbankan akurasi yang signifikan. Metode ini mampu melakukan deteksi multi objek secara *Real-time* dengan kinerja yang baik, sehingga cocok untuk berbagai aplikasi yang membutuhkan respons cepat seperti kendaraan otonom pengawasan keamanan, dan pengenalan wajah (Alfarizi et al., 2023).

2.1 Perancangan Sistem

Perancangan sistem dapat dialurkan seperti pada flowchart pada Gambar 1. Tahapan dimulai dari *dataset construction* sampai pada testing deteksi APD. *Dataset construction* merupakan langkah awal yang sangat krusial dalam pengembangan model pembelajaran mesin berbasis citra. Kualitas dan kuantitas *dataset* yang baik akan sangat mempengaruhi kinerja model yang dihasilkan. Proses ini melibatkan beberapa tahapan penting, yaitu: (1) pengumpulan data, yakni mengumpulkan citra yang relevan, bervariasi, dan berkualitas; (2) pre-processing data, seperti pembersihan, augmentasi, dan normalisasi untuk meningkatkan kualitas citra; (3) pelabelan data, yakni memberikan label pada objek atau kelas yang terdapat dalam citra; (4) pembagian data menjadi *training set*, *validation set*, dan *test set*; serta (5) penyimpanan data dengan format dan struktur yang efisien.

Visi komputer menggunakan subset dari instans data, yang dikenal sebagai data pelatihan, untuk menyesuaikan parameter suatu model. Subset lain dari data dikenal sebagai data pengujian, digunakan untuk mengevaluasi model dan memperkirakan kemampuan model untuk menggeneralisasi ke gambar yang tidak terlihat selama pelatihan (Scheuerman et al., 2021). Penelitian ini akan mengembangkan sistem deteksi otomatis APD pada area konstruksi

menggunakan algoritma Deep-learning YOLOv8 yang dilatih pada dataset citra berjumlah 3.400 citra yang akan diberi label. Dataset tersebut akan dibagi menjadi tiga kelompok, yaitu data latih (60%), data validasi (20%), dan data uji (20%) untuk meningkatkan akurasi dan generalisasi model (Hayat & Morgado-Dias, 2022).



Gambar 1 Flowchart Sistem

Tahapan model training adalah proses pelatihan kumpulan dataset citra, sehingga model dapat menemukan pola data dan memprediksi data baru. Dalam proses ini terdapat 3 tahapan utama yang akan dilakukan sistem, yaitu *image enhancement*, *parameter settings*, dan *model training*. Pelatihan model dalam visi komputer adalah proses mengajarkan komputer untuk 'memahami' gambar. Proses ini dimulai dengan mempersiapkan kumpulan data gambar yang akan digunakan untuk melatih model. Gambar-gambar ini kemudian diolah untuk meningkatkan kualitasnya agar fitur-fitur penting lebih terlihat. Setelah itu, model akan diatur parameternya, seperti jumlah lapisan dan tingkat pembelajaran. Tahap akhir adalah proses pelatihan itu sendiri, di mana model akan mempelajari pola-pola dalam data gambar secara berulang-ulang. Semakin banyak data yang digunakan dan semakin baik pengaturan parameternya, maka model akan semakin akurat dalam memprediksi atau mengklasifikasikan objek dalam gambar baru (Santos et al., 2022). Pada tahap *testing*, sistem mengambil frame video *real-time* dari lingkungan baru. Model YOLOv8 menganalisis objek dan memberi label sesuai penggunaan APD. Jika terdeteksi tidak memakai APD, sistem segera mengaktifkan alarm peringatan.

2.2 Teknik Pengumpulan Data

Penelitian ini bertujuan mengembangkan model deteksi objek *real-time* berbasis deep learning YOLOv8 untuk mendeteksi penggunaan APD pada area konstruksi. Dataset yang terdiri dari 3.400 citra berlabel akan digunakan untuk melatih, memvalidasi, dan menguji model. Pembagian dataset ini mengikuti proporsi umum dalam penelitian deep learning, yaitu 60% untuk pelatihan, 20% untuk validasi, dan 20% untuk pengujian. Setelah model dianggap memiliki akurasi yang memadai, model akan dievaluasi lebih lanjut menggunakan data sampel *real-time*. Pengumpulan data sampel akan dilakukan terhadap 3, 5, dan 7 sampel citra *real-time* objek (random menggunakan APD/tidak) sekaligus dalam 1 kali pengambilan data yang diulang tiap 30 detik sebanyak 30 kali pada frame dalam waktu tertentu secara random, sehingga akan diperoleh 90 set hasil data uji sistem terhadap precision, recall, dan F1-measure beserta rata-rata akhir dari ketiga jenis pengujian terhadap seluruh data hasil uji sistem (Hand et al., 2021).

2.3 Teknik Analisis Data

Pengujian riset akan dihitung berdasarkan metrik precision*,* recall*, dan* F1-measure. Precision, dalam konteks evaluasi model klasifikasi, adalah metrik yang mengukur proporsi dari prediksi positif yang benar terhadap total prediksi positif. Secara matematis, precision didefinisikan sebagai rasio antara *true positive* dan jumlah dari *true positive*.dan *false positive* (Prabowo, 2021). Recall adalah metrik yang mengukur proporsi dari contoh positif yang benar-benar diklasifikasikan sebagai positif terhadap total jumlah contoh positif yang sebenarnya ada. Secara matematis, recall didefinisikan sebagai rasio antara *true positive* dan jumlah dari *true positive* dan *false negative* (Wijaya et al., 2022). F1-measure adalah metrik evaluasi yang menggabungkan precision dan recall. Ini memberikan kita gambaran yang lebih komprehensif tentang kinerja model kita, terutama ketika kita ingin menyeimbangkan antara kemampuan model untuk memberikan prediksi positif yang benar (precision) dan kemampuan model untuk menemukan semua contoh positif yang sebenarnya ada (recall) (Rahma et al., 2021).

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

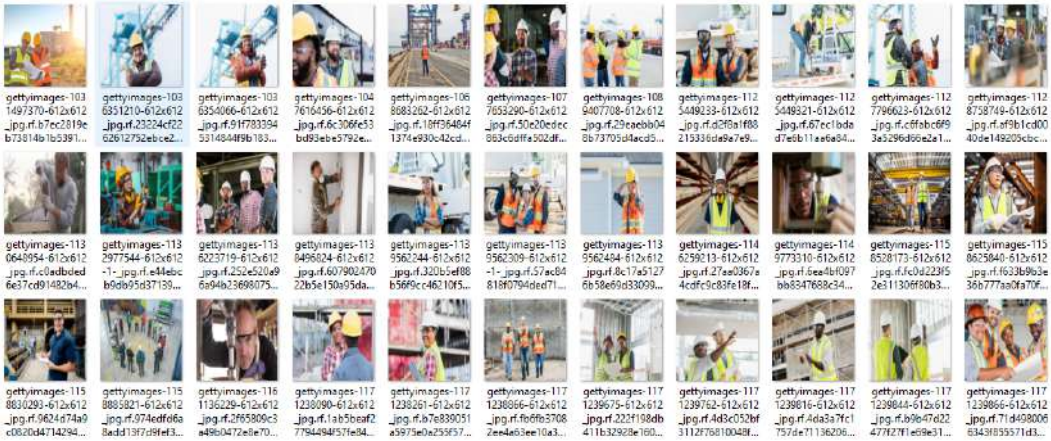
$$F1measure = \frac{2 \times precision \times recall}{precision + recall} \tag{3}$$

Cara penafsiran dan penyimpulan kebenaran hasil prediksi sistem usulan ini akan dihitung berdasarkan 3 metrik perhitungan dapat dilihat pada Pers. (1) untuk perhitungan precision, recall pada Pers. (2), dan F1-measure pada Pers. (3), yang akan digunakan untuk menghitung hasil dari data uji. Nilai-nilai tersebut berdasarkan hasil identifikasi dari: a) TP (True Positive), yaitu jumlah objek sebenarnya yang terdeteksi benar oleh sistem; b) FP (False Positive), yaitu jumlah objek noise yang terdeteksi benar oleh sistem; c) FN (False Negative), yaitu jumlah objek sebenarnya yang tidak terdeteksi benar oleh sistem (Surbakti et al., 2021).

3. HASIL DAN PEMBAHASAN

3.1 Pengumpulan Dataset

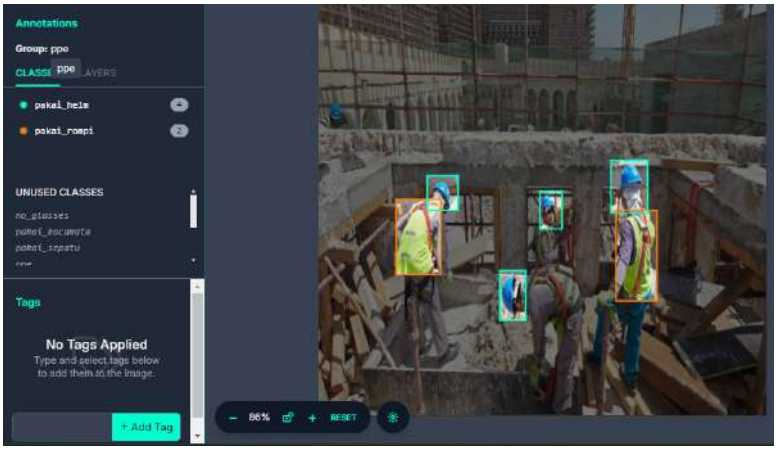
Dataset yang digunakan dalam pelatihan model berjumlah 1.740 data citra. Dataset ini diambil dari sumber pustaka https://bit.ly/CHVG_Dataset, yang menyediakan kumpulan data penggunaan APD yang relevan dan sesuai dengan kebutuhan penelitian. Contoh data dalam dataset tersebut dapat dilihat pada Gambar 2, yang menunjukkan variasi penggunaan APD di lingkungan konstruksi.



Gambar 2 Kumpulan Dataset

3.2 Anotasi Data

Anotasi dimulai dengan menentukan kelas-kelas objek yang akan dideteksi. Penentuan kelas dilakukan secara jelas agar model dapat mengidentifikasi objek dengan akurat. Gambar 3 memperlihatkan tahap awal dalam pembangunan model deteksi objek, yaitu proses anotasi data. Pada tahap ini, tujuan utama yang ingin dicapai adalah memberikan label yang sesuai dengan kelas yang ingin dideteksi sehingga model dapat belajar membedakan perbedaan antar objek (Supriyanto & Siahaan, 2024).



Gambar 3 Proses Anotasi Data

Tabel 1 Index Kelas

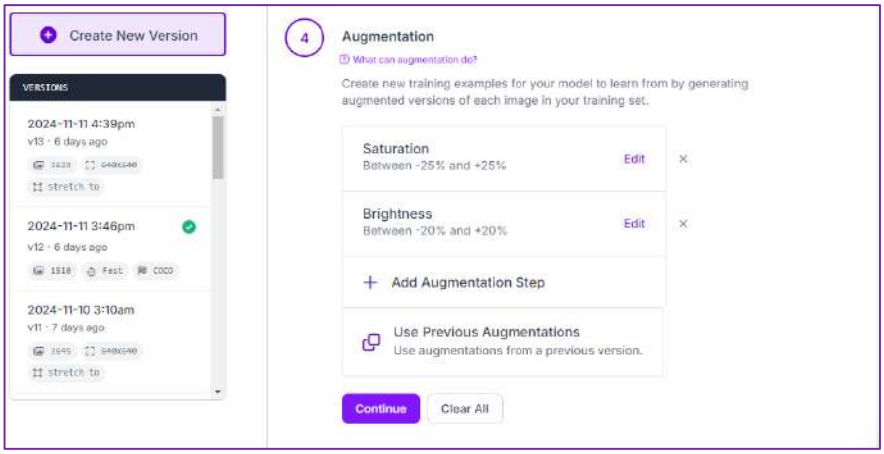
Index	Class
0	pakai_helm
1	pakai_kacamata
2	pakai_rompi
3	pakai_sepatu
4	tidak_pakai_helm
5	tidak_pakai_kacamata
6	tidak_pakai_rompi
7	tidak_pakai_sepatu

Proses anotasi dilakukan untuk seluruh kelas objek yang tercantum pada Tabel 1, dengan total 8 kelas. Setiap kelas diberikan indeks unik yang dimulai dari 0 untuk memudahkan proses

identifikasi dan klasifikasi oleh model. Proses pelabelan menghasilkan informasi berupa koordinat objek dalam bentuk *bounding box* beserta nama kelasnya. Nilai koordinat dinyatakan dalam skala 0 hingga 1. Struktur data yang dihasilkan terdiri dari lima bagian, yaitu: kolom pertama berisi indeks kelas objek; kolom kedua dan ketiga menunjukkan posisi sumbu x dan y dari titik tengah *bounding box*; kolom keempat menunjukkan lebar; dan kolom kelima menunjukkan tinggi *bounding box* (Yasen et al., 2023). Informasi indeks dan nama kelas inilah yang nantinya akan ditampilkan pada *bounding box* ketika objek berhasil terdeteksi sesuai kelasnya masing-masing.

3.3 Augmentasi Data

Untuk meningkatkan kualitas serta kuantitas data pelatihan, dilakukan proses augmentasi pada 1.500 citra awal. Augmentasi ini dilakukan dengan memanipulasi tingkat kecerahan (*brightness*) dan saturasi warna (*saturation*) pada citra, sehingga menghasilkan 3.569 citra baru yang memiliki variasi pencahayaan dan warna yang lebih beragam. Variasi tersebut diharapkan dapat meningkatkan kemampuan model dalam mengenali objek dalam berbagai kondisi pencahayaan (Ganda & Bunyamin, 2021). Proses penambahan augmentasi melalui platform Roboflow ditampilkan pada Gambar 4.



Gambar 4 Proses Augmentasi Data

3.4 Training Data

Setelah proses pengolahan data selesai, langkah selanjutnya adalah melakukan pelatihan data. Proses pelatihan dilakukan menggunakan Google Colab, sebuah platform yang menyediakan akses gratis ke GPU berbasis cloud dengan durasi hingga dua belas jam per hari. Penggunaan GPU sangat membantu karena perangkat ini dirancang untuk melakukan komputasi paralel dalam jumlah besar, sehingga secara signifikan mempercepat proses pelatihan model *machine learning* (Arip et al., 2024).

Model	size (pixels)	mAP^{val} 50-95	Speed CPU ONNX (ms)	Speed A100 TensorRT (ms)	params (M)	FLOPs (B)
YOLOv8n	640	37.3	80.4	0.99	3.2	8.7
YOLOv8s	640	44.9	128.4	1.20	11.2	28.6
YOLOv8m	640	50.2	234.7	1.83	25.9	78.9
YOLOv8l	640	52.9	375.2	2.39	43.7	165.2
YOLOv8x	640	53.9	479.1	3.53	68.2	257.8

Gambar 5 Varian Model YOLOv8

Pada tahap pelatihan dataset, model YOLO mulai mempelajari seluruh dataset yang telah melalui proses prapengolahan. Dalam penelitian ini digunakan model YOLOv8m (versi medium), yang merupakan salah satu dari lima varian YOLOv8 seperti ditunjukkan pada Gambar 5. Setiap varian memiliki karakteristik berbeda terkait ukuran model, kecepatan deteksi, serta akurasi, sehingga pemilihan varian disesuaikan dengan kebutuhan sistem. Perbandingan karakteristik antarvarian dapat dilihat pada Tabel 2.

Setelah memastikan data siap, langkah berikutnya adalah menginstal modul Python Ultralytics, yaitu modul resmi yang dikembangkan oleh tim pembuat algoritma YOLO. Modul ini menyediakan berbagai fungsi yang dibutuhkan untuk pelatihan dan pengujian model deteksi objek. Instalasi dilakukan melalui perintah *pip install ultralytics* pada terminal atau *command prompt*, seperti ditunjukkan pada Gambar 6.

```
Downloading ultralytics_thop-2.0.11-py3-none-any.whl (26 kB)
Installing collected packages: ultralytics-thop, ultralytics
Successfully installed ultralytics-8.3.29 ultralytics-thop-2.0.11
```

Gambar 6 Instalasi Modul Ultralyitics

Proses pelatihan model dilakukan dengan mengulang pembelajaran terhadap dataset sebanyak 100 kali, yang disebut dengan 100 epoch. Pada setiap epoch, model menyesuaikan parameter internalnya agar semakin akurat dalam mendeteksi objek pada citra. *Script* untuk menjalankan proses pelatihan ditunjukkan pada Gambar 7, sedangkan hasil pelatihan selama 100 epoch ditampilkan pada Tabel 3, yang memuat informasi mengenai perkembangan performa model sepanjang proses pelatihan.

```
import os

from ultralytics import YOLO


# Load a model
model = YOLO("yolov8m.yaml")  # build a new model from scratch

# Use the model
results = model.train(data=os.path.join(ROOT_DIR, "config.yaml"), epochs=100,
                      loss_weights=[1.5, 0.3, 0.2])  # train the model
```

Gambar 7 *Script* Pelatihan Data

Tabel 2 Hasil Iterasi 100 *Epoch*

Epoch	box_loss	cls_loss	dfl_loss
10	2,135	2,598	2,261
20	1,873	2,004	1,95
30	1,785	1,811	1,871
40	1,694	1,623	1,79
50	1,629	1,464	1,699
60	1,58	1,381	1,665
70	1,504	1,275	1,603
80	1,446	1,209	1,574
90	1,391	1,121	1,515
100	1,286	0,9271	1,481

3.5 Validasi Data

Validasi data dilakukan untuk menghindari kesalahan interpretasi serta hasil analisis yang bias akibat penggunaan data yang tidak valid. Melalui proses validasi, hasil dari proses *training* dapat

dievaluasi kembali sebelum dibentuk menjadi model akhir. Gambar 8 menunjukkan hasil validasi data yang dijalankan setelah proses *training* selesai. Pada tahap ini, dataset validasi memiliki peran krusial sebagai tolok ukur independen untuk mengevaluasi kinerja model setelah melalui proses pembelajaran. Dataset ini dipisahkan secara eksplisit dari dataset pelatihan dan digunakan untuk menguji kemampuan generalisasi model terhadap data yang belum pernah ditemui sebelumnya, sehingga dapat mengurangi risiko *overfitting*. Visualisasi dan informasi hasil validasi secara lebih rinci ditampilkan pada Tabel 4.

```
Validating runs/detect/train/weights/best.pt...
WARNING ⚠ validating an untrained model YAML will result in 0 mAP.
Ultralytics 8.3.29 🚀 Python-3.10.12 torch-2.5.0+cu121 CUDA:0 (Tesla T4, 15102MiB)
YOLOv8m summary (fused): 218 layers, 25,844,392 parameters, 0 gradients, 78.7 GFLOPs
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████|
                   all       2117       7928      0.769       0.83      0.889      0.565
           pakai_helm        588       1187      0.812      0.953      0.969      0.668
       pakai_kacamata        428        629      0.729      0.722      0.842      0.487
          pakai_rompi        382        760       0.81      0.949      0.974      0.686
         pakai_sepatu        326        751       0.82      0.883      0.920      0.605
     tidak_pakai_helm        359       1539      0.711      0.873      0.897      0.548
  tidak_pakai_kacamata       352       1313      0.675      0.624      0.727      0.366
     tidak_pakai_rompi       446       1254      0.764      0.891      0.945      0.602
    tidak_pakai_sepatu       361        595      0.831      0.746      0.836      0.555
Speed: 0.3ms preprocess, 9.8ms inference, 0.0ms loss, 2.5ms postprocess per image
Results saved to runs/detect/train
```

Gambar 8 Hasil Validasi Data

Tabel 3 Hasil Validasi Data

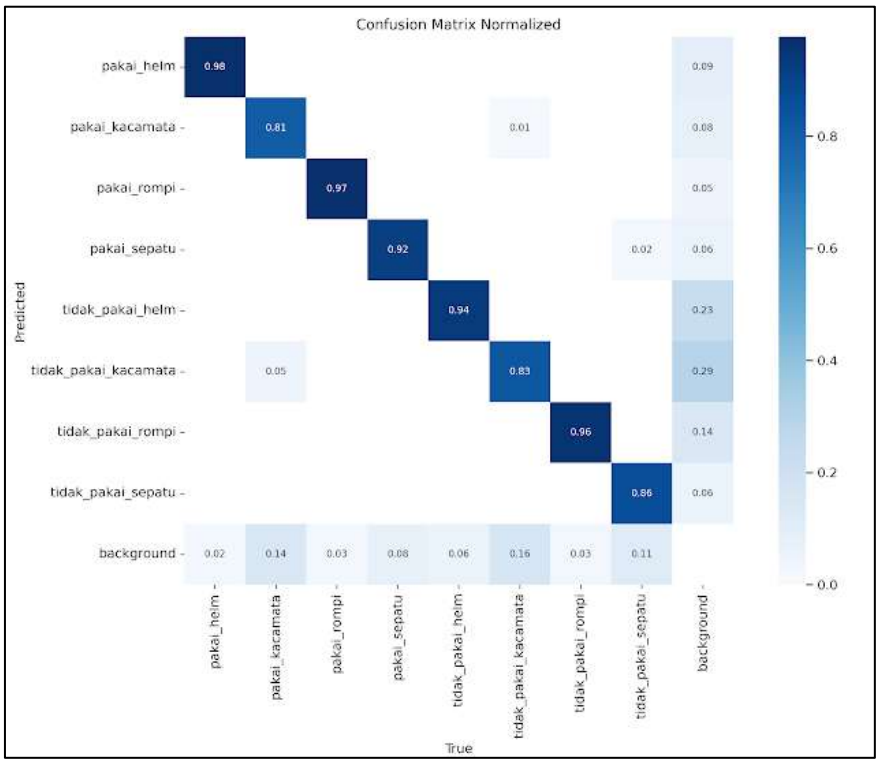
Class	images	instances	mAP
All	2117	7928	0,889
pakai_helm	588	1187	0,969
pakai_kacamata	428	629	0,842
pakai_rompi	382	760	0,974
pakai_sepatu	326	751	0,92
tidak_pakai_helm	359	1539	0,897
tidak_pakai_kacamata	352	1313	0,727
tidak_pakai_rompi	446	1254	0,945
tidak_pakai_sepatu	361	595	0,836

3.6 Evaluasi Hasil Model YOLOv8

Evaluasi model YOLOv8 merupakan langkah penting untuk memastikan bahwa model yang dibangun dapat diandalkan dan diterapkan dalam kondisi nyata. Melalui proses evaluasi, peneliti dapat memahami kinerja model, mengidentifikasi kelemahan, serta mengambil langkah yang diperlukan untuk meningkatkan performa agar sesuai dengan kebutuhan aplikasi (Muhlashin & Stefanie, 2023). Salah satu metode evaluasi yang digunakan adalah *confusion matrix*, yaitu matriks yang menggambarkan kinerja model pada data uji dengan membandingkan hasil prediksi model dengan label sebenarnya (Normawati & Prayogi, 2021). *Confusion matrix* dari hasil pelatihan model divisualisasikan pada Gambar 7. Tahap *testing* ini dilakukan setelah seluruh proses *training* dan *validation* selesai, dengan tujuan utama mengukur kemampuan model dalam mengenali serta mengklasifikasikan objek yang belum pernah ditemui sebelumnya. Dengan demikian, data *testing* berfungsi sebagai tolok ukur independen untuk mengevaluasi kemampuan generalisasi model terhadap data baru (Azhar et al., 2021).

Untuk memberikan gambaran yang lebih jelas, nilai numerik dari *confusion matrix* pada Gambar 9 ditampilkan dalam Tabel 5. Berdasarkan hasil pada tabel tersebut, evaluasi model YOLOv8 menunjukkan adanya variasi kinerja pada masing-masing kelas. Kelas dengan performa sangat baik meliputi *pakai_helm*, *pakai_rompi*, dan *tidak_pakai_rompi*, ditandai dengan nilai precision,

recall, dan F1-measure yang tinggi serta jumlah kesalahan deteksi yang rendah. Kelas *pakai_sepatu* dan *tidak_pakai_helm* memiliki performa moderat, dengan nilai precision yang perlu ditingkatkan karena jumlah *false positive* (FP) yang cukup besar, meskipun nilai recall-nya relatif baik. Sementara itu, beberapa kelas seperti *pakai_kacamata*, *tidak_pakai_kacamata*, dan *tidak_pakai_sepatu* menunjukkan performa yang kurang optimal. Rendahnya nilai precision pada kelas-kelas tersebut menunjukkan bahwa model sering melakukan kesalahan deteksi, ditambah jumlah *false negative* (FN) yang besar mengindikasikan banyak sampel relevan tidak terdeteksi. Secara khusus, kelas *tidak_pakai_kacamata* memiliki nilai precision terendah, yaitu 0,597, sehingga menjadi fokus penting untuk peningkatan performa pada penelitian selanjutnya.



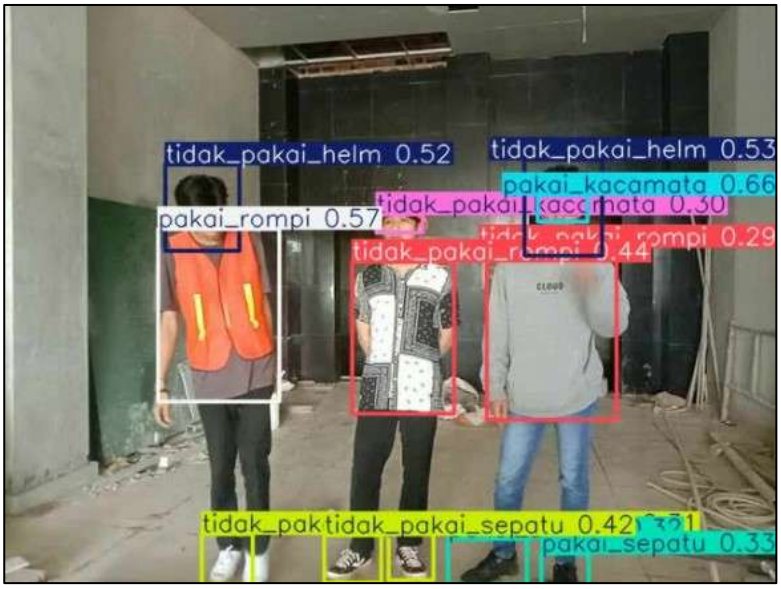
Gambar 9 Confusion Matrix

Tabel 4 Nilai Metrik Hasil Evaluasi Pelatihan

Nama Kelas	Precision	Recall	F1-measure
pakai_helm	0,839	0,978	0,903
pakai_kacamata	0,709	0,808	0,755
pakai_rompi	0,865	0,972	0,916
pakai_sepatu	0,731	0,919	0,814
tidak_pakai_helm	0,719	0,936	0,813
tidak_pakai_kacamata	0,597	0,826	0,693
tidak_pakai_rompi	0,773	0,960	0,856
tidak_pakai_sepatu	0,639	0,864	0,735

3.7 Pengujian Sistem Deteksi APD

Untuk mengevaluasi kinerja model deteksi APD pada skenario nyata, digunakan dataset uji berupa 90 frame hasil deteksi real-time yang diambil di area konstruksi gedung Fakultas Kedokteran Universitas Borneo Tarakan. Dataset ini mencakup variasi jumlah objek, kelengkapan APD, serta berbagai kondisi lingkungan yang menantang, seperti pencahayaan yang acak, jarak pengamatan berkisar 3–5 meter, dan pergerakan objek (pekerja). Gambar 10 menampilkan contoh frame dari data uji yang digunakan.



Gambar 10 Frame Data Uji

3.8 Analisis Hasil Data Uji

Meskipun model YOLOv8 telah mencapai rata-rata precision sebesar 0,935, recall sebesar 0,806, dan F1-measure sebesar 0,862 dalam mendeteksi delapan kelas APD sebagaimana ditunjukkan pada Tabel 6, masih terdapat ruang untuk peningkatan, terutama pada nilai recall. Nilai precision yang tinggi mengindikasikan bahwa ketika model memprediksi suatu objek sebagai APD, prediksinya benar sekitar 93,5% dari waktu. Namun, nilai recall sebesar 0,806 menunjukkan bahwa model hanya mampu mendeteksi sekitar 80,6% dari seluruh penggunaan APD yang terdapat dalam data uji. Kombinasi kedua metrik tersebut menghasilkan nilai F1-measure sebesar 0,862, yang menggambarkan kinerja keseluruhan yang baik. Untuk memperoleh sistem yang lebih optimal dalam menangkap seluruh kasus positif, peningkatan nilai recall menjadi fokus yang penting.

Secara keseluruhan, dengan rata-rata precision 0,935, recall 0,806, dan F1-measure 0,862, hasil evaluasi menunjukkan bahwa sistem memiliki kinerja yang sangat baik dalam mendeteksi penggunaan APD multiobjek secara *real-time*. Nilai precision yang tinggi meminimalkan kemungkinan terjadinya *false positive* (deteksi objek yang sebenarnya tidak ada), sedangkan nilai recall menunjukkan kemampuan model dalam meminimalkan *false negative* (kegagalan mendeteksi objek yang sebenarnya ada). Sementara itu, nilai F1-measure sebagai rata-rata harmonis dari precision dan recall menunjukkan adanya keseimbangan yang baik antara kedua metrik tersebut, mengindikasikan kinerja sistem yang optimal secara keseluruhan.

Tabel 5 Hasil Evaluasi 90 Data Uji

Nama Kelas	Precision	Recall	F1-measure
pakai_helm	1	0,834	0,909
pakai_rompi	1	0,897	0,946
pakai_kacamata	0,811	0,785	0,798
pakai_sepatu	0,89	0,782	0,833
tidak_pakai_helm	0,984	0,891	0,935
tidak_pakai_rompi	0,924	0,965	0,944
tidak_pakai_kacamata	0,922	0,673	0,778
tidak_pakai_sepatu	0,948	0,619	0,749
Overall	0,935	0,806	0,862

4. KESIMPULAN

Model YOLOv8 dilatih menggunakan 3.569 data dengan 100 epoch dan pembagian 60% untuk training, 20% validasi, serta 20% testing. Evaluasi pada 20% data testing menunjukkan kelas pakai_helm, pakai_rompi, dan tidak_pakai_rompi memiliki performa optimal dengan precision, recall, dan F1-measure tinggi serta tingkat kesalahan deteksi rendah. Kelas pakai_sepatu dan tidak_pakai_helm menunjukkan recall tinggi tetapi precision rendah akibat tingginya false positive. Sebaliknya, kelas pakai_kacamata, tidak_pakai_kacamata, dan tidak_pakai_sepatu memiliki performa suboptimal, dengan precision rendah dan false negative tinggi.

Pada pengujian dengan 90 frame dari data video *real-time* yang diambil secara primer, model YOLOv8 telah menunjukkan performa yang baik dalam mendeteksi 8 kelas penggunaan APD. Dengan nilai rata-rata precision 0,935, model ini cukup akurat dalam mendeteksi objek yang terdeteksi sebagai APD. Namun, nilai rata-rata recall yang masih di bawah 1 (0,806) mengindikasikan bahwa model masih melewatkan beberapa deteksi kelas pada model, terutama pada kelas tidak_pakai_kacamata dan tidak_pakai_sepatu. Nilai rata-rata F1-measure sebesar 0,862 menunjukkan bahwa model mencapai keseimbangan yang baik antara kemampuan untuk mengklasifikasikan objek dengan benar (precision) dan kemampuan untuk menemukan semua objek yang ada (recall).

Pengujian model YOLOv8 dengan menggunakan data primer sebaiknya dilakukan dengan jumlah data yang lebih banyak dan bervariasi. Data tersebut perlu mencakup berbagai skenario penggunaan Alat Pelindung Diri (APD), baik dari segi jenis maupun kondisi pemakaiannya, serta dalam beragam posisi tubuh, misalnya posisi duduk, berdiri, berjalan, maupun aktivitas lain yang relevan. Hal ini penting agar data uji memiliki karakteristik yang sebanding dengan data latih yang digunakan pada proses pengembangan model YOLOv8.

Dalam penelitian ini porsi data latih mencapai sekitar 20% dari total keseluruhan data (1.740 gambar), yaitu kurang lebih 348 gambar. Oleh karena itu, jumlah data primer yang digunakan untuk pengujian tidak ideal apabila hanya terbatas pada 90 gambar. Jumlah tersebut sebaiknya ditingkatkan agar minimal setara dengan jumlah data uji pada dataset sekunder, sehingga hasil evaluasi model menjadi lebih valid, representatif, dan dapat dipertanggungjawabkan secara ilmiah.

DAFTAR PUSTAKA

Alfarizi, D. N., Pangestu, R. A., Aditya, D., Setiawan, M. A., & Rosyani, P. (2023). Penggunaan Metode YOLO pada Deteksi Objek: Sebuah Tinjauan Literatur Sistematis. *AI dan SPK: Jurnal Artificial Intelligent dan Sistem Penunjang*, *1*(1), 54–63. https://jurnalmahasiswa.com/index.php/aidanspk/article/view/144

Alfidyani, K. S., Lestantyo, D., & Wahyuni, I. (2020). Hubungan Pelatihan K3, Penggunaan APD, Pemasangan Safety Sign, dan Penerapan SOP dengan Terjadinya Risiko Kecelakaan Kerja (Studi pada Industri Garmen Kota Semarang). *Jurnal Kesehatan Masyarakat*, *8*(4), 478–483. https://doi.org/10.14710/jkm.v8i4.27531

Arip, A. A. S., Sazali, N., Kadirgama, K., Jamaludin, A. S., Turan, F. M., & Razak, N. Ab. (2024). Object Detection for Safety Attire Using YOLO (You Only Look Once). *Journal of Advanced Research in Applied Mechanics*, *113*(1), 37–51. https://doi.org/10.37934/aram.113.1.3751

Azhar, K. M., Santoso, I., & Soetrisno, Y. A. A. (2021). Implementasi Deep Learning Menggunakan Metode Convolutional Neural Network dan Algoritma YOLO dalam Sistem Pendeteksi Uang Kertas Rupiah Bagi Penyandang Low Vision. *Transient: Jurnal Ilmiah Teknik Elektro*, *10*(3), 502–509. https://doi.org/10.14710/transient.v10i3.502-509

Ganda, L. H., & Bunyamin, H. (2021). Penggunaan Augmentasi Data pada Klasifikasi Jenis Kanker Payudara dengan Model ResNet-34. *Jurnal Strategi*, *3*(1), 187–193. https://strategi.it.maranatha.edu/index.php/strategi/article/view/246

Hand, D. J., Christen, P., & Kirielle, N. (2021). F*: An Interpretable Transformation of the F-Measure. *Machine Learning*, *110*(3), 451–456. https://doi.org/10.1007/s10994-021-05964-1

Hayat, A., & Morgado-Dias, F. (2022). Deep Learning-Based Automatic Safety Helmet Detection System for Construction Safety. *Applied Sciences*, *12*(16), Article ID: 8268. https://doi.org/10.3390/app12168268

Julianti, I., & Wibawa, Y. (2024). *Angka Kecelakaan Kerja Naik 15 Kasus*. Benuanta. https://benuanta.co.id/index.php/2024/02/07/angka-kecelakaan-kerja-naik-15-kasus/134245/18/24/08/

Muhlashin, M. N. I., & Stefanie, A. (2023). Klasifikasi Penyakit Mata Berdasarkan Citra Fundus Menggunakan YOLO V8. *JATI (Jurnal Mahasiswa Teknik Informatika)*, *7*(2), 1363–1368. https://doi.org/10.36040/jati.v7i2.6927

Normawati, D., & Prayogi, S. A. (2021). Implementasi Naïve Bayes Classifier dan Confusion Matrix pada Analisis Sentimen Berbasis Teks pada Twitter. *Jurnal Sains Komputer & Informatika (J-SAKTI)*, *5*(2), 697–711. https://doi.org/10.30645/J-SAKTI.V5I2.369

Peraturan Menteri Tenaga Kerja dan Transmigrasi Republik Indonesia Nomor PER.08/MEN/VII/2010 Tentang Alat Pelindung Diri, Kemenakertrans RI (2010). https://jdih.kemnaker.go.id/peraturan/detail/158/peraturan-menteri-nomor-8-tahun-2010

Prabowo, T. T. (2021). Efektivitas Sistem Temu Kembali Informasi Perpustakaan Digital Institut Seni Indonesia (ISI) Yogyakarta dalam Tinjauan Recall dan Precision. *Media Pustakawan*, *28*(1), 37–48. https://doi.org/10.37014/medpus.v28i1.1087

Rahma, L., Syaputra, H., Mirza, A. H., & Purnamasari, S. D. (2021). Objek Deteksi Makanan Khas Palembang Menggunakan Algoritma YOLO (You Only Look Once). *Jurnal Nasional Ilmu Komputer*, *2*(3), 213–232. https://doi.org/10.47747/jurnalnik.v2i3.534

Santos, C., Aguiar, M., Welfer, D., & Belloni, B. (2022). A New Approach for Detecting Fundus Lesions Using Image Processing and Deep Neural Network Architecture Based on YOLO Model. *Sensors*, *22*(17), Article ID: 6441. https://doi.org/10.3390/s22176441

Scheuerman, M. K., Hanna, A., & Denton, R. (2021). Do Datasets Have Politics? Disciplinary Values in Computer Vision Dataset Development. *Proceedings of the ACM on Human-Computer Interaction*, *5*(CSCW2), 1–37. https://doi.org/10.1145/3476058

Supriyanto, R., & Siahaan, D. O. (2024). Sistem Informasi Anotasi Data Penelitian : Pengolahan Data. *Jurnal Teknik ITS*, *13*(1), 98–103. https://doi.org/10.12962/j23373539.v13i1.128602

Surbakti, A. Q., Hayami, R., & Amien, J. Al. (2021). Analisa Tanggapan Terhadap PSBB di Indonesia dengan Algoritma Decision Tree pada Twitter. *Jurnal CoSciTech (Computer Science and Information Technology)*, *2*(2), 91–97. https://doi.org/10.37859/coscitech.v2i2.2851

Wijaya, D. P., Murti, L. D., & Rachman, M. R. (2022). Recall dan Precision pada Online Public Access Catalog (OPAC) Dinas Arsip dan Perpustakaan Kota Bandung. *VISI PUSTAKA: Buletin Jaringan Informasi Antar Perpustakaan*, *24*(1), 81–91. https://doi.org/10.37014/visipustaka.v24i1.2915

Wulandari, S. (2023). Memastikan Keselamatan dan Kesehatan di Industri Konstruksi: Tantangan dan Solusi K3 yang Efektif. *ARRAZI: Scientific Journal of Health*, *1*(2), 103–112. https://journal.csspublishing.com/index.php/arrazi/article/view/255

Yasen, N. M., Rifka, S., Vitria, R., & Yulindon, Y. (2023). Pemanfaatan YOLO untuk Deteksi Hama dan Penyakit pada Daun Cabai Menggunakan Metode Deep Learning. *Elektron: Jurnal Ilmiah*, *15*, 63–71. https://doi.org/10.30630/eji.0.0.397

JISKA (Jurnal Informatika Sunan Kalijaga)
Vol. 11, No. 1, JANUARI, 2026, Pp. 56 – 69
ISSN: 2527 – 5836 (print) | 2528 – 0074 (online)

Uji Efektifitas Kompresi Golomb-Rice dan Huffman untuk Metadata EXIF dalam File JPEG

Yasir Hasan
Departemen Informatika, STMIK Mulia Darma, Labuhanbatu, Indonesia
e-mail : yasirhasan.kom@gmail.com.
* Penulis korespondensi.
Artikel ini diajukan 6 Maret 2025, direvisi 29 Mei 2025, diterima 17 Juni 2025, dan dipublikasikan 25 Januari 2026.

Abstract

Compression algorithms are now called modern compression algorithms. This improvement is characterized by the combination of various classical techniques and is even based on machine learning and AI. However, the important part of compression is not only the algorithm, but also knowledge of the internal structure and metadata of the file is required. Like JPEG has a file structure that can be changed, cannot be changed, every marker (header), and EXIF metadata. Lack of knowledge of the file structure can cause data damage and file corruption. This study evaluates the compression of EXIF metadata of JPEG files using the Golomb-Rice and Huffman algorithms. Golomb-Rice can produce compression that affects the k parameter, while Huffman is optimal based on symbol frequency, but requires a code table. This study measures the effectiveness of both algorithms based on the compression ratio (CR). The test results of Golomb-Rice are more effective than those of Huffman. So, it can be concluded that the Golomb-Rice algorithm is superior in the context of compressing EXIF JPEG metadata, while Huffman shows lower efficiency in the tested scenarios.

Keywords: Golomb-Rice, Huffman, EXIF, JPEG, Compression Ratio

Abstrak

Algoritma kompresi sekarang disebut algoritma kompresi modern. Peningkatan ini dicirikan dengan penggabungkan berbagai teknik klasik dan bahkan berbasis machine learning dan AI. Namun, bagian penting kompresi bukan hanya algoritmanya, tetapi pengetahuan tentang struktur internal dan metadata file harus diketahui. Seperti JPEG memiliki struktur file yang bisa diubah, tidak bisa diubah, setiap penanda-penanda (header) dan metada EXIF. Kurangnya pengetahuan struktur file dapat menyebabkan kerusakan data dan file korup. Penelitian ini mengevaluasi kompresi metadata EXIF file JPEG menggunakan algoritma Golomb-Rice dan Huffman. Golomb-Rice dapat menghasilkan kompresi yang berpengaruh pada parameter k, sementara Huffman optimal berdasarkan frekuensi simbol, namun membutuhkan tabel kode. Penelitian ini mengukur efektivitas kedua algoritma berdasarkan rasio kompresi (CR). Hasil pengujian Golomb-Rice lebih efektif dibandingkan Huffman. Sehingga dapat disimpulkan algoritma Golomb-Rice lebih unggul dalam konteks pengompresian metadata EXIF JPEG, sementara Huffman menunjukkan efisiensi yang lebih rendah dalam skenario yang diuji.

Kata Kunci: Golomb-Rice, Huffman, EXIF, JPEG, Rasio Kompresi

1. PENDAHULUAN

Teknik kompresi yang salah dapat menyebabkan hilangnya data atau bahkan kerusakan file yang tidak dapat diperbaiki. Dalam banyak masalah, kompresi yang tidak memperhitungkan struktur file akan mengubah susunan bit secara tidak semestinya, menyebabkan file menjadi korup atau tidak dapat dibaca oleh perangkat lunak standar (Liu et al., 2022). Hal ini sering terjadi ketika kompresi dilakukan dengan algoritma yang tidak sesuai dengan karakteristik data yang dikompresi, seperti menggunakan metode lossy pada informasi yang seharusnya tetap utuh (Cai et al., 2024; Cappello et al., 2025; Mahmood & Wagner, 2023). Selain itu, beberapa teknik kompresi dapat menyebabkan metadata penting dalam file hilang, yang mengakibatkan informasi seperti tanggal pengambilan gambar atau lokasi geografis tidak lagi tersedia. Kesalahan dalam memahami format penyimpanan metadata juga dapat menyebabkan penghapusan atau

pengubahan bagian dari file yang penting untuk kompatibilitas (Doménech Fons & Pegueroles Vallés, 2021; Zheng et al., 2023). Oleh karena itu, pemilihan metode kompresi yang tepat sangat penting untuk memastikan bahwa file tetap dapat digunakan dengan baik setelah dikompresi (Martini, 2025).

File gambar memiliki struktur yang kompleks, dengan bagian-bagian yang yang dapat diubah (*editable*) dan yang tidak dapat diubah (*non-editable*). Metadata yang dapat diubah, seperti tanggal pengambilan gambar, informasi hak cipta, atau komentar pengguna, dapat dimodifikasi tanpa memengaruhi kualitas visual gambar (Mani et al., 2022; Stoilov, 2022). Namun, metadata yang tidak dapat diubah, seperti tabel kuantisasi, tabel Huffman, atau *marker* SOF (*Start of Frame*), bersifat kritis untuk merekonstruksi gambar. Kesalahan dalam memodifikasi metadata non-editable dapat menyebabkan file JPEG menjadi rusak atau tidak dapat dibaca (Itier et al., 2022; Mills, 2018). Oleh karena itu, teknik kompresi yang digunakan harus memperhatikan struktur file JPEG secara keseluruhan, termasuk metadata EXIF, untuk memastikan integritas data tetap terjaga. File gambar JPEG sebenarnya merupakan hasil kompresi dari format lain yang lebih besar, dengan menggunakan metode *Discrete Cosine Transform* (DCT) (Kumar & Kumar, 2021). Selain itu, komposisi warna dalam JPEG dikompresi dengan konversi ke ruang warna YCbCr. Meskipun telah terkompresi, file JPEG masih dapat dikompresi lebih lanjut, terutama jika mengandung metadata EXIF yang besar. Metadata ini tidak termasuk dalam kompresi utama JPEG dan sering kali menggunakan format teks atau biner yang kurang optimal dalam hal penyimpanan namun sangat berguna jika membutuhkan informasi khusus tersendiri (Acharya R et al., 2023; Ardiansyah et al., 2020; Wijayanto et al., 2016).

Teknik kompresi dapat dibagi menjadi dua kategori utama, yaitu kompresi yang mengubah format file dan kompresi yang mempertahankan format aslinya. Kompresi yang mengubah format file, seperti mengonversi JPEG ke format lain yang lebih efisien seperti PNG, WebP, atau HEIF, dapat menghasilkan pengurangan ukuran yang signifikan tetapi sering kali menyebabkan hilangnya kompatibilitas dengan perangkat lunak yang hanya mendukung format asli (Dhawan, 2011; Jamil, 2024). Sebaliknya, kompresi yang mempertahankan format awal, seperti kompresi *lossless* pada JPEG, memungkinkan metadata EXIF tetap utuh. Namun, teknik kompresi *lossless* seperti *Deflate* atau LZW tidak selalu efisien untuk metadata EXIF, yang seringkali memiliki distribusi data yang unik. Dalam konteks metadata EXIF, kompresi yang ideal adalah yang dapat mengurangi ukuran metadata tanpa mengubah format file JPEG secara keseluruhan (Agnihotri et al., 2024; Hwang et al., 2021). Dengan demikian, pengguna tetap dapat mengakses dan menggunakan gambar seperti biasa tanpa perlu mengonversi kembali ke format asli.

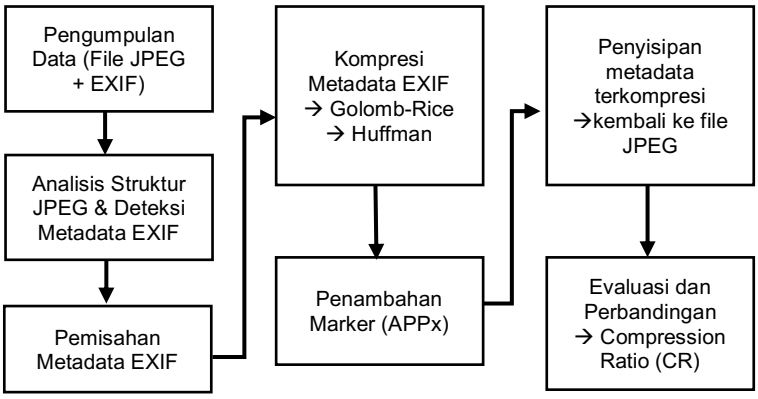
Huffman Coding dan Golomb-Rice Coding adalah dua metode kompresi data yang memiliki prinsip kerja berbeda. Golomb-Rice Coding mengandalkan pembagian nilai dengan parameter $m = 2^k$, di mana data dikodekan berdasarkan *quotient* dan *remainder*, sehingga lebih efisien untuk data dengan pola eksponensial yang sering mengandung nilai kecil berulang (Nousheen & Kumar, 2023; Žalik et al., 2021). Namun, keefektifan Golomb-Rice sangat bergantung pada pemilihan parameter k, yang jika tidak optimal, justru dapat menghasilkan ukuran kompresi yang lebih besar dibandingkan data aslinya (Himalyan & Gupta, 2023). Sebaliknya, Huffman Coding selalu memberikan kompresi paling optimal untuk data dengan distribusi acak (Kadhim et al., 2024). Huffman Coding bekerja dengan membangun pohon Huffman berdasarkan frekuensi kemunculan simbol, di mana simbol yang lebih sering muncul akan diberikan kode yang lebih pendek, sehingga cocok untuk data dengan distribusi yang tidak merata, seperti dalam format JPEG, MP3, dan ZIP. Oleh karena itu, pemilihan metode kompresi bergantung pada pola distribusi data yang dikompresi (Kumar & Kumar, 2021).

Ide untuk menerapkan kompresi Golomb-Rice dan Huffman pada metadata EXIF dalam file JPEG muncul untuk melihat performa metode klasik tersebut mengurangi ukuran file tanpa mengubah struktur file atau merusak metadata serta memperhatikan struktur file JPEG. Pendekatan ini memastikan bahwa metadata *non-editable*, yang kritis untuk integritas file, tetap tidak tersentuh. Selain itu, kompresi Golomb-Rice dan Huffman dapat diintegrasikan dengan teknik kompresi JPEG yang ada, sehingga menghasilkan gambaran yang komprehensif untuk optimasi ukuran

file dan penerapan metode klasik (Jamil, 2024). Urgensi penelitian ini terletak pengujian kompresi Golomb-Rice dan Huffman untuk mengoptimalkan ukuran file JPEG tanpa mengorbankan metadata EXIF. Oleh karena itu, penelitian ini menguji dan membandingkan efektivitas kedua metode ini dalam mengompresi metadata JPEG. Studi ini menjadi penting untuk mengetahui apakah pendekatan tradisional masih relevan atau apakah diperlukan modifikasi dan adaptasi algoritma agar sesuai dengan karakteristik metadata yang lebih spesifik. Dengan memahami keunggulan dan keterbatasan masing-masing metode.

2. METODE PENELITIAN

Penyajian diagram alur metode penelitian yang menggambarkan proses secara sistematis mulai dari pengumpulan data hingga evaluasi hasil. Penelitian ini diawali dengan pengumpulan file JPEG yang memiliki variasi ukuran metadata EXIF. Selanjutnya dilakukan analisis terhadap struktur internal file JPEG untuk mengidentifikasi letak metadata. File JPEG memiliki struktur yang terdiri dari berbagai segmen yang diawali dengan *marker* FF diikuti oleh *Byte* lain yang menentukan jenis segmen. Metadata EXIF terletak di antara *marker* FF D8 (*Start of Image*, SOI) dan sebelum *marker* FF DB (*Start of Quantization Table*, DQT) (Azeem & Fatima, 2022; Taha et al., 2022). Metadata EXIF kemudian diekstraksi dan dikompresi menggunakan dua algoritma kompresi klasik berbeda, yaitu Golomb-Rice dan Huffman. Hasil kompresi disisipkan kembali ke dalam file JPEG dengan penambahan marker khusus masing-masing algoritma untuk menandai metode kompresi yang digunakan. Langkah terakhir adalah evaluasi efektivitas kompresi berdasarkan rasio kompresi (*Compression Ratio*) untuk menilai keunggulan masing-masing metode. Diagram berikut memvisualisasikan keseluruhan proses tersebut secara runtut.



Gambar 1 Tahapan Penelitian

2.1 Identifikasi file Input

Identifikasi file input dilakukan pada tahap awal, dimulai dari pengumpulan data, analisis struktur JPEG, dan deteksi metadata EXIF hingga pemisahan metadata EXIF, sebagaimana terlihat pada Gambar 1. Bagian ini bertujuan untuk memastikan bahwa hanya metadata EXIF yang akan dikompresi, sedangkan bagian lain dari file JPEG tetap utuh agar kompatibilitas format tidak terganggu.

Langkah-langkah identifikasi file input adalah sebagai berikut:

1. Membaca file JPEG dalam format heksadesimal.
2. Mencari marker dari FF D8 hingga FF DB yang pertama muncul.
3. Mendeteksi penanda kompresi Golomb-Rice atau Huffman. Jika marker sudah ada, proses kompresi tidak dilakukan; jika marker belum ada, maka kompresi dilanjutkan.
4. Menyalin data yang berada di antara FF D8 hingga sebelum FF DB yang pertama.

2.2 Kompresi Terhadap Metadata EXIF

Setelah metadata *editable* diekstrak, langkah selanjutnya adalah menerapkan kompresi menggunakan algoritma Golomb-Rice atau Huffman. Proses kompresi Golomb-Rice terdiri dari beberapa tahapan (Boddu & Mandal, 2022). Pertama, metadata dikonversi ke dalam bentuk biner. Selanjutnya, ditentukan parameter m, seperti pada Pers. (1), di mana k merupakan parameter yang dioptimalkan. Data kemudian dibagi menjadi *quotient* (q) dan *remainder* (r) menggunakan rumus pada Pers. (2) dan (3). *Quotient* di-*encode* menggunakan *unary code*, sedangkan *remainder* di-*encode* menggunakan *binary code* sepanjang k bit. Setelah encoding selesai, data diintegrasikan kembali, diberikan penanda APP dan atribut kompresi Golomb-Rice pada awal metadata, dan disimpan sebagai metadata terkompresi.

$$m = 2^k \tag{1}$$

$$q = \lfloor n/m \rfloor \tag{2}$$

$$r = n \bmod m \tag{3}$$

Proses kompresi Huffman melibatkan beberapa langkah (Aria & Sanjaya, 2018). Pertama, dilakukan perhitungan frekuensi kemunculan setiap simbol pada metadata. Berdasarkan frekuensi tersebut, dibangun pohon Huffman untuk menentukan kode unik bagi setiap simbol. Metadata kemudian di-*encode* menggunakan kode Huffman yang telah ditetapkan. Hasil encoding diberikan penanda APP dan atribut kompresi Huffman pada awal metadata, kemudian disimpan sebagai data terkompresi.

2.3 Evaluasi

Untuk menilai efektivitas metode yang digunakan, evaluasi dilakukan dengan menggunakan perbandingan ukuran metadata file awal dan file hasil kompresi yaitu *Compression Ratio* (CR). Metrik ini mengukur seberapa banyak data dikompresi dibandingkan dengan ukuran aslinya, seperti yang ditunjukkan pada Pers. (4) (Boddu & Mandal, 2022; Hwang et al., 2021). Jika CR > 1 menunjukkan kompresi efektif (file lebih kecil setelah dikompresi) dan jika CR < 1 berarti ukuran file hasil kompresi malah lebih besar, yang menunjukkan kompresi tidak efektif.

$$CR = \frac{Ukuran\ Data\ Asli}{Ukuran\ Data\ Terkompresi} \tag{4}$$

3. HASIL DAN PEMBAHASAN

Hasil evaluasi kompresi menggunakan algoritma Golomb-Rice dan Huffman pada tiga file gambar yang diuji, dengan daftar file ditunjukkan pada Tabel 1, adalah sebagai berikut:

1) File AMAN_O.JPG
   a) Hasil kompresi Golomb-Rice dari 18 *Byte* → 25 *Byte* justru lebih besar dibandingkan ukuran aslinya. Ini menunjukkan bahwa Golomb-Rice kurang efisien untuk ukuran data yang sangat kecil atau data dengan distribusi nilai yang tidak sesuai untuk skema kompresi ini.
   b) Hasil Kompresi Huffman dari 18 *Byte* → 34 *Byte* tidak efektif, karena ukuran file setelah kompresi justru bertambah. Hal ini menunjukkan bahwa data awal mungkin sudah sangat terkompresi atau memiliki struktur yang tidak menguntungkan untuk Huffman.
2) File ME_O.JPG
   a) Hasil kompresi Golomb-Rice dari 560 *Byte* → 172 *Byte* cukup signifikan, mengurangi ukuran file menjadi sekitar 30.7% dari ukuran aslinya. Ini menunjukkan bahwa file memiliki pola yang dapat dimanfaatkan oleh Golomb-Rice untuk menghilangkan redundansi.

b) Hasil Kompresi Huffman dari 560 *Byte* → 347 *Byte* efektif, karena ada pengurangan ukuran sekitar 22%. Ini menunjukkan bahwa algoritma Huffman mampu mengoptimalkan representasi data dengan baik pada file.

3) File KUNO_O.JPG
   a) Hasil kompresi Golomb-Rice dari 616 *Byte* → 167 *Byte*. Setelah dikompresi hanya sekitar 27.1% dari ukuran awal. Ini menunjukkan bahwa data dalam file ini sangat sesuai untuk dikompresi dengan metode Golomb-Rice.
   b) Hasil Kompresi Huffman dari 616 *Byte* → 385 *Byte* efektif, dengan pengurangan sekitar 40%. Ini menunjukkan bahwa file ini memiliki banyak redundansi yang bisa dikompresi dengan Huffman.

Tabel 1 File Gambar JPEG yang diuji

No.	Nama file	File Gambar	Size file	Size EXIF
1	AMAN_O.JPG		59.8 KB	18 Byte
2	ME_O.JPG		3.45 KB	560 Byte
3	KUNO_O.JPG		42.2 KB	616 Byte

Compression Ratio (CR) dari ketiga file gambar yang telah dikompresi menggunakan Golomb-Rice dan Huffman terdapat perbedaan hasil dari panjang metadata yang digunakan untuk kompresi. Hasil CR dari ketiga file gambar yang diuji dapat dilihat pada Tabel 2, sebagai berikut.

1) Golomb-rice:
   AMAN_O.JPG → $CR = \frac{18}{25} = 0.72$ → (tidak efektif)
   ME_O.JPG→ $CR = \frac{560}{172} = 3.26$ → (efektif)
   KUNO_O.JPG→ $CR = \frac{616}{167} = 3.69$ → (efektif)
2) Huffman:
   AMAN_O.JPG → $CR = \frac{18}{34} = 0.53$ → (tidak efektif)
   ME_O.JPG→ $CR = \frac{560}{347} = 1.61$ → (efektif)
   KUNO_O.JPG→ $CR = \frac{616}{385} = 1.60$ → (efektif)

Tabel 2 Perbandingan *Compression Ratio*

Kompresi Golomb-Rice	Kompresi Huffman	CR Golomb-Rice	CR Huffman	Selisih CR
25 Byte	34 Byte	0.72	0.53	0.19
172 Byte	347 Byte	3.26	1.61	1.65
167 Byte	365 Byte	3.69	1.60	2.09

Dari hasil ini terlihat bahwa Golomb-Rice lebih unggul dibandingkan Huffman, terutama pada data dengan CR tinggi seperti ME_O.JPG dan KUNO_O.JPG. Namun pada file Aman_O.JPG, kedua metode tidak memberikan hasil yang baik karena metadata yang digunakan sangat kecil jumlahnya sehingga kompresi menjadi tidak efisien. Berdasarkan tabel perbandingan

Compression Ratio pada Tabel 2, maka dapat dijabarkan kelebihan dan keterbatasan dari metode Golomb-Rice dan Huffman pada Tabel 3.

Tabel 3 Kelebihan dan Keterbatasan Golomb-Rice dan Huffman

No.	Algoritma	Kelebihan	Keterbatasan
1	Golomb-Rice	- Efisiensi tinggi pada data berukuran sedang hingga besar. - Struktur sederhana	- Kurang efektif pada metadata kecil - Sensitif terhadap distribusi data
2	Huffman	- Kompresi stabil di berbagai ukuran data - Optimal berdasarkan frekuensi simbol	- Kurang efektif untuk data kecil atau terbatas jenis simbol - Membutuhkan tambahan header/tabel

3.1 Kelayakan Kompresi Terhadap Hasil Uji

Untuk menentukan kelayakan pemanfaatan media penyimpanan dengan kompresi pada file gambar, dapat melihat rasio antara ukuran file asli dan ukuran metadata EXIF. Perhitungan persentase metadata terhadap ukuran file dituliskan pada Pers. (5).

$$\text{Persentase Metadata} = \frac{Metada\ Exif}{Ukuran\ File} * 100\% \quad (5)$$

1) AMAN_O.JPG 59.8 KB → Metadata: 18 *Byte* → $= \frac{18}{61235} * 100\% = 0.03\%$
2) ME_O.JPG 3.45 KB → Metadata: 560 *Byte* → $= \frac{560}{3532} * 100\% = 15.86\%$
3) KUNO_O.JPG 42.2 KB → Metadata: 616 *Byte* →$= \frac{616}{43264} * 100\% = 1.42\%$

Pada gambar dengan ukuran besar AMAN_O.JPG (59.8 KB) dan KUNO_O.JPG (42.2 KB), metadata hanya mengambil porsi kecil dari total file (0.03% dan 1.42%), sehingga kompresi metadata saja tidak memberikan penghematan signifikan. Pada gambar kecil ME_O.JPG (3.45 KB), metadata diambil 15.86% dari ukuran file, sehingga kompresi metadata bisa lebih berarti dalam menghemat ruang penyimpanan.

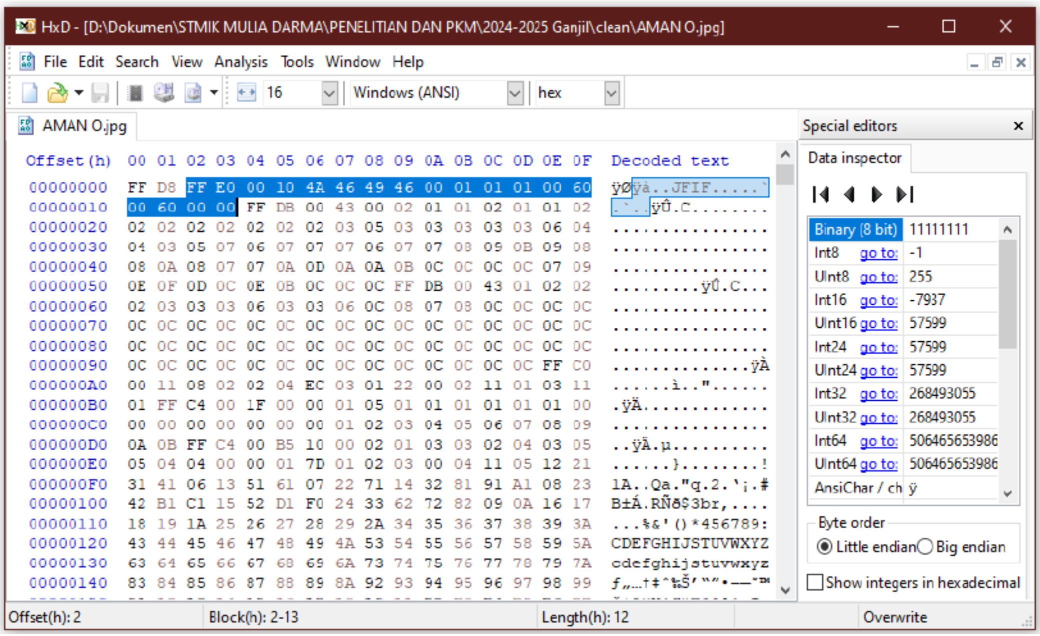
Jika tujuan utama adalah menghemat ruang penyimpanan secara keseluruhan, kompresi metadata saja tidak terlalu efektif untuk gambar berukuran besar. Lebih baik menerapkan kompresi ke seluruh gambar. Namun, untuk gambar kecil yang memiliki metadata besar, kompresi metadata bisa berkontribusi dalam mengurangi ukuran file. Jika metadata EXIF penting untuk dipertahankan, maka teknik kompresi metadata dapat berguna untuk mengurangi *overhead* penyimpanan. Implikasi praktis dari penelitian ini adalah potensi penerapan kompresi metadata EXIF menggunakan Golomb-Rice dan Huffman dalam sistem pengelolaan arsip foto digital, terutama untuk menghemat ruang penyimpanan tanpa memengaruhi kualitas visual file JPEG.

3.2 Analisis Data

Pada tahap awal, dilakukan ekstraksi metadata EXIF dari file JPEG menggunakan aplikasi HxD berguna untuk analisis Metadata Hex secara (Sunardi et al., 2020). File gambar yang digunakan untuk analisa sebanyak tiga file gambar, yaitu file yang bernama AMAN_O.JPG ukuran file 59.8 KB, ME_O.JPG ukuran file 3.45 KB, dan KUNO_O.JPG 42.2 KB. Struktur file dari semua file JPEG dianalisis untuk menentukan bagian metadata yang dapat diedit tanpa mengubah struktur utama file.

Berdasarkan hasil analisis, metadata yang dapat dikompresi terletak pada segmen dengan alamat offset 00000002h, yang diawali dengan FF E0, yaitu marker untuk segmen APP0 (JFIF header). Metadata yang dipilih dapat dilihat pada Gambar 2, dengan seleksi berwarna biru. Metadata ini berada setelah marker FF D8 (Start of Image) yang menandai awal file JPEG, hingga

marker FF DB (Define Quantization Table) yang menandai awal dari tabel kuantisasi yang bersifat tetap dan tidak dapat diubah (Hazel, 2008). Untuk perhitungan kompresi berikutnya, hanya digunakan metadata dari file AMAN_O.JPG, karena jumlah data heksadesimalnya lebih pendek dibandingkan file lainnya, yaitu 18 Byte atau 32 karakter heksadesimal. Metadata yang diperoleh dari setiap file JPEG input adalah sebagai berikut.



Gambar 2 Pembacaan file Gambar AMAN_O.JPG

1) File AMAN_O.JPG length 18 Byte:
   FF E0 00 10 4A 46 49 46 00 01 01 01 00 60 00 60 00 00
2) File ME_O.JPG length 560 Byte:
   FF E0 00 10 4A 46 49 46 00 01 01 01 00 60 00 60 00 00 FF E2 02 1C 49 43 43 5F 50 52 4F 46 49 4C 45 00 01 01 00 00 02 0C 6C 63 6D 73 02 10 00 00 6D 6E 74 72 52 47 42 20 58 59 5A 20 07 DC 00 01 00 19 00 03 00 29 00 39 61 63 73 70 41 50 50 4C 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 F6 D6 00 01 00 00 00 00 D3 2D 6C 63 6D 73 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 0A 64 65 73 63 00 00 00 FC 00 00 00 5E 63 70 72 74 00 00 01 5C 00 00 00 0B 77 74 70 74 00 00 01 68 00 00 00 14 62 6B 70 74 00 00 01 7C 00 00 00 14 72 58 59 5A 00 00 01 90 00 00 00 14 67 58 59 5A 00 00 01 A4 00 00 00 14 62 58 59 5A 00 00 01 B8 00 00 00 14 72 54 52 43 00 00 01 CC 00 00 00 40 67 54 52 43 00 00 01 CC 00 00 00 40 62 54 52 43 00 00 01 CC 00 00 00 40 64 65 73 63 00 00 00 00 00 00 00 03 63 32 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 74 65 78 74 00 00 00 00 46 42 00 00 58 59 5A 20 00 00 00 00 00 00 F6 D6 00 01 00 00 00 00 D3 2D 58 59 5A 20 00 00 00 00 00 00 03 16 00 00 03 33 00 00 02 A4 58 59 5A 20 00 00 00 00 00 00 6F A2 00 00 38 F5 00 00 03 90 58 59 5A 20 00 00 00 00 00 00 62 99 00 00 B7 85 00 00 18 DA 58 59 5A 20 00 00 00 00 00 00 24 A0 00 00 0F 84 00 00 B6 CF 63 75 72 76 00 00 00 00 00 00 00 1A 00 00 00 CB 01 C9 03 63 05 92 08 6B 0B F6 10 3F 15 51 1B 34 21 F1 29 90 32 18 3B 92 46 05 51 77 5D ED 6B 70 7A 05 89 B1 9A 7C AC 69 BF 7D D3 C3 E9 30 FF FF
3) File KUNO_O.JPG length 616 Byte:
   FF E0 00 10 4A 46 49 46 00 01 02 00 00 01 00 01 00 00 FF ED 00 36 50 68 6F 74 6F 73 68 6F 70 20 33 2E 30 00 38 42 49 4D 04 04 00 00 00 00 00 19 1C 02 67 00 14 73 62 4F 62 61 45 70 36 34 49 31 36 33 51 45 48 51 46 43 68 00 FF E2 02 1C 49 43 43 5F 50 52 4F 46 49

4C 45 00 01 01 00 00 02 0C 6C 63 6D 73 02 10 00 00 6D 6E 74 72 52 47 42 20 58 59 5A 20 07 DC 00 01 00 19 00 03 00 29 00 39 61 63 73 70 41 50 50 4C 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 F6 D6 00 01 00 00 00 00 D3 2D 6C 63 6D 73 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 0A 64 65 73 63 00 00 00 FC 00 00 00 5E 63 70 72 74 00 00 01 5C 00 00 00 0B 77 74 70 74 00 00 01 68 00 00 00 14 62 6B 70 74 00 00 01 7C 00 00 00 14 72 58 59 5A 00 00 01 90 00 00 00 14 67 58 59 5A 00 00 01 A4 00 00 00 14 62 58 59 5A 00 00 01 B8 00 00 00 14 72 54 52 43 00 00 01 CC 00 00 00 40 67 54 52 43 00 00 01 CC 00 00 00 40 62 54 52 43 00 00 01 CC 00 00 00 40 64 65 73 63 00 00 00 00 00 00 00 03 63 32 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 74 65 78 74 00 00 00 00 46 42 00 00 58 59 5A 20 00 00 00 00 00 00 F6 D6 00 01 00 00 00 00 D3 2D 58 59 5A 20 00 00 00 00 00 00 03 16 00 00 03 33 00 00 02 A4 58 59 5A 20 00 00 00 00 00 00 6F A2 00 00 38 F5 00 00 03 90 58 59 5A 20 00 00 00 00 00 00 62 99 00 00 B7 85 00 00 18 DA 58 59 5A 20 00 00 00 00 00 00 24 A0 00 00 0F 84 00 00 B6 CF 63 75 72 76 00 00 00 00 00 00 00 1A 00 00 00 CB 01 C9 03 63 05 92 08 6B 0B F6 10 3F 15 51 1B 34 21 F1 29 90 32 18 3B 92 46 05 51 77 5D ED 6B 70 7A 05 89 B1 9A 7C AC 69 BF 7D D3 C3 E9 30 FF

3.3 Proses Kompresi dan Penyisipan Marker Golomb-Rice

Setelah metadata *editable* diekstrak, langkah berikutnya adalah melakukan kompresi menggunakan algoritma Golomb-Rice. Tahapan ini dimulai dengan konversi nilai heksadesimal ke biner. Sebagai contoh, metadata dari file gambar AMAN_O.JPG yang dipilih, yaitu FF E0 00 10 4A 46 49 46 00 01 01 01 00 60 00 60 00 00, dikonversi menjadi biner, dan hasilnya ditampilkan pada Tabel 4. Setelah data biner diperoleh, langkah penting berikutnya adalah memilih parameter k untuk menentukan nilai m. Pada Golomb-Rice, m biasanya dipilih sebagai pangkat dari 2 (Pers. 1) (Hamilton et al., 2023). Dalam perhitungan ini dipilih $k = 4$, sehingga $m = 2^4 = 16$. Nilai m ini digunakan untuk menghitung *Quotient* (q) dan *Remainder* (r) dengan rumus pada Pers. (2) dan (3). Misalnya, untuk nilai metadata pertama $n_1 = 255$, diperoleh $q = \left\lfloor \frac{255}{16} \right\rfloor = 15$ dan $r = 255 \bmod 16 = 15$.

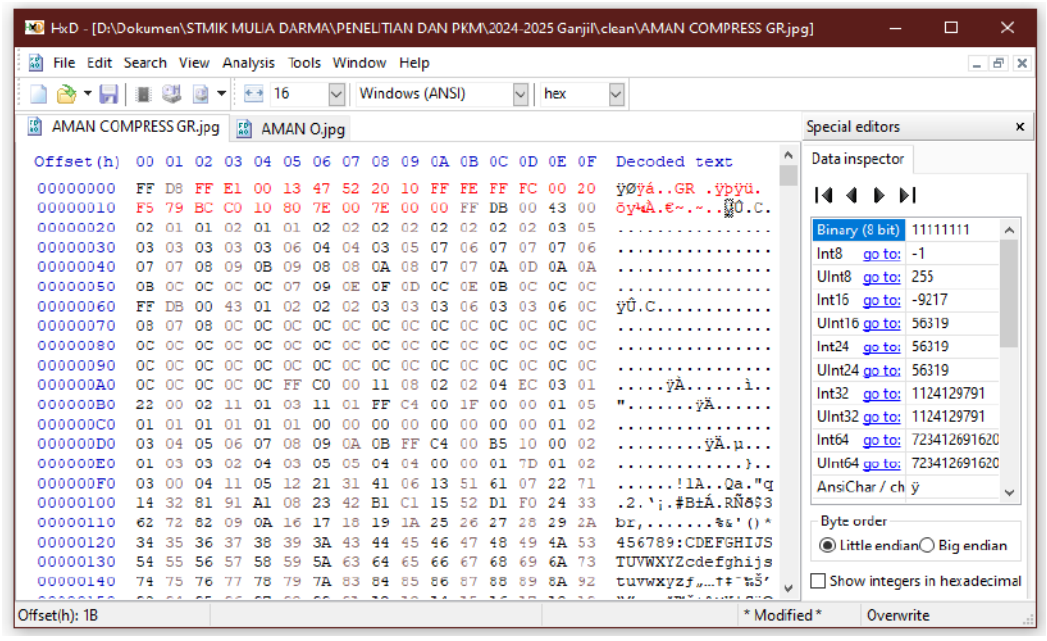
Tabel 4 Konversi Heksadesimal, Desimal ke Biner

No.	Hex	Decimal	Bin
1	FF	255	11111111
2	E0	224	11100000
3	00	0	00000000
4	10	16	00010000
5	4A	74	01001010
6	46	70	01000110
7	49	73	01001001
8	46	70	01000110
9	00	0	00000000
10	01	1	00000001
11	01	1	00000001
12	01	1	00000001
13	00	0	00000000
14	60	96	01100000
15	00	0	00000000
16	60	96	01100000
17	00	0	00000000
18	00	0	00000000

Proses yang sama diterapkan pada seluruh metadata. Nilai 255 (FF) dikompresi menjadi q = 15, yang di-*encode* dalam bentuk unary sebagai 1111111111111110, dan r = 15 di-*encode* dalam bentuk biner sebagai 1111. Kedua kode ini kemudian digabung menjadi 11111111111111101111. Encoding setiap pasangan (q, r) untuk seluruh metadata ditunjukkan pada Tabel 5. Seluruh *encoded bits* dari setiap pasangan digabungkan, dibagi menjadi 8-bit, dan dikonversi kembali ke bentuk heksadesimal. Jika terdapat kekurangan bit di akhir, ditambahkan padding 0. Hasil akhir penggabungan unary dan binary diubah menjadi heksadesimal: FF FE FF FC 00 20 F5 79 BC C0 10 80 7E 00 7E 00 00. Pada hasil akhir ini diberikan penanda (*marker*) Golomb-Rice agar file dapat dikenali untuk proses dekompresi selanjutnya, sebagaimana ditunjukkan pada Tabel 6. Proses penggantian metadata EXIF dengan hasil kompresi Golomb-Rice divisualisasikan pada Gambar 3 (Harvey, 2025).

Tabel 5 Pembentukkan Unary dan Binary

No.	q	Unary	r	Binary	Hasil
1	15	1111111111111110	15	1111	11111111111111101111
2	14	11111111111110	0	0000	11111111111100000
3	0	0	0	0000	00000
4	1	10	0	0000	100000
5	4	11110	10	1010	111101010
6	4	11110	6	0110	111100110
7	4	11110	9	1001	111101001
8	4	11110	6	0110	111100110
9	0	0	0	0000	00000
10	0	0	1	0001	00001
11	0	0	1	0001	00001
12	0	0	1	0001	00001
13	0	0	0	0000	00000
14	6	1111110	0	0000	11111100000
15	0	0	0	0000	00000
16	6	1111110	0	0000	11111100000
17	0	0	0	0000	00000
18	0	0	0	0000	00000



Gambar 3 Penggantian Metadata EXIF Dengan Hasil Kompresi Golomb-Rice

Tabel 6 Marker Golomb-Rice

Deskripsi	Heksadesimal
Marker APP1	FFE1
Panjang Data 19 Byte (2 Byte identifier + hasil kompresi)	0013
Identifier "GR " (Golomb-Rice, k=4, m=16)	47 52 20 10
Hasil kompresi Golomb-Rice	FF FE FF FC 00 20 F5 79 BC C0 10 80 7E 00 7E 00 00
Deskripsi	Heksadesimal

3.4 Proses Kompresi dan Penyisipan Marker Huffman

Langkah-langkah kompresi Huffman dimulai dengan mengonversi data heksadesimal FF E0 00 10 4A 46 49 46 00 01 01 01 00 60 00 60 00 00 menjadi bentuk biner. Selanjutnya, dilakukan perhitungan frekuensi kemunculan setiap byte, dan hasilnya disajikan pada Tabel 7. Tahap penting berikutnya adalah pembuatan pohon Huffman, yang membentuk pemetaan untuk proses kompresi. Byte dengan frekuensi paling kecil digabungkan, setiap simpul baru memiliki bobot total dari dua simpul digabungkan, dengan cabang kiri diberikan kode 0 dan cabang kanan diberikan kode 1. Hasil *encoding* dari setiap byte ditampilkan pada Tabel 8, kemudian setiap kode disusun per 8-bit. Jika kode kurang dari 8-bit, dilakukan *padding* 0 hingga mencukupi satu byte, kemudian dikonversi ke bentuk heksadesimal.

Tabel 7 Frekuensi Kemunculan *Byte*

Byte	FF	E0	00	10	4A	46	49	01	60
Frek	1	1	6	1	1	2	1	3	2

Tabel 8 Simbol Kode

Byte	Huffman Code	Panjang
0	0	1-bit
1	10	2-bit
46	1100	4-bit
60	1101	4-bit
FF	11100	5-bit
E0	11101	5-bit
10	11110	5-bit
4A	111110	6-bit
49	111111	6-bit

Tabel 9 Penyusunan data Huffman dalam APP1 (FFE1)

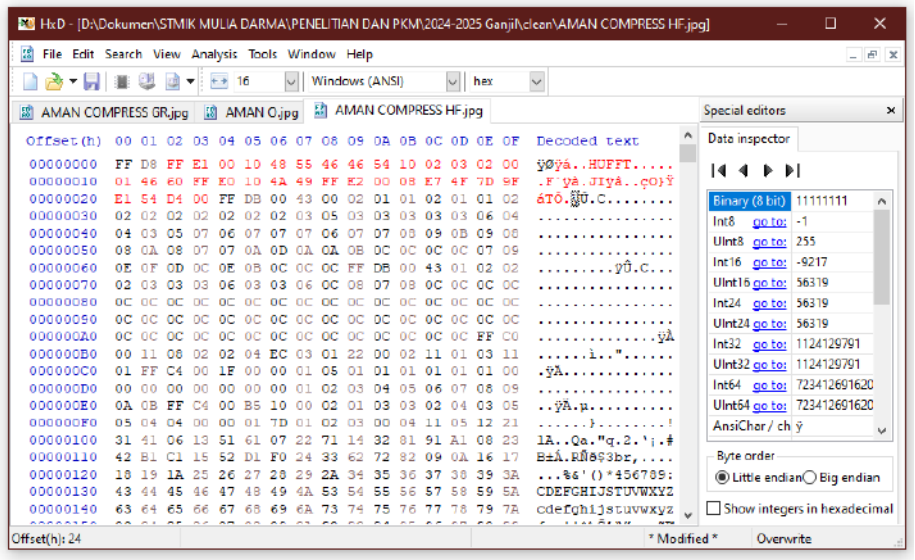
Deskripsi	Heksadesimal
Marker APP1	FFE1
Panjang Data (18 Byte)	0012
Identifier "HUFF"	48554646
Jenis Huffman DC Chrominance → 010100	54
Jumlah kode untuk setiap panjang → 010000	10
Daftar panjang kode	020302
Daftar simbol sesuai panjang kode	00 01 46 60 FF E0 10 4A 49

Untuk memungkinkan metadata JPEG dikompresi dan didekompresi dengan benar, penyisipan hasil kompresi Huffman dan tabel Huffman dilakukan ke dalam bagian metadata JPEG. Marker khusus digunakan untuk menandai bagian metadata, yaitu FF E1 sebagai Marker APP1 untuk tabel Huffman (Tabel 9) dan FF E2 sebagai Marker APP2 untuk data Huffman yang telah dikompresi (Tabel 10) (Accusoft, 2021; Harvey, 2025). Hasil penyusunan byte Huffman setelah proses pengurutan berjumlah 12 Byte, yaitu FF E2 00 0A E7 4F 7D 9F E1 54 D4 00. Hasil akhir

setelah digabungkan dengan struktur metadata JPEG menjadi 34 Byte, yaitu FF D8 FF E1 00 12 48 55 46 46 54 10 02 03 02 00 01 46 60 FF E0 10 4A 49 FF E2 00 08 E7 4F 7D 9F E1 54 D4 00 FF DB. Visualisasi proses penggantian metadata EXIF dengan hasil kompresi Huffman dapat dilihat pada Gambar 4.

Tabel 10 Hasil Kompresi ke APP2 (FFE2)

Deskripsi	Heksadesimal
Marker APP2 2 Byte	FFE2
Panjang Data (Hasil kompresi + 2 Byte panjang)	000A
Data hasil kompresi 8 Byte	E7 4F 7D 9F E1 54 D4 00



Gambar 4 Penggantian Metadata EXIF Dengan Hasil Kompresi Huffman

3.5 Hasil Kompresi ME_O.JPG DAN KUNO_O.JPG

Berikut ditampilkan hasil dari file gambar ME_O.JPG dan KUNO_O.JPG yang telah disiapkan untuk proses kompresi menggunakan algoritma Golomb-Rice dan Huffman. Pada hasil kompresi Golomb-Rice, marker ditandai dengan warna merah untuk mempermudah identifikasi lokasi marker dalam metadata. Sedangkan pada hasil kompresi Huffman, marker APP1 ditandai dengan warna merah dan marker APP2 ditandai dengan warna hijau. Pemberian warna ini bertujuan untuk memudahkan pemahaman mengenai penambahan marker pada metadata setelah kompresi, sehingga mempermudah identifikasi dan proses dekompresi pada tahap berikutnya.

1) Kompresi Golomb-Rice

File KUNO_O.JPG 8 + 164 = 172 *Byte* kompresi Golomb-Rice

FF E1 00 A6 47 52 20 10 00 44 32 14 E8 52 D8 F8 24 96 6A 29 AA BB 30 61 C9 A7 6E EE 0E 1E 2E 3E 4E 6E 8E 9E BE FF 07 87 C5 E3 F2 F9 BC FE 8F 4F AB D9 ED F7 FC 1F 0F C5 F2 7D 1F 4F D5 F6 7D BF 77 DF F8 7F 17 E3 FC 9F 97 F3 FE 8F D3 FA FF 67 ED FD DF BF F8 3F 8B F8 FF 93 F9 7F 9B F9 FF A3 FA BF B3 FB 7F C9 FE 5F F4 FF C1 FF 17 FD 3F F5 7F E0 FF C5 FF 93 FF 67 FF 0F FF 37 FF 3F FF 47 FF 7F FF 8F FF D3 FF EB FF F6 7F FB FF FE 3F FF 9B FF EA FF FB 3F FF 07 FF E2 FF FD 3F FF B7 FF F8 7F FF 97 FF F9 BF FF B3 FF FB 03

File ME_O.JPG 8 + 159 = 167 *Byte* Kompresi Golomb-Rice

FF E1 00 A1 47 52 20 10 00 44 32 9D 0A 5B 1F 04 92 CD 45 35 57 66 0C 39 34 ED E0 E2 E3 E4 E8 E9 EB EF F0 78 7C 5E 3F 2F 9B CF E9 F5 7B 3D FF 07 C3 F1 7C 9F 47 D3 F5 7D 9F 6F DD F7 FE 0F C3 F8 BF 1F E4 FC BF 9F F4 7E 9F D7 FB 3F 6F EE FD FF C1 FC 5F C7 FC 9F CB FC DF CF FD 1F D5 FD 9F DB FE 4F F2 FF A7 FE 0F F8 BF E9 FF AB FF 07

FE 2F FC 9F FB 3F F8 7F F9 BF F9 FF FA 3F FB FF FC 7F FE 9F FF 5F FF B3 FF DF FF F1 FF FC DF FF 57 FF D9 FF F8 3F FF 17 FF E9 FF FD BF FF C3 FF FC BF FF CD FF FD 9F FF 6F

2) Kompresi Huffman

File KUNO_O.JPG 32 + 405 = 437 *Byte* kompresi huffman

FF E1 00 12 48 55 46 46 54 10 02 03 02 01 02 03 04 05 06 07 08 09 0f 0A 0D 0C 0E 0B FF E2 01 97 42 D3 BE 8B A1 22 90 BB EE FB FB FC 21 68 B6 24 72 01 05 88 41 60 90 08 2C BA 22 E6 83 C4 08 E4 52 2D 4A 7F EF 53 DB B8 99 BD 18 24 E4 09 38 9E 86 C8 90 A2 28 5E 24 22 6F 43 40 1B D0 90 48 18 42 D1 D7 3D B4 52 0A 09 84 E6 E4 0A 12 29 1B 43 EF BF EE B6 4D 90 92 CC 1B 9F F2 58 9A 31 18 E3 72 31 1C E9 80 D4 6B BB 62 CD F7 EF 57 1D CA C5 44 F2 16 0B 28 F3 9D 1B FF FF FF FF FF FF F0 92 C5 DF FE B0 B9 62 6C 84 96 60 FF FF FF FF FF FF FF FF FF FF FF FF 5C 90 8D 82 43 F8 5B FC DA 10 B2 C7 31 3E F3 6F FB 56 31 88 CB 13 EF 20 7E F4 27 13 56 58 9F 7B 1B FD E8 C7 18 0D 46 BF DE AF EF 42 61 80 D4 6B FD EB A7 EF 42 71 80 D4 6B FD ED 41 FB D1 8E 34 37 20 FD ED B7 FA 49 86 86 E4 1F BD B6 FF 49 38 D0 DC 83 F7 B6 DF E9 24 23 60 90 FF FF 12 11 77 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF B1 08 D8 06 27 FA 12 3B CC 06 A3 5D DF FF 09 2C 5D FF EB 0B 96 30 1A 8D 77 7F FE 2F 2F 88 8F DC BA 18 0D 46 BB BF FE 41 5D DE 20 09 FC 55 30 1A 8D 77 7F FC 9C A5 7B 56 00 FD E0 59 73 01 A8 D7 77 FF DC 8B FE 10 27 B5 26 C2 42 C3 63 98 5F FF BD 7F DB 6B 7B 6A 89 0C D4 E8 09 AD A8 49 F1 0B CC DE F6 A2 8E 78 5E E5 2A 2E 78 05 AA 9C 85 33 7B 18 6B 34 58 9A B2 C5 E6 05 35 7A 97 63 6B B6 4A 6A 16 2C B0 B6 2D 14 30 84 21

File ME_O.JPG 32 + 353 = 385 *Byte* kompresi huffman

FF E1 00 12 48 55 46 46 54 10 02 03 02 01 02 03 04 05 06 07 08 09 0f 0A 0D 0C 0E 0B FF E2 01 63 21 3C E1 27 C8 C9 44 7C 21 0C F9 FC 42 78 DB 40 64 A2 71 39 04 94 69 08 8C 94 43 24 E1 0F B6 33 19 B8 D6 63 AD 0F 9A C6 F1 84 C6 A3 49 84 D6 D1 74 52 3E DD 8B 0F 86 0A EE DA AD CA 30 37 31 CC 90 25 24 3F FF FF FF FF FF FF 11 AC 7C 3F D6 73 56 31 9B 8D 66 3B FF FF FF FF FF FF FF FF FF FF FF FB E6 46 8C 71 BB F1 0F F4 78 6E 65 8D 61 7C 10 7F DE B1 8C 26 58 5F 03 5F F0 23 69 BD 65 85 F0 60 FF 02 63 51 74 52 3F F0 57 F0 23 62 2E 8A 47 FE 0F 97 E0 46 D4 5D 14 8F FC 1E AF F8 13 1A 84 8D 27 7C 06 3F CA 6C 42 46 93 BE 03 1F E5 36 A1 23 49 DF 01 8F F2 99 1A 31 C6 EF FF EE 37 39 BF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FE C2 34 62 EC 2F F2 32 6F A2 E8 A4 7D BF FF 11 AC 7C 3F D6 73 56 45 D1 48 FB 7F FF 70 1F DC E7 7D AF 92 2E 8A 47 DB FF F3 13 ED F7 2E 24 FB 95 45 D1 48 FB 7F FE 6D 52 BD EB 17 4F 05 D6 7D 17 45 23 ED FF FB 49 FF C4 B9 7B D3 18 8D CC 46 35 87 FF F8 3F F8 DE C0 6A B8 DD 45 36 B9 BD BD 11 85 C2 04 40 07 AE 26 80 81 AA 55 CD 05 DC F5 4D 23 A2 03 18 8B 3C 58 DE B2 C7 E8 BA 9E 82 9F 60 DF 19 A9 E8 98 B2 CE 1B 9E 29 D1 08 42

4. KESIMPULAN

Berdasarkan hasil kompresi yang diperoleh, metode Golomb-Rice menunjukkan rasio kompresi (CR) yang lebih tinggi dibandingkan metode Huffman pada sebagian besar file, terutama untuk file dengan ukuran metadata yang besar. Namun, untuk file dengan metadata kecil, kedua metode tidak memberikan kompresi yang signifikan. Hal ini menunjukkan bahwa efektivitas Golomb-Rice lebih unggul dalam kondisi tertentu, sedangkan Huffman masih memberikan hasil kompresi yang lebih stabil di berbagai skenario.

Metadata EXIF memiliki kontribusi yang berbeda terhadap ukuran file tergantung pada ukuran gambar. Pada gambar dengan ukuran besar, metadata hanya mengambil porsi kecil dari keseluruhan file, sehingga kompresinya tidak memberikan penghematan yang signifikan. Sebaliknya, pada gambar kecil, metadata dapat mencapai lebih dari 15% dari total ukuran file, sehingga kompresi metadata menjadi lebih relevan untuk mengurangi overhead penyimpanan. Selain itu pemanfaatan metadata EXIF sangat tepat untuk digunakan sebagai bagian kompresi yang mana dalam file JPEG memilki metadata yang tidak bisa diubah.

Dengan kemajuan teknologi penyimpanan dan kompresi gambar saat ini, metode Golomb-Rice dan Huffman masih memiliki manfaat dalam skenario tertentu. Namun, dengan meningkatnya efisiensi algoritma kompresi berbasis JPEG 2000, WebP, dan AVIF, kompresi metadata EXIF

saja kurang efektif untuk mengoptimalkan ruang penyimpanan secara keseluruhan. Oleh karena itu, dalam aplikasi modern, kombinasi metode kompresi metadata dengan algoritma kompresi gambar yang lebih canggih lebih disarankan untuk efisiensi penyimpanan yang optimal. Penelitian selanjutnya berdasarkan temuan ini dapat menjadi dasar untuk pengembangan metode hybrid yang menggabungkan kedua metode dan lebih adaptif dalam implementasi teknologi kompresi metadata JPEG dan file lainnya di masa depan.

DAFTAR PUSTAKA

Acharya R, S., J, S., S, S., S Aithal, V., & G S, H. (2023). Geo-Locating an Image Using EXIF Data. *International Journal of Engineering Applied Sciences and Technology*, *8*(1), 43–47. https://doi.org/10.33564/IJEAST.2023.v08i01.007

Agnihotri, S., Rameshan, R. M., Ghosal, R., & Gupta, P. (2024). Lossless Binary Image Compression Using Learned Multi-Level Dictionaries. *2024 60th Annual Allerton Conference on Communication, Control, and Computing*, 1–8. https://doi.org/10.1109/Allerton63246.2024.10735272

Ardiansyah, A., Hardi, N., & Gata, W. (2020). Identifikasi dan Recovery File JPEG dengan Metode Signature-Based Carving dalam Model Automata. *Komputika: Jurnal Sistem Komputer*, *9*(1), 75–83. https://doi.org/10.34010/komputika.v9i1.2733

Aria, M., & Sanjaya, A. (2018). Teknik Kompresi pada Transmisi Data Citra Payload KOMURINDO. *Komputika: Jurnal Sistem Komputer*, *7*(2), 103–111. https://doi.org/10.34010/komputika.v7i2.1512

Azeem, E. A., & Fatima. (2022). The Data Carving - The Art of Retrieving Deleted Data as Evidence. *International Journal for Electronic Crime Investigation*, *6*(2), 8. https://doi.org/10.54692/ijeci.2022.0602101

Boddu, M., & Mandal, S. K. (2022). Quantum-Dot Cellular Automata Based Lossless CFA Image Compression Using Improved and Extended Golomb-Rice Entropy Coder. *International Journal of Intelligent Engineering and Systems*, *15*(2), 12–25. https://doi.org/10.22266/ijies2022.0430.02

Cai, S., Liang, X., Cao, S., Yan, L., Zhong, S., Chen, L., & Zou, X. (2024). Powerful Lossy Compression for Noisy Images. *2024 IEEE International Conference on Multimedia and Expo (ICME)*, 1–6. https://doi.org/10.1109/ICME57554.2024.10687468

Cappello, F., Acosta, M., Agullo, E., Anzt, H., Calhoun, J., Di, S., Giraud, L., Grützmacher, T., Jin, S., Sano, K., Sato, K., Singh, A., Tao, D., Tian, J., Ueno, T., Underwood, R., Vivien, F., Yepes, X., Kazutomo, Y., & Zhang, B. (2025). Multifacets of Lossy Compression for Scientific Data in the Joint-Laboratory of Extreme Scale Computing. *Future Generation Computer Systems*, *163*, 107323. https://doi.org/10.1016/j.future.2024.05.022

Dhawan, S. (2011). A Review of Image Compression and Comparison of Its Algorithms. *IJECT*, *2*(1). https://www.iject.org/vol2issue1/sachindhawan.pdf

Doménech Fons, J., & Pegueroles Vallés, J. R. (2021). *Study and Development of an Autopsy Module for Automated Analysis of Image Metadata* [Universitat Politècnica de Catalunya]. https://hdl.handle.net/2117/356815

Hamilton, A., El-Hajjar, M., & Maunder, R. G. (2023). Reordered Exponential Golomb Error Correction Code for Universal Near-Capacity Joint Source-Channel Coding. *IEEE Access*, *11*, 93619–93634. https://doi.org/10.1109/ACCESS.2023.3310824

Harvey, P. (2025). JPEG Tags. In *EXIFTool*. https://EXIFtool.org/TagNames/JPEG.html

Hazel, T. (2008, September). *JPEG Marker Codes*. TechStumbler. https://techstumbler.blogspot.com/2008/09/jpeg-marker-codes.html

Himalyan, S., & Gupta, V. (2023). Golomb–Rice Coder-Based Hybrid Electrocardiogram Compression System. *ECSA 2023*, *2023*, Article ID: 10. https://doi.org/10.3390/ecsa-10-16209

Hwang, I., Yun, J., Chung, W., Lee, J., Kim, C.-G., Kim, Y., & Park, W.-C. (2021). Lossless Compression Algorithm and Architecture for Reduced Memory Bandwidth Requirement with Improved Prediction Based on the Multiple DPCM Golomb-Rice Algorithm. *Journal of Web Engineering*, *20*(6), 1813–1828. https://doi.org/10.13052/jwe1540-9589.2065

Itier, V., Puteaux, P., & Puech, W. (2022). Image Crypto-Compression. In M. Security (Ed.), *Multimedia Security 2* (pp. 91–128). Wiley. https://doi.org/10.1002/9781119987390.ch4

Jamil, S. (2024). Review of Image Quality Assessment Methods for Compressed Images. *Journal of Imaging*, *10*(5), Article ID: 113. https://doi.org/10.3390/jimaging10050113

Kadhim, D. J., Mosleh, M. F., & Abed, F. A. (2024). Exploring Text Data Compression: A Comparative Study of Adaptive Huffman and LZW Approaches. *BIO Web of Conferences*, *97*, Article ID: 00035. https://doi.org/10.1051/bioconf/20249700035

Kumar, G., & Kumar, R. (2021). Analysis of Arithmetic and Huffman Compression Techniques by Using DWT-DCT. *International Journal of Image, Graphics and Signal Processing*, *13*(4), 63–70. https://doi.org/10.5815/ijigsp.2021.04.05

Liu, X., An, P., Chen, Y., & Huang, X. (2022). An Improved Lossless Image Compression Algorithm Based on Huffman Coding. *Multimedia Tools and Applications*, *81*(4), 4781–4795. https://doi.org/10.1007/s11042-021-11017-5

Mahmood, A., & Wagner, A. B. (2023). Lossy Compression with Universal Distortion. *IEEE Transactions on Information Theory*, *69*(6), 3552–3573. https://doi.org/10.1109/TIT.2023.3247601

Mani, R. G., Parthasarathy, R., Eswaran, S., & Honnavalli, P. (2022). A Survey on Digital Image Forensics: Metadata and Image Forgeries. *WAC-2022: Workshop on Applied Computing*, 22. https://ceur-ws.org/Vol-3142/PAPER_03.pdf

Martini, M. G. (2025). Measuring Objective Image and Video Quality: On the Relationship Between SSIM and PSNR for DCT-Based Compressed Images. *IEEE Transactions on Instrumentation and Measurement*, *74*, 1–13. https://doi.org/10.1109/TIM.2025.3529045

Mills, R. (2018). *The Metadata in JPEG Files*. https://dev.exiv2.org/projects/exiv2/wiki/The_Metadata_in_JPEG_files

Nousheen, M. S., & Kumar, T. V. (2023). Implementation of Lossless Image Compression Using Fuzzy Based Modified Golomb Rice Encoding. *Journal of Engineering Sciences*, *14*(02), 242–253. https://doi.org/10.15433.JES.2023.V14I2.43P.29

Stoilov, E. P. (2022). Discovery and Analysis of EXIF Data in Images. *61 St Annual Scientific Conference - University of Ruse and Union of Scientists*, 52–58. https://conf.uni-ruse.bg/bg/docs/cp22/bp/bp-6.pdf

Sunardi, S., Riadi, I., & Akbar, M. H. (2020). Steganalisis Bukti Digital pada Media Penyimpanan Menggunakan Metode Static Forensics. *Jurnal Nasional Teknologi dan Sistem Informasi*, *6*(1), 1–8. https://doi.org/10.25077/TEKNOSI.v6i1.2020.1-8

Taha, H., Luisa, S., Nasir, V., & Editors, M. (2022). *Multimedia Forensics* (H. T. Sencar, L. Verdoliva, & N. Memon, Eds.). Springer Singapore. https://doi.org/10.1007/978-981-16-7621-5

Wijayanto, H., Riadi, I., & Prayudi, Y. (2016). Encryption EXIF Metadata for Protection Photographic Image of Copyright Piracy. *IJRCCT*, *5*(5), 237–243.

Žalik, B., Mongus, D., Žalik, K. R., Podgorelec, D., & Lukač, N. (2021). Lossless Chain Code Compression with an Improved Binary Adaptive Sequential Coding of Zero-Runs. *Journal of Visual Communication and Image Representation*, *75*, Article ID: 103050. https://doi.org/10.1016/j.jvcir.2021.103050

Zheng, C., Shrivastava, A., & Owens, A. (2023). EXIF as Language: Learning Cross-Modal Associations Between Images and Camera Metadata. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 6945–6956. https://doi.org/10.1109/CVPR52729.2023.00671

JISKA (Jurnal Informatika Sunan Kalijaga)
Vol. 11, No. 1, JANUARI, 2026, Pp. 70 – 82
ISSN: 2527 – 5836 (print) | 2528 – 0074 (online)

Perbandingan Kinerja MobileNetV2 dan VGG16 dalam Klasifikasi Penyakit pada Citra Daun Tanaman Cabai

Itsnaini Irvina Khoirunnisa [(1)*], Abdul Fadlil [(2)], Herman Yuliansyah [(3)]
[1] Departemen Magister Informatika, Universitas Ahmad Dahlan, Yogyakarta, Indonesia
[2] Departemen Teknik Elektro, Universitas Ahmad Dahlan, Yogyakarta, Indonesia
[3] Departemen Informatika, Universitas Ahmad Dahlan, Yogyakarta, Indonesia
e-mail : itsnainiirvina@gmail.com, fadlil@mti.uad.ac.id, herman.yuliansyah@tif.uad.ac.id.
* Penulis korespondensi.
Artikel ini diajukan 13 Maret 2025, direvisi 24 Juni 2025, diterima 7 Juli 2025, dan dipublikasikan 25 Januari 2026.

Abstract

Chili peppers play a crucial role in the Indonesian economy, serving as a significant source of income for many farmers. Price fluctuations influenced by weather conditions make this crop vulnerable to diseases that can impact productivity. However, leaves are key indicators of plant health, revealing early disease symptoms before they spread. This research focuses on detecting diseases in chili plants using neural network architectures via transfer learning, specifically MobileNetV2 and VGG16, to classify chili leaf images. The study aims to identify three disease classes: begomovirus, leaf spots, and healthy leaves. The dataset comprises 3,150 leaf images, split into 70% for training and 30% for testing. Results show that MobileNetV2 achieved an accuracy of 99.47% and VGG16 98.62%, with evaluation using a confusion matrix indicating good performance in disease identification, where MobileNetV2 offers better computational efficiency. Thus, transfer learning can effectively identify leaf diseases in chili plants.

Keywords: Chili Diseases, Deep Learning, Image Classification, Neural Network, Transfer Learning

Abstrak

Cabai memilki peranan penting dalam perekonomian Indonesia dan menjadi sumber pendapatan bagi banyak petani. Fluktuasi harga cabai yang sering dipengaruhi oleh faktor cuaca membuat tanaman ini rentan terhadap penyakit, yang dapat berakibat signifikan terhadap penurunan produktivitas. Namun, daun menjadi indikator utama kesehatan tanaman karena menunjukkan gejala awal penyakit sebelum menyebar ke bagian lain. Penelitian ini bertujuan untuk mengusulkan model deteksi penyakit pada tanaman cabai berdasarkan citra daun menggunakan arsitektur *neural network* melalui pendekatan *transfer learning,* khususnya MobileNetV2 dan VGG16. Dalam penelitian ini, terdapat tiga kelas penyakit yang akan dideteksi, yaitu begomovirus, bercak daun, dan daun sehat. Dataset yang digunakan terdiri dari 3.150 citra daun, yang dibagi menjadi 70% data *training* dan 30% data *testing*. Hasil menunjukkan MobileNetV2 mencapai akurasi 99.47% dan VGG16 98.62% dengan evaluasi menggunakan *confusion matrix* yang mengindikasikan kinerja baik dalam identifikasi penyakit, di mana MobileNetV2 menawarkan efisiensi waktu komputasi yang lebih baik. Dengan demikian, *transfer learning* dapat berperan penting dalam mencapai hasil optimal dalam mengidentifikasi penyakit daun tanaman cabai.

Kata Kunci: Penyakit Cabai, *Deep Learning*, Klasifikasi Gambar, *Neural Network*, *Transfer Learning*

1. PENDAHULUAN

Indonesia dikenal sebagai negara agraris, sehingga masyarakatnya masih bergantung pada sektor pertanian untuk memenuhi kebutuhan hidup. Salah satu komoditas pertanian yang signifikan di Indonesia adalah cabai (Ramadhani et al., 2023). Cabai adalah tanaman yang memiliki peran penting dan ditanam di berbagai belahan dunia karena manfaatnya dalam kuliner dan sebagai obat (Siddiqui, 2023). Terdapat berbagai varietas tanaman cabai, salah satunya adalah cabai rawit (*Capsicum frutescens L.*). Berdasarkan Pusat Data dan Sistem Informasi

Pertanian, rata-rata konsumsi cabai rawit di Indonesia pada tahun 2023 mencapai 2,192 kg per kapita per tahun, menunjukkan peningkatan sebesar 5,76% dibandingkan dengan tahun sebelumnya (Sekretariat Jenderal Kementerian Pertanian, 2023).

Cabai rawit termasuk sayuran yang mengalami fluktuasi harga yang signifikan, sehingga berpengaruh terhadap tingkat inflasi (K. Prasetyo et al., 2023). Fluktuasi harga tersebut dipicu oleh berbagai faktor, salah satunya adalah kondisi cuaca (Andini et al., 2024). Menanam cabai pada musim hujan dapat menghadapi tantangan cuaca yang berisiko bagi pertumbuhan tanaman. Curah hujan yang tinggi dan kelembaban udara yang meningkat dapat menyebabkan munculnya penyakit pada tanaman cabai rawit, seperti busuk buah antraknosa (*Colletotrichum sp.*), virus kuning (*Begomovirus*), virus *mosaic*, dan bercak daun (Syukur, 2018). Upaya untuk mengidentifikasi penyakit secara dini penting untuk meningkatkan ketahanan pangan nasional.

Penyakit cabai rawit yang tidak terdeteksi dan dibiarkan berkembang dapat menyebabkan kerusakan pada tanaman. Hal ini berpotensi menurunkan kualitas dan kuantitas panen cabai rawit, yang dapat berdampak negatif pada perekonomian negara (Ramadhani et al., 2023). Beberapa penelitian terdahulu telah mengevaluasi kualitas cabai berdasarkan daun, tanah, kandungan air, dan suhu (Pramudhita et al., 2023). Namun, kerugian yang dialami petani akibat kurangnya kemampuan dalam mengidentifikasi dan menangani penyakit tanaman cabai rawit secara dini tidak hanya berdampak pada pendapatan, tetapi juga dapat memengaruhi ketahanan pangan nasional. Adanya fluktuasi harga dan tantangan cuaca, penting untuk menemukan solusi yang efektif untuk mendeteksi penyakit ini secara otomatis.

Berdasarkan permasalahan yang ada, diperlukan solusi untuk memudahkan petani cabai dalam mengklasifikasikan penyakit tanaman cabai. Pemanfaatan teknik *deep learning* dan pemrosesan citra dapat memberikan jawaban untuk masalah ini. Beberapa tahun terakhir, telah muncul berbagai metode otomatis untuk membantu petani dalam mengidentifikasi jenis penyakit yang menyerang tanaman cabai. Salah satu proyek penelitian paling populer dalam klasifikasi penyakit tanaman adalah penggunaan pendekatan *deep learning*, khususnya *Convolutional Neural Network* (CNN), karena efisien dalam menghemat waktu dan hemat biaya (Pramudhita et al., 2023).

Terdapat beberapa penelitian sebelumnya yang membahas klasifikasi penyakit tanaman cabai. Ramadhani et al. (2023) membandingkan kemampuan CNN dan MobileNetV2 dalam mengklasifikasikan penyakit pada tanaman cabai. Hasil penelitian menunjukkan MobileNetV2 mencapai akurasi 90% yang lebih optimal dibandingkan dengan model CNN. Gulzar (2023) melakukan klasifikasi citra buah menggunakan TL-MobileNetV2 melalui teknik *transfer learning* untuk mengidentifikasi 40 jenis buah. Model TL-MobileNetV2 menunjukkan akurasi 99% lebih tinggi dibandingkan MobileNetV2. Maftukhah et al. (2024) melakukan klasifikasi citra kupu-kupu menggunakan CNN dengan arsitektur AlexNet. Model tersebut menunjukkan peningkatan akurasi seiring dengan ukuran citra yang lebih besar.

Alkanan & Gulzar (2024) melakukan klasifikasi penyakit biji jagung menggunakan *deep learning* arsitektur MobileNetV2. Model yang diusulkan mencapai akurasi sekitar 96% menunjukkan performa yang baik dibandingkan dengan model lainnya. Winiarti & Khoirunnisa (2024) melakukan deteksi penyakit cabai menggunakan aplikasi *mobile*, di mana model MobileNetV2 memperoleh akurasi 94%. Membuktikan CNN efektif dalam memproses gambar beresolusi kecil tanpa kehilangan informasi penting. Nguyen et al. (2022) mengklasifikasikan penyakit daun tomat menggunakan VGG-19 untuk mendeteksi penyakit secara akurat dan meningkatkan hasil panen. Hasil penelitian menunjukkan akurasi sebesar 99,72% dalam mengklasifikasikan citra daun tomat, serta mengurangi waktu pelatihan secara signifikan. Muis et al. (2024) melakukan klasifikasi citra tumor otak menggunakan pendekatan CNN menggunakan arsitektur AlexNet dan GoogLeNet. Temuan ini membantu mengurangi beban komputasi saat pelatihan model.

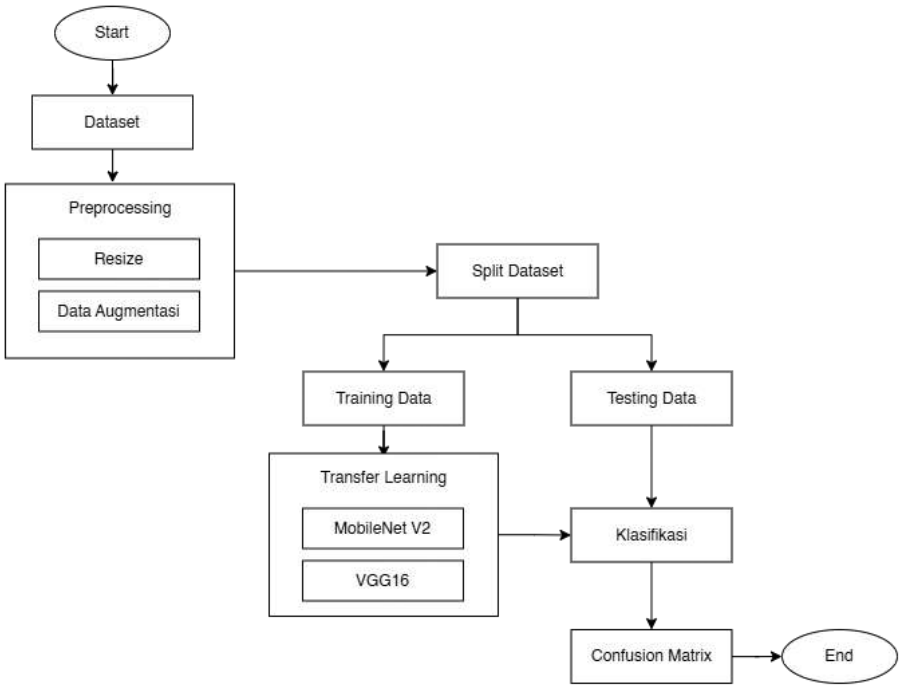
Tujuan penelitian ini adalah mengoptimalkan arsitektur model dalam mengidentifikasi penyakit pada citra daun tanaman cabai. Daun merupakan bagian tanaman yang responsif terhadap

perubahan iklim (Y. Li et al., 2020) dan serangan pathogen (Mostafa et al., 2022). Gejala penyakit cabai umumnya muncul pertama kali pada daun sebelum menyebar ke bagian lain seperti batang atau buah (Rustanto et al., 2024). Penelitian ini menerapkan metode CNN melalui pendekatan *transfer learning* dengan pemanfaatan arsitektur yang telah dilatih sebelumnya pada dataset besar. Penelitian ini tidak hanya meningkatkan akurasi dan efisiensi dalam klasifikasi citra, tetapi juga menghemat waktu dan sumber daya dalam pengembangan model.

Berdasarkan informasi yang diberikan, sangat penting untuk mengidentifikasi secara akurat jenis penyakit yang menyerang tanaman cabai sedini mungkin. Hal ini penting untuk memastikan bahwa penyakit ditangani dengan cepat dan akurat, mencegahnya menyebar ke tanaman lain di sekitarnya dan mengurangi risiko gagal panen. Oleh karena itu, penelitian ini bertujuan untuk meningkatkan konsep pemrosesan gambar digital dengan memanfaatkan *transfer learning*.

2. METODE PENELITIAN

Penelitian ini dimulai dari tahapan yang ditunjukkan pada Gambar 1 yang menjelaskan pengumpulan data. Kemudian *pre-processing* dilakukan dengan mengubah ukuran citra dan melakukan penambahan data. Dataset tersebut dibagi menjadi dua bagian yaitu data *training* dan data *testing*. Selanjutnya, memilih model berdasarkan arsitektur MobileNetV2 dan VGG16. Proses berikutnya adalah *training* model, kemudian model tersebut disimpan dan dilakukan *testing* model. Model dari masing-masing arsitektur kemudian dibandingkan berdasarkan hasil evaluasi untuk menentukan model yang terbaik.



Gambar 1 Metode Klasifikasi Penyakit Daun Tanaman Cabai

2.1 Dataset

Citra yang digunakan dalam penelitian ini berasal dari dataset citra daun penyakit tanaman cabai. Citra yang digunakan adalah citra berwarna RGB dalam format JPG. Penelitian ini difokuskan pada daun yang terinfeksi oleh berbagai jenis penyakit, antara lain virus kuning *begomovirus*, penyakit ini disebabkan oleh virus yang menyebar dengan cepat melalui serangga vektor seperti kutu kebul (*Bemisia tabaci*). Hal ini mengakibatkan tanaman cabai menjadi tidak produktif dan tidak menghasilkan buah (Syukur, 2018). Bercak daun antraknosa muncul sebagai bercak-bercak berwarna pucat yang awalnya kecil dan perlahan membesar (A. D. Prasetyo & Agustinur, 2022).

Bagian tengah bercak berwarna putih muda, sedangkan tepinya lebih gelap. Bercak yang sudah tua dapat menyebabkan terbentuknya lubang. Setelah terinfeksi, daun akan layu dan rontok. Virus *mosaic* ditandai dengan pola belang-belang hijau muda dan tua pada daun, yang mengakibatkan ukuran daun menjadi lebih kecil. Akibatnya, tanaman tidak dapat tumbuh dengan baik bahkan tidak mampu berproduksi sama sekali (Syukur, 2018). Kumpulan data yang digunakan dalam penelitian ini sebanyak 3.150 citra yang dibagi menjadi tiga kelas, di mana masing-masing kelas terdiri dari 1.050 citra. Kumpulan data ditunjukkan pada Gambar 2.

(a) *Begomovirus* (b) Bercak daun (c) Daun sehat

Gambar 2 Contoh Dataset Tanaman Cabai

2.2 Preprocessing Data

Preprocessing adalah proses yang dilakukan sebelum memulai *training* model. Pada tahap ini, citra digital diproses untuk meningkatkan kualitasnya, sehingga menghasilkan hasil yang lebih optimal saat data digunakan dalam pelatihan model (Singh et al., 2021). Preprocessing terdiri dari dua tahap yaitu rezise dan augmentasi data. Pada tahap resize, ukuran citra diubah menjadi resolusi yang lebih kecil supaya prosesnya lebih efisien (Pramudhita et al., 2023). Resize dilakukan dengan menyamakan ukuran piksel dalam citra dan mengubah dimensi citra asli menjadi 300×300 piksel. Setelah proses resize, dilakukan augmentasi data untuk kemudian diperbanyak menggunakan ImageDataGenerator dengan menerapkan *flipping* vertikal dan horizontal, *cropping* citra, dan *zoom* citra untuk meningkatkan jumlah data.

2.3 Pembagian Data

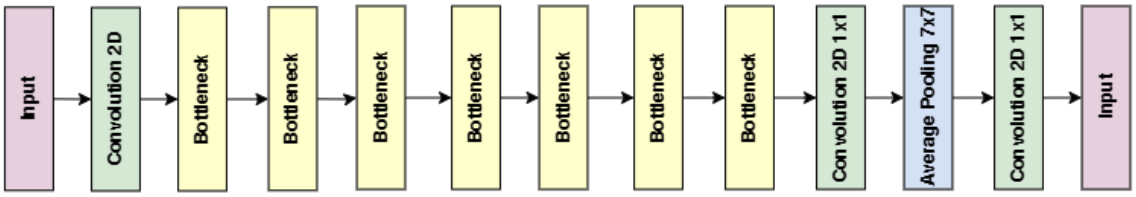
Pembagian dataset dilakukan dengan 70% untuk data *training* dan 30% untuk data *testing*, sehingga model dapat dilatih dengan cukup data dan juga dievaluasi secara efektif menggunakan data yang terpisah untuk mengukur kinerjanya. Pada tahap *training* model, dilakukan inisialisasi untuk mengatur parameter seperti *learning rate* dan *optimizer* Adam (*Adaptive Moment Estimation*). Model dilatih menggunakan *training* data untuk menghitung prediksi dan loss, kemudian dihitung dengan membandingkan prediksi dengan label sebenarnya. Proses ini diulang selama n *epoch* hingga model mencapai performa yang memuaskan. Setelah *training* selesai, model disimpan untuk digunakan *testing* model, sehingga menghindari kebutuhan untuk melatih ulang model setiap kali ingin melakukan prediksi. Tahap selanjutnya *testing* model, di mana model yang telah dilatih sebelumnya digunakan untuk melakukan prediksi pada data *testing*. Proses ini mirip dengan *training* tetapi tidak ada pembaruan bobot. Data *testing* tidak digunakan selama *training*, sehingga tahap ini bertujuan untuk menguji kemampuan model dalam melakukan deteksi penyakit daun tanaman cabai.

2.4 Pemilihan Model

Pemilihan model dilakukan untuk menentukan arsitektur yang paling efektif dalam mengidentifikasi penyakit pada daun tanaman cabai. Proses ini melibatkan evaluasi berbagai arsitektur model yang ada, seperti MobileNetV2 dan VGG16. Tujuan utama dari pemilihan model ini adalah untuk memaksimalkan hasil akurasi klasifikasi, sehingga model dapat secara akurat mendeteksi dan mengklasifikasikan berbagai jenis penyakit yang mungkin menyerang tanaman cabai. Selain itu, waktu komputasi pelatihan juga menjadi pertimbangan dalam tahap ini.

2.4.1 MobileNetV2

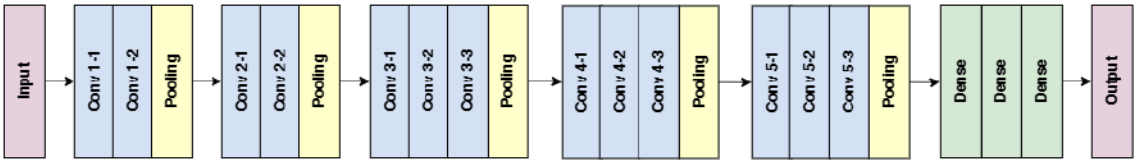
MobileNetV2 adalah jenis *neural network* berbasis *mobile* yang digunakan untuk menangani kebutuhan sumber daya komputer yang berlebihan. Ketebalan filter di MobileNetV2 disesuaikan dengan ketebalan gambar *input* (Ramadhani et al., 2023). Arsitektur ini merupakan pengembangan dari konsep MobileNetV1. Meskipun MobileNetV2 membutuhkan waktu komputasi lebih lama dibandingkan MobileNetV1, arsitektur ini tetap mampu mencapai akurasi yang lebih tinggi. Gambar 3 menunjukkan arsitektur MobileNetV2.



Gambar 3 Arsitektur MobileNetV2

2.4.2 VGG16

VGG16 adalah jenis *neural network* yang terdiri dari 16 lapisan konvolusional, terbagi menjadi 11 lapisan konvolusional dan lima lainnya. Ukuran gambar *input* ditetapkan pada 224×224 piksel (Paymode & Malode, 2022). Menggunakan filter kecil (3×3) untuk menangkap fitur detail secara efektif, dan memiliki struktur yang dalam untuk meningkatkan kemampuan klasifikasi citra. Gambar 4 menunjukkan arsitektur VGG16.



Gambar 4 Arsitektur VGG16

2.5 Klasifikasi

Klasifikasi dilakukan untuk memproses model yang telah dilatih supaya dapat memprediksi kelas dari data uji dengan akurasi tinggi. Proses klasifikasi memungkinkan model untuk mengidentifikasi dan mengkategorikan data citra berdasarkan fitur-fitur tertentu yang telah dipelajari selama pelatihan. Sehingga model dapat mengenali berbagai jenis penyakit yang menyerang tanaman.

2.6 Evaluasi Kinerja Model

Hasil evaluasi model akan ditampilkan dalam bentuk *confusion matrix* yang digunakan untuk menghitung nilai akurasi, presisi, *recall,* dan F1-*score* (Hicks et al., 2022). Hal ini bertujuan untuk menilai seberapa baik model melakukan klasifikasi pada dataset yang digunakan. Pers. (1) digunakan untuk menghitung nilai akurasi, yang memberikan gambaran jelas tentang seberapa baik model dalam melakukan klasifikasi dengan benar. Pers. (2) digunakan untuk menghitung presisi, yang mengukur tingkat ketepatan sistem dalam mengklasifikasikan data sesuai dengan label yang seharusnya. Pers. (3) digunakan untuk menghitung *recall*, yang mengukur seberapa efektif model dalam mengurangi jumlah FN. Pers. (4) digunakan untuk menghitung F1-*score*, yang digunakan untuk melihat perbandingan rata-rata antara presisi dan *recall* sehingga memberikan gambaran menyeluruh tentang kinerja model.

$$akurasi\ (\%) = \frac{TP + TN}{TP + TN + FP + FN} \times 100\% \quad (1)$$

$$presisi = \frac{TP}{TP + FP} \quad (2)$$

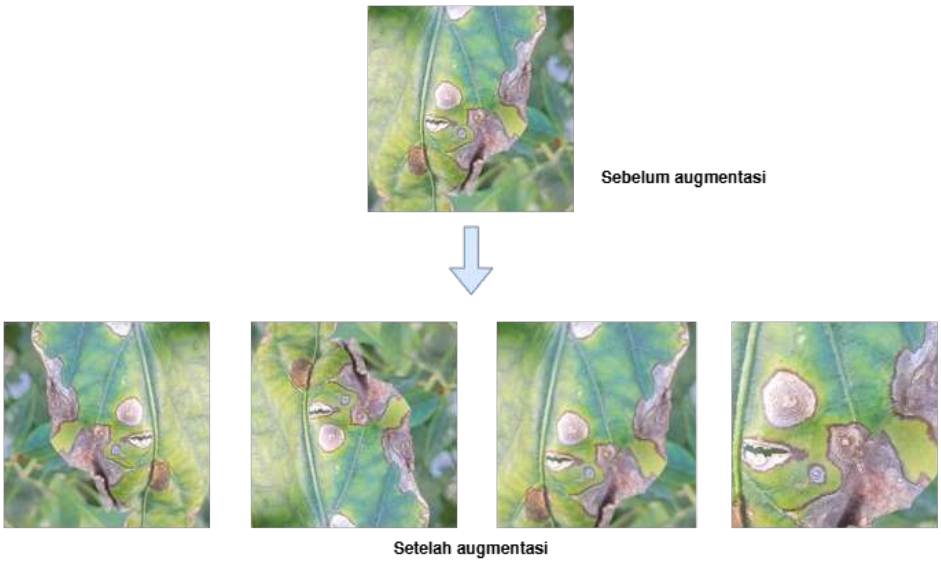
$$recall = \frac{TP}{TP + FN} \quad (3)$$

$$F1 - Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (4)$$

3. HASIL DAN PEMBAHASAN

3.1 Preprocessing Data

Setelah pemodelan dilakukan, data melalui serangkaian proses augmentasi meliputi *resize*, *zoom*, *cropping*, dan *flipping*. Proses ini bertujuan untuk meningkatkan variasi dalam dataset dan membantu model belajar dari berbagai kondisi. Tabel 1 menyajikan perbandingan jumlah citra sebelum dan sesudah proses augmentasi. *Resize* digunakan untuk mengubah ukuran gambar agar sesuai dengan input yang diperlukan oleh model, sementara *flipping* memungkinkan model untuk mengenali objek dari sudut pandang yang berbeda. *Cropping* dan *zoom* membantu dalam menekankan fitur-fitur penting dengan memperbesar bagian tertentu dari gambar. Hasil dari proses augmentasi dapat dilihat pada Gambar 5, yang menunjukkan berbagai transformasi yang diterapkan pada data untuk meningkatkan kinerja model. Pada tahap ini, terdapat kendala yang sering muncul saat melakukan *cropping*, di mana tidak akurat dalam penentuan parameter area *cropping* dapat menyebabkan hasil tidak sesuai dengan area yang diharapkan. Untuk mengatasi hal ini ini, dilakukan uji coba berulang dengan berbagai pengaturan untuk menemukan parameter *cropping* yang efektif.



Gambar 5 Hasil Preprocessing Data

Tabel 1 Variasi Dataset

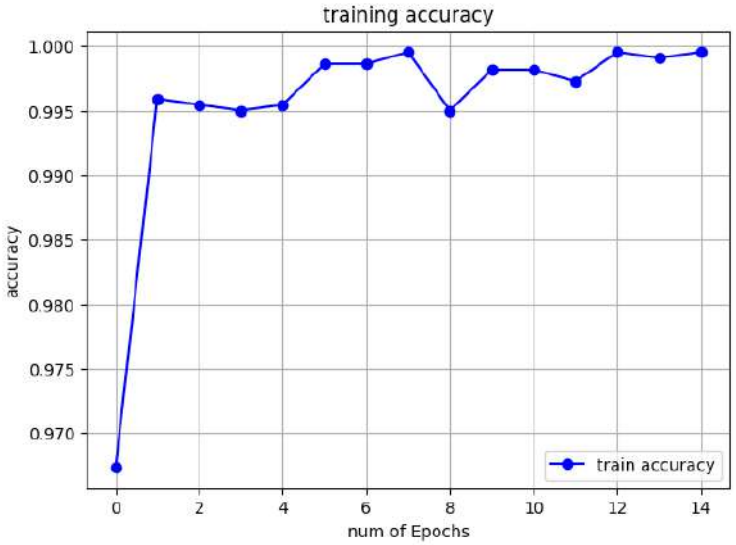
No.	Class	Sample	
		Sebelum Augmentasi	Setelah Augmentasi
1	Begomovirus	210	1.050
2	Bercak daun	210	1.050
3	Daun sehat	210	1.050
Total		**630**	**3.150**

3.2 Pelatihan Model

Bagian ini menyajikan hasil dan pembahasan penelitian ini, termasuk hasil pelatihan model, evaluasi model, perbandingan model arsitektur MobileNetV2 dan VGG16 dengan menggunakan presisi, recall, dan F1-Score. Sehingga dapat mengevaluasi kelebihan dan kekurangan masing-masing arsitektur dalam konteks pengenalan penyakit pada tanaman. Selain itu, juga dilakukan perbandingan kinerja penelitian ini dengan penelitian lainnya untuk menyoroti keunggulan atau kekurangan model yang digunakan.

3.2.1 MobileNetV2

Akurasi klasifikasi model MobileNetV2 dapat dilihat pada Gambar 6. Dalam penelitian ini, model dilatih selama 15 *epoch*, yang merupakan jumlah iterasi di mana model diperbarui menggunakan data *training*. *Learning rate* yang digunakan adalah 0.0001, yang merupakan parameter penting yang menentukan seberapa besar perubahan bobot model dilakukan di setiap iterasi. *Learning rate* yang kecil ini dirancang untuk membantu menghindari *overfitting* pada minimum fungsi *loss*, memungkinkan model untuk belajar dengan lebih stabil dan mengurangi risiko terjebak pada solusi lokal yang tidak optimal. Optimasi *hyperparameter* dilakukan menggunakan algoritma Adam, yang merupakan salah satu algoritma optimasi paling banyak digunakan dalam *deep learning*.



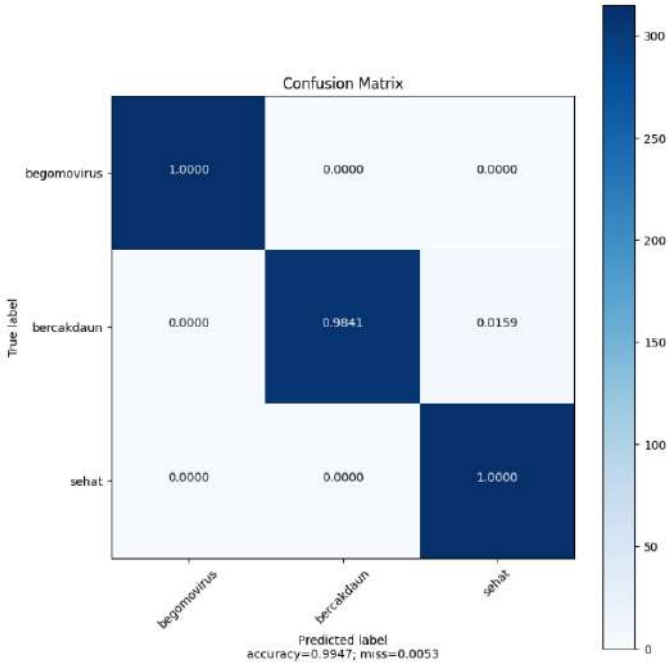
Gambar 6 Grafik Training MobileNetV2

Grafik yang ditampilkan menunjukkan akurasi *training* sepanjang 15 *epoch*. Pada *epoch* pertama akurasi awal mungkin relatif rendah, namun seiring berjalannya *epoch* akurasi meningkat secara signifikan. Model mencapai akurasi 99,96% pada akhir pelatihan dengan rata-rata waktu komputasi sekitar 222 detik per iterasi. Peningkatan akurasi yang stabil ini menunjukkan bahwa model berhasil belajar dari data *training* dan mampu mengklasifikasikan data dengan baik.

Setelah proses *training* selesai, model diuji menggunakan data *testing* untuk mengevaluasi kinerjanya. Hasil *testing* menunjukkan bahwa model mencapai akurasi sebesar 99,47%, menandakan tingkat keakuratan yang sangat tinggi dalam mendeteksi penyakit pada daun tanaman cabai. Untuk memberikan analisis yang lebih mendalam tentang performa model, evaluasi dilakukan dengan menggunakan *confusion matrix*, seperti yang ditunjukkan pada Gambar 7.

Gambar 7 menunjukkan kinerja model MobileNetV2 dalam mengklasifikasikan penyakit daun tanaman cabai. Secara keseluruhan, model berhasil mengklasifikasikan dengan baik, terutama

kelas begomovirus dan daun sehat. Namun, terdapat kesalahan klasifikasi pada kelas bercak daun yang salah diidentifikasi sebagai daun sehat. *Confusion matrix* yang ditampilkan memberikan rincian lebih lanjut mengenai prediksi model. Tabel 2 menyajikan *classification report* yang memberikan rincian kinerja model dalam mendeteksi penyakit pada daun tanaman cabai, termasuk presisi, recall, F1-score, dan support untuk masing-masing kelas. Secara keseluruhan, tabel ini menunjukkan bahwa model memiliki kinerja yang sangat baik dalam mendeteksi berbagai jenis penyakit pada daun tanaman cabai.



Gambar 7 Confusion Matrix MobileNetV2

Tabel 2 Hasil Testing MobileNetV2

Kelas	Presisi	Recall	F1-Score	Support
Begomovirus	1,00	1,00	1,00	315
Bercak daun	1,00	0,98	0,99	315
Daun sehat	0,98	1,00	0,99	315
Akurasi	-	-	0,99	945
Macro Avg	0,99	0,99	0,99	945
WeightedAvg	0,99	0,99	0,99	945

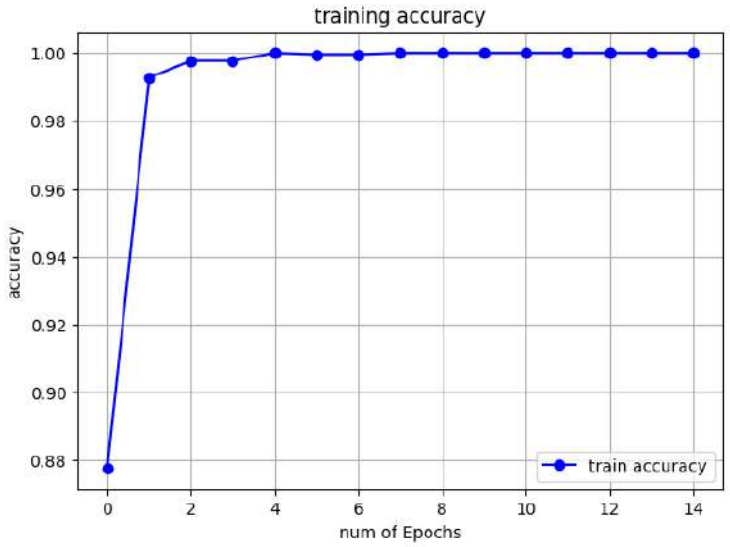
3.2.2 VGG16

Akurasi klasifikasi model VGG16 ditampilkan pada Gambar 8. Model ini dilatih selama 15 *epoch*, dengan menggunakan *learning rate* yang sama yaitu 0.0001. *Learning rate* ini berfungsi untuk mengatur besarnya perubahan bobot model pada setiap iterasi dan pengaturan ini mirip dengan yang diterapkan pada model MobileNetV2. Tujuan utama dari pengaturan ini adalah untuk mencegah *overfitting* serta memastikan proses pembelajaran yang stabil dan efektif.

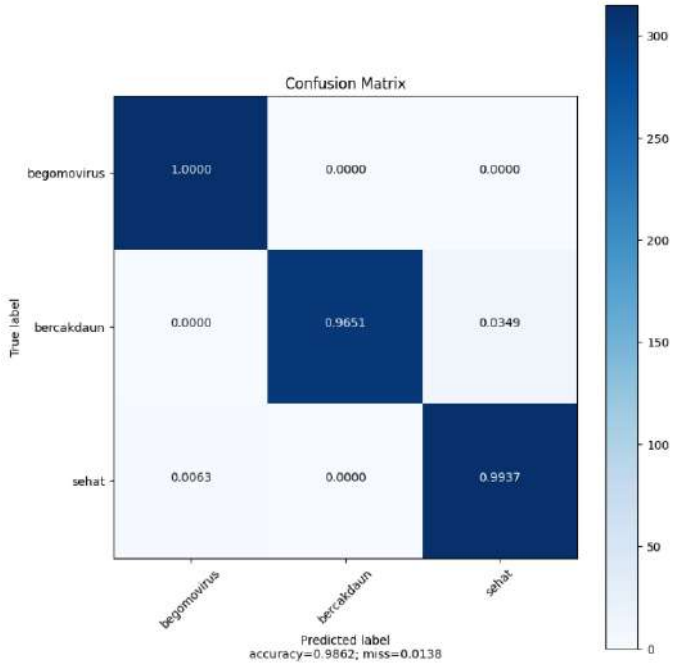
Selama *training*, akurasi model menunjukkan peningkatan yang signifikan dari *epoch* pertama hingga akhir *training*. Pada akhir proses *training*, model VGG16 berhasil mencapai akurasi sempurna sebesar 100%. Meskipun hasil ini menunjukkan kemampuan model yang sangat baik dalam mengenali dan mengklasifikasikan data, rata-rata waktu komputasi yang dibutuhkan sekitar 1.092 detik per iterasi sehingga tergolong cukup lama. Meskipun VGG16 menawarkan akurasi yang tinggi, efisiensi waktu yang kurang optimal menjadi salah satu kekurangan yang

perlu dipertimbangkan, terutama dalam skenario yang memerlukan respon cepat. Peningkatan akurasi yang konsisten selama pelatihan menunjukkan bahwa model mampu belajar dengan efektif dari data *training*, meskipun optimasi lebih lanjut masih diperlukan untuk meningkatkan efisiensi komputasi.

Setelah menyelesaikan proses *training*, model diuji menggunakan data *testing* untuk menilai kinerjanya. Hasil evaluasi menunjukkan bahwa model berhasil mencapai akurasi sebesar 98,62%, yang mencerminkan tingkat keakuratan yang sangat baik dalam mendeteksi penyakit pada daun tanaman cabai. Untuk mendapatkan analisis yang lebih mendalam mengenai performa model, evaluasi dilakukan dengan memanfaatkan *confusion matrix*, seperti yang ditunjukkan pada Gambar 9.



Gambar 8 Grafik Training VGG16



Gambar 9 Confusion Matrix VGG16

Gambar 9 menunjukkan kinerja model VGG16 dalam mengklasifikasikan penyakit daun tanaman cabai. Secara keseluruhan, model berhasil mengklasifikasikan dengan baik, terutama kelas begomovirus. Namun, terdapat kesalahan klasifikasi pada kelas bercak daun yang salah diidentifikasi sebagai daun sehat dan kelas daun sehat yang salah diidentifikasi sebagai begomovirus. Tabel 3 menyajikan *classification report* yang mencakup metrik penting seperti presisi, recall, F1-score, dan support untuk setiap kelas penyakit. Secara keseluruhan, hasil yang ditampilkan dalam tabel ini menunjukkan bahwa model memiliki performa yang sangat baik dalam mengidentifikasi berbagai jenis penyakit pada daun tanaman cabai.

Tabel 3 Hasil Testing VGG16

Kelas	Presisi	Recall	F1-Score	Support
Begomovirus	1,00	1,00	1,00	315
Bercak daun	1,00	0,95	0,98	315
Daun sehat	0,95	1,00	0,98	315
Akurasi	-	-	0,98	945
Macro Avg	0,98	0,98	0,98	945
WeightedAvg	0,98	0,98	0,98	945

3.3 Perbandingan Model

Tabel 4 menyajikan perbandingan kinerja antara model MobileNetV2 dan VGG16 dalam hal presisi, recall, dan F1-score. MobileNetV2 menunjukkan kinerja yang efisien dibandingkan dengan model VGG16, di mana menawarkan kecepatan pemrosesan yang jauh lebih baik dan menggunakan ruang yang lebih sedikit. Secara keseluruhan, akurasi tertinggi yang dicapai adalah 99,47% untuk MobileNetV2, sedangkan VGG16 mencapai akurasi sebesar 98,62%. Meskipun kedua model menunjukkan akurasi yang tinggi, waktu komputasi untuk MobileNetV2 jauh lebih efisien dibandingkan dengan VGG16 yang memakan waktu lebih lama dalam proses *training* dan *testing*. Dari hasil ini, dapat disimpulkan bahwa MobileNetV2 tidak hanya unggul dalam hal akurasi, tetapi juga dalam efisiensi, menjadikannya pilihan yang lebih baik untuk klasifikasi penyakit pada daun tanaman cabai. Kelebihan ini menjadikan MobileNetV2 sebagai model yang lebih praktis dan optimal.

Tabel 4 Perbandingan Model MobileNetV2 dan VGG16

Model	Presisi	Recall	F1-Score
MobileNetV2	**0,99**	**0,99**	**0,99**
VGG16	0,98	0,98	0,98

3.4 Perbandingan Kinerja

Tabel 5 menyajikan perbandingan kinerja berbagai model dalam deteksi penyakit tanaman berdasarkan beberapa penelitian terkini. Tabel ini mencakup berbagai dataset, jumlah dataset, jumlah kelas, metode yang digunakan, serta akurasi yang dicapai. Sehingga, memberikan wawasan yang lebih luas mengenai efektivitas model yang diterapkan dalam penelitian ini dibandingkan dengan penelitian lain yang telah dilakukan sebelumnya.

Dalam konteks deteksi penyakit tanaman, penting untuk mempertimbangkan tidak hanya akurasi, tetapi juga kompleksitas model dan waktu komputasi. Hasil yang diperoleh dalam penelitian ini menunjukkan bahwa MobileNetV2 maupun VGG16 dapat mencapai akurasi yang sangat tinggi dalam mendeteksi penyakit pada daun tanaman cabai, dengan MobileNetV2 mencapai 99,47% dan VGG16 mencapai 98,62%. Dibandingkan dengan penelitian (Ramadhani et al., 2023) yang mencapai akurasi 90% dengan dataset yang lebih besar, temuan ini menunjukkan kemajuan signifikan dalam efektivitas model. Penelitian ini menunjukkan bahwa baik MobileNetV2 maupun VGG16 tidak hanya memberikan akurasi yang tinggi, tetapi juga efisiensi yang lebih baik dalam penggunaan sumber daya, menjadikannya pilihan yang lebih praktis untuk aplikasi di lapangan.

Waktu komputasi yang digunakan juga menjadi faktor penting, di mana MobileNetV2 menunjukkan efisiensi yang lebih unggul, seperti pada penelitian (Hermanto et al., 2024) di mana MobileNetV2 lebih efisien dan mengoptimalkan proses *transfer learning* dengan baik. Dengan demikian, perbandingan ini tidak hanya menyoroti keberhasilan model yang diterapkan dalam penelitian ini, tetapi juga memberikan konteks yang lebih luas tentang kemajuan dalam deteksi penyakit tanaman, serta potensi untuk pengembangan lebih lanjut dalam teknologi pertanian berbasis kecerdasan buatan.

Tabel 5 Perbandingan Kinerja Deteksi

Referensi	Dataset	Jumlah Dataset	Kelas	Metode	Akurasi (%)
Ramadhani et al. (2023)	Penyakit daun cabai	5.000	5	MobileNetV2	90
Pramudhita et al. (2023)	Penyakit daun strawberry	1.336	4	MobileNetV3-Large	92,14
D. Li et al. (2024)	Penyakit daun cabai	2.500	5	MCCM CNN	93,5
Raghuram & Borah (2025)	Penyakit daun tomat	14.000	8	DRL-TL	99,23
Arnob et al. (2025)	Penyakit daun kembang kol	7.360	4	Custom CNN, InceptionV3, ResNet50, VGG16	83,94 76,53 90,85 65,58
Chaithanya & Rachana (2023)	Penyakit daun pepaya	2.159	5	CNN, VGG16, VGG19, Inception V3, DenseNet 121, MobileNet V2, EfficientNet B0, ResNet50	72,76 84,82 83,48 73,67 86,16 83,93 86,61 87,95
Raufer et al. (2025)	Penyakit daun timun	1.289	8	VGG19	93,87
Penelitian ini	**Penyakit daun cabai**	**3.150**	**3**	**MobileNetV2 dan VGG16**	**99,47 dan 98,62**

4. KESIMPULAN

Penelitian ini berhasil melatih dua model, yaitu MobileNetV2 dan VGG16, untuk mendeteksi penyakit pada daun tanaman cabai. MobileNetV2 mencapai akurasi 99,47%, sedangkan VGG16 mencapai akurasi 98,62%. Kedua model menunjukkan kemampuan yang sangat baik dalam klasifikasi, namun MobileNetV2 unggul dalam efisiensi waktu komputasi. Temuan ini memberikan kontribusi positif terhadap pengembangan teknologi pertanian berbasis kecerdasan buatan, menunjukkan potensi besar dalam penggunaan model *transfer learning* untuk deteksi penyakit tanaman secara efektif dan efisien. Disarankan supaya penelitian selanjutnya dapat mengeksplorasi penggunaan model lain yang lebih canggih dan melakukan optimasi lebih lanjut pada *hyperparameter* untuk meningkatkan efisiensi dan akurasi.

DAFTAR PUSTAKA

Alkanan, M., & Gulzar, Y. (2024). Enhanced Corn Seed Disease Classification: Leveraging MobileNetV2 with Feature Augmentation and Transfer Learning. *Frontiers in Applied Mathematics and Statistics*, *9*(1), Article ID:1320177. https://doi.org/10.3389/fams.2023.1320177

Andini, W., Zahra, S. K., Abdurrahman, M., & Sebayang, V. B. (2024). Analisis Fluktuasi Harga Terhadap Faktor-Faktor yang Mempengaruhi Produktivitas Usaha Tani Cabai Merah di

Indonesia. *Jurnal Riset dan Inovasi Manajemen*, *2*(2), 162–172. https://doi.org/10.59581/jrim-widyakarya.v2i2.3526

Arnob, A. S., Kausik, A. K., Islam, Z., Khan, R., & Bin Rashid, A. (2025). Comparative Result Analysis of Cauliflower Disease Classification Based on Deep Learning Approach VGG16, Inception V3, ResNet, and a Custom CNN Model. *Hybrid Advances*, *10*, Article ID: 100440. https://doi.org/10.1016/j.hybadv.2025.100440

Chaithanya, A. S., & Rachana, M. (2023). Identification of Diseased Papaya Leaf Through Transfer Learning. *Indian Journal of Science and Technology*, *16*(48), 4676–4687. https://doi.org/10.17485/IJST/v16i48.2690

Gulzar, Y. (2023). Fruit Image Classification Model Based on MobileNetV2 with Deep Transfer Learning Technique. *Sustainability*, *15*(3), Article ID: 1906. https://doi.org/10.3390/su15031906

Hermanto, A. R., Aziz, A., & Sudianto, S. (2024). Perbandingan Arsitektur MobileNetV2 dan RestNet50 untuk Klasifikasi Jenis Buah Kurma. *Jurnal Sistem dan Teknologi Informasi (JustIN)*, *12*(4), 630–637. https://doi.org/10.26418/justin.v12i4.80358

Hicks, S. A., Strümke, I., Thambawita, V., Hammou, M., Riegler, M. A., Halvorsen, P., & Parasa, S. (2022). On Evaluation Metrics for Medical Applications of Artificial Intelligence. *Scientific Reports*, *12*(1), Article ID: 5979. https://doi.org/10.1038/s41598-022-09954-8

Li, D., Zhang, C., Li, J., Li, M., Huang, M., & Tang, Y. (2024). MCCM: Multi-Scale Feature Extraction Network for Disease Classification and Recognition of Chili Leaves. *Frontiers in Plant Science*, *15*, 1–19. https://doi.org/10.3389/fpls.2024.1367738

Li, Y., Zou, D., Shrestha, N., Xu, X., Wang, Q., Jia, W., & Wang, Z. (2020). Spatiotemporal Variation in Leaf Size and Shape in Response to Climate. *Journal of Plant Ecology*, *13*(1), 87–96. https://doi.org/10.1093/jpe/rtz053

Maftukhah, A., Fadlil, A., & Sunardi, S. (2024). Butterfly Image Classification Using Convolution Neural Network with AlexNet Architecture. *Jurnal INFOTEL*, *16*(1), 82–95. https://doi.org/10.20895/infotel.v16i1.1004

Mostafa, S., Wang, Y., Zeng, W., & Jin, B. (2022). Plant Responses to Herbivory, Wounding, and Infection. *International Journal of Molecular Sciences*, *23*(13), Article ID: 7031. https://doi.org/10.3390/ijms23137031

Muis, A., Sunardi, S., & Yudhana, A. (2024). CNN-based Approach for Enhancing Brain Tumor Image Classification Accuracy. *International Journal of Engineering*, *37*(5), 984–996. https://doi.org/10.5829/IJE.2024.37.05B.15

Nguyen, T.-H., Nguyen, T.-N., & Ngo, B.-V. (2022). A VGG-19 Model with Transfer Learning and Image Segmentation for Classification of Tomato Leaf Disease. *AgriEngineering*, *4*(4), 871–887. https://doi.org/10.3390/agriengineering4040056

Paymode, A. S., & Malode, V. B. (2022). Transfer Learning for Multi-Crop Leaf Disease Image Classification Using Convolutional Neural Network VGG. *Artificial Intelligence in Agriculture*, *6*(1), 23–33. https://doi.org/10.1016/j.aiia.2021.12.002

Pramudhita, D. A., Azzahra, F., Arfat, I. K., Magdalena, R., & Saidah, S. (2023). Strawberry Plant Diseases Classification Using CNN Based on MobileNetV3-Large and EfficientNet-B0 Architecture. *Jurnal Ilmiah Teknik Elektro Komputer dan Informatika*, *9*(3), 522–534. https://doi.org/10.26555/jiteki.v9i3.26341

Prasetyo, A. D., & Agustinur, A. (2022). Inventarisasi Penyakit pada Tanaman Cabai Merah (Capsicum annum l.) di Kebun Warga Gampong Suak Raya Kecamatan Johan Pahlawan Kabupaten Aceh Barat. *Jurnal Agrotek Lestari*, *8*(1), 70–75. https://doi.org/10.35308/jal.v8i1.5293

Prasetyo, K., Putri, D. D., Kartika Eka Wijayanti, I., & Zulkifli, L. (2023). Forecasting of Red Chilli Prices in Banyumas Regency: The ARIMA Approach. *E3S Web of Conferences*, *444*, Article ID: 02017. https://doi.org/10.1051/e3sconf/202344402017

Raghuram, K., & Borah, M. D. (2025). A Hybrid Learning Model for Tomato Plant Disease Detection Using Deep Reinforcement Learning with Transfer Learning. *Procedia Computer Science*, *252*, 341–354. https://doi.org/10.1016/j.procs.2024.12.036

Ramadhani, A. N. M., Saraswati, G. W., Agung, R. T., & Santoso, H. A. (2023). Performance Comparison of Convolutional Neural Network and MobileNetV2 for Chili Diseases

Classification. *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, *7*(4), 940–946. https://doi.org/10.29207/resti.v7i4.5028

Raufer, L., Wiedey, J., Mueller, M., Penava, P., & Buettner, R. (2025). A Deep Learning-Based Approach for the Detection of Cucumber Diseases. *PLOS ONE*, *20*(4), Article ID: e0320764. https://doi.org/10.1371/journal.pone.0320764

Rustanto, D. W., Liantoni, F., & Prakisya, N. P. T. (2024). Identifikasi Penyakit Daun pada Tanaman Padi Menggunakan Ekstraksi Fitur Gray Level Co-occurrence Matrix (GLCM) dan Metode K-Nearest Neighbour (KNN). *Jurnal Sistem dan Teknologi Informasi (JustIN)*, *12*(1), 100–106. https://doi.org/10.26418/justin.v12i1.69752

Sekretariat Jenderal Kementerian Pertanian. (2023). *Statistik Konsumsi Pangan Tahun 2023* (Mas'ud & S. Wahyuningsih, Eds.). Pusat Data dan Sistem Informasi Pertanian.

Siddiqui, F. M. (2023). Chili Leaf Disease Prediction Using CNN. *International Journal for Research in Applied Science and Engineering Technology*, *11*(5), 4791–4797. https://doi.org/10.22214/ijraset.2023.52757

Singh, T., Kumar, K., & Bedi, S. (2021). A Review on Artificial Intelligence Techniques for Disease Recognition in Plants. *IOP Conference Series: Materials Science and Engineering*, *1022*(1), Article ID: 012032. https://doi.org/10.1088/1757-899X/1022/1/012032

Syukur, M. (2018). *8 Kiat Sukses Panen Cabai Sepanjang Musim*. AgroMedia Pustaka.

Winiarti, S., & Khoirunnisa, I. I. (2024). Mobile Application Development for Chili Disease Detection with Convolutional Neural Network. *International Journal of Informatics and Computation*, *6*(2), 97–112. https://doi.org/10.35842/ijicom.v6i2.93

JISKA (Jurnal Informatika Sunan Kalijaga)
Vol. 11, No. 1, JANUARI, 2026, Pp. 83 – 97
ISSN: 2527 – 5836 (print) | 2528 – 0074 (online)

Sistem Deteksi Gerakan Kecurangan UTBK *Real-Time* dengan YOLOv8 dan Optical Flow

Muhammad Naufal Ardiansyah [(1)*], Farrel Zikri Suryahadi [(2)], Hendrico Edhent Surya Pratama [(3)], Anggraini Puspita Sari [(4)]
Departemen Informatika, Universitas Pembangunan Nasional Veteran Jawa Timur, Surabaya, Indonesia
e-mail : {23081010047,23081010213,23081010070}@student.upnjatim.ac.id, anggraini.puspita.if@upnjatim.ac.id.
* Penulis korespondensi.
Artikel ini diajukan 15 Juli 2025, direvisi 16 Oktober 2025, diterima 17 Oktober 2025, dan dipublikasikan 25 Januari 2025.

Abstract

Integrity and honesty are fundamental aspects of education, including the implementation of the Computer-Based Written Examination (UTBK). Conventional exam supervision is considered less effective in monitoring participants' behavior due to the limitations in human observation capabilities and consistency. This study develops a real-time cheating-detection system based on camera input by integrating the YOLOv8 algorithm with Farnebäck optical flow. The YOLOv8 algorithm identifies participants' body poses and activities directly from video footage, while Optical Flow analyzes the direction and motion patterns between frames over time. The system is designed to recognize various suspicious poses such as head-turning, bowing, and cheating-related gestures that indicate potential dishonesty. All detection results are automatically recorded in an SQLite database, complete with timestamps and visual evidence. Experimental results show that the system achieves 94.3% accuracy in detecting suspicious movements. The combination of both methods also helps maintain detection stability when keypoints are not consistently captured in some frames. Additionally, the system is equipped with a graphical user interface (GUI) to facilitate easier monitoring and analysis. These results demonstrate that a pose-and-motion analysis-based approach offers an intelligent and efficient solution for enhancing digital supervision of UTBK examinations.

Keywords: UTBK Cheating, Motion Detection, Pose Estimation, YOLOv8, Optical Flow

Abstrak

Integritas dan kejujuran menjadi aspek krusial dalam pendidikan, termasuk pelaksanaan Ujian Tulis Berbasis Komputer (UTBK). Pengawasan ujian secara konvensional dinilai belum optimal dalam memantau perilaku peserta akibat keterbatasan kemampuan dan konsistensi pengamatan manusia. Penelitian ini mengembangkan sistem deteksi kecurangan secara real-time berbasis kamera dengan mengintegrasikan algoritma YOLOv8 dan Optical Flow Farneback. Metode algoritma YOLOv8 dimanfaatkan untuk mengidentifikasi pose dan aktivitas tubuh peserta secara langsung dari rekaman video, sedangkan Optical Flow digunakan untuk menganalisis arah dan pola pergerakan antar frame secara temporal. Sistem ini dirancang untuk mengenali berbagai pose mencurigakan seperti menoleh, menunduk, dan gerakan mencontek yang mengindikasikan potensi kecurangan. Seluruh hasil deteksi dicatat secara otomatis dalam basis data SQLite dengan informasi waktu dan bukti visual. Hasil pengujian menunjukkan bahwa sistem mampu mencapai akurasi sebesar 94,3% dalam mendeteksi gerakan mencurigakan. Kombinasi kedua metode tersebut juga membantu menjaga stabilitas deteksi saat *keypoint* tidak terbaca secara konsisten pada frame tertentu. Selain itu, sistem dilengkapi antarmuka GUI untuk mendukung kemudahan dalam proses pemantauan dan analisis. Hasil ini menunjukkan bahwa pendekatan berbasis analisis pose dan gerakan memberikan solusi cerdas dan efisien dalam pengawasan digital UTBK.

Kata Kunci: Kecurangan UTBK, Deteksi Gerakan, Estimasi Pose, YOLOv8, Optical Flow

1. PENDAHULUAN

Kejujuran akademik merupakan pilar penting dalam membentuk pendidikan yang berintegritas (Tjahyanti & Gitakarma, 2024). Nilai ini sangat krusial diterapkan dalam berbagai aspek pendidikan, termasuk dalam pelaksanaan Ujian Tulis Berbasis Komputer (UTBK) sebagai metode seleksi masuk perguruan tinggi (Simarmata et al., 2022). Meskipun pelaksanaannya diawasi langsung oleh pengawas di ruang ujian, efektivitas sistem pengawasan konvensional ini masih dipertanyakan (Pangestu et al., 2024). Ketergantungan penuh pada pengamatan manusia membuat pengawasan rentan terhadap faktor kelelahan, kelengahan, dan keterbatasan jangkauan (Thohari et al., 2025; Tjahyanti & Gitakarma, 2024). Pelaksanaan UTBK secara luring justru bisa meningkatkan potensi terjadinya kecurangan, seperti menoleh, mencontek, gerakan menunduk mencurigakan, hingga berkomunikasi dengan peserta lain (Gopane et al., 2024; Iskandar et al., 2024; Karim et al., 2020). Meskipun teknologi terus berkembang, tantangan dalam mendeteksi dan mencegah berbagai gerakan kecurangan tersebut masih menjadi hambatan dalam menjaga integritas dan keadilan seleksi UTBK.

Sistem pengawasan berbasis kamera atau webcam kini umum digunakan dalam pelaksanaan UTBK untuk membantu pengawas memantau aktivitas peserta secara lebih menyeluruh. Pendekatan ini dinilai masih belum optimal jika tidak didukung oleh teknologi cerdas yang mampu mengidentifikasi perilaku mencurigakan secara otomatis (Wicaksono & Yamasari, 2025). Salah satu bentuk inovasi yang dapat diterapkan adalah pemanfaatan kecerdasan buatan (AI) melalui integrasi kamera pada komputer ujian (Hadibrata & Rochadiani, 2024a; Sari et al., 2024a). Teknologi ini berpotensi mendeteksi berbagai bentuk kecurangan, seperti gerakan menoleh, keberadaan orang lain dalam frame, serta indikasi gerakan mencontek (Erdem & Karabatak, 2025). Pengembangan sistem berbasis AI menjadi langkah strategis untuk meningkatkan efektivitas pengawasan, sekaligus menjaga kejujuran dan kredibilitas proses seleksi UTBK secara menyeluruh.

Beberapa penelitian sebelumnya telah mengembangkan sistem deteksi kecurangan ujian menggunakan algoritma YOLO. Penelitian pertama memanfaatkan YOLOv8 untuk mendeteksi pose mencurigakan seperti menoleh ke kanan, kiri, dan belakang, dengan akurasi tinggi (*precision* 0.952, *recall* 0.966, dan mAP50 0.984) serta mampu berjalan real-time pada 28 FPS, namun masih terbatas pada lingkungan data yang spesifik (Thohari et al., 2025). Penelitian lainnya menggunakan YOLO untuk mendeteksi penggunaan ponsel dan interaksi mencurigakan antar peserta ujian dengan akurasi 85–90% dan kecepatan 25 FPS, meskipun masih ada beberapa kasus kecurangan yang luput terdeteksi, menunjukkan perlunya peningkatan pada aspek *recall* dan generalisasi model (Tjahyanti & Gitakarma, 2024).

Beberapa penelitian sebelumnya juga telah mengembangkan sistem deteksi gerakan, khususnya gerakan kepala, menggunakan metode optical flow. Salah satu penelitian memanfaatkan Lucas-Kanade Optical Flow dan facial landmarks untuk membantu penyandang disabilitas memilih menu tanpa sentuhan, dengan akurasi tinggi (94,6%–98,6%) dan waktu komputasi di bawah 2,5 detik. Optical Flow dipilih karena efisien mendeteksi pergerakan antar frame tanpa tergantung pada ukuran objek (Pratama et al., 2021). Penelitian lain mengembangkan sistem serupa berbasis NVIDIA Jetson Nano untuk pemilihan menu makanan *touchless* guna mengurangi kontak fisik selama pandemi COVID-19. Sistem ini juga menggunakan facial landmarks dan menunjukkan akurasi rata-rata 85,7% serta waktu komputasi cepat (sekitar 0,041 detik), menjadikannya solusi efektif untuk interaksi tanpa sentuhan (Hidayatullah et al., 2022).

Berdasarkan penelitian-penelitian yang telah disebutkan dalam upaya mengembangkan sistem deteksi kecurangan dan deteksi pergerakan, maka bisa diketahui bahwa penggunaan metode *You Only Look Once* (YOLO) terbukti berhasil membuat sistem deteksi gerakan secara *real-time* (Nur et al., 2023). Tidak hanya itu, penggunaan metode *Optical Flow* juga terbukti berhasil dalam mendeteksi sebuah pergerakan. Dengan demikian, integrasi metode YOLOv8 dan Optical Flow dipandang sesuai untuk digunakan dalam penelitian ini. Kombinasi kedua metode tersebut

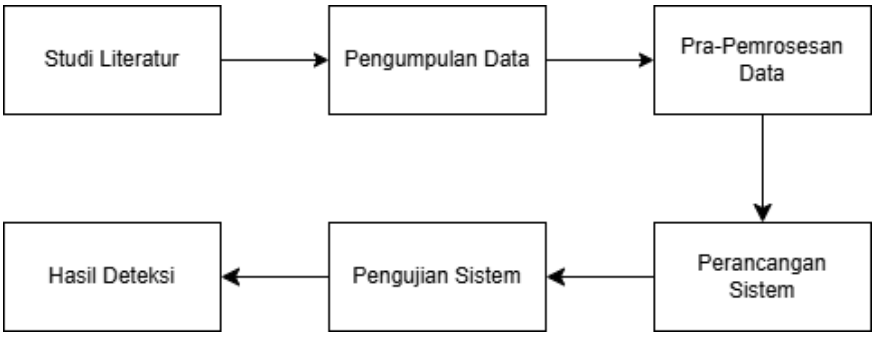
berperan dalam mendukung proses pengawasan dengan tujuan meminimalkan potensi kecurangan selama pelaksanaan UTBK.

Metode YOLOv8 merupakan model algoritma berbasis deep learning yang dirancang untuk mendeteksi sebuah gerakan atau postur tubuh secara real-time dengan akurasi tinggi pada setiap frame (Bimantoro et al., 2024; Saepudin et al., 2024). Model ini memiliki keunggulan dalam kecepatan pemrosesan serta efisien dalam mengenali berbagai entitas visual, termasuk postur tubuh peserta dan klasifikasi gerakan mencurigakan, sehingga sangat relevan diterapkan dalam sistem pemantauan otomatis. Dalam konteks deteksi pose, YOLOv8 mampu mengidentifikasi titik-titik kunci (*keypoints*) pada tubuh manusia yang merepresentasikan postur atau gerakan spesifik, seperti menoleh, menunduk, hingga sikap kondusif. Peningkatan akurasi dalam analisis arah dan dinamika gerakan dapat dicapai melalui kombinasi YOLOv8 dengan metode Optical Flow. Optical Flow merupakan teknik pemrosesan citra yang menghitung perubahan posisi piksel antar frame secara berurutan, sehingga mampu memetakan arah dan kecepatan gerakan secara lebih halus dan kontinu (Syaharuddin et al., 2023). Integrasi antara kemampuan deteksi spasial dari YOLOv8 dan analisis gerak temporal melalui Optical Flow menciptakan sistem yang lebih komprehensif dan adaptif, khususnya dalam mendeteksi aktivitas tidak wajar dalam konteks pemantauan real-time.

Pada konteks pelaksanaan UTBK, integrasi antara metode YOLOv8 dan Optical Flow menunjukkan tingkat relevansi dan efektivitas yang tinggi dalam mendeteksi gerakan mencurigakan secara real-time. YOLOv8 memiliki kapabilitas untuk mengidentifikasi gerakan peserta serta postur tubuh secara langsung pada setiap frame, sehingga mampu mengenali beragam klasifikasi gerakan yang berpotensi mencurigakan (Zuo et al., 2024). Di sisi lain, Optical Flow berfungsi untuk menganalisis pergerakan peserta antar frame melalui pelacakan perubahan posisi piksel secara kontinu, sehingga menghasilkan informasi yang lebih rinci mengenai arah dan intensitas gerakan (Pratama et al., 2021). Sinergi antara kedua metode ini memungkinkan sistem untuk menangkap dan memahami pola gerakan peserta secara komprehensif, mencakup aspek spasial maupun temporal. Dengan demikian, penerapan gabungan YOLOv8 dan Optical Flow dinilai tepat untuk mendukung sistem pemantauan otomatis yang lebih akurat, efisien, dan objektif dalam konteks pengawasan ujian. Penelitian ini bertujuan mengembangkan sistem deteksi kecurangan UTBK dengan mengintegrasikan YOLOv8 untuk identifikasi pose gerakan secara real-time, serta Optical Flow untuk menganalisis arah dan intensitas pergerakan peserta. Sistem ini dirancang untuk mengenali perilaku mencurigakan seperti menoleh, menunduk, atau mencontek. Adapun pengembangan sistem deteksi berbasis YOLO dan *Optical Flow* telah digunakan dalam berbagai studi sebelumnya, namun sebagian besar penerapannya masih secara terpisah dan belum spesifik pada konteks pengawasan UTBK. Oleh karena itu, integrasi keduanya dalam penelitian ini menawarkan pendekatan baru yang lebih adaptif dan efektif dalam mendeteksi potensi kecurangan saat ujian berbasis komputer.

2. METODE PENELITIAN

Penelitian ini menguraikan tahapan-tahapan sistematis untuk merancang dan membangun sebuah sistem cerdas yang mampu mengidentifikasi perilaku tidak wajar selama ujian (Jannah, 2025). Untuk mencapai tujuan tersebut, penelitian ini mengadopsi pendekatan teknis yang menggabungkan dua algoritma utama, yaitu YOLOv8 dan Optical Flow. Kolaborasi kedua teknologi ini bertujuan untuk menekan angka kecurangan selama UTBK dengan memanfaatkan kemampuan keduanya, sistem yang dikembangkan dapat secara langsung (real-time) menganalisis rekaman video dari kamera untuk memantau pergerakan peserta ujian. Sistem ini juga dirancang untuk dapat mengenali kehadiran individu lain di dalam area pantauan, sambil tetap memprioritaskan peserta ujian sebagai subjek pengawasan utama. Adapun tahapan penelitian yang dilakukan demi mewujudkan sistem deteksi gerakan kecurangan UTBK yang terdapat pada Gambar 1.



Gambar 1 Alur Kerja Penelitian

2.1 Studi Literatur

Pada penelitian ini dimulai dengan studi literatur mengenai metode deteksi gerakan mencurigakan guna mengurangi kecurangan pada UTBK berbasis deteksi pose. Berdasarkan dari *research* tersebut, penelitian memfokuskan pada gerakan peserta yang diduga mencurigakan melakukan kecurangan pada ujian UTBK. Dalam kasus ini, diperlukan rancangan sistem yang mampu mendeteksi kecurangan tersebut. Beberapa studi literatur yang digunakan yaitu algoritma YOLOv8 (You Only Look Once version 8), algoritma Optical Flow Farneback, dan pengukuran akurasi deteksi gerakan. Beberapa skema literatur telah dikaji dalam penelitian yang membahas deteksi gerakan mencurigakan pada pelaksanaan ujian UTBK.

Algoritma YOLOv8 dipilih karena kemampuannya yang telah teruji dalam bidang Computer Vision, khususnya dalam analisis postur dan deteksi pose tubuh manusia secara real-time. Model ini unggul dalam mengidentifikasi titik-titik kunci tubuh (*keypoints*) dengan presisi tinggi pada setiap frame video, yang memungkinkan sistem mengenali berbagai pola gerakan seperti menoleh, menunduk, atau posisi tubuh lainnya (Bimantoro et al., 2024; Saepudin et al., 2024). Proses deteksi pose pada YOLOv8 mencakup identifikasi koordinat anatomi tubuh berdasarkan anotasi per gambar yang telah dilabeli sebelumnya (Wulandari & Rosandy, 2024). Selain itu, pembagian input gambar dilakukan dalam format grid untuk mempermudah prediksi letak *keypoint* tubuh. Hal ini membuat YOLOv8 pose ideal untuk aplikasi pemantauan aktivitas atau perilaku berdasarkan pergerakan tubuh peserta dalam konteks ujian. Nilai confidence score dalam deteksi pose ditentukan menggunakan Pers. (1), di mana $\Pr(Pose)$ merupakan probabilitas gerakan yang terdeteksi, sedangkan IoU merupakan perbandingan antara prediksi model dengan ground truth. Selanjutnya, untuk memperoleh nilai skor objektif, digunakan Pers. (2), sehingga confidence akhir mempertimbangkan probabilitas objek sekaligus kualitas overlap prediksinya.

$$C = \Pr(Pose) \times IoU_{predict}^{truth} \quad (1)$$

$$Score_{(obj)} = c \times p_{(obj)} \quad (2)$$

Algoritma Optical Flow Farneback digunakan untuk mendeteksi perubahan postur tubuh yang tidak terduga dan berpotensi mencerminkan tindakan curang, seperti gerakan menoleh ke samping, menunduk, atau mencontek (Syaharuddin et al., 2023). Teknik ini bekerja dengan melacak perubahan pola piksel antar dua bingkai gambar (frame) berturut-turut untuk memperkirakan arah dan kecepatan gerakan tubuh. Perhitungan dilakukan berdasarkan pergeseran piksel pada sumbu horizontal (X) dan vertikal (Y), di mana dominasi pergerakan pada sumbu horizontal dapat mengindikasikan gerakan menoleh, sementara pergeseran signifikan ke arah vertikal negatif dapat menunjukkan aktivitas menunduk.

Metode *Optical Flow Farneback* menghitung dua frame secara penuh dengan dua peta vektor yaitu vektor u dan v. Vektor u mewakili garis horizontal dan vektor v mewakili garis vertikal dalam diagram kartesius. Estimasi aliran optik klasik dapat dinyatakan pada Pers. (3), yang

menunjukkan bahwa intensitas piksel dianggap konstan antar frame. Bentuk diferensial dari pendekatan ini dituliskan dalam Pers. (4), sehingga aliran gerakan dapat didekati melalui gradien spasial dan temporal.

$$Metode\ Klasik = l(x, y, t) - I(x + u, y + v, t\ + 1) = 0 \quad (3)$$

$$Metode\ Dasar = \frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v + \frac{\partial I}{\partial t} = 0 \quad (4)$$

Sebagai upaya mendefinisikan gerakan tiba-tiba secara vertikal dan horizontal, diperlukan magnitudo dan arah gerakan sesuai dengan kriteria pergerakan mencurigakan. Besarnya magnitudo dihitung menggunakan Persamaan (5), yang menunjukkan kekuatan pergerakan dalam dua dimensi. Sementara itu, arah gerakan (angle) diperoleh dari Persamaan (6), sehingga dapat dianalisis apakah gerakan peserta cenderung horizontal (menoleh) atau vertikal (menunduk). Jika u lebih besar dari v maka dianggap menoleh, sedangkan jika v lebih besar dari u dianggap menunduk.

$$Magnitude = \sqrt{u^2 + v^2} \quad (5)$$

$$Angle = tan^{-1}\left(\frac{u}{v}\right) \quad (6)$$

2.2 Pengukuran Akurasi Deteksi

Penilaian kinerja keandalan sistem akan didasarkan pada dua parameter evaluasi utama: presisi (precision) dan perolehan (*recall*) (Prabowo, 2021). Presisi mengukur tingkat akurasi dari deteksi yang dinyatakan positif; artinya, seberapa banyak dari semua kejadian yang ditandai sebagai pelanggaran oleh sistem yang memang merupakan pelanggaran nyata, dengan perhitungan presisi menggunakan Pers. (7). Di sisi lain, perolehan atau *recall* menunjukkan sensitivitas sistem dalam menemukan semua kasus pelanggaran yang sesungguhnya terjadi, termasuk yang mungkin luput dari deteksi, dihitung menggunakan Persamaan (8). Selain itu, akurasi digunakan sebagai ukuran tambahan untuk melihat seberapa besar proporsi deteksi yang dilakukan sistem sesuai dengan kondisi sebenarnya, baik untuk kasus pelanggaran maupun bukan pelanggaran, dihitung menggunakan Pers. (9). Dengan demikian, kombinasi ketiga metrik ini membantu memberikan gambaran yang lebih menyeluruh terkait kemampuan sistem dalam mendeteksi pelanggaran secara efektif dan konsisten.

$$Precision = \frac{True\ Positive\ (TP)}{True\ Positive\ (TP) + False\ Positive\ (FP)} \quad (7)$$

$$Recall = \frac{True\ Positive\ (TP)}{True\ Positive\ (TP) + False\ Negative\ (FN)} \quad (8)$$

$$Akurasi = \frac{TP + TN}{TP + TN + FP + FN} \quad (9)$$

2.3 Teknik Pengumpulan Data

Pengumpulan data pada penelitian ini dikumpulkan data gambar gerakan orang yang diklasifikasikan kedalam beberapa label pose. Subjek eksperimen untuk pengujian data melibatkan sembilan orang partisipan dengan suasana ruang kelas yang sesuai dengan suasana UTBK. Masing-masing partisipan diminta untuk melakukan berbagai jenis aktivitas, baik yang tergolong sebagai perilaku normal maupun mencurigakan, seperti menoleh kanan, kiri, gerakan menunduk, atau mencontek melihat kertas catatan. Terdapat 180 gambar yang diperoleh sebagai data citra aktual, di mana setiap individu direkam dalam lima pose aktivitas yang dilakukan sebanyak dua kali dari dua sudut kamera. Dengan demikian, total citra gambar awal yang

dihasilkan berjumlah 180. Selanjutnya, data citra tersebut dibagi (split) ke dalam tiga subset, yaitu 69% untuk data pelatihan, 19% untuk data validasi, dan 11% untuk data pengujian. Hasil dari pembagian ini menghasilkan 125 citra untuk pelatihan, 35 citra untuk validasi, dan 20 citra untuk pengujian. Untuk meningkatkan keragaman data serta memperkaya kemungkinan variasi kondisi, dilakukan proses augmentasi berupa penyesuaian saturasi, kecerahan, eksposur, penambahan blur, serta noise. Proses augmentasi ini diterapkan pada data pelatihan dengan melipatgandakan jumlah citra sebanyak tiga kali. Dengan demikian, total keseluruhan dataset citra yang diperoleh setelah augmentasi adalah 430 citra, terdiri atas 375 citra pelatihan, 35 citra validasi, dan 20 citra pengujian.

2.4 Pra-Pemrosesan Data

Pemrosesan data dilakukan guna untuk menyediakan gambar agar bisa digunakan sebagai dataset pose estimasi gerakan mencurigakan UTBK. Gerakan tersebut diklasifikasikan dengan labelnya masing-masing. Hal itu mempermudah model YOLOv8 mengenali data yang sudah dikelompokkan berdasarkan labelnya. Pada Gambar 2 menampilkan sejumlah skema pemrosesan data.

Gambar 2 Pre-Processing Data

1) Ekstraksi
   Pada tahap ini dilakukan proses ekstraksi untuk memperoleh ciri-ciri penting dari data peserta ujian, seperti posisi tubuh dan arah gerakan. Ekstraksi ini berperan penting dalam mengidentifikasi perilaku mencurigakan, seperti menoleh, menunduk, atau mencontek, sehingga sistem mampu mengklasifikasikan tindakan tidak wajar secara tepat. Informasi visual yang dihasilkan merepresentasikan aktivitas peserta secara menyeluruh.

2) Segmentasi
   Setelah proses ekstraksi, dilakukan segmentasi untuk memisahkan dan mengelompokkan bagian-bagian penting dalam citra, seperti area tubuh yang terdeteksi. Segmentasi dilakukan dengan cara memberikan area box deteksi pada data gambar, sehingga sistem dapat membatasi fokus analisis hanya pada wilayah yang relevan.

3) Anotasi
   Tahap selanjutnya adalah anotasi, yaitu proses pemberian label pada data gambar untuk menandai bagian-bagian tertentu yang merepresentasikan pose atau gerakan tubuh. Pada proses ini digunakan titik *keypoint* yang diletakkan pada area tubuh seperti kepala, bahu, siku, dan lutut. Titik-titik ini merepresentasikan struktur tubuh peserta dan menjadi acuan dalam pelatihan model untuk mengenali pola gerakan secara akurat.

2.5 Perancangan dan Uji Sistem

Perancangan sistem dimulai dengan estimasi pose berbasis YOLOv8 yang telah dikustomisasi, serta algoritma Optical Flow Farneback. Proses pelatihan model dilakukan menggunakan dataset khusus yang mengklasifikasikan lima jenis gerakan mencurigakan selama UTBK dengan pelabelan secara manual. Dataset tersebut mencakup 430 citra dengan pembagain 375 gambar pelatihan, 35 gambar validasi, dan 20 gambar pengujian, yang semulanya berjumlah 180 gambar. Langkah perancangan selanjutnya, Optical Flow akan diintegrasikan untuk menangkap arah serta pola pergerakan tubuh secara temporal, di mana hasilnya dipadukan dengan bounding box dari YOLOv8 untuk memperkuat analisis gerakan antar frame. Setiap deteksi pelanggaran pada frame video akan memicu sistem untuk secara otomatis menyimpan bukti visual berupa gambar ke dalam folder *snapshots*.

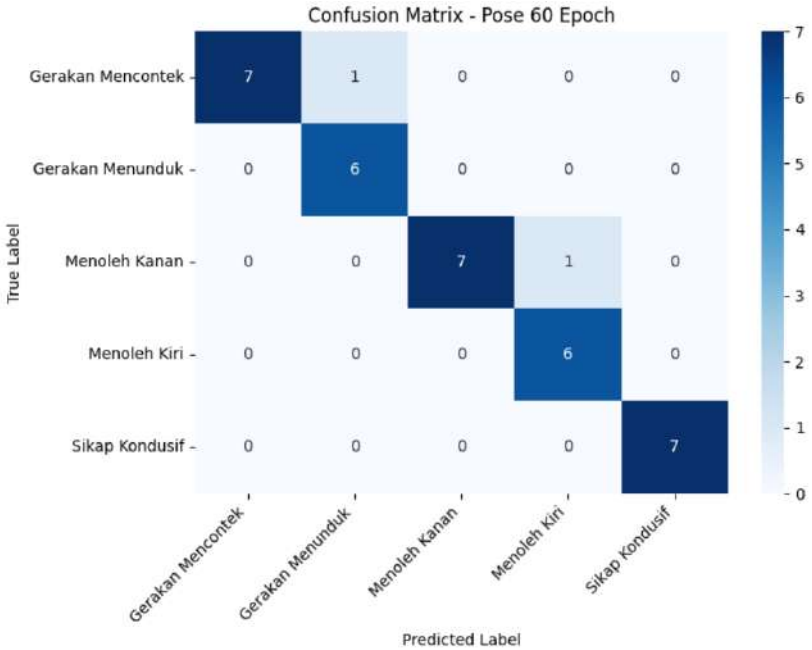
Dokumentasi ini didukung dengan sistem yang berjalan secara real-time menggunakan kamera yang terhubung dengan database lokal SQLite bernama log_gerakan.db. Seluruh data deteksi, termasuk jenis gerakan, waktu kejadian, dan identitas peserta, dicatat secara otomatis ke dalam database tersebut. Agar data ini mudah dikelola dan dianalisis, sistem dilengkapi antarmuka berbasis Tkinter yang menampilkan informasi secara terstruktur, status gerakan terkini, serta histogram kejadian. Antarmuka ini juga memungkinkan peninjauan kembali bukti gambar dari folder snapshots dan menyediakan fungsi ekspor data ke format CSV. Uji coba sistem dilakukan dalam skenario pengawasan UTBK untuk memastikan keandalan integrasi model dan akurasi pencatatan data gerakan.

3. HASIL DAN PEMBAHASAN

Evaluasi data model pose yang dilatih menggunakan YOLOv8 dengan beberapa klasifikasi gerakan disusun dengan anotasi yang sesuai dengan standar *pose estimation*. Kemudian, data dilatih selama 60 dan 120 *epoch* menggunakan file konfigurasi pose.yaml yang akan dibuat perbandingan berdasarkan *epoch*. Model best.pt digunakan dalam sistem karena menunjukkan performa validasi terbaik dibandingkan model last.pt. Hasil pelatihan dievaluasi berdasarkan metrik *precision*, *recall*, dan mean *Average Precision* (mAP). Secara umum, model mampu mengenali pola gerakan dengan baik dan akurat pada data uji, ditunjukkan oleh peningkatan akurasi selama training.

3.1 Evaluasi Skema 60 Epoch

Evaluasi terhadap skema pelatihan selama 60 *epoch* dilakukan untuk menilai peningkatan kinerja model dalam mendeteksi pose gerakan mencurigakan secara lebih optimal. Pengamatan dilakukan terhadap sejumlah matrik evaluasi, seperti akurasi, presisi, *recall*, serta visualisasi hasil pelatihan. Hasil evaluasi disajikan melalui Tabel 1 yang merepresentasikan performa model secara visual dan kuantitatif selama proses pelatihan berlangsung. Selain itu, pada Gambar 3 merupakan confusion matrix yang digunakan sebagai dasar perhitungan classification report dan akurasi seperti pada Tabel 2.



Gambar 3 Confusion Matrix 60 Epoch

Confusion matrix pada Gambar 3 menunjukkan hasil dari setiap gerakan yang berhasil dideteksi secara benar. Pembagian gerakan tersebut didasarkan dataset validasi yang berjumlah 35 data.

Adapun masih ada data gerakan yang gagal dideteksi secara benar, yaitu gerakan menunduk yang diprediksi gerakan mencontek, dan gerakan menoleh kiri yang diprediksi gerakan menoleh kanan.

Dalam menunjukkan bahwa model mampu mengklasifikasikan sebagian besar pose gerakan dengan benar ditunjukkan pada Tabel 1, dengan dominasi prediksi pada diagonal utama confusion matrix. Gerakan mencontek dan sikap kondusif memiliki akurasi tertinggi, sementara misklasifikasi masih terjadi pada gerakan menunduk yang sering tertukar dengan mencontek. Hal ini diduga akibat kemiripan pose atau terbatasnya data pelatihan.

Untuk menampilkan hasil *precision*, Tabel 2 menyajikan nilai nominal *recall*, dan *F1-score* yang tinggi di hampir seluruh kelas, dengan nilai sempurna 1.00 pada label sikap kondusif. *Recall* pada gerakan menoleh kanan dan gerakan mencontek sedikit menurun, menandakan masih ada deteksi yang terlewat. Secara keseluruhan, model menunjukkan kinerja baik dengan akurasi deteksi mencapai 94,3%.

Tabel 1 Hasil Evaluasi Training Data 60 Epoch

Epoch	Time (sec)	Metrics precision (B)	Metrics recall (B)	Metrics mAP50 (B)	Metrics mAP50-95 (B)
1	177.169	0.23782	0.91429	0.47186	0.38817
2	346.04	0.30361	0.93667	0.51299	0.34844
3	512.557	0.40689	0.78123	0.55974	0.39996
4	679.235	0.40763	0.91274	0.64495	0.52852
5	846.289	0.38554	0.91429	0.55699	0.46893
…	…	…	…	…	…
56	9249.51	0.78571	1.0	0.89079	0.78249
57	9444.56	0.79957	0.97143	0.87992	0.77446
58	9645.43	0.83202	0.9588	0.94478	0.83048
59	9828.39	0.8847	0.94286	0.95793	0.84287
60	10014.9	0.89119	0.94286	0.95548	0.84243

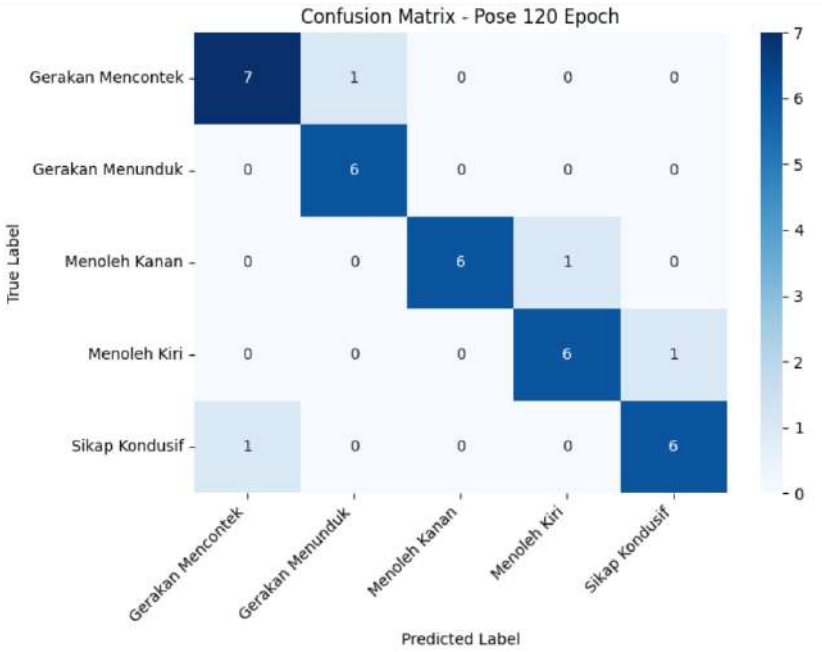
Tabel 2 Classification Pose Data 60 Epoch

Label	Precision	Recall	F1-Score	Support
Gerakan Mencontek	1.0000	0.8750	0.9333	8
Gerakan Menunduk	0.8571	1.0000	0.9231	6
Menoleh Kanan	1.0000	0.8750	0.9333	8
Menoleh Kiri	0.8571	1.0000	0.9231	6
Sikap Kondusif	1.0000	1.0000	1.0000	7
Total Accuracy		**94.2857%**		

3.2 Evaluasi Skema 120 Epoch

Evaluasi terhadap skema pelatihan selama 120 *epoch* dilakukan untuk menilai peningkatan kinerja model dalam mendeteksi pose gerakan mencurigakan secara lebih optimal. Pengamatan dilakukan terhadap sejumlah matrik evaluasi, seperti akurasi, presisi, *recall*, serta visualisasi hasil pelatihan. Hasil evaluasi disajikan melalui Tabel 3 yang merepresentasikan performa model secara visual dan kuantitatif selama proses pelatihan berlangsung. Selain itu, pada Gambar 4 merupakan confusion matrix yang digunakan sebagai dasar perhitungan classification report dan akurasi seperti pada Tabel 4.

Dominasi prediksi pada diagonal utama confusion matrix yang ditampilkan pada Gambar 4 menunjukkan bahwa sebagian besar pose gerakan telah berhasil diklasifikasikan secara akurat oleh model. Pembagian gerakan tersebut didasarkan dataset validasi yang berjumlah 35 data. Adapun ada sedikit data gerakan yang membuat model bingung memprediksi citra, namun sebagian besar sudah memprediksi gerakan yang benar.



Gambar 4 Confusion Matrix 120 Epoch

Tabel 3 Hasil Evaluasi Training Data 120 Epoch

Epoch	Time (sec)	Metrics precision (B)	Metrics recall (B)	Metrics mAP50 (B)	Metrics mAP50-95 (B)
1	224.29	0.96362	0.92296	0.995	0.86753
2	445.496	0.9058	0.88609	0.90438	0.76477
3	695.599	0.87542	0.91436	0.94132	0.83025
4	929.614	0.93802	0.94286	0.9796	0.84603
5	1160.06	0.93577	0.9452	0.9915	0.87366
…	…	…	…	…	…
116	55368.8	0.94953	0.9417	0.9915	0.9073
117	56461.6	0.94227	0.94477	0.9915	0.90312
118	56732.7	0.94123	0.94332	0.9915	0.90608
119	56997.1	0.93437	0.93772	0.9915	0.90623
120	57298.6	0.93945	0.93946	0.9915	0.90322

Tabel 4 Classification Pose Data 120 Epoch

Label	Precision	Recall	F1-Score	Support
Gerakan Mencontek	0.8750	0.8750	0.8750	8
Gerakan Menunduk	0.8571	1.0000	0.9231	6
Menoleh Kanan	1.0000	0.8571	0.9231	7
Menoleh Kiri	0.8571	0.8571	0.8571	7
Sikap Kondusif	0.8571	0.8571	0.8571	7
Total Accuracy		**88.5714%**		

Hal ini turut diperkuat oleh data pada Tabel 3 yang memperlihatkan tingkat akurasi klasifikasi yang tinggi untuk sebagian besar kategori gerakan. Gerakan mencontek dan menunduk memiliki jumlah klasifikasi benar tertinggi, sementara misklasifikasi masih terjadi pada gerakan menoleh kiri yang sering tertukar dengan menoleh kanan, kemungkinan karena kemiripan pose atau keterbatasan data. Selain itu, sudah mulai terlihat beberapa perbedaan dari 60 *epoch* yang ditandai dengan mulai meningkatnya *precision*, mAP50, dan mAP50-95. Di samping peningkatan tersebut, durasi pelatihan data juga memerlukan waktu yang cukup lama.

Klasifikasi pose pada Tabel 4 menampilkan nilai *precision*, *recall*, dan *F1-score* yang cukup tinggi, di mana gerakan menunduk mencapai *recall* sempurna sebesar 1.00. Beberapa label lain mengalami sedikit penurunan *recall*, mengindikasikan masih adanya deteksi yang terlewat. Secara keseluruhan, model menunjukkan performa baik dengan akurasi deteksi mencapai 88,6%.

3.3 Perbandingan Skema Pelatihan

Penelitian ini melakukan pengujian skema dengan berdasarkan split *epoch* untuk melatih dataset estimasi gerakan kecurangan UTBK. Skema split pelatihan pertama menggunakan 60 *epoch*, sementara skema split pelatihan kedua menggunakan 120 *epoch*. Perbandingan ini dimaksudkan untuk mengevaluasi sejauh mana jumlah *epoch* memengaruhi performa model dalam mendeteksi gerakan secara akurat. Tabel 5 menyajikan perbandingan hasil dari kedua skema tersebut, yang menunjukkan perbedaan performa model berdasarkan skema split pelatihan yang digunakan.

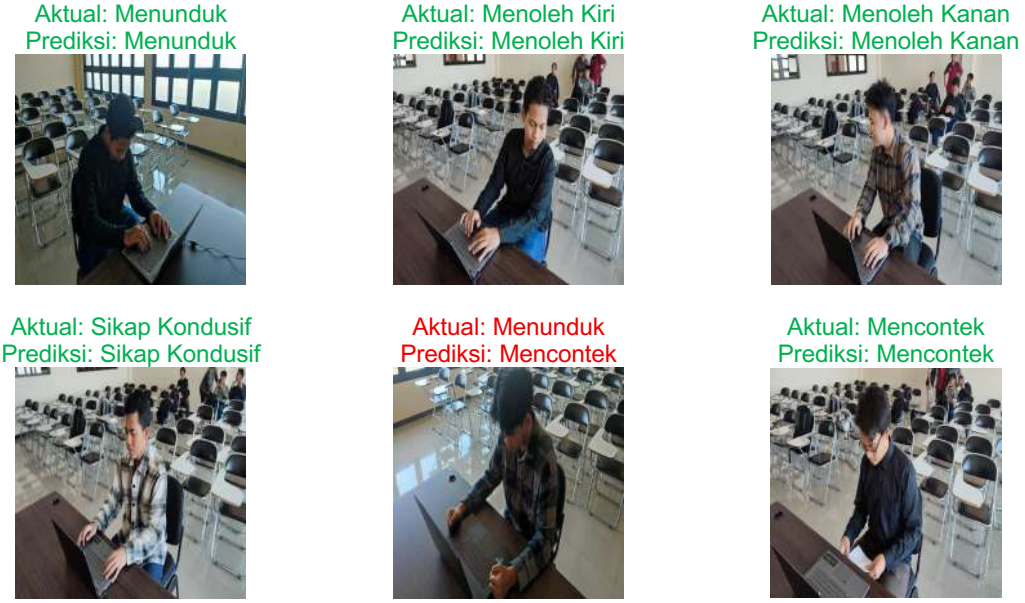
Perolehan pada Tabel 5 menunjukkan skema pelatihan 60 *epoch* memiliki tingkat keakuratan sebesar 94.3%. Sedangkan, skema pelatihan 120 *epoch* hanya memiliki tingkat keakuratan sebesar 88.6%. Apabila melihat hasil skema paling unggul dalam hal akurasi, skema pelatihan 60 *epoch* yang terpilih sebagai metode pelatihan data terbaik penelitian ini.

Tabel 5 Perbandingan Pelatihan Berdasarkan Epoch

Skema Split (Epoch)	Akurasi (%)
60	94.2857
120	88.5714

3.4 Kinerja Hasil Deteksi 60 Epoch

Perolehan hasil deteksi diperoleh dari model yang telah dilatih selama 60 *epoch*. Sebagai ilustrasi kemampuan klasifikasi gerakan tersebut, ditampilkan beberapa contoh citra yang memperlihatkan hasil prediksi model. Setiap citra dilengkapi dengan label ground truth serta prediksi yang dihasilkan, seperti yang ditunjukkan pada Gambar 5.



Gambar 5 Visualisasi Hasil Deteksi Aktual dan Prediksi

Hasil deteksi pada Gambar 5 menampilkan representasi visual dari hasil klasifikasi gerakan oleh model, di mana setiap citra disertai label aktual dan prediksi. Sistem ini menerapkan pewarnaan

kode untuk membedakan hasil klasifikasi: warna hijau menunjukkan kecocokan antara label aktual dan prediksi (klasifikasi benar), sementara warna merah menunjukkan kesalahan klasifikasi. Dari enam citra yang ditampilkan, lima berhasil diklasifikasikan dengan tepat dan satu mengalami misklasifikasi. Sehingga, hal tersebut memberikan wawasan dalam analisis kesalahan serta mengidentifikasi karakteristik citra yang menyulitkan bagi model.

3.5 Hasil Sistem Deteksi

Penggunaan metode YOLOv8 dan aliran optik pada sistem deteksi ini dipandang sudah sesuai dengan tujuan untuk mendeteksi gerakan mencurigakan yang bisa menimbulkan kecurangan pada UTBK. Pernyataan ini didukung dengan penelitian sebelumnya yang menggunakan kedua metode tersebut untuk suatu sistem deteksi. Meskipun, kurang banyaknya penelitian secara langsung yang menggabungkan YOLOv8 dan juga Optical Flow, tetapi kedua metode bisa memberikan kegunaan yang saling menyempurnakan suatu sistem deteksi.

Keselarasan kedua metode ini telah dibuktikan oleh penelitian Chavan et al. (2024), yang mengevaluasi kombinasi YOLO dan Optical Flow Farneback untuk deteksi objek bergerak dan melaporkan skor F1 sebesar 90,50% dengan waktu rata-rata 0,025 detik per bingkai. Selain itu, penerapan ROI berbasis kotak deteksi YOLO mampu menekan positif palsu dan meningkatkan efisiensi komputasi, sehingga metode gabungan ini lebih adaptif dan akurat dibandingkan pemindaian bingkai secara keseluruhan. Lebih lanjut, penerapan ROI berbasis kotak deteksi YOLO mampu menekan tingkat positif palsu serta meningkatkan efisiensi komputasi. Fokus wilayah relevan ini membuat metode gabungan dinilai lebih adaptif dan akurat dibandingkan pendekatan yang hanya memindai keseluruhan bingkai secara independen. Hasil eksperimen juga menunjukkan bahwa metode kombinasi ini bisa mendeteksi dengan presisi tinggi.

Berdasarkan hasil perbandingan antar-epoch, diperoleh perbedaan tingkat akurasi yang signifikan. Peningkatan jumlah *epoch* tidak selalu berbanding lurus dengan kenaikan akurasi atau metrik evaluasi. Kondisi ini disebabkan oleh terjadinya *overfitting*, di mana model terlalu menyesuaikan diri dengan data latih sehingga performa pada data validasi maupun data uji menurun. Pada *epoch* ke-60, model diduga telah mencapai titik optimal dalam menyeimbangkan kesesuaian terhadap data latih dan kemampuan generalisasi terhadap data validasi.

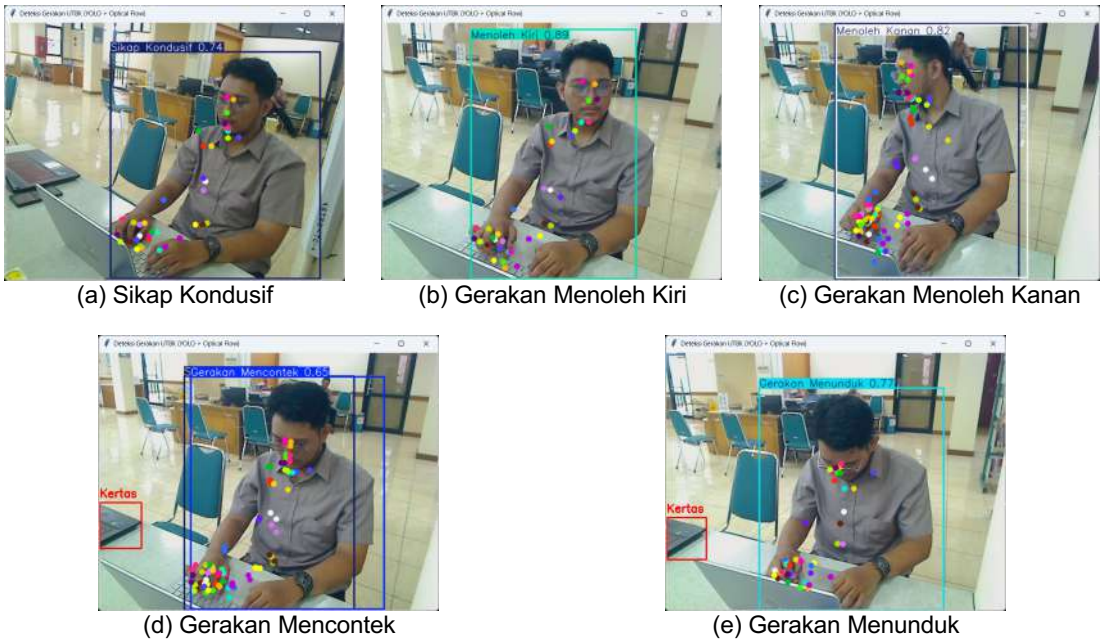
Pengintegrasian algoritma YOLOv8 dan Optical Flow dalam sistem deteksi menunjukkan kinerja yang baik dalam mengidentifikasi gerakan mencurigakan selama pelaksanaan UTBK. Model YOLOv8 memberikan kontribusi signifikan dalam mengenali pose gerakan peserta, dengan tingkat akurasi 94,3% dan *confidence* 0,75 yang dinilai cukup tinggi. Kemampuannya dalam mendeteksi pose secara real-time menjadikannya efektif dalam pengamatan gerakan.

Dalam model ini, algoritma Optical Flow berperan dalam menjaga kontinuitas interpretasi gerakan, khususnya ketika *keypoint* dari YOLOv8 bisa saja mengalami ketidakstabilan antar frame akibat gangguan pencahayaan atau gerakan yang terlalu cepat. Optical Flow membantu mengatasi kekosongan data yang muncul akibat hilangnya deteksi *keypoint*, sehingga menghasilkan analisis gerakan yang lebih konsisten. Hasil integrasi kedua metode ini dapat ditinjau pada Gambar 6.

Hasil deteksi gerakan pada Gambar 6 memperlihatkan beberapa hasil keluaran sistem deteksi yang menampilkan jenis gerakan beserta tingkat confidence-nya. Pada Gambar opsi (a), sistem mendeteksi gerakan kondusif dengan tingkat confidence sebesar 0,74, yang dianggap sebagai gerakan normal (default) dan tidak dicatat sebagai perilaku mencurigakan. Adapun gerakan yang diklasifikasikan sebagai indikasi kecurangan mencakup aktivitas mencontek, menoleh ke kanan, menoleh ke kiri, dan menunduk. Gerakan-gerakan tersebut secara otomatis direkam dan disimpan dalam log aktivitas.

Secara keseluruhan, integrasi antara algoritma YOLOv8 dan Optical Flow mampu menghasilkan performa deteksi yang baik dalam mengidentifikasi pose gerakan mencurigakan. Metode ini menggabungkan deteksi visual statis dan analisis gerakan dinamis, yang secara eksperimental

terbukti meningkatkan akurasi dibandingkan penggunaan YOLO atau Optical Flow secara terpisah. Pada Gambar 6, ditunjukkan *keypoint* berwarna yang merepresentasikan hasil Optical Flow dalam menjaga stabilitas deteksi pada area tertentu, sehingga dapat mendukung YOLOv8 dalam mengurangi ketidakstabilan antar-frame.



(a) Sikap Kondusif (b) Gerakan Menoleh Kiri (c) Gerakan Menoleh Kanan

(d) Gerakan Mencontek (e) Gerakan Menunduk

Gambar 6 Hasil Deteksi Gerakan

3.6 Sistem Logging dan Basis Data SQLite

Setiap kali sistem mendeteksi aktivitas mencurigakan, seperti gerakan tidak wajar. Informasi deteksi gerakan tersebut secara otomatis tercatat ke dalam basis data lokal SQLite. Pencatatan dilakukan melalui fungsi log_gerakan(), yang dipanggil setelah proses penyimpanan gambar dengan cv2.imwrite(). Fungsi ini merekam waktu kejadian, identitas peserta (secara default tercatat sebagai "Peserta"), jenis pelanggaran yang terdeteksi, serta jalur file dari gambar tangkapan (snapshot) yang diambil. Adapun contoh isi log dari folder snapshots terdapat pada Gambar 7.

(a) Bukti Gerakan Menoleh Kiri (b) Bukti Gerakan Mencontek (c) Bukti Gerakan Menoleh Kanan (d) Bukti Gerakan Menoleh Kiri

Gambar 7 Isi Log Folder Snapshots

Data hasil logging dapat diakses melalui antarmuka pengguna grafis (GUI) dan diekspor dalam format CSV untuk keperluan evaluasi dan dokumentasi. Contoh hasil pencatatan dapat dilihat pada Gambar 7, yang menunjukkan struktur data dalam file log_gerakan.db sebagai arsip digital pelanggaran yang tersimpan secara sistematis. Gambar dan hasil pelanggaran dapat ditinjau

kembali dengan melakukan *double-click* pada pelanggaran yang ingin ditinjau kembali. Dengan begitu pemantauan terhadap deteksi gerakan saat UTBK mudah terawasi karena adanya bukti log snapshots.

3.7 Pencatatan Otomatis dan Visualisasi

Pencatatan juga menjadi poin tambahan dalam memonitoring para peserta terhadap segala kecurigaan aktivitas saat UTBK. Selain mencatat data ke dalam basis data, sistem ini juga dilengkapi dengan fitur visualisasi interaktif melalui dashboard GUI yang dikembangkan menggunakan pustaka Tkinter. Salah satu fitur utama adalah tampilan tabel log gerakan yang memungkinkan pengguna mengakses bukti visual secara langsung. Cukup dengan melakukan *double-click* pada entri tertentu di kolom gambar, sistem akan secara otomatis membuka file citra yang terkait dengan kejadian tersebut.

Tampilan dashboard GUI yang menunjukkan informasi hasil log gerakan bisa dilihat pada Tabel 6. Fitur ini sangat membantu dalam proses verifikasi visual oleh pengawas atau administrator ujian, karena memungkinkan pemeriksaan langsung terhadap kondisi peserta pada saat pelanggaran terjadi tanpa perlu mencari file secara manual di direktori penyimpanan. Dengan demikian, proses identifikasi dan evaluasi pelanggaran menjadi lebih efisien, cepat, dan akurat. Apabila ingin melihat Ilustrasi antarmuka kita bisa melakukan *double-click* , misal pada kolom ID nomor 118 Tabel 6, sistem akan secara otomatis membuka file citra opsi (c) yang ada pada Gambar 7.

Tabel 6 Dashboard GUI Gerakan UTBK

ID	Waktu	Peserta	Jenis Gerakan	Gambar
117	2025-06-19 12:53:15	Peserta	Gerakan Menoleh Kiri	snapshots\GerakanMenolehKiri_20250619_125308_580150
116	2025-06-19 12:53:21	Peserta	Gerakan Menoleh Kiri	snapshots\GerakanMenolehKiri_20250619_125309_242193
117	2025-06-19 12:53:25	Peserta	Gerakan Menoleh Kiri	snapshots\GerakanMenolehKiri_20250619_125310_615292
118	2025-06-19 12:53:30	Peserta	Gerakan Menoleh Kiri	snapshots\GerakanMenolehKiri_20250619_125311_717058
…	…	…	…	…
137	2025-06-19 12:54:35	Peserta	Gerakan Menoleh Kanan	snapshots\GerakanMenolehKanan_20250619_125330_539255

4. KESIMPULAN

Penelitian ini berhasil merancang sistem deteksi gerakan mencurigakan yang berpotensi menjadi indikasi kecurangan saat UTBK berbasis kamera, dengan mengintegrasikan YOLOv8 untuk estimasi pose gerakan dan algoritma Optical Flow untuk analisis arah pergerakan *keypoint* secara real-time. Sistem ini dikembangkan melalui percobaan dengan perbandingan jumlah *epoch* 60 dan 120 untuk mengevaluasi performa model, serta dirancang untuk meningkatkan efektivitas pengawasan ujian berbasis komputer yang sebelumnya masih bergantung pada pemantauan manual. Berdasarkan hasil pengujian, pelatihan dengan 60 *epoch* terbukti mampu mendeteksi gerakan mencurigakan dengan akurasi mencapai 94,3%. Setiap pelanggaran yang teridentifikasi akan tercatat secara otomatis dalam basis data lokal (SQLite), dilengkapi dengan waktu kejadian, identitas peserta, jenis gerakan, dan bukti visual. Secara keseluruhan, integrasi antara deteksi pose dan analisis gerakan menghasilkan sistem pemantauan yang andal dan efisien untuk mendukung pelaksanaan ujian secara digital.

Hasil yang telah dicapai menunjukkan kinerja sistem yang andal, namun pengembangan di masa mendatang perlu diarahkan pada peningkatan kapabilitas dalam mendeteksi pose gerakan mencurigakan, mengingat performa saat ini masih memiliki ruang untuk perbaikan. Distribusi data validasi dan pengujian yang relatif terbatas juga perlu diperluas guna meningkatkan kemampuan

generalisasi model terhadap berbagai skenario. Selain itu, optimalisasi metode Optical Flow dengan memperkuat integrasinya bersama hasil deteksi pose dari YOLOv8 diharapkan dapat menghasilkan analisis gerakan yang lebih detail dan berkesinambungan. Dengan pengembangan tersebut, sistem diproyeksikan mampu memberikan deteksi yang lebih akurat, adaptif, dan responsif terhadap berbagai bentuk pelanggaran selama ujian berlangsung.

DAFTAR PUSTAKA

Bimantoro, F., Wijaya, I. G. P. S., & Aohana, M. R. (2024). Pendeteksian Kecurangan Ujian Melalui CCTV Menggunakan Algoritma YOLOv5. *Seminar Nasional Teknologi & Sains*, *3*(1), 109–117. https://doi.org/10.29407/stains.v3i1.4360

Chavan, R., Gulge, A., & Bhandare, S. (2024). Moving Object Detection and Classification Using Deep Learning Techniques. *Tuijin Jishu/Journal of Propulsion Technology*, *45*(1), 1001–4055. https://www.propulsiontechjournal.com/index.php/journal/article/view/5000

Erdem, B., & Karabatak, M. (2025). Cheating Detection in Online Exams Using Deep Learning and Machine Learning. *Applied Sciences*, *15*(1), Article ID: 400. https://doi.org/10.3390/app15010400

Gopane, S., Kotecha, R., Obhan, J., & Pandey, R. K. (2024). Cheat Detection in Online Examinations Using Artificial Intelligence. *ASEAN Engineering Journal*, *14*(1), 121–128. https://doi.org/10.11113/aej.v14.20188

Hadibrata, L., & Rochadiani, T. H. (2024). Deteksi Potensi Menyontek Menggunakan Feedforward Neural Network pada Ujian Daring. *SINTECH (Science and Information Technology) Journal*, *7*(2), 92–100. https://doi.org/10.31598/sintechjournal.v7i2.1585

Hidayatullah, D., Ardiansah, T., & Styawati, S. (2022). Sistem Informasi Reservasi Pelayanan dan Penyewaan Fasilitas Lapangan Futsal Berbasis Web dengan Metode Waterfall. *Jurnal Teknologi dan Sistem Informasi (JTSI)*, *3*(3), 64–68. https://doi.org/10.33365/jtsi.v3i3.1994

Iskandar, A. P., Thohir, M. I., Kharisma, I. L., Kamdan, & Fergina, A. (2024). Implementasi Deteksi Langsung pada Sistem Ujian Online Menggunakan Algoritma Convolutional Neural Network. *Jurnal CoSciTech (Computer Science and Information Technology)*, *5*(2), 483–492. https://doi.org/10.37859/coscitech.v5i2.7270

Jannah, M. (2025). Pendekatan Penelitian Pendidikan Metode Penelitian Kualitatif, Metode Penelitian Kuantitatif dan Metode Penelitian Kombinasi. *Shofiya: Jurnal Pendidikan Islam*, *1*(1), 23–40. https://nuruledukasi.com/index.php/shofiya

Karim, I. N., Kadir, D. A., & Ali, F. (2020). Development of Detection System on Suspicious Behaviour During Exam. *Journal of Computing Technologies and Creative Content*, *5*(2), 75–81. https://www.jtec.org.my/index.php/JTeC/article/view/434

Nur, T., Huzaeni, H., & Khadafi, M. (2023). Implementasi Metode Object Detection dengan Algoritma You Only Look Once (YOLO) untuk Deteksi Kecurangan di dalam Ruang Ujian. *Jurnal Teknologi Rekayasa Informasi dan Komputer*, *6*(1), 28–33. https://doi.org/10.30811/jtrik.v6i1.4699

Pangestu, M. P., Wiyono, S., & Af'idah, D. I. (2024). Platform Ujian Online Berbasis Pendeteksi Gerakan Kecurangan Menggunakan Kamera. *Infomatek*, *26*(1), 55–62. https://doi.org/10.23969/infomatek.v26i1.11208

Prabowo, T. T. (2021). Efektivitas Sistem Temu Kembali Informasi Perpustakaan Digital Institut Seni Indonesia (ISI) Yogyakarta dalam Tinjauan Recall dan Precision. *Media Pustakawan*, *28*(1), 37–48. https://doi.org/10.37014/medpus.v28i1.1087

Pratama, K. A., Subagio, R. T., Hatta, M., & Asih, V. (2021). Implementasi Laod Balancing pada Web Server Menggunakan Apache dengan Server Mirror Data Secara Real Time. *Jurnal Digit*, *11*(2), 178. https://doi.org/10.51920/jd.v11i2.203

Saepudin, S., Sujana, N., Mutoffar, M. M., & Haryanto, A. A. (2024). Analisis Kinerja YOLOv8 Optimalisasi Roboflow untuk Deteksi Ekspresi Wajah Emosional dengan Machine Learning. *Naratif: Jurnal Nasional Riset, Aplikasi dan Teknik Informatika*, *6*(2), 115–124. https://doi.org/10.53580/naratif.v6i2.292

Sari, A. P., Haromainy, M. M. Al, & Purnomo, R. (2024). Implementasi Metode Rapid Application Development pada Aplikasi Sistem Informasi Monitoring Santri Berbasis Website. *Decode: Jurnal Pendidikan Teknologi Informasi*, *4*(1), 316–325. https://doi.org/10.51454/decode.v4i1.348

Simarmata, J. E., Laja, Y. P. W., Salsinha, C. N., Kehi, Y. J., Laki, A. G., Gomes, M. R., Asa, J. M. P., Bano, E. N., Muanley, Y. Y., & Meti, H. Y. (2022). Pelatihan Tes Kemampuan Akademik Bagi Siswa SMA Kelas XII untuk Persiapan UTBK SBMPTN 2022. *Jurnal Abdi Insani*, *9*(2), 471–479. https://doi.org/10.29303/abdiinsani.v9i2.557

Syaharuddin, A. Z., Endang, A. H., & Alfarizi, D. A. (2023). Rancang Bangun Sistem Monitoring Aktivitas Pelanggan Berbasis Citra Menggunakan Optical Flow. *Journal of Embedded Systems, Security and Intelligent Systems*, *4*(2), 126–131. https://doi.org/10.59562/jessi.v4i2.699

Thohari, A. N. A., Lathief, M. F., Triyono, L., & Santoso, K. (2025). Deteksi Kecurangan Ujian pada Ruangan Tertutup Menggunakan Algoritma YOLOv8. *Journal of Computer Science and Informatics Engineering*, *4*(2), 61–71. https://doi.org/10.55537/cosie.v4i2.1100

Tjahyanti, L. P. A. S., & Gitakarma, M. S. (2024). Sistem Pendeteksi Kecurangan Ujian Menggunakan YOLO: Identifikasi Penggunaan Ponsel dan Interaksi Mencurigakan. *KOMTEKS*, *3*(2), 25–32. https://doi.org/10.37637/komteks.v3i2.2293

Wicaksono, F. B., & Yamasari, Y. (2025). Pengembangan Model Pengawas Ujian Berbasis Kecerdasaan Buatan untuk Ujian Online. *Journal of Informatics and Computer Science (JINACS)*, *6*(03), 882–890. https://doi.org/10.26740/jinacs.v6n03.p882-890

Wulandari, Y. T., & Rosandy, T. (2024). Implementasi Computer Vision dalam Sistem Deteksi Gerakan Disiplin Kampus. *Jurnal Teknika*, *18*(2), 343–354. https://doi.org/https://doi.org/10.5281/zenodo.11139336

Zuo, Y., Chai, S. S., & Goh, K. L. (2024). Cheating Detection in Examinations Using Improved YOLOv8 with Attention Mechanism. *Journal of Computer Science*, *20*(12), 1668–1680. https://doi.org/10.3844/jcssp.2024.1668.1680

JISKA (Jurnal Informatika Sunan Kalijaga)
Vol. 11, No. 1, JANUARY, 2026, Pp. 98 – 113
ISSN: 2527 – 5836 (print) | 2528 – 0074 (online)

Comparative Analysis of Camellia and AES Across File Sizes and Types

Teja Endra Eng Tju [(1)*], Fauzi Alfadhillah [(2)]
Department of Information Systems, Budi Luhur University, Jakarta, Indonesia
e-mail : teja.endraengtju@budiluhur.ac.id, 2311501767@student.budiluhur.ac.id.
* Corresponding author.
This article was submitted on 30 July 2025, revised on 15 December 2025, accepted on 16 December 2025, and published on 25 January 2026.

Abstract

Data security is a critical aspect of modern information systems that requires processing cryptographic efficiency and resilience. This study compares two widely used symmetric encryption algorithms, named Camellia and AES, based on their performance and resistance to standard attack methods. An experimental approach was applied using 72 files across eight commonly used formats (.mp3, *.jpg, *.png, *.pdf, *.docx, *.xls, *.pptx, and .txt) in three predefined sizes: 100 KB, 1 MB, and 10 MB. Each file underwent encryption and decryption in a controlled environment, with metrics such as processing time, CPU usage, and RAM consumption recorded. Simulated Dictionary, Birthday, and Brute-Force attacks were conducted to assess algorithm robustness. Results show that AES performs faster, especially on large files, but with higher memory usage. Camellia demonstrated more consistent RAM usage and stronger resistance, successfully withstanding all attacks except one brute-force case on a small plaintext file. AES suffered multiple breaches on structured files of smaller sizes. The findings suggest that algorithm selection should consider workload characteristics and system constraints. The main contribution of this research lies in its comprehensive dataset and empirical comparison, providing practical insights to support encryption algorithm choices in real-world applications.*

Keywords: Cryptographic Efficiency, Data Encryption, Cryptanalysis, Algorithm Security, Encryption Performance

Abstrak

Keamanan data merupakan aspek krusial dalam sistem informasi modern yang menuntut efisiensi pemrosesan sekaligus ketahanan kriptografis. Penelitian ini membandingkan dua algoritma enkripsi simetris yang banyak digunakan, yaitu Camellia dan AES, berdasarkan kinerja dan ketahanannya terhadap metode serangan umum. Pendekatan eksperimental diterapkan pada 72 file dalam delapan format umum (*.mp3, *.jpg, *.png, *.pdf, *.docx, *.xls, *.pptx, dan .txt) dalam tiga ukuran: 100 KB, 1 MB, dan 10 MB. Setiap file dienkripsi dan didekripsi dalam lingkungan terkontrol, dengan pencatatan waktu proses, penggunaan CPU, dan konsumsi RAM. Serangan Dictionary, Birthday, dan Brute-Force disimulasikan untuk menilai ketahanan masing-masing algoritma. Hasil menunjukkan bahwa AES lebih cepat, terutama pada file berukuran besar, namun memerlukan memori lebih tinggi. Camellia menunjukkan penggunaan RAM yang lebih stabil dan ketahanan lebih kuat, berhasil menahan semua serangan kecuali satu kasus brute-force pada file teks kecil. AES mengalami beberapa kebocoran pada file terstruktur dengan ukuran kecil. Temuan ini menunjukkan bahwa pemilihan algoritma harus mempertimbangkan karakteristik beban kerja dan batasan sistem. Kontribusi utama dari penelitian ini adalah penyediaan dataset komprehensif dan perbandingan empiris yang memberikan wawasan praktis untuk mendukung pemilihan algoritma enkripsi dalam berbagai skenario aplikasi.

Kata Kunci: Efisiensi Kriptografi, Enkripsi Data, Kriptanalisis, Keamanan Algoritma, Kinerja Enkripsi

1. INTRODUCTION

Data security has become a primary concern as information technology continues to evolve and connects various global systems. The use of cryptographic algorithms to protect sensitive data is the main way to maintain the confidentiality and integrity of information (Manullang, 2023).

Cryptographic algorithms such as Camellia and AES (Advanced Encryption Standard) are widely used to protect sensitive data by ensuring confidentiality and integrity (Sulaiman & Hammood, 2025). Camellia, developed by NTT and Mitsubishi Electric, provides a flexible and efficient structure applicable across diverse platforms (Matsui et al., 2015), while AES, standardized by NIST, is known for its simplicity, strong security, and widespread adoption (Azhari et al., 2022). This research specifically focuses on comparing the performance of Camellia and AES in securing small-to-medium-sized files on specific devices. The study evaluates encryption and decryption efficiency in terms of run time, resource consumption, and robustness against common cyberattacks, such as Dictionary Attack (Adams, 2021; Alkhwaja et al., 2023; Hranický et al., 2025), Birthday Attack (Chavan et al., 2024; Fahdi & Ahmed, 2020), and Brute-force Attack (Hamza & Al-Janabi, 2024; Sarmila & Manisekaran, 2022; Verma et al., 2022). The choice of these two algorithms is justified by their prevalence in modern cryptographic applications and contrasting design philosophies, offering insight into optimal algorithm selection for practical data protection. In practical deployments, encryption is rarely evaluated only by theoretical strength; it must also meet operational constraints such as throughput, memory budget, and predictable behavior across heterogeneous data formats. A system that encrypts files quickly but exhibits higher memory demand or inconsistent robustness under common password-guessing and collision-based attack scenarios may create hidden security and performance risks in real applications. Therefore, evaluating efficiency and robustness together across diverse file structures is essential for making defensible design choices rather than relying on one-size-fits-all assumptions.

Previous research on Camellia and AES cryptographic algorithms has extensively covered their security and resilience against cryptographic attacks. AES has become the international standard due to its strong encryption capabilities and broad acceptance in various security applications (Bibiola et al., 2023). Camellia offers comparable security levels with a more flexible and modular structure, facilitating implementation across different platforms and scenarios (Li et al., 2024). Applications of Camellia encryption have been explored in areas such as authentication for wireless robotics and mobile platforms (Nithishma et al., 2022; Rasheed et al., 2023), quantum implementations with S-box optimizations (ZhenQiang et al., 2023), and security models for e-banking and Internet of Things (IoT) devices (Mohammed et al., 2025; Rashidi, 2021). Meanwhile, AES has been applied in securing digital images (Haris et al., 2023), medical records (Tanjung, 2022), financial transactions (Andriyanto & Sukmasetya, 2022), cloud computing (Dwina et al., 2023), and IoT healthcare networks (Rachmayanti & Wirawan, 2022). Comparative studies between Camellia and AES include quantum circuit designs (Wei et al., 2023), digital image encryption evaluations (Hidayati et al., 2021), and security implementations in 5G networks and IoT frameworks (Arabul et al., 2023; Muthavhine & Sumbwanyambe, 2023; Ning et al., 2020; Panahi & Bayılmış, 2023; Rasheed et al., 2024). Although prior studies have compared Camellia and AES in specific contexts (e.g., images, IoT frameworks, or specialized implementations), many evaluations are scoped to a narrow data type, focus on a single dimension (performance *or* security), or do not jointly examine how file structure and size interact with both resource usage and attack outcomes. This study addresses that gap by conducting a controlled and reproducible experiment across multiple real-world file formats and standardized size categories, reporting both system-level efficiency metrics and empirical robustness under common attack simulations. The resulting workflow is described in the Methods section. This study investigates efficiency differences between Camellia and AES (runtime, CPU usage, and RAM consumption) across file types and sizes, examines how file structure and size relate to empirical outcomes in dictionary, birthday, and brute-force attack simulations, and derives actionable guidelines for algorithm selection under different operational constraints.

The contributions of this paper are threefold: it provides a controlled empirical dataset and test protocol covering 72 files across eight standard formats and three standardized size categories to enable consistent comparison; it conducts a joint evaluation of efficiency (run time, CPU usage, and RAM consumption) and empirical robustness under dictionary, birthday, and brute-force attack simulations within a single experimental pipeline; and it derives evidence-based insights into how file type or structure and size shape performance–security trade-offs, translating these

findings into practical recommendations for encryption algorithm selection in real-world systems implementation.

2. METHODS

This study employs an experimental approach to evaluate the performance of the Camellia and AES algorithms based on the stages illustrated in Figure 1. The process involves collecting data or files comprising various file types, followed by the encryption and decryption of these files using Camellia and AES. Next, Attack Testing on Encrypted Files is performed to assess resistance against different cryptographic attacks. Then, the Performance Analysis of both algorithms is conducted to compare their efficiency and robustness. The study concludes with an Evaluation and Conclusion stage that provides insights into the strengths and limitations of each algorithm in the context of data protection.

The stage begins with the Data or Files Collection, which involves 72 files across eight widely used formats: *.mp3, *.jpg, *.png, *.pdf, *.docx, *.xls, *.pptx, and *.txt. This diverse selection aims to assess how the Camellia and AES algorithms handle data with varying structures in sizes and types. Files were randomly collected from various sources. For each file type, three predefined sizes were prepared: 100 KB, 1 MB, and 10 MB, including audio (.mp3), image (.jpg, .png), document (.pdf, .docx, .xls, .pptx), and plaintext (.txt).

Figure 1 Research Method Stages

Table 1 presents the distribution of file types and predefined sizes (100 KB, 1 MB, and 10 MB), which will be utilized in the encryption and decryption experiments to evaluate the efficiency and robustness of Camellia and AES across various data types and file structures. This structured dataset ensures a balanced and systematic representation of file categories and size variations, providing a consistent basis for fair comparison and reliable performance measurement across all test cases.

After data collection, the study proceeds to the File Encryption and Decryption stage using Camellia and AES. All tests were conducted on a laptop with an AMD Ryzen 7 5800H processor, 16GB RAM, and an NVIDIA GeForce RTX 3050 Ti GPU, running Windows 11. The files used in the experiment vary in format and size, each encrypted and decrypted using a 256-bit key, selected to comply with modern cryptographic standards against various cryptanalysis attacks (Housley & Schaad, 2020; Kato & Moriai, 2013). A strong password was used to derive each

encryption key (Patra & Patra, 2021). The encryption and decryption processes are carried out using a simple web-based application developed in-house with the PyCryptodome library (A et al., 2022; G et al., 2022), which records run time, CPU Usage, and RAM Usage. These procedures allow observation of how each algorithm handles different file types and workloads, highlighting differences in processing efficiency and consistency. Figure 2 presents the encryption flow for both algorithms.

Table 1 Distribution Files

Types	Count by Predefined Sizes			Total
	100 KB	1 MB	10 MB	
*.mp3	3	3	3	9
*.jpg	3	3	3	9
*.png	3	3	3	9
*.pdf	3	3	3	9
*.docx	3	3	3	9
*.xls	3	3	3	9
*.pptx	3	3	3	9
*.txt	3	3	3	9
Total Data				**72**

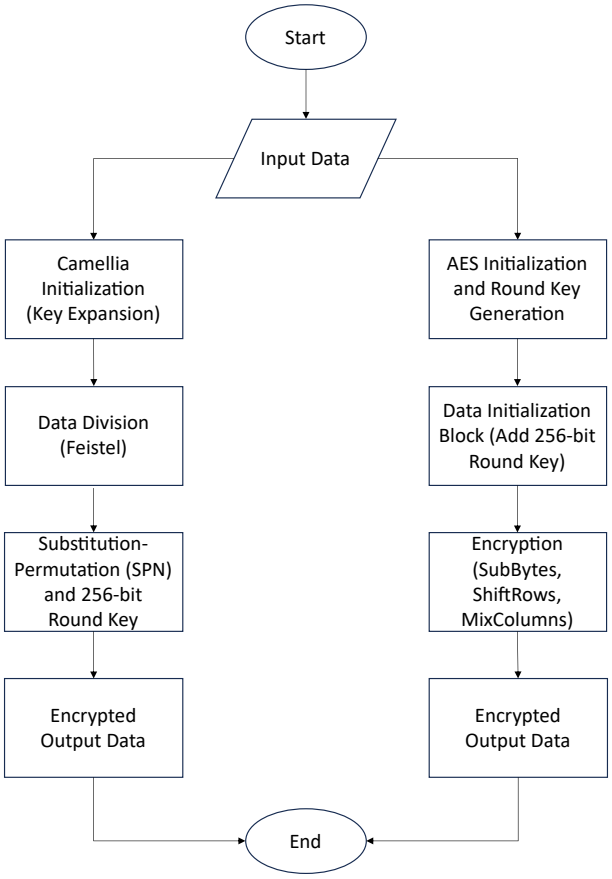


Figure 2 Encryption Process Flow of Camellia and AES Algorithms

The next stage is Attack Testing on Encrypted Files to assess each algorithm's robustness against common cryptographic threats. This study simulated Dictionary, Birthday, and Brute-Force attacks and recorded the attack time and outcome for each test case. These simulations identify potential weaknesses and determine how each algorithm can withstand unauthorized decryption

attempts. Specifically, the study examines whether specific patterns in the encryption process can be exploited to accelerate decryption without the proper key.

The fourth stage is Performance Analysis of Camellia and AES from both efficiency and robustness perspectives. Efficiency was measured using run time, CPU usage, and RAM consumption during encryption and decryption. Robustness was evaluated based on whether encrypted outputs could be successfully compromised under the three simulated attack settings. The results provide a comprehensive view of each algorithm's capabilities in handling diverse data efficiently and securely under varying conditions.

The final stage is Evaluation and Conclusion based on all obtained results. In this stage, the study will summarize the strengths and weaknesses of each algorithm in terms of efficiency and security. The evaluation results will identify the most suitable algorithm for data protection purposes. These conclusions are expected to provide a solid foundation for selecting a cryptographic algorithm that aligns with specific security requirements.

3. RESULTS AND DISCUSSION

The data collection results comprise 72 files across eight file types, with three predefined size categories. Table 2 presents three sample files for each predefined size category, labeled File 1, File 2, and File 3, with actual sizes varying within a ±5% tolerance range. The use of three files per size category serves multiple purposes. It ensures consistent and reliable results within each size group, captures minor variations in file structure and content that may affect processing, and helps mitigate potential bias from relying on a single file.

Table 2 Files Distribution and Variation

Types	Predefined Sizes	Actual Sizes		
		File 1	File 2	File 3
*.mp3	100 KB	102 KB	103 KB	102 KB
	1 MB	1.00 MB	1.04 MB	1.00 MB
	10 MB	10.03 MB	10.03 MB	10.04 MB
*.jpg	100 KB	105 KB	100 KB	100 KB
	1 MB	1.05 MB	1.05 MB	1.00 MB
	10 MB	10.03 MB	10.01 MB	10.02 MB
*.png	100 KB	101 KB	101 KB	101 KB
	1 MB	1.01 MB	1.00 MB	1.01 MB
	10 MB	10.02 MB	10.02 MB	10.04 MB
*.pdf	100 KB	100 KB	100 KB	101 KB
	1 MB	1.04 MB	1.05 MB	1.00 MB
	10 MB	10.00 MB	10.01 MB	10.01 MB
*.docx	100 KB	100 KB	104 KB	105 KB
	1 MB	1.02 MB	1.01 MB	1.01 MB
	10 MB	10.01 MB	10.05 MB	10.04 MB
*.xls	100 KB	100 KB	104 KB	100 KB
	1 MB	1.00 MB	1.01 MB	1.01 MB
	10 MB	10.04 MB	10.02 MB	10.01 MB
*.pptx	100 KB	104 KB	105 KB	100 KB
	1 MB	1.00 MB	1.05 MB	1.05 MB
	10 MB	10.04 MB	10.04 MB	10.04 MB
*.txt	100 KB	101 KB	103 KB	100 KB
	1 MB	1.01 MB	1.00 MB	1.01 MB
	10 MB	10.05 MB	10.05 MB	10.02 MB

The encryption and decryption efficiency of Camellia and AES was evaluated using Run Time, CPU usage, and RAM consumption. Run Time refers to the time taken to complete the encryption or decryption process, CPU Usage indicates the percentage of processor resources used during

execution, and RAM Consumption reflects the amount of memory utilized throughout the operation. To ensure a representative comparison, the values for each metric were averaged per file type using data from three files in every predefined size category, since the evaluation results from File 1, File 2, and File 3 showed consistent values that indicate stable performance. Measurements across formats and sizes were also relatively uniform within each group, with no significant deviations, which supports the validity of the averaged values.

The encryption results are presented in Table 3, which shows the average Run Time, CPU Usage, and RAM Consumption for each file type and predefined size. To maintain unit consistency and simplify numerical comparison, file sizes were defined as 100 KB, 1024 KB, and 10240 KB instead of mixed units with 1MB and 10MB. These metrics illustrate how each algorithm's performance responds to increasing file sizes, offering a clearer view of their efficiency and resource usage during encryption.

Table 3 Summary of Encryption Process (Averaged)

Types	Predefined Sizes	Run Time (s)		CPU Usage (%)		RAM Usage (MB)	
		Camellia	AES	Camellia	AES	Camellia	AES
*.mp3	100 KB	260.61	75.83	22.37	17.00	24.72	13.49
	1024 KB	458.64	140.65	28.49	21.22	31.76	24.33
	10240 KB	646.16	217.75	32.11	27.59	34.81	26.66
*.jpg	100 KB	137.59	77.30	18.34	21.12	21.54	16.46
	1024 KB	349.24	147.29	20.64	24.28	24.30	22.83
	10240 KB	524.70	197.32	23.62	27.75	25.57	26.77
*.png	100 KB	104.53	93.44	17.93	20.80	22.87	17.96
	1024 KB	305.27	136.36	22.13	24.59	25.43	24.85
	10240 KB	479.15	203.79	24.32	29.04	27.87	27.64
*.pdf	100 KB	152.16	192.21	23.17	25.39	21.39	25.61
	1024 KB	200.60	256.51	25.11	31.47	26.35	32.83
	10240 KB	392.14	340.52	29.29	36.25	27.99	35.03
*.docx	100 KB	91.98	94.97	22.13	20.88	20.25	23.51
	1024 KB	156.01	149.13	24.37	26.25	23.41	26.67
	10240 KB	409.58	206.79	28.81	30.49	24.68	28.47
*.xls	100 KB	96.38	107.52	22.99	22.37	21.78	25.47
	1024 KB	234.36	259.62	24.08	27.21	23.46	29.20
	10240 KB	384.74	338.61	25.61	33.40	25.59	30.26
*.pptx	100 KB	111.67	95.38	25.04	21.73	25.88	23.38
	1024 KB	185.69	156.13	28.36	24.59	28.55	27.39
	10240 KB	278.82	209.68	30.46	28.76	31.51	30.61
*.txt	100 KB	94.74	88.50	22.57	21.35	25.46	23.23
	1024 KB	199.99	157.52	26.55	24.50	28.58	27.54
	10240 KB	325.68	234.29	31.80	30.16	30.72	31.59

Based on Table 3, Figure 3 presents the graphical visualization of encryption. Each graph uses a logarithmic scale on the x-axis to represent predefined file sizes of 100 KB, 1024 KB, and 10240 KB. This allows more precise observation of efficiency trends as file size increases. The graphs illustrate the efficiency differences between the two encryption algorithms across file types and sizes. In general, larger file sizes lead to increased run time, CPU usage, and RAM usage, with algorithm efficiency varying depending on the file types. Across most formats and sizes, AES shows lower run time (higher throughput), while Camellia tends to maintain more stable memory behavior. Importantly, the magnitude of the gap is not uniform: differences become more pronounced at larger sizes and vary by file structure, indicating that efficiency conclusions depend on workload composition rather than file size alone. Similarly, the decryption metrics for Run Time, CPU Usage, and RAM Consumption were recorded and averaged across the predefined size, as shown in Table 4. Refer to Table 4, Figure 4 illustrates the decryption efficiency of Camellia and AES. Quite similar to the encryption process, larger file sizes generally result in increased run

time, CPU usage, and RAM usage, with efficiency varying depending on both the encryption algorithm and the file types.

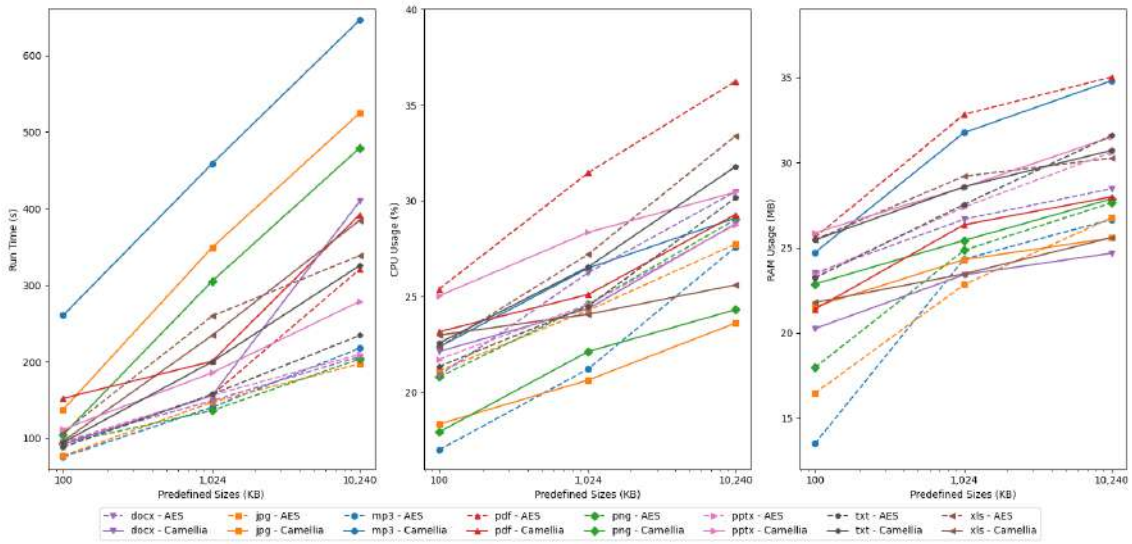


Figure 3 Visualization of Encryption Metrics

Table 4 Summary of Decryption Process (Averaged)

Types	Predefined Sizes	File 1		File 2		File 3	
		Camellia	AES	Camellia	AES	Camellia	AES
*.mp3	100 KB	282.39	78.89	24.82	23.50	29.72	21.49
	1024 KB	441.79	138.52	25.53	25.53	32.07	23.59
	10240 KB	630.44	268.60	27.50	27.51	33.57	24.48
*.jpg	100 KB	181.40	96.32	18.64	21.24	21.16	19.50
	1024 KB	240.79	154.13	21.17	23.44	24.41	23.44
	10240 KB	320.44	275.45	23.58	24.60	25.57	26.66
*.png	100 KB	121.40	73.74	28.31	21.22	23.16	23.58
	1024 KB	300.79	144.99	32.17	24.57	26.07	25.64
	10240 KB	530.44	208.12	34.59	26.29	26.90	26.64
*.pdf	100 KB	181.40	107.08	33.98	25.71	32.49	29.79
	1024 KB	340.79	325.69	35.17	29.50	35.74	32.22
	10240 KB	510.44	504.91	39.59	34.29	37.90	33.23
*.docx	100 KB	121.40	121.40	22.64	21.95	21.16	22.78
	1024 KB	260.79	219.12	24.17	25.55	23.74	24.73
	10240 KB	390.44	442.44	29.59	28.48	24.57	25.56
*.xls	100 KB	121.40	131.40	28.31	24.31	23.49	23.15
	1024 KB	230.79	277.45	30.50	28.84	24.74	24.07
	10240 KB	410.44	440.11	34.59	31.25	26.57	25.57
*.pptx	100 KB	131.40	131.40	24.98	24.31	26.16	23.49
	1024 KB	200.79	277.45	28.50	25.50	28.74	25.74
	10240 KB	330.44	440.11	30.59	29.59	31.56	27.90
*.txt	100 KB	121.40	131.40	24.64	28.64	23.16	28.49
	1024 KB	220.79	210.78	27.50	30.50	27.74	30.56
	10240 KB	410.44	330.44	31.92	32.59	30.57	32.74

Evaluating both encryption and decryption processes highlights consistent performance differences between Camellia and AES across various file types and predefined sizes (100 KB, 1024 KB, and 10240 KB). In the encryption phase, AES consistently achieved faster run times. For example, encrypting a 10 MB *.mp3 file took 217.75 seconds with AES, compared to 646.16

seconds with Camellia—nearly three times longer. Similar trends appeared in *.pdf files, where AES processed the largest size in 340.52 seconds, while Camellia required 392.14 seconds. However, this higher speed was often accompanied by increased memory usage. Encrypting a 10 MB *.pdf file consumed 35.03 MB of RAM with AES, while Camellia used only 27.99 MB. Similar patterns were found in other formats, such as *.xls and *.jpg, where AES consistently required more memory during encryption. Camellia generally showed higher processor load for CPU utilization during encryption, especially with larger files. For instance, processing a 10 MB *.png file required 24.32% CPU with Camellia and 29.04% with AES. This pattern, however, varied depending on the file format. With *.docx files, AES demonstrated slightly better CPU efficiency at 30.49%, compared to Camellia's 28.81%. In the decryption phase, AES continued to show faster processing times. Decrypting a 10 MB *.png file took 208.12 seconds with AES, while Camellia required 530.44 seconds. In terms of RAM usage, Camellia was generally more efficient. For example, decrypting a 10 MB *.xls file used 25.57 MB of memory with AES and 26.57 MB with Camellia—a relatively small difference. On the other hand, AES consumed less memory than Camellia when decrypting *.pdf files of the same size, recording 33.23 MB versus 37.90 MB. These differences suggest that file size and type influence memory usage and how each algorithm handles data internally.

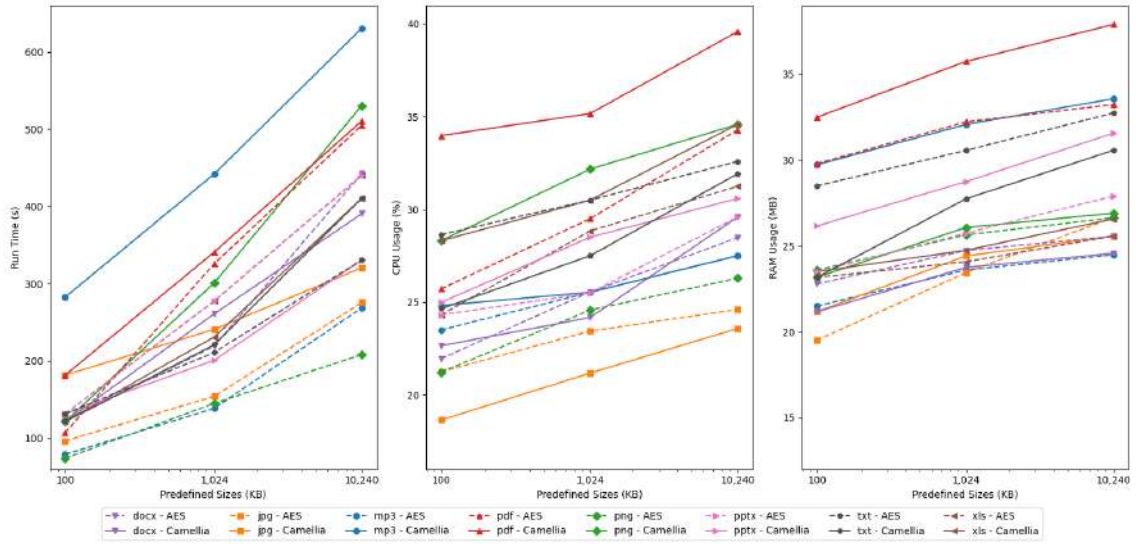


Figure 4 Visualization of Decryption Metrics

During the attack simulation phase, the outcomes varied depending on the file type, encryption algorithm, and specific method. Table 5 presents the results of the Dictionary Attack Test for both Camellia and AES. Each entry includes the Attack Time (AT) in seconds and the Attack Outcome (AO), where 'R' indicates the attack was resisted and 'B' signifies a successful breach. Shows that Camellia consistently resisted all dictionary attack attempts across various file types and sizes, with no violations observed in any scenario. In contrast, AES demonstrated vulnerabilities in specific formats, particularly in *.pdf and *.docx files at 100 KB and 1024 KB. Attack times (AT) in this table varied significantly depending on file size and type; for example, smaller files like *.pptx at 100 KB showed the shortest durations, with Camellia completing attacks in 306–326 seconds and AES in 781–801 seconds. Conversely, larger files, such as *.pdf and *.png at 10240 KB, required over 30,000 seconds for AES, whereas Camellia consistently remained under that threshold. These variations suggest that both file size and internal structure—such as compressibility or embedded metadata—play a key role in determining the efficiency of dictionary-based attacks. Figure 5 visualizes the average attack times from Table 5, offering a more precise comparison between the two algorithms. Averaging was considered valid and representative because the three attack time values for each test case were closely aligned. Moreover, the outcomes across those files were consistent, with each case resulting in complete resistance (3R) or complete breach (3B), indicating uniform behavior within each test condition. The 3B pattern,

as indicated by the '0R 3B' (zero file resisted and three files breached) outcomes, appeared exclusively in AES for *.pdf and *.docx files at 100 KB and 1024 KB, highlighting an apparent vulnerability when handling structured document formats. In contrast, Camellia maintained a consistent 3R outcome across all file types and sizes, reinforcing its resistance to dictionary attacks and efficiency across varying file characteristics.

Table 5 Dictionary Attack Results on Camellia and AES

Types	Predefined Sizes	File 1				File 2				File 3			
		Camellia		AES		Camellia		AES		Camellia		AES	
		AT (s)	AO	AT (s)	AO	AT (s)	AO	AT (s)	AO	AT (s)	AO	AT (s)	AO
*.mp3	100 KB	2657	R	1587	R	2677	R	1617	R	2637	R	1577	R
	1024 KB	32751	R	10002	R	2765	R	10042	R	2735	R	9972	R
	10240 KB	14808	R	32801	R	14828	R	33051	R	14798	R	33051	R
*.jpg	100 KB	726	R	426	R	736	R	436	R	726	R	416	R
	1024 KB	1739	R	9957	R	1759	R	9917	R	1719	R	10117	R
	10240 KB	21019	R	11624	R	21039	R	11724	R	20999	R	11574	R
*.png	100 KB	3648	R	15301	R	3668	R	15341	R	3628	R	15281	R
	1024 KB	23552	R	17269	R	23582	R	17319	R	23532	R	17219	R
	10240 KB	29844	R	26569	R	29794	R	26656	R	29914	R	26556	R
*.pdf	100 KB	3160	R	8934	B	3180	R	8954	B	3140	R	8914	B
	1024 KB	3616	R	27722	B	3636	R	27692	B	3596	R	27772	B
	10240 KB	25445	R	30757	R	25475	R	30787	R	25425	R	30767	R
*.docx	100 KB	639	R	7200	B	659	R	7230	B	619	R	7180	B
	1024 KB	13063	R	26408	B	13083	R	26438	B	13043	R	26378	B
	10240 KB	27309	R	52224	R	27329	R	52194	R	27289	R	52294	R
*.xls	100 KB	4900	R	8757	R	4920	R	8787	R	4880	R	8727	R
	1024 KB	22510	R	10099	R	22540	R	10069	R	22470	R	10139	R
	10240 KB	24259	R	11160	R	24229	R	11190	R	24309	R	11130	R
*.pptx	100 KB	306	R	791	R	326	R	801	R	306	R	781	R
	1024 KB	9553	R	11150	R	9573	R	11180	R	9533	R	11130	R
	10240 KB	27111	R	16929	R	27131	R	16959	R	27091	R	16899	R
*.txt	100 KB	6858	R	10058	R	6878	R	10088	R	6838	R	10028	R
	1024 KB	8185	R	10482	R	8195	R	10512	R	8165	R	10482	R
	10240 KB	10426	R	16492	R	10446	R	16522	R	10406	R	16472	R

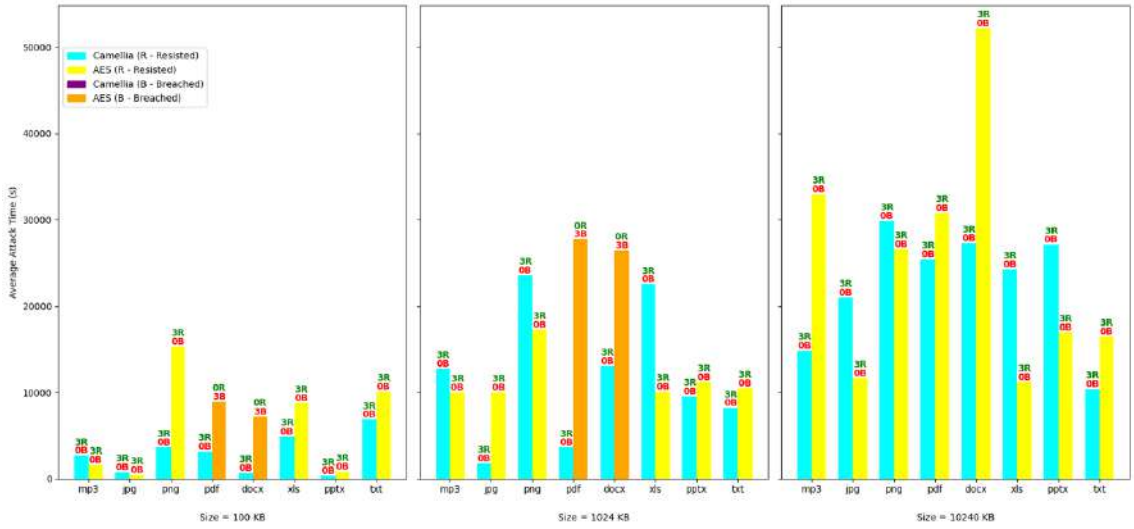


Figure 5 Visualization of Dictionary Attack

Table 6 indicates that Camellia consistently resisted all Birthday Attack attempts across every file type and size, with no breaches observed. In contrast, AES was breached in several cases, particularly in *.mp3, *.jpg, *.png, *.pdf, and *.docx at 100 KB and *.docx at 1024 KB. Attack times varied depending on file type and size—smaller files like *.jpg and *.mp3 at 100 KB had shorter durations (around 6100–11600 seconds in AES), while larger files such as *.docx and *.txt at

10240 KB required over 25,000 seconds in AES and over 24,000 seconds in Camellia. These findings indicate that while Camellia maintained complete resistance, file characteristics and algorithm behavior under collision-based attacks still influenced the computational effort involved. Figure 6 further illustrates this comparison by visualizing the average attack times between AES and Camellia, calculated from three predefined sizes of consistent test repetitions. The data confirms that AES was breached (3B) on several smaller files—specifically *.mp3, *.pdf, *.docx, and *.txt at 100 KB, as well as *.docx at 1024 KB—while Camellia maintained complete resistance (3R) across all tested files. Interestingly, even when both algorithms resisted attacks, AES consistently required longer durations, with the gap becoming most prominent at 10240 KB, suggesting that Camellia offers consistent resistance and performs more efficiently across increasing file sizes.

Table 6 Result of the Birthday Attack

Types	Predefined Sizes	File 1				File 2				File 3			
		Camellia		AES		Camellia		AES		Camellia		AES	
		AT (s)	AO	AT (s)	AO	AT (s)	AO	AT (s)	AO	AT (s)	AO	AT (s)	AO
*.mp3	100 KB	3616	R	11624	B	3711	R	11634	B	3602	R	11604	B
	1024 KB	6858	R	15301	R	6768	R	15331	R	6804	R	15281	R
	10240 KB	10426	R	17269	R	10403	R	17229	R	10503	R	17299	R
*.jpg	100 KB	4400	R	6132	B	4344	R	6102	B	4435	R	6162	B
	1024 KB	8757	R	18370	R	8816	R	18340	R	8803	R	18400	R
	10240 KB	21019	R	21307	R	20969	R	21337	R	20965	R	21277	R
*.png	100 KB	4900	R	8934	B	4890	R	8964	B	4993	R	8904	B
	1024 KB	11150	R	13500	R	11060	R	13530	R	11075	R	13490	R
	10240 KB	21247	R	13699	R	21206	R	13729	R	21208	R	13669	R
*.pdf	100 KB	5733	R	11150	B	5772	R	11180	B	5721	R	11120	B
	1024 KB	6858	R	12342	R	6792	R	12372	R	6926	R	12312	R
	10240 KB	8934	R	17272	R	9018	R	17302	R	8927	R	17242	R
*.docx	100 KB	13063	R	10058	B	13044	R	10088	B	13069	R	10028	B
	1024 KB	16492	R	21247	B	16449	R	21277	B	16475	R	21217	B
	10240 KB	17269	R	25608	R	17309	R	25638	R	17280	R	25578	R
*.xls	100 KB	10002	R	4486	R	10076	R	4516	R	9930	R	4456	R
	1024 KB	10099	R	21019	R	10172	R	21049	R	10027	R	20989	R
	10240 KB	15301	R	23552	R	15347	R	23582	R	15209	R	23522	R
*.pptx	100 KB	8185	R	3616	R	8283	R	3586	R	8096	R	3646	R
	1024 KB	10077	R	4400	R	10098	R	4430	R	10055	R	4370	R
	10240 KB	11624	R	21622	R	11655	R	21652	R	11697	R	21592	R
*.txt	100 KB	4486	R	4900	R	4414	R	4930	R	4386	R	4870	R
	1024 KB	9553	R	13063	R	9457	R	13093	R	9590	R	13033	R
	10240 KB	24259	R	22510	R	24292	R	22540	R	24208	R	22480	R

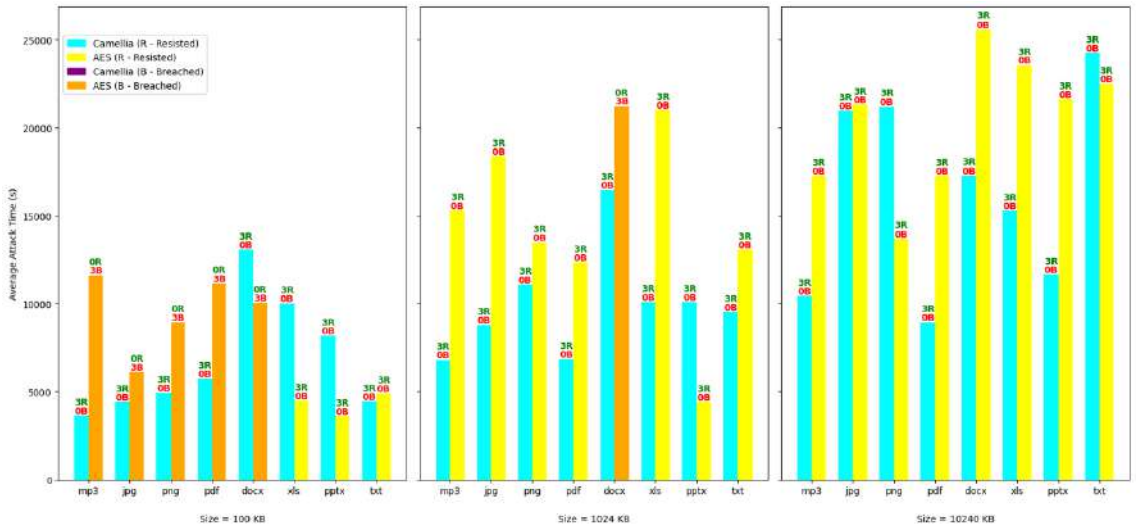


Figure 6 Visualization of Birthday Attack

Table 7 reveals a clear performance contrast between the two algorithms, particularly at the 100 KB level. AES consistently required longer attack durations than Camellia, especially for files like *.docx, *.jpg, and *.xls, which surpassed 18,000 seconds. Camellia maintained lower attack times and more consistent results across all file types and sizes. Both algorithms exhibited longer durations as file sizes increased to 10240 KB, though AES remained slower overall. These patterns suggest that Camellia handled brute-force attempts more efficiently, while certain 100 KB files appeared more susceptible, especially under AES.

Table 7 Brute-Force Attack Test Results

Types	Predefined Sizes	File 1				File 2				File 3			
		Camellia		AES		Camellia		AES		Camellia		AES	
		AT (s)	AO	AT (s)	AO	AT (s)	AO	AT (s)	AO	AT (s)	AO	AT (s)	AO
*.mp3	100 KB	3943	R	8340	B	3900	R	8344	B	3843	R	8331	B
	1024 KB	11580	R	15720	R	11573	R	15604	R	11653	R	15663	R
	10240 KB	12960	R	16380	R	12983	R	16272	R	12892	R	16364	R
*.jpg	100 KB	6960	R	10260	B	6910	R	10373	B	6947	R	10111	B
	1024 KB	18360	R	14160	B	18302	R	14038	B	18361	R	14307	B
	10240 KB	19980	R	14760	R	19922	R	14746	R	20045	R	14858	R
*.png	100 KB	4860	R	5520	B	4886	R	5416	B	4805	R	5518	B
	1024 KB	9720	R	7440	R	9698	R	7538	R	9743	R	7350	R
	10240 KB	12060	R	17580	R	12057	R	17508	R	12099	R	17596	R
*.pdf	100 KB	7260	R	8220	B	7231	R	8188	B	7288	R	8303	B
	1024 KB	13380	R	11520	R	13356	R	11468	R	13346	R	11554	R
	10240 KB	13620	R	19680	R	13611	R	19670	R	13631	R	19812	R
*.docx	100 KB	5340	R	18240	B	5305	R	18210	B	5300	R	18296	B
	1024 KB	5460	R	20880	R	5471	R	20902	R	5479	R	20902	R
	10240 KB	10680	R	21022	R	10739	R	21600	R	10713	R	21517	R
*.xls	100 KB	4260	R	8760	B	4300	R	8727	B	4233	R	8718	B
	1024 KB	10320	R	21000	R	10226	R	21010	R	10299	R	21012	R
	10240 KB	15000	R	21420	R	15005	R	21336	R	15039	R	21437	R
*.pptx	100 KB	6120	R	4740	B	6031	R	4590	B	6101	R	4844	B
	1024 KB	7080	R	6000	R	7045	R	6105	R	7063	R	6024	R
	10240 KB	20520	R	18780	R	20462	R	18788	R	20478	R	18697	R
*.txt	100 KB	3660	B	4080	B	3661	B	4193	B	3595	B	4170	B
	1024 KB	8460	R	8040	R	8458	R	7911	R	8507	R	7945	R
	10240 KB	17760	R	14640	R	17782	R	14564	R	17712	R	14782	R

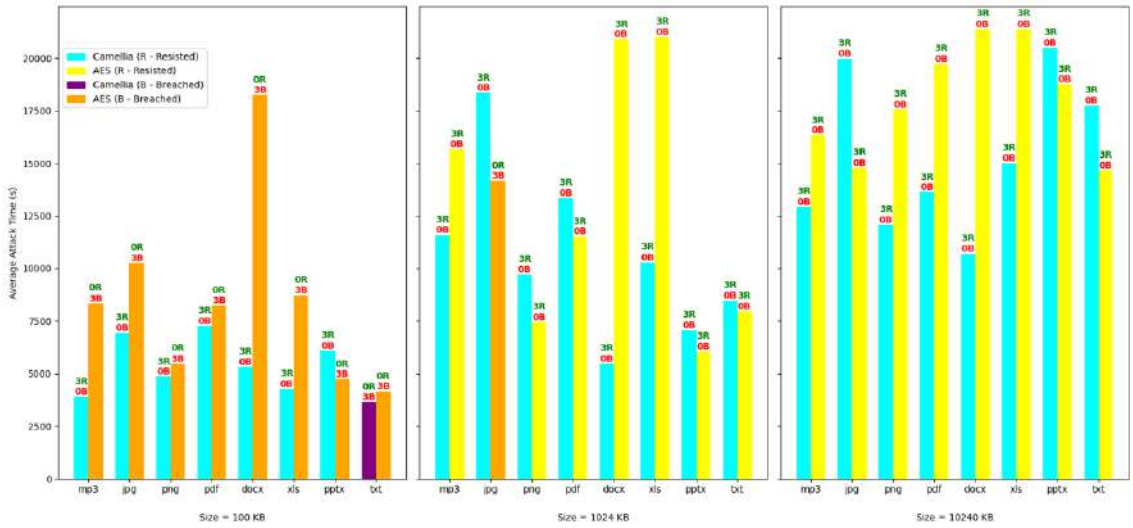


Figure 7 Visualization of Brute-Force Attack

Figure 7 presents the average brute-force attack times for AES and Camellia across various file types and sizes, based on consistent results from three predefined files per type and size in Table 7. At 100 KB, AES was breached in all file types (0R 3B), including .jpg, while Camellia showed

stronger resistance with 3R 0B—except for the .txt file, which recorded 0R 3B. Both algorithms fully resisted all attacks (3R 0B across the board) for larger sizes, with attack times increasing along with file size. These results suggest higher resilience at larger sizes and highlight a specific vulnerability in Camellia for small .txt files.

Importantly, none of these outcomes resulted from timeouts or execution limits. The breaches occurred due to actual key discovery during simulation, not from premature halting or insufficient time allocation, which confirms that algorithmic resilience—not processing duration—was the determining factor. Table 8 presents the total number of Resisted (R) and Breached (B) cases for each algorithm across all file types and attack types, summarizing the outcomes from all attack simulations. The attack simulation phase highlights distinct contrasts in the security performance of Camellia and AES across three common cryptographic threats. In the Dictionary Attack, Camellia consistently resisted all attempts across all file types and sizes. In contrast, AES was breached in specific cases—particularly for *.pdf and *.docx files at 100 KB and 1024 KB—suggesting that the key derivation or encryption patterns in AES may have been more susceptible to wordlist-based guessing in smaller or structured documents. In the Birthday Attack, which targets hash collisions and probabilistic vulnerabilities, AES again exhibited more weaknesses than Camellia. Successful breaches were recorded in AES for file types like *.mp3, *.jpg, *.png, *.pdf, and *.docx, mainly at smaller sizes. Meanwhile, Camellia resisted every test, indicating stronger protection against entropy-based collision exploitation. The Brute-Force Attack produced different observations. AES failed in all 100 KB file types tested, showing total vulnerability at smaller sizes. While more resistant overall, Camellia experienced a breach only in the 100 KB *.txt file. This suggests that straightforward file structures may reduce the number of key permutations needed to match the plaintext, making brute-force attempts more effective. These outcomes should be interpreted as empirical behavior under the defined attack configuration. The observed breaches highlight that robustness can differ across file structures at smaller sizes, reinforcing that algorithm choice should be aligned with the expected data profile and the system's threat assumptions, not solely with raw encryption speed.

Table 8 Summary of Attack Outcomes

Types	Predefined Sizes	Dictionary Attack				Birthday Attack				Brute-Force Attack				Total	
		Camellia		AES		Camellia		AES		Camellia		AES			
		B	R	B	R	B	R	B	R	B	R	B	R	B	R
*.mp3	100 KB	0	3	0	3	0	3	3	0	0	3	3	0	6	12
	1024 KB	0	3	0	3	0	3	0	3	0	3	0	3	0	18
	10240 KB	0	3	0	3	0	3	0	3	0	3	0	3	0	18
*.jpg	100 KB	0	3	0	3	0	3	3	0	0	3	3	0	6	12
	1024 KB	0	3	0	3	0	3	0	3	0	3	0	3	0	18
	10240 KB	0	3	0	3	0	3	0	3	0	3	0	3	0	18
*.png	100 KB	0	3	0	3	0	3	3	0	0	3	3	0	6	12
	1024 KB	0	3	0	3	0	3	0	3	0	3	0	3	0	18
	10240 KB	0	3	0	3	0	3	0	3	0	3	0	3	0	18
*.pdf	100 KB	0	3	3	0	0	3	3	0	0	3	3	0	9	9
	1024 KB	0	3	0	3	0	3	0	3	0	3	0	3	0	18
	10240 KB	0	3	0	3	0	3	0	3	0	3	0	3	0	18
*.docx	100 KB	0	3	3	0	0	3	3	0	0	3	3	0	9	9
	1024 KB	0	3	3	0	0	3	3	0	0	3	0	3	6	12
	10240 KB	0	3	0	3	0	3	0	3	0	3	0	3	0	18
*.xls	100 KB	0	3	0	3	0	3	0	3	0	3	3	0	3	15
	1024 KB	0	3	0	3	0	3	0	3	0	3	0	3	0	18
	10240 KB	0	3	0	3	0	3	0	3	0	3	0	3	0	18
*.pptx	100 KB	0	3	0	3	0	3	0	3	0	3	3	0	3	15
	1024 KB	0	3	0	3	0	3	0	3	0	3	0	3	0	18
	10240 KB	0	3	0	3	0	3	0	3	0	3	0	3	0	18
*.txt	100 KB	0	3	0	3	0	3	0	3	3	0	3	0	6	12
	1024 KB	0	3	0	3	0	3	0	3	0	3	0	3	0	18
	10240 KB	0	3	0	3	0	3	0	3	0	3	0	3	0	18
Total		**0**	**72**	**9**	**63**	**0**	**72**	**18**	**54**	**3**	**69**	**24**	**48**	**54**	**378**

The performance comparison between Camellia and AES reveals distinct characteristics in terms of both computational efficiency and cryptographic robustness. From the encryption and

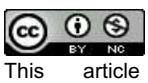
decryption phases, AES consistently demonstrated superior speed, completing tasks in significantly shorter run times than Camellia across all file types and sizes. This advantage is especially apparent in larger files, where AES achieved nearly threefold faster processing, making it more suitable for time-sensitive applications. However, this speed benefit comes at the cost of higher memory usage. AES generally consumes more RAM during encryption, particularly for complex file types such as *.pdf and *.xls. In contrast, Camellia maintained more stable and lower RAM consumption, suggesting its advantage in memory-constrained environments. CPU utilization, on the other hand, varied depending on file type. However, Camellia often exhibited slightly higher processor loads during encryption, while AES showed more efficient CPU use in specific formats like *.docx. Regarding robustness, the attack simulation phase highlighted AES's relative weaknesses compared to Camellia. Camellia resisted all attacks in the Dictionary and Birthday categories and was only breached in one case during Brute-Force testing (the 100 KB *.txt file). AES, meanwhile, recorded multiple breaches, particularly in smaller file sizes. Notably, AES failed to withstand Dictionary and Birthday Attacks in formats like *.pdf and *.docx at 100 KB and 1024 KB, and was breached across all file types in the Brute-Force Attack at the smallest size. Despite its speed, these findings suggest that AES may exhibit reduced resistance against specific cryptanalytic techniques in small or structured file types. Conversely, Camellia provides more consistent security, particularly at smaller sizes, although it is marginally slower.

The variations observed across run time, CPU usage, and memory consumption confirm that the performance characteristics of both algorithms are highly dependent on file type and workload scale. As file size increases, the differences between AES and Camellia become more pronounced, offering valuable insight into their suitability for various system requirements and processing environments. The attack simulations highlight significant disparities in how each algorithm responds to cryptographic threats. Resistance levels varied across Dictionary, Birthday, and Brute-Force attack scenarios based on file structure and size. While AES exhibited multiple breaches—especially on smaller and structured files—Camellia consistently demonstrated stronger resilience in most cases, indicating more robust default protection in diverse conditions. Camellia and AES are evaluated based on the combined performance efficiency and cryptographic robustness assessment. Rather than favoring one algorithm absolutely, the findings underscore the importance of aligning algorithm selection with system priorities. One algorithm may offer operational advantages for applications requiring rapid data handling, while for scenarios prioritizing security under sustained attack, another may provide more consistent resilience. The results highlight that performance and security trade-offs are not uniform across file types or sizes, emphasizing the need for contextual decision-making in cryptographic implementations. Additionally, isolated vulnerabilities in specific structured files suggest that further refinement or hybrid encryption strategies may be required in practice.

Practical implications and system design recommendations. The findings suggest several actionable guidelines for system architects: for throughput-prioritized workflows dominated by large files and time-sensitive processing, AES is a practical choice due to consistently lower run time; for memory-constrained environments where RAM budgets are tight or multi-process encryption is expected, Camellia may be preferable because it shows more stable memory behavior across workloads; for mixed-format repositories handling heterogeneous document formats (e.g., office and PDF files), selection should consider not only speed but also empirical robustness under the assumed attack model, particularly for smaller structured files; for security-first deployments where the threat model includes sustained password-guessing attempts, system designers should prioritize secure configurations and operational controls (strong credential policies, rate limiting, and key management hardening) and consider Camellia when results indicate more consistent resistance; and rather than enforcing a single algorithm globally, a policy-based selection approach can be implemented to choose encryption based on file size/type and system constraints (time budget versus memory budget versus threat level).

4. CONCLUSIONS

This study presented a comparative performance analysis of Camellia and AES across various file types and sizes. AES consistently achieved faster encryption and decryption times, especially



on large files, making it suitable for time-sensitive applications. However, it consumed more memory and was more vulnerable to cryptanalytic attacks. Camellia exhibited more stable resource usage and greater resistance to attacks, experiencing only one breach during brute-force testing on a small plaintext file. At the same time, AES recorded multiple breaches in both dictionary and birthday attacks. Therefore, algorithm selection should be based on system constraints and required security levels. AES is recommended for high-speed environments, while Camellia is more appropriate for systems prioritizing security and resource efficiency. Overall, this study emphasizes that performance and robustness trade-offs are workload-dependent and can vary by file structure and size under practical attack conditions. By providing a controlled multi-format dataset and a unified evaluation of efficiency and empirical robustness, this paper supports more defensible, context-aware encryption decisions in real-world system design. Future research may include testing additional algorithms, expanding file variations, and evaluating performance under real-time operational conditions.

REFERENCES

A, H. M., S, F. M., & G, E. J. (2022). Implementation of Password Hashing on Embedded Systems with Cryptographic Acceleration Unit. *International Journal of Advanced Computer Science and Applications*, *13*(2), 171–175. https://doi.org/10.14569/IJACSA.2022.0130221

Adams, C. (2021). Dictionary Attack. In *Encyclopedia of Cryptography, Security and Privacy* (pp. 1–2). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-27739-9_74-2

Alkhwaja, I., Albugami, M., Alkhwaja, A., Alghamdi, M., Abahussain, H., Alfawaz, F., Almurayh, A., & Min-Allah, N. (2023). Password Cracking with Brute Force Algorithm and Dictionary Attack Using Parallel Programming. *Applied Sciences*, *13*(10), 5979. https://doi.org/10.3390/app13105979

Andriyanto, M. R., & Sukmasetya, P. (2022). Penerapan Algoritma Advanced Encryption Standard (AES) untuk Keamanan Data Transaksi pada Sistem E-Marketplace. *Journal of Computer System and Informatics (JoSYC)*, *4*(1), 179–187. https://doi.org/10.47065/josyc.v4i1.2451

Arabul, E., Oliveira, R. D., Emami, A., Typos, S., Vrontos, C., Wang, R., Nejabati, R., & Simeonidou, D. (2023). 100 Gbps Quantum-Secured and O-RAN-Enabled Programmable Optical Transport Network for 5G Fronthaul. *Journal of Optical Communications and Networking*, *15*(8), C223–C231. https://doi.org/10.1364/JOCN.483644

Azhari, M., Mulyana, D. I., Perwitosari, F. J., & Ali, F. (2022). Implementasi Pengamanan Data pada Dokumen Menggunakan Algoritma Kriptografi Advanced Encryption Standard (AES). *Jurnal Pendidikan Sains dan Komputer*, *2*(01), 163–171. https://doi.org/10.47709/jpsk.v2i01.1390

Bibiola, F., Kalsum, T. U., & Alamsyah, H. (2023). Penerapan Algoritma Advance Encryption Standard (AES) untuk Pengamanan File pada Aplikasi Berbasis WEB. *Jurnal Surya Energy*, *8*(1), 35. https://doi.org/10.32502/jse.v8i1.6461

Chavan, R., Gulge, A., & Bhandare, S. (2024). Moving Object Detection and Classification Using Deep Learning Techniques. *Tuijin Jishu/Journal of Propulsion Technology*, *45*(1), 1001–4055. https://www.propulsiontechjournal.com/index.php/journal/article/view/5000

Dwina, N., Khairani, N., Nasir, M., & Indrawati, I. (2023). Penerapan Metode Advanced Encryption Standard pada Sistem Penyimpanan Data Menggunakan Cloud Computing Sebagai Software-as-a-Service. *Journal of Artificial Intelligence and Software Engineering (J-AISE)*, *3*(1), 25–29. https://doi.org/10.30811/jaise.v3i1.4183

Fahdi, H. Al, & Ahmed, M. (2020). Cryptographic Attacks, Impacts and Countermeasures. In *Security Analytics for the Internet of Everything* (pp. 215–230). CRC Press. https://doi.org/10.1201/9781003010463-13

G, E. J., A, H. M., & S, F. H. M. (2022). Enhanced Security: Implementation of Hybrid Image Steganography Technique Using Low-Contrast LSB and AES-CBC Cryptography. *International Journal of Advanced Computer Science and Applications*, *13*(8), 899–905. https://doi.org/10.14569/IJACSA.2022.01308104

Hamza, A. A., & Al-Janabi, R. J. surayh. (2024). Detecting Brute Force Attacks Using Machine Learning. *BIO Web of Conferences*, *97*, Article ID: 00045. https://doi.org/10.1051/bioconf/20249700045

Haris, M., Lydia, M. S., & Sutarman, S. (2023). Pengamanan pada Citra Digital dengan Menggunakan Modifikasi Blok Data Algoritma AES - Rijndael. *Jurnal Media Informatika Budidarma*, *7*(1), 444–453. https://doi.org/10.30865/mib.v7i1.5458

Hidayati, L. N., Fitriana, G. F., & Adam, I. F. (2021). Perbandingan Keacakan Citra Enkripsi Algoritma AES dan Camelia Uji NPCR dan UACI. *JURIKOM (Jurnal Riset Komputer)*, *8*(6), 274–283. https://doi.org/10.30865/jurikom.v8i6.3624

Housley, R., & Schaad, J. (2020). *RFC 3394 - Advanced Encryption Standard (AES) Key Wrap Algorithm*. IETF Datatracker. https://datatracker.ietf.org/doc/rfc3394/

Hranický, R., Šírová, L., & Rucký, V. (2025). Beyond the Dictionary Attack: Enhancing Password Cracking Efficiency Through Machine Learning-Induced Mangling Rules. *Forensic Science International: Digital Investigation*, *52*, Article ID: 301865. https://doi.org/10.1016/j.fsidi.2025.301865

Kato, A., & Moriai, S. (2013). *RFC 3657 - Use of the Camellia Encryption Algorithm in Cryptographic Message Syntax (CMS)*. IETF Datatracker. https://datatracker.ietf.org/doc/rfc3657/

Li, Y., Wang, Q., Huang, D., Liu, J., & Xie, H. (2024). Quantum Chosen-Cipher Attack on Camellia. *Cryptology EPrint Archive, Paper 2024/1804*. https://eprint.iacr.org/2024/1804

Manullang, S. (2023). Implementasi Kriptografi Pengamanan Data File Dokumen Menggunakan Algoritma Advanced Encryption Standard Mode Chiper Block Chaining. *Jurnal Nasional Teknologi Komputer*, *3*(2), 87–95. https://doi.org/10.61306/jnastek.v3i2.69

Matsui, M., Moriai, S., & Nakajima, J. (2015). *RFC 3713 - A Description of the Camellia Encryption Algorithm*. IETF Datatracker. https://datatracker.ietf.org/doc/rfc3713/

Mohammed, A. A., Rahma, A. M. S., & Abdul Wahab, H. B. (2025). Security Encryption Model of E-Bank Based Camellia Algorithm Using Cubic Curve. *AIP Conference Proceedings*, *3264*(1), Article ID: 030030. https://doi.org/10.1063/5.0263610

Muthavhine, K. D., & Sumbwanyambe, M. (2023). Blocking Linear Cryptanalysis Attacks Found on Cryptographic Algorithms Used on Internet of Thing Based on the Novel Approaches of Using Galois Field (GF (232)) and High Irreducible Polynomials. *Applied Sciences*, *13*(23), Article ID: 12834. https://doi.org/10.3390/app132312834

Ning, L., Ali, Y., Ke, H., Nazir, S., & Huanli, Z. (2020). A Hybrid MCDM Approach of Selecting Lightweight Cryptographic Cipher Based on ISO and NIST Lightweight Cryptography Security Requirements for Internet of Health Things. *IEEE Access*, *8*, 220165–220187. https://doi.org/10.1109/ACCESS.2020.3041327

Nithishma, A., Vijayan, A., Nanda, D., Kumar, Ch. A., Thaseen, S., & Ahmad, A. (2022). Remote User Authentication Using Camellia Encryption for Network-Based Applications. In *Security and Privacy-Preserving Techniques in Wireless Robotics* (pp. 255–279). CRC Press. https://doi.org/10.1201/9781003156406-19

Panahi, U., & Bayılmış, C. (2023). Enabling Secure Data Transmission for Wireless Sensor Networks Based IoT Applications. *Ain Shams Engineering Journal*, *14*(2), Article ID: 101866. https://doi.org/10.1016/j.asej.2022.101866

Patra, R., & Patra, S. (2021). Cryptography: A Quantitative Analysis of the Effectiveness of Various Password Storage Techniques. *Journal of Student Research*, *10*(3). https://doi.org/10.47611/jsrhs.v10i3.1764

Rachmayanti, A., & Wirawan, W. (2022). Implementasi Algoritma Advanced Encryption Standard (AES) pada Jaringan Internet of Things (IoT) untuk Mendukung Smart Healthcare. *Jurnal Teknik ITS*, *11*(3). https://doi.org/10.12962/j23373539.v11i3.97042

Rasheed, A., Baza, M., Badr, Mahmoud. M., Alshahrani, H., & Choo, K.-K. R. (2024). Efficient Crypto Engine for Authenticated Encryption, Data Traceability, and Replay Attack Detection Over CAN Bus Network. *IEEE Transactions on Network Science and Engineering*, *11*(1), 1008–1025. https://doi.org/10.1109/TNSE.2023.3312545

Rasheed, A., Baza, M., Khan, M., Karpoor, N., Varol, C., & Srivastava, G. (2023). Using Authenticated Encryption for Securing Controller Area Networks in Autonomous Mobile Platforms. *2023 26th International Symposium on Wireless Personal Multimedia Communications (WPMC)*, 76–82. https://doi.org/10.1109/WPMC59531.2023.10338834

Rashidi, B. (2021). Flexible and High-Throughput Structures of Camellia Block Cipher for Security of the Internet of Things. *IET Computers & Digital Techniques*, *15*(3), 171–184. https://doi.org/10.1049/cdt2.12025

Sarmila, K. B., & Manisekaran, S. V. (2022). Honey Encryption and AES based Data Protection Against Brute Force Attack. *2022 Sixth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, 187–190. https://doi.org/10.1109/I-SMAC55078.2022.9987304

Sulaiman, A. S., & Hammood, M. M. (2025). Enhancing Data Security by Using Hybrid Encryption Technique Based on AES and Camellia. In *Learning and Analytics in Intelligent Systems* (Vol. 45, pp. 173–182). https://doi.org/10.1007/978-3-031-82706-8_18

Tanjung, P. Y. (2022). Penerapan Algoritma AES 625 dalam Pengamanan Data Rekam Medis. *Journal Global Technology Computer*, *1*(3), 77–83. https://doi.org/10.47065/jogtc.v1i3.2054

Verma, R., Dhanda, N., & Nagar, V. (2022). Enhancing Security with In-Depth Analysis of Brute-Force Attack on Secure Hashing Algorithms. In *Lecture Notes in Networks and Systems* (Vol. 376, pp. 513–522). https://doi.org/10.1007/978-981-16-8826-3_44

Wei, Z., Sun, S., Hu, L., Wei, M., & Peralta, R. (2023). Searching the Space of Tower Field Implementations of the $\mathbb{F}28$ Inverter - with Applications to AES, Camellia and SM4. *International Journal of Information and Computer Security*, *20*(1–2), 1–26. https://doi.org/10.1504/IJICS.2023.127999

ZhenQiang, L., Fei, G., SuJuan, Q., & QiaoYan, W. (2023). Quantum Circuit for Implementing Camellia S-Box with Low Costs. *SCIENTIA SINICA Physica, Mechanica & Astronomica*, *53*(4), Article ID: 240313. https://doi.org/10.1360/SSPMA-2022-0485

JISKA (Jurnal Informatika Sunan Kalijaga)
Vol. 11, No. 1, JANUARI, 2026, Pp. 114 – 126
ISSN: 2527 – 5836 (print) | 2528 – 0074 (online)

Model Prediksi Risiko Kanker Serviks dengan Pendekatan Support Vector Machine

Juwita Stefany Hutapea [(1)*], Nisa Hanum Harani [(2)], Cahyo Prianto [(3)]
Departemen Teknik Informatika, Universitas Logistik dan Bisnis Internasional, Bandung, Indonesia
e-mail : juwitastefany13@gmail.com, {nisa,cahyo}@ulbi.ac.id.
* Penulis korespondensi.
Artikel ini diajukan 5 Agustus 2025, direvisi 15 Januari 2026, diterima 15 Januari 2026, dan dipublikasikan 25 Januari 2026.

Abstract

Cervical cancer is one of the leading causes of death in women, especially in developing countries due to delays in early diagnosis. Developing a risk prediction model based on the Support Vector Machine (SVM) algorithm is one way to support a more accurate and efficient early detection process. The research object is medical records of female patients obtained from hospitals in Medan City, with a total of 164 patient data. The development process was carried out through the CRISP-DM stages, which include data cleaning, feature transformation, class balancing with SMOTE, and dimensionality reduction using PCA. The evaluation results showed that the best model was obtained with a PCA configuration with 9 principal components (90% variance) and a test size of 80:20, resulting in an accuracy of 88%, a precision of 88%, a recall of 84%, and an F1-score of 86%. Cross-validation evaluation with 5 folds provided the best average performance and the smallest standard deviation, indicating model stability. The final model was implemented in a web-based system to facilitate digital early detection. This study shows that SVM with the SMOTE and PCA approaches is effective in predicting cervical cancer risk accurately and efficiently.

Keywords: Cervical Cancer, Prediction, Support Vector Machine, SMOTE, PCA

Abstrak

Kanker serviks merupakan salah satu penyebab kematian tertinggi pada wanita terutama di negara berkembang akibat keterlambatan dalam diagnosis dini. Pengembangan model prediksi risiko berbasis algoritma *Support Vector Machine* (SVM) menjadi salah satu cara dalam mendukung proses deteksi dini yang lebih akurat dan efisien. Objek penelitian berupa data rekam medis pasien wanita yang diperoleh dari rumah sakit di Kota Medan, dengan total 164 data pasien. Proses pengembangan dilakukan melalui tahapan CRISP-DM yang mencakup pembersihan data, transformasi fitur, penyeimbangan kelas dengan SMOTE dan reduksi dimensi menggunakan PCA. Hasil evaluasi menunjukkan bahwa model terbaik diperoleh pada konfigurasi PCA dengan 9 komponen utama (90% variansi) dan test size 80:20 yang menghasilkan akurasi sebesar 88%, precision 88%, recall 84%, dan F1-score 86%. Evaluasi *cross validation* dengan fold 5 memberikan performa rata-rata terbaik dan standar deviasi terkecil, menandakan kestabilan model. Model akhir diimplementasikan dalam sistem berbasis web untuk mempermudah deteksi dini secara digital. Penelitian ini menunjukkan bahwa SVM dengan pendekatan SMOTE dan PCA efektif untuk memprediksi risiko kanker serviks secara akurat dan efisien.

Kata Kunci: Kanker Serviks, Prediksi, Support Vector Machine, SMOTE, PCA

1. PENDAHULUAN

Kanker serviks merupakan salah satu jenis kanker yang memiliki dampak signifikan secara global. Penyakit ini berkembang dari sel-sel abnormal pada leher rahim (serviks). Berdasarkan data dari *Global Cancer Observatory* (Globocan) yang dikeluarkan oleh *World Health Organization* (WHO) pada tahun 2020, terdapat 604.127 kasus dan 341.831 orang meninggal karena kanker serviks di seluruh dunia (Singh et al., 2023; Sung et al., 2021). Hampir 90% dari kasus kematian tersebut terjadi di negara-negara berkembang. Di Indonesia, kanker serviks

menjadi penyakit kedua yang paling sering terjadi pada wanita, dengan perkiraan 36.633 kasus baru dan 21.003 kematian setiap tahun (Indarti, 2023). Hal ini menunjukkan bahwa kanker serviks masih menjadi masalah kesehatan yang serius dan membutuhkan solusi tepat dalam medeteksi secara dini.

Salah satu hambatan dalam penanganan kanker serviks adalah keterlambatan dalam diagnosis yang menyebabkan angka kematian tinggi. Metode pemeriksaan konvensional seperti *Pap smear* dan inspeksi visual dengan asam asetat (IVA) sering dibatasi karena interpretasi hasil yang bersifat subjektif, ketersediaan tenaga medis yang terbatas, serta akses yang kurang merata di berbagai wilayah (Ekawati et al., 2024). Oleh karena itu, diperlukan pendekatan berbasis teknologi yang dapat mendukung sistem bantuan keputusan untuk membantu memprediksi kanker serviks lebih objektif, efisien, dan akurat.

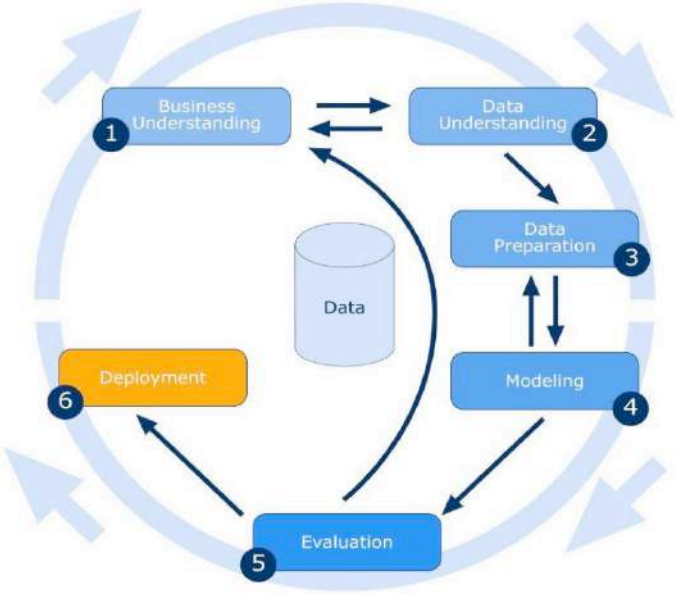
Perkembangan teknologi kecerdasan buatan terutama *machine learning* menjadi peluang baru dalam analisis data medis untuk memprediksi penyakit. Algoritma *Support Vector Machine* (SVM) dikenal sebagai metode klasifikasi yang baik dalam menangani data berdimensi tinggi dan pola yang tidak linear, sehingga cocok untuk analisis data rekam medis yang kompleks. Sejumlah penelitian terdahulu telah menunjukkan potensi besar machine learning dalam prediksi kanker serviks. Studi yang dilakukan oleh (Binanto et al., 2024) berfokus pada perbandingan efektivitas algoritma *Random Forest* (RF) dan *Support Vector Machine* (SVM) dalam menilai risiko kanker serviks. Penelitian ini menggunakan dataset dari Kaggle dan menyoroti tahapan preprocessing, pembagian data, serta evaluasi performa model menggunakan ROC curve dan metrik seperti akurasi, precision, recall, dan F1-score. Hasilnya menunjukkan bahwa RF mencapai akurasi sebesar 97,16% dengan AUC 0,996, sementara SVM hanya mencapai akurasi 96,01% dan AUC 0,543, menjadikan RF lebih unggul dalam konteks ini.

Penelitian oleh (Alsmariy et al., 2020) mengusulkan model prediksi kanker serviks dengan mengombinasikan metode voting classifier. Dataset UCI digunakan bersama teknik SMOTE untuk penyeimbangan kelas dan PCA untuk reduksi dimensi. Model ini menunjukkan peningkatan signifikan dalam akurasi dan sensitivitas (hingga lebih dari 5%) dibandingkan model-model sebelumnya. Penelitian ini menegaskan bahwa kombinasi SMOTE dan PCA efektif dalam meningkatkan kinerja klasifikasi dan efisiensi komputasi untuk prediksi kanker serviks. Berdasarkan hasil penelitian-penelitian sebelumnya menunjukkan bahwa SVM mampu memberikan hasil yang baik dalam klasifikasi berbagai penyakit, termasuk kanker serviks terutama ketiga digunakan dengan teknik pra-pemrosesan seperti SMOTE untuk menyeimbangkan data dan PCA untuk mengurangi dimensi.

Penelitian ini bertujuan untuk mengembangkan model prediksi risiko kanker serviks dengan menggunakan data rekam medis pasien kanker serviks di salah satu rumah sakit di Kota Medan dengan menggunakan algoritma *Support Vector Machine* dan penerapan teknik SMOTE serta PCA. Proses pengembangan model dijalani melalui tahapan CRISP-DM, yaitu pembersihan data, transformasi fitur, penyeimbangan kelas, pemodelan, dan evaluasi. Diharapkan model yang dihasilkan dapat menjadi dasar sistem bantuan keputusan untuk memprediksi risiko kanker serviks secara dini dan membantu meningkatkan kualitas layanan kesehatan wanita di Indonesia.

2. METODE PENELITIAN

Penelitian ini menggunakan pendekatan CRISP-DM (*Cross Industry Standard Process for Data Mining*) sebagai kerangka kerja utama dalam pengembangan model prediksi risiko kanker serviks berbasis algoritma *Support Vector Machine* (SVM) (Pinto et al., 2020). Metodologi ini dipilih karena merupakan standar dalam data mining dan memiliki struktur yang sistematis dalam proses pengolahan data (Hasanah et al., 2021). Alur proses penelitian dapat dilihat pada Gambar 1. Secara umum, tahapan penelitian ini meliputi pemahaman bisnis, pemahaman data, persiapan data, pemodelan, evaluasi, dan deployment.



Gambar 1 Diagram Alur CRISP-DM

2.1 Support Vector Machine (SVM)

Support Vector Machine (SVM) adalah salah satu algoritma machine learning berbasis *supervised learning* yang dirancang untuk menyelesaikan permasalahan klasifikasi dan regresi. Teknik ini bertujuan untuk mencari sebuah bidang pemisah (*hyperlane*) yang paling optimal dalam memisahkan dua kelas data. Prinsip utama dari SVM adalah memilih bidang pemisah yang memiliki margin terbesar antara dua kelompok data, di mana margin tersebut merupakan jarak antara *hyperlane* dengan titik-titik data terdekat dari masing-masing kelas yang dikenal sebagai support vectors (Jalil et al., 2024).

2.2 Synthetic Minority Over-sampling Technique (SMOTE)

Synthetic Minority Over-sampling Technique (SMOTE) adalah metode oversampling yang digunakan untuk mengatasi masalah ketidakseimbangan kelas dalam kumpulan data (Fatmawati et al., 2025). Kumpulan data yang tidak seimbang, di mana satu kelas (kelas mayoritas) memiliki jumlah data yang jauh lebih banyak daripada kelas lainnya (kelas minoritas) dapat menghasilkan model klasifikasi yang bias dan berkinerja buruk dalam mengidentifikasi kelas minoritas (Sulistiyono et al., 2021). SMOTE bekerja dengan cara menghasilkan data sintetis baru dari kelas minoritas melalui interpolasi antar titik data yang saling berdekatan (nearest neighbors), bukan hanya menggandakan data yang sudah ada (Parman et al., 2024).

2.3 Principal Component Analysis (PCA)

Principal Component Analysis (PCA) adalah teknik statistik yang digunakan untuk mengurangi jumlah fitur atau variabel dalam sebuah dataset tanpa kehilangan informasi penting. Metode ini sangat bermanfaat ketika data memiliki banyak atribut yang saling berkorelasi sehingga dapat menyederhanakan kompleksitas model dan mempercepat proses pelatihan algoritma seperti *Support Vector Machine* (SVM) (Oktafiani & Itje Sela, 2024). PCA memiliki tujuan untuk menemukan pola dan struktur tersembunyi dalam data berdimensi tinggi dengan mentransformasikannya ke dalam ruang berdimensi lebih rendah, namun tetap mempertahankan informasi yang paling penting. Proses ini dilakukan dengan menghilangkan korelasi antar variabel independent melalui transformasi ke dalam variabel-variabel baru yang saling tidak berkorelasi yang dikenal sebagai principal components. Hasilnya adalah representasi data dalam dimensi yang lebih rendah namun tetap mempertahankan informasi paling penting dari data asli (Mulla et al., 2021).

2.4 Pemahaman Bisnis (*Business Understanding*)

Tahap pertama dalam penelitian ini adalah business understanding untuk mengetahui sebuah masalah (Cahyaningtyas et al., 2022). Tingginya angka kematian akibat kanker serviks disebabkan oleh keterlambatan dalam proses diagnosis dini. Metode skrining konvensional seperti *Pap smear* dan IVA memiliki keterbatasan dalam hal akurasi dan aksesibilitas. Penelitian ini bertujuan mengembangkan model prediktif berbasis *Support Vector Machine* (SVM) untuk mengidentifikasi pasien dengan risiko tinggi menggunakan data rekam medis. Model ini diharapkan dapat mendukung proses skrining yang lebih cepat, objektif, dan akurat dalam konteks deteksi dini kanker serviks.

2.5 Pemahaman Data (*Data Understanding*)

Setelah tujuan penelitian ditetapkan, tahap selanjutnya adalah memahami data yang akan digunakan. Tahap ini diawali dengan pengumpulan data yang menjadi dasar untuk analisis dan pengambilan kesimpulan (Hasanah et al., 2021). Dataset yang digunakan dalam penelitian ini berasal dari data rekam medis pasien di salah satu rumah sakit di Kota Medan dengan total 164 data yang berisi informasi klinis seperti usia, gejala utama, riwayat *pap smear*, dan lainnya yang disajikan pada Tabel 1.

Tabel 1 Dataset

No.	Atribut	Keterangan
1	Nomor_Rekam_Medis	Nomor identitas unik pasien di rumah sakit
2	Umur	Usia pasien
3	Status_Perkawinan	Status pernikahan pasien (contoh: Belum menikah, Menikah)
4	Gejala_Utama	Gejala utama yang dirasakan pasien saat pertama kali memeriksakan diri
5	Durasi_Gejala_Minggu	Lama gejala berlangsung dalam minggu
6	Riwayat_Pap_Smear	Riwayat pemeriksaan Pap Smear sebelumnya (Pernah/Tidak Pernah)
7	Riwayat_Kontrasepsi	Riwayat penggunaan kontrasepsi (Pernah/Tidak Pernah)
8	Jumlah_Kehamilan	Total jumlah kehamilan yang pernah dialami pasien
9	Tes_HPV	Riwayat melakukan tes HPV (Ya/Tidak)
10	Jumlah_Pasangan_Seksual	Jumlah pasangan seksual yang pernah dimiliki pasien
11	Risiko_Jatuh_1th	Risiko pasien mengalami jatuh dalam 1 tahun terakhir (Ya/Tidak)
12	Psikologis	Kondisi psikologis pasien (Cemas/Tenang)
13	Berat_Badan	Berat badan pasien dalam satuan kilogram (kg)
14	Tinggi_Badan	Tinggi badan pasien dalam satuan sentimeter (cm)
15	BMI	Indeks massa tubuh pasien (Body Mass Index)
16	Tekanan_Sistole	Tekanan darah sistolik pasien (mmHg)
17	Tekanan_Diastole	Tekanan darah diastolik pasien (mmHg)
18	Frekuensi_Nadi	Denyut nadi per menit (bpm - beats per minute)
19	Frekuensi_Nafas	Jumlah nafas per menit
20	Suhu_Tubuh	Suhu tubuh pasien dalam derajat Celcius
21	Skor_GCS	Skor Glasgow Coma Scale untuk menilai kesadaran pasien
22	Skor_Karnofsky	Skor untuk menilai kemampuan fungsional pasien
23	Tipe_Perawatan	Jenis perawatan yang diterima (Rawat Inap/Rawat Jalan)
24	Tahun	Tahun dilakukannya pemeriksaan atau pengambilan data
25	Hasil_Biopsi	Hasil pemeriksaan biopsi (Positif/Negatif untuk kanker serviks)

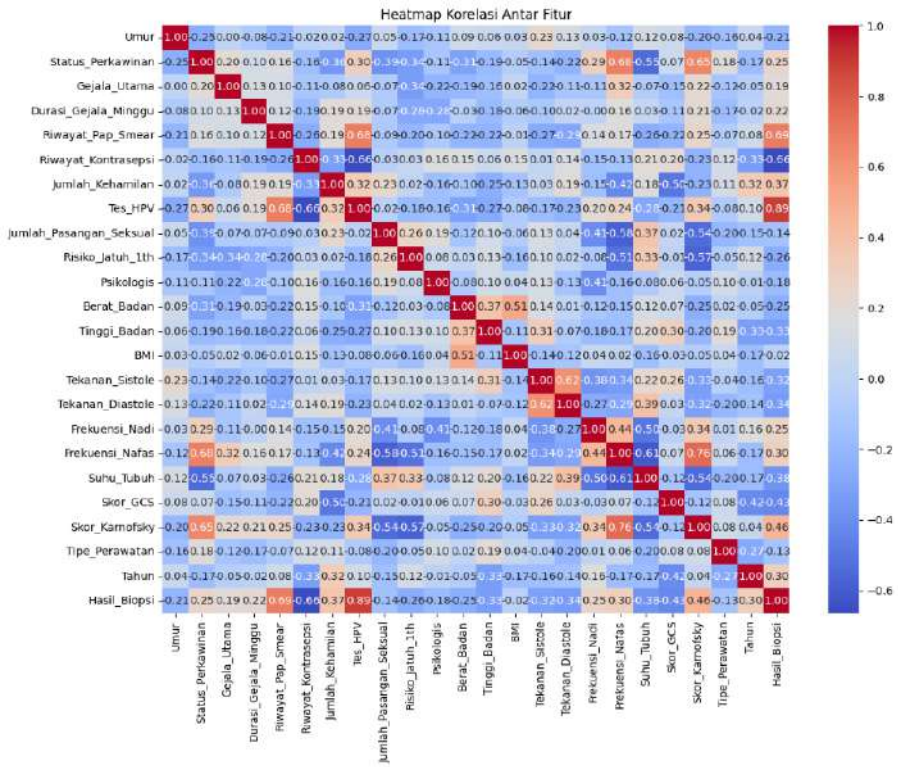
Tahap eksplorasi awal dilakukan untuk melihat struktur dan isi dataset, termasuk beberapa baris awal, tipe data, dan statistik deskriptif. Salah satunya dengan menggunakan fungsi *head* yang

menampilkan baris pertama data. Pada Gambar 2, menampilkan data rekam medis 5 pasien yang berisi informasi demografis (usia, status perkawinan), keluhan (gejala utama, durasi), riwayat kesehatan (*Pap Smear*, kontrasepsi, jumlah kehamilan), dan hasil tes HPV untuk setiap pasien.

	Nomor_Rekam_Medis	Umur	Status_Perkawinan	Gejala_Utama	Durasi_Gejala_Minggu	Riwayat_Pap_Smear	Riwayat_Kontrasepsi	Jumlah_Kehamilan	Tes_HPV
0	01.22.90.03	50	Menikah	Pendarahan	4	Tidak Pernah	Pernah	4	Ya
1	03.77.34.73	63	Menikah	Nyeri Perut	6	Pernah	Tidak Pernah	2	Tidak
2	01.22.85.54	48	Menikah	Pendarahan	2	Tidak Pernah	Tidak Pernah	1	Tidak
3	01.23.39.32	42	Menikah	Pendarahan	16	Tidak Pernah	Pernah	3	Ya
4	01.22.81.55	45	Menikah	Pendarahan	4	Tidak Pernah	Tidak Pernah	4	Ya
5	01.23.36.99	69	Menikah	Nyeri Perut	4	Tidak Pernah	Tidak Pernah	6	Ya

Gambar 2 Data Baris Pertama

Sebelum memilih fitur yang akan dipakai dalam pelatihan, dilakukan uji korelasi untuk memahami hubungan antar fitur dengan target (Hasil_Biopsi). Korelasi dihitung menggunakan korelasi *pearson* yang mengukur kekuatan dan arah hubungan linear antara dua variabel dengan nilai antara -1 hingga 1. Korelasi *pearson* memberikan gambaran tentang hubungan linear antar variabel yang berguna dalam memilih fitur yang relevan untuk model. Diagram batang yang memvisualisasikan korelasi antar berbagai fitur (variabel independent) terhadap variabel target yaitu Hasil_Biopsi ditunjukkan pada Gambar 3. mengurutkan berbagai faktor (fitur) berdasarkan kekuatan hubungannya (korelasi) dengan Hasil Biopsi. Batang yang mengarah ke kanan menunjukkan korelasi positif, sedangkan batang ke kiri menunjukkan korelasi negatif. Panjang batang merepresentasikan kekuatan hubungan, semakin panjang batangnya semakin kuat korelasinya.



Gambar 3 Korelasi Antar Fitur

Berdasarkan hasil uji korelasi terhadap target dalam Gambar 3, variabel Umur, Gejala_Utama, Durasi_Gejala_Utama, Riwayat_Pap_Smear, Riwayat_Kontrasepsi, Jumlah_Kehamilan, Tes_HPV, Jumlah_Pasangan_Seksual, Psikologis, Risiko_Jatuh_1th, dan Tahun dipilih karena menunjukkan hubungan yang signifikan, baik secara positif maupun negatif terhadap target. Misalnya, Tes_HPV dan Riwayat_Pap_Smear memiliki korelasi positif tinggi yang menunjukkan bahwa variabel ini berhubungan erat dengan Hasil_Biopsi yang positif. Jumlah_Kehamilan dan Gejala_Utama juga menunjukkan korelasi positif yang sedang sehingga relevan untuk dimasukkan dalam model prediksi. Sementara, Riwayat_Kontrasepsi memiliki korelasi negatif tinggi yang artinya kemungkinan untuk mendapatkan Hasil_Biopsi positif cenderung lebih rendah. Variabel lain seperti Umur, Tahun, dan Risiko_Jatuh_1th memiliki nilai korelasi yang lebih rendah, tetapi tetap dipertimbangkan karena masih memiliki sedikit pengaruh terhadap hasil akhir prediksi dan relevan secara klinis. Pemilihan variabel ini menunjukkan keseimbangan antara kekuatan korelasi dan relevansi kontekstual dalam prediksi risiko kanker serviks.

2.6 Persiapan Data (*Data Preparation*)

Tahap ini bertujuan untuk mengubah data mentah menjadi dataset yang bersih dan siap untuk diolah sebelum diterapkan ke dalam model pembelajaran mesin (Studer et al., 2021). Proses ini mencakup pembersihan, transformasi, dan penyusunan data agar siap digunakan dalam proses modeling. Persiapan data dilakukan untuk menghilangkan noise, mengisi nilai yang hilang, mengubah data kategorikal menjadi numerik, melakukan normalisasi, dan mengatasi ketidakseimbangan kelas agar model prediksi dapat dilatih dengan optimal.

Dataset yang digunakan terdiri atas data bertipe kategorikal dan numerik. Untuk memungkinkan pemrosesan oleh algoritma machine learning, data kategorikal perlu dikonversi terlebih dahulu ke dalam bentuk numerik melalui proses encoding, sehingga seluruh fitur dapat dikenali dan diolah secara optimal oleh model. Agar interpretasi data tetap jelas, dilakukan pencetakan hasil mapping dari nilai numerik hasil encoding ke label aslinya yang ditunjukkan dalam Gambar 4. Pada Gambar 4 menunjukkan proses mapping data, di mana setiap kategori data yang berupa teks diubah menjadi format angka untuk keperluan analisis oleh komputer. Sebagai contoh, pada kolom 'Hasil_Biopsi', nilai 'negatif' diubah menjadi 0 dan 'positif' diubah menjadi 1.

```
Mapping untuk kolom 'Status_Perkawinan':
  0 → Belum Menikah
  1 → Menikah

Mapping untuk kolom 'Gejala_Utama':
  0 → Keputihan
  1 → Nyeri Perut
  2 → Pendarahan

Mapping untuk kolom 'Riwayat_Pap_Smear':
  0 → Pernah
  1 → Tidak Pernah

Mapping untuk kolom 'Hasil_Biopsi':
  0 → negatif
  1 → positif

Mapping untuk kolom 'Riwayat_Kontrasepsi':
  0 → Pernah
  1 → Tidak Pernah
```

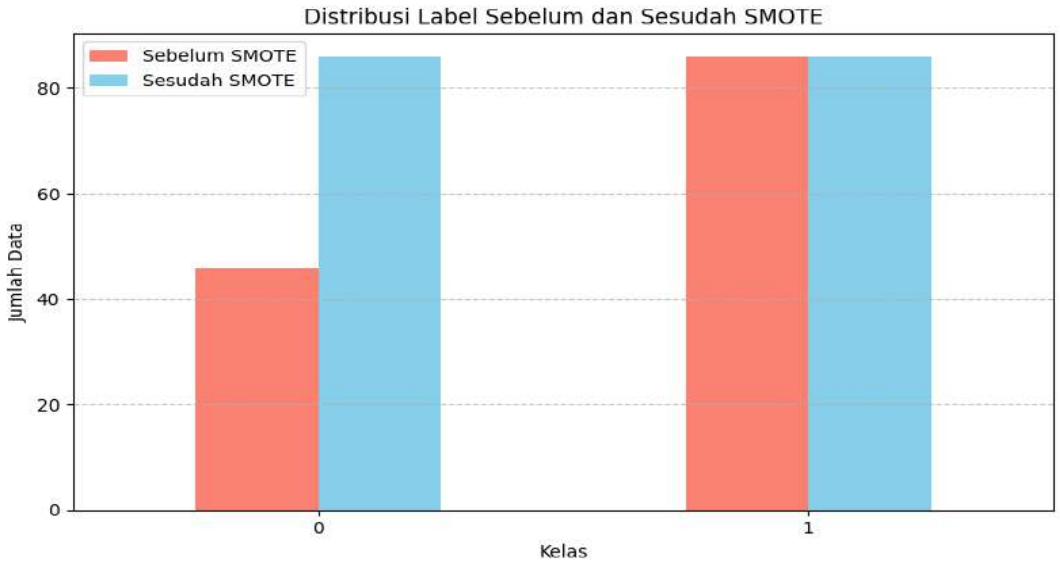
Gambar 4 Korelasi Fitur Terhadap Target

Setelah data dibersihkan dan dikonversi, langkah selanjutnya adalah memisahkan fitur (X) dan label (y) untuk keperluan pelatihan model pada tahap selanjutnya. Fitur yang digunakan meliputi variabel umur, gejala utama, durasi gejala dalam minggu, riwayat pap smear, riwayat penggunaan kontrasepsi, jumlah kehamilan, hasil tes HPV, jumlah pasangan seksual, kondisi psikologis, risiko jatuh dalam satu tahun terakhir, dan tahun pemeriksaan. Sementara itu, label yang digunakan sebagai target prediksi adalah hasil biopsi.

Data yang telah melalui tahap praproses dan dipisahkan antara fitur dan target selanjutnya dibagi menjadi dua bagian, yaitu 131 data latih (*training*) dan 33 data (*testing*). Pembagian ini bertujuan untuk memastikan bahwa model dilatih menggunakan sebagian data, sementara performanya diuji menggunakan data yang belum pernah dilihat sebelumnya. Dalam penelitian ini, digunakan beberapa variasi proporsi data pengujian (test size), yaitu 0.2, 0.3, dan 0.5 dari total 164 dataset untuk mengevaluasi konsistensi dan stabilitas performa model pada berbagai skenario pembagian data. Proporsi test size sebesar 0,2 menghasilkan 33 data uji dan 131 data latih, test size 0,3 menghasilkan 49 data uji dan 115 data latih, sedangkan test size 0,5 menghasilkan 82 data uji dan 82 data latih.

Setelah data dibagi menjadi data pelatihan dan pengujian, tahap selanjutnya adalah membangun pipeline untuk mendukung proses pelatihan model secara terstruktur dan sistematis. Penggunaan pipeline memungkinkan integrasi beberapa tahapan seperti normalisasi, penyeimbangan kelas, reduksi dimensi, dan pelatihan model secara berurutan dan konsisten setiap kali model dijalankan. Pendekatan ini tidak hanya meningkatkan efisiensi pemrosesan, tetapi juga meminimalkan potensi kesalahan akibat ketidakkonsistenan dalam penanganan data. Selain itu, pipeline memastikan bahwa seluruh transformasi hanya diterapkan pada data pelatihan dalam setiap lipatan validasi silang, sehingga mencegah terjadinya kebocoran data (*data leakage*) ke dalam proses pelatihan.

Pada tahap oversampling, dilakukan penanganan terhadap ketidakseimbangan kelas pada data target. SMOTE digunakan untuk mengatasi masalah ini dengan cara membuat sampel sintetik baru pada kelas minoritas sehingga distribusi antar kelas menjadi lebih seimbang. Hasil dari proses penyeimbangan data ditampilkan pada Gambar 5. Awalnya, dataset tidak seimbang dengan 86 data untuk kelas '1' dan hanya 46 data untuk kelas '0'. Setelah diterapkan SMOTE, jumlah data pada kelas minoritas '0' ditingkatkan sehingga kini kedua kelas menjadi seimbang dengan masing-masing memiliki 86 data.



Gambar 5 Penerapan SMOTE

Penerapan PCA dilakukan untuk mereduksi jumlah fitur menjadi sejumlah komponen utama yang mampu menjelaskan total varians data. Teknik ini digunakan untuk menyaring informasi paling relevan serta mengurangi noise atau redundansi yang tidak signifikan terhadap klasifikasi. Untuk menentukan jumlah komponen utama yang optimal tanpa mengorbankan terlalu banyak informasi penting, dilakukan pengujian terhadap 2 variasi nilai n_components, sehingga diperoleh konfigurasi terbaik dalam mereduksi dimensi sekaligus mempertahankan kinerja model secara maksimal.

JISKA (Jurnal Informatika Sunan Kalijaga)
Vol. 11, No. 1, JANUARI, 2026: 114 – 126

2.7 Pemodelan (*Modeling*)

Pada tahap ini merupakan inti dari proses data mining, di mana dilakukan penerapan algoritma yang sesuai untuk membangun model prediktif berdasarkan data yang telah dipersiapkan (Schröer et al., 2021). Model yang digunakan dalam penelitian ini adalah *Support Vector Machine* (SVM) dengan kernel *Radial Basis Function* (RBF). SVM dipilih karena kemampuannya yang andal dalam menangani klasifikasi dengan dimensi tinggi dan data yang tidak terpisah secara linear. Selain itu, pendekatan ini dilengkapi dengan teknik *cross-validation* dan penyesuaian bobot kelas (*class weighting*) untuk meningkatkan generalisasi dan mengurangi bias akibat ketidakseimbangan data.

2.8 Evaluasi Model (*Evaluation*)

Setelah model dibangun, kinerjanya dievaluasi secara objektif menggunakan data uji yang belum pernah digunakan selama proses pelatihan. Evaluasi ini bertujuan untuk mengukur seberapa baik model dapat menggeneralisasi pengetahuannya pada data baru (Dzulhijjah et al., 2024). Kinerja model dievaluasi menggunakan beberapa metrik klasifikasi yaitu akurasi, precision, recall, F1-score, dan confusion matrix. Metrik recall menjadi perhatian utama untuk memastikan model mampu mendeteksi sebanyak mungkin kasus risiko tinggi kanker serviks.

Penilaian kinerja model selanjutnya dilakukan dengan beberapa metrik. Pertama, akurasi, yaitu metrik sederhana namun penting yang menunjukkan persentase prediksi model yang benar (baik positif maupun negatif) dari total keseluruhan data (Tangkelayuk & Mailoa, 2022). Akurasi sangat bermanfaat jika distribusi data antar kelas seimbang. Rumus akurasi dapat dihitung menggunakan Pers. (1). Selanjutnya, precision, yaitu metrik yang mengukur seberapa banyak dari prediksi untuk suatu kelas yang benar. Precision yang tinggi menunjukkan bahwa model menghasilkan sedikit kesalahan (*false positive*) (Admojo & Ahsanawati, 2020). Di mana rumus precision dapat dilihat pada Pers. (2). Metrik lainnya adalah recall, metrik ini mengukur kemampuan model untuk mendeteksi semua data yang sebenarnya termasuk dalam suatu kelas. Recall yang tinggi berarti model jarang melewatkan data yang penting (*false negative*) (Prasetyo et al., 2023). Persamaan untuk menghitung recall dapat dihitung menggunakan Pers. (3). Terakhir, F1-score, yaitu metrik kombinasi antara precision dan recall yang memberikan keseimbangan antara keduanya (Sylvia et al., 2024). F1-score berguna pada saat mempertimbangkan baik *false positives* maupun *false negatives*. Di mana rumus F1-score dapat dilihat pada Pers. (4).

$$Akurasi = \frac{TP + TN}{TP + TN + FP + FN} \times 100 \quad (1)$$

$$Presisi = \frac{TP}{TP + FP} \quad (2)$$

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

$$F1Score = 2 \times \frac{Presisi \times Recall}{Presisi + Recall} \quad (4)$$

2.9 Implementasi (*Deployment*)

Sebagai tahap terakhir dalam kerangka CRISP-DM adalah deployment. Dalam penelitian ini, tahap ini berfokus pada implementasi sistem (Triginandri & Subhiyakto, 2024). Dilakukan pembangunan sistem prototipe berbasis web menggunakan framework *Flask* dan *database* MySQL. Sistem ini memungkinkan pengguna menginputkan data dan memperoleh hasil prediksi risiko secara langsung, serta visualisasi hasil prediksi melalui antarmuka pengguna yang sederhana.

3. HASIL DAN PEMBAHASAN

Penelitian ini menghasilkan model prediksi menggunakan algoritma *Support Vector Machine* (SVM) yang dibangun untuk memprediksi risiko kanker serviks berdasarkan data rekam medis. Model diuji pada berbagai scenario pengolahan data dan evaluasi untuk mendapatkan konfigurasi terbaik yang menghasilkan kinerja optimal yang ditunjukkan pada Tabel 2. Berdasarkan evaluasi awal terhadap variasi proporsi pembagian data pada Tabel 2, diperoleh hasil akurasi sebesar 0.88 untuk test_size 0.2, 0.82 untuk test_size 0.3, dan 0.85 untuk test_size 0.5. Proporsi 80:20 dipilih sebagai konfigurasi terbaik karena memberikan akurasi tertinggi sekaligus mempertahankan keseimbangan antara data pelatihan dan data pengujian.

Tabel 2 Akurasi Berdasarkan Test Size

Test_size	Hasil Akurasi
0.2	0.88
0.3	0.82
0.5	0.85

Pengujian terhadap nilai n_components pada PCA dilakukan dengan dua variasi nilai yang diuji untuk menentukan jumlah komponen utama yang dapat dipertahankan. Hasil pengujian menunjukkan bahwa pemilihan komponen yang sesuai mampu meningkatkan efisiensi tanpa menurunkan akurasi karena fitur-fitur yang kurang relevan telah dieliminasi. PCA berhasil membantu menyederhanakan kompleksitas data sekaligus mempertahankan informasi penting untuk prediksi.

Tabel 3 Evaluasi Model pada Berbagai Nilai PCA

PCA	Principal Component	Evaluasi						
		Accuracy	Precision		Recall		F1-Score	
			0	1	0	1	0	1
0.90	9	0.88	0.89	0.88	0.73	0.95	0.80	0.91
0.75	7	0.85	0.80	0.87	0.73	0.91	0.76	0.89

Berdasarkan hasil pengujian pada Tabel 3, n_components 0.90 dipilih karena mempertahankan 9 komponen dan memberikan hasil evaluasi terbaik secara keseluruhan. Akurasi model pada komponen ini mencapai 0.88 tertinggi dibandingkan komponen lain, selain itu nilai precision, recall dan F1-score untuk masing-masing kelas juga menunjukkan performa yang sangat baik dan seimbang. Sementara n_components 0.75 hasil evaluasinya lebih rendah dibandingkan dengan 0.90.

Tabel 4 Perbandingan Cross Validation

Test_size	Fold (cv)	Akurasi	Rata-Rata	Standar Deviasi
0.2	2	0.88	0.8106	0.0379
	3		0.7879	0.0214
	5		0.8023	0.0665
0.3	2	0.82	0.7564	0.0195
	3		0.8345	0.0264
	5		0.8348	0.0696
0.5	2	0.85	0.7561	0.0000
	3		0.7685	0.0330
	5		0.7691	0.0404

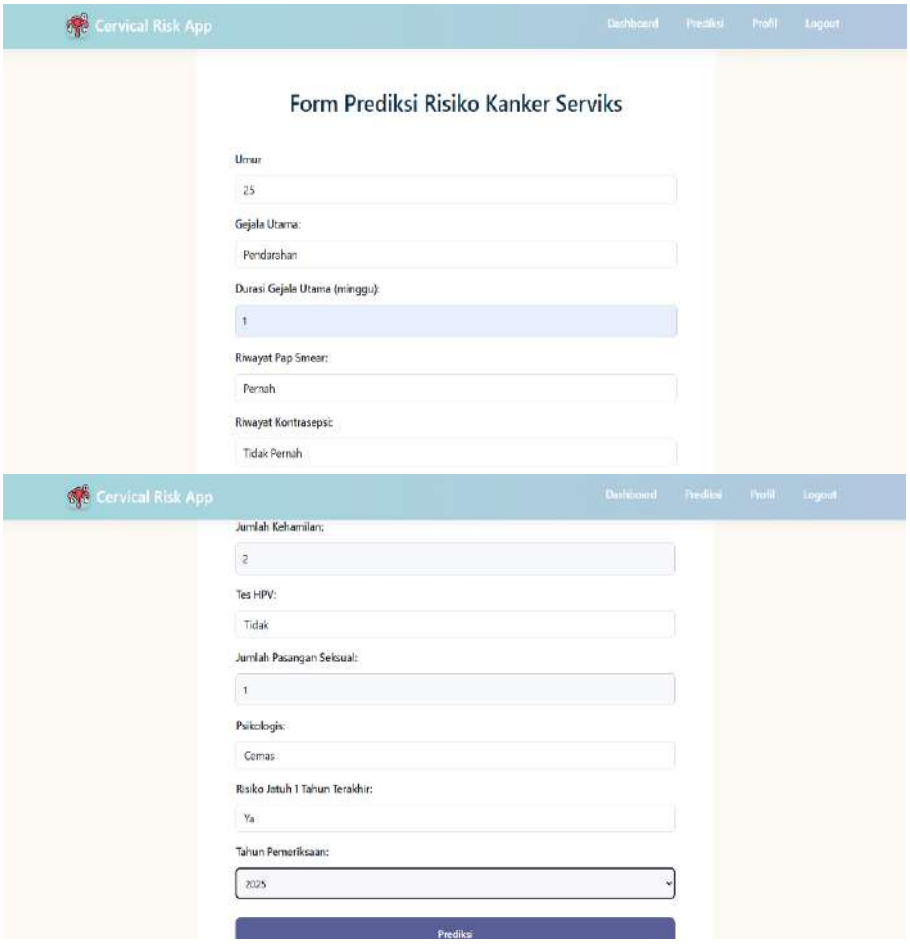
Untuk menguji konsistensi model, dilakukan evaluasi menggunakan teknik *cross validation* dengan nilai fold sebanyak 2, 3, dan 5 yang dikombinasikan dengan berbagai test size. Hasil evaluasi ditunjukkan dalam Tabel 4. Berdasarkan hasil evaluasi di Tabel 4, dapat dilihat bahwa penggunaan cv = 5 dengan test_size 0.2 menghasilkan performa yang paling baik dengan nilai

rata-rata yang diperoleh cukup baik yaitu sebesar 0.8023 dan standar deviasi yaitu 0.0665 dibandingkan dengan cv = 2 atau cv = 3 pada data test_size yang sama. Penggunaan 5 fold ini menunjukkan bahwa model cukup konsisten dalam melakukan prediksi pada data yang berbeda-beda.

Model akhir yang telah dibentuk melalui pipeline kemudian dievaluasi menggunakan *confusion matrix* untuk melihat kinerja klasifikasi secara rinci. Model menghasilkan akurasi sebesar 0.88, precision sebesar 0.88, recall sebesar 0.84, dan F1-score sebesar 0.86. Hasil evaluasi lengkap disajikan dalam Tabel 5. Model yang telah dibangun dan dianalisis hingga mencapai tingkat akurasi sebesar 88%, kemudian diimplementasikan dalam bentuk sistem sederhana berbasis web. Dashboard ini memungkinkan pengguna untuk menginputkan data dan memperoleh hasil prediksi risiko kanker serviks secara *real-time*, serta melihat visualisasi data. Implementasi ini bertujuan untuk mempermudah proses skrining awal secara digital, mempercepat pengambilan keputusan, dan meningkatkan efisiensi layanan deteksi dini. Tampilan dari dashboard dapat dilihat pada Gambar 6.

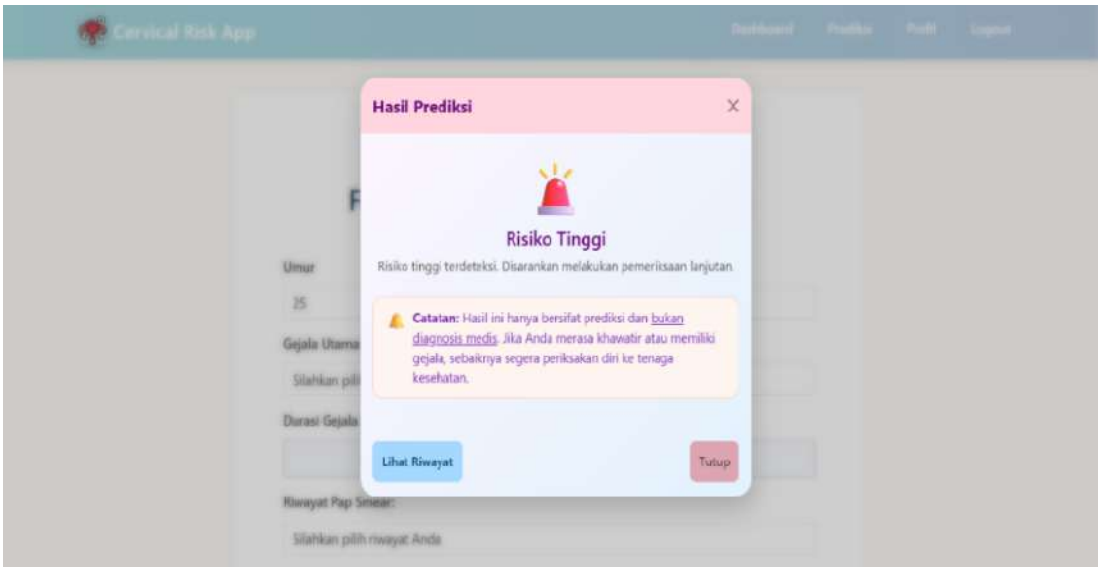
Tabel 5 Performa Kinerja Model

Kelas	Precision	Recall	F1-score	Support
0	0.89	0.73	0.80	11
1	0.88	0.95	0.91	22
Accuracy			**0.88**	**33**
Macro Avg	**0.88**	**0.84**	**0.86**	**33**
Weighted Avg	**0.88**	**0.88**	**0.88**	**33**



Gambar 6 Form Prediksi

Dapat dilihat Gambar 6 menampilkan antarmuka sistem yang dirancang untuk memprediksi risiko kanker serviks berdasarkan data yang diinputkan. Pengguna dapat memasukkan informasi seperti usia, gejala utama, durasi gejala, riwayat pap smear, penggunaan kontrasepsi, jumlah kehamilan, hasil tes HPV, jumlah pasangan seksual, kondisi psikologis, serta faktor risiko lainnya. Setelah data lengkap diinputkan, pengguna dapat menekan tombol Prediksi untuk memperoleh hasil prediksi risiko. Sistem akan menghasilkan output berupa prediksi risiko kanker serviks yang memetakan data pasien ke dalam kategori tinggi atau rendah, sehingga dapat membantu proses skrining awal secara efisien. Hasil visualisasi dari prediksi tersebut dapat dilihat pada Gambar 7 berikut ini.



Gambar 7 Hasil Prediksi

Model yang digunakan dalam penelitian ini adalah algoritma *Support Vector Machine* (SVM) yang telah dioptimalkan melalui tahapan preprocessing, balancing data, dan reduksi dimensi. Sebagaimana ditampilkan pada Gambar 7, sistem prediksi berbasis web yang dikembangkan menghasilkan visualisasi hasil prediksi risiko dalam bentuk modal pop-up yang interaktif. Dalam contoh ini, sistem mendeteksi bahwa pengguna memiliki risiko tinggi terhadap kanker serviks dan memberikan rekomendasi untuk melakukan pemeriksaan lanjutan. Visualisasi ini juga dilengkapi dengan catatan penting yang menekankan bahwa hasil prediksi bukan merupakan diagnosis medis, melainkan alat bantu skrining awal. Penyajian hasil prediksi dalam format visual ini tidak hanya mempermudah interpretasi bagi pengguna, tetapi juga mempercepat pengambilan keputusan terhadap langkah selanjutnya yang harus dilakukan. Implementasi antarmuka ini bertujuan untuk mendukung proses prediksi dini kanker serviks secara digital dan efisien.

4. KESIMPULAN

Penelitian ini menunjukkan bahwa penerapan algoritma *Support Vector Machine* (SVM) dalam prediksi risiko kanker serviks mampu menghasilkan model yang akurat dan andal. Dengan bantuan teknik SMOTE untuk menyeimbangkan jumlah data antar kelas dan PCA untuk menyederhanakan jumlah fitur, model mampu mencapai akurasi sebesar 88%, dengan precision 0.88, recall 0.84, dan F1-score 0.86. Nilai tersebut menunjukkan bahwa model bekerja dengan baik dalam mengenali pasien yang berisiko tinggi. Uji validasi silang (*cross-validation*) dengan 5 fold dan pembagian data 80:20 juga membuktikan bahwa model cukup stabil dan konsisten. Hasil penelitian ini menunjukkan bahwa penggabungan metode pembelajaran mesin dengan teknik praproses yang tepat dapat meningkatkan akurasi prediksi pada data kesehatan. Sebagai rekomendasi, model ini dapat dikembangkan lebih lanjut dengan jumlah data yang lebih besar dan diintegrasikan dalam sistem digital di fasilitas kesehatan untuk mendukung proses skrining awal kanker serviks secara lebih luas.

DAFTAR PUSTAKA

Admojo, F. T., & Ahsanawati. (2020). Klasifikasi Aroma Alkohol Menggunakan Metode KNN. *Indonesian Journal of Data and Science*, *1*(2), 34–38. https://doi.org/10.33096/ijodas.v1i2.12

Alsmariy, R., Healy, G., & Abdelhafez, H. (2020). Predicting Cervical Cancer Using Machine Learning Methods. *International Journal of Advanced Computer Science and Applications*, *11*(7), 173–184. https://doi.org/10.14569/IJACSA.2020.0110723

Binanto, I., B, J. P. K., & Leokadja, L. (2024). Perbandingan Metode Klasifikasi Random Forest dan Support Vector Machine Terhadap Dataset Resiko Kanker Serviks. *JTRISTE*, *11*(1), 60–66. https://doi.org/10.55645/jtriste.v11i1.507

Cahyaningtyas, C., Manongga, D., & Sembiring, I. (2022). Algorithm Comparison and Feature Selection for Classification of Broiler Chicken Harvest. *Jurnal Teknik Informatika (Jutif)*, *3*(6), 1717–1727. https://doi.org/10.20884/1.jutif.2022.3.6.493

Dzulhijjah, D. A., Herlambang, M. B., & Haifan, M. (2024). Implementasi Framework CRISP-DM untuk Proses Data Mining Aplikasi Credit Scoring PT. XYZ. *Seminar Nasional Sains dan Teknologi "SainTek" Seri I*, *1*(1), 238–251. https://conference.ut.ac.id/index.php/saintek/article/view/2337

Ekawati, F. M., Listiani, P., Idaiani, S., Thobari, J. A., & Hafidz, F. (2024). Cervical Cancer Screening Program in Indonesia: Is It Time for HPV-DNA Tests? Results of a Qualitative Study Exploring the Stakeholders' Perspectives. *BMC Women's Health*, *24*(1), Article ID: 125. https://doi.org/10.1186/s12905-024-02946-y

Fatmawati, A., Latifah, U. B., S, A. S., & Zuhriya, T. K. (2025). Klasifikasi Emosi Teks Pengguna Twitter Menggunakan Metode SVM. *SEMNAS INOTEK (Seminar Nasional Inovasi Teknologi)*, *9*, 1999–2007. https://doi.org/10.29407/hq9jyy12

Hasanah, M. A., Soim, S., & Handayani, A. S. (2021). Implementasi CRISP-DM Model Menggunakan Metode Decision Tree dengan Algoritma CART untuk Prediksi Curah Hujan Berpotensi Banjir. *Journal of Applied Informatics and Computing*, *5*(2), 103–108. https://doi.org/10.30871/jaic.v5i2.3200

Indarti, J. (2023). The Role of Social Obstetrics and Gynecology in the Coverage of Cervical Cancer Screening in the Era of Health Transformation in Indonesia. *Indonesian Journal of Obstetrics and Gynecology*, 198–200. https://doi.org/10.32771/inajog.v11i4.2181

Jalil, A., Homaidi, A., & Fatah, Z. (2024). Implementasi Algoritma Support Vector Machine untuk Klasifikasi Status Stunting pada Balita. *G-Tech: Jurnal Teknologi Terapan*, *8*(3), 2070–2079. https://doi.org/10.33379/gtech.v8i3.4811

Mulla, G. A. A., Demir, Y., & Hassan, M. (2021). Combination of PCA with SMOTE Oversampling for Classification of High-Dimensional Imbalanced Data. *Bitlis Eren Üniversitesi Fen Bilimleri Dergisi*, *10*(3), 858–869. https://doi.org/10.17798/bitlisfen.939733

Oktafiani, R., & Itje Sela, E. (2024). Breast Cancer Classification with Principal Component Analysis and SMOTE Using Random Forest Method and Support Vector Machine. *Article in International Journal of Computer Applications*, *186*(16), 1–8. https://doi.org/10.5120/ijca2024923537

Parman, N. H., Hassan, R., & Zakaria, N. H. (2024). Breast Cancer Prediction Using Support Vector Machine Ensemble with PCA Feature Selection Method. *International Journal of Innovative Computing*, *14*(1), 15–19. https://doi.org/10.11113/ijic.v14n1.461

Pinto, A., Ferreira, D., Neto, C., Abelha, A., & Machado, J. (2020). Data Mining to Predict Early Stage Chronic Kidney Disease. *Procedia Computer Science*, *177*, 562–567. https://doi.org/10.1016/j.procs.2020.10.079

Prasetyo, S. D., Hilabi, S. S., & Nurapriani, F. (2023). Analisis Sentimen Relokasi Ibukota Nusantara Menggunakan Algoritma Naïve Bayes dan KNN. *Jurnal KomtekInfo*, *10*, 1–7. https://doi.org/10.35134/komtekinfo.v10i1.330

Schröer, C., Kruse, F., & Gómez, J. M. (2021). A Systematic Literature Review on Applying CRISP-DM Process Model. *Procedia Computer Science*, *181*, 526–534. https://doi.org/10.1016/j.procs.2021.01.199

Singh, D., Vignat, J., Lorenzoni, V., Eslahi, M., Ginsburg, O., Lauby-Secretan, B., Arbyn, M., Basu, P., Bray, F., & Vaccarella, S. (2023). Global Estimates of Incidence and Mortality of

Cervical Cancer in 2020: A Baseline Analysis of the WHO Global Cervical Cancer Elimination Initiative. *The Lancet Global Health*, *11*(2), e197–e206. https://doi.org/10.1016/S2214-109X(22)00501-0

Studer, S., Bui, T. B., Drescher, C., Hanuschkin, A., Winkler, L., Peters, S., & Müller, K.-R. (2021). Towards CRISP-ML(Q): A Machine Learning Process Model with Quality Assurance Methodology. *Machine Learning and Knowledge Extraction*, *3*(2), 392–413. https://doi.org/10.3390/make3020020

Sulistiyono, M., Pristyanto, Y., Adi, S., & Gumelar, G. (2021). Implementasi Algoritma Synthetic Minority Over-Sampling Technique untuk Menangani Ketidakseimbangan Kelas pada Dataset Klasifikasi. *SISTEMASI*, *10*(2), 445–459. https://doi.org/10.32520/stmsi.v10i2.1303

Sung, H., Ferlay, J., Siegel, R. L., Laversanne, M., Soerjomataram, I., Jemal, A., & Bray, F. (2021). Global Cancer Statistics 2020: GLOBOCAN Estimates of Incidence and Mortality Worldwide for 36 Cancers in 185 Countries. *CA: A Cancer Journal for Clinicians*, *71*(3), 209–249. https://doi.org/10.3322/caac.21660

Sylvia, P. H., Arifin, O., Arpan, A., Permata, R., Handoko, D., & Fitriyah. (2024). Evaluasi Kinerja Algoritma Naïve Bayes, KNN, dan SVM dalam Analisis Sentimen Media Sosial. *Jusikom: Jurnal Sistem Komputer Musi Rawas*, *9*(2), 157–166.

Tangkelayuk, A., & Mailoa, E. (2022). Klasifikasi Kualitas Air Menggunakan Metode KNN, Naïve Bayes, dan Decision Tree. *JATISI (Jurnal Teknik Informatika dan Sistem Informasi)*, *9*(2), 1109–1119. https://doi.org/10.35957/jatisi.v9i2.2048

Triginandri, R., & Subhiyakto, E. R. (2024). Deteksi Dini Cacar Monyet menggunakan Convolutional Neural Network (CNN) dalam Aplikasi Mobile. *Edumatic: Jurnal Pendidikan Informatika*, *8*(2), 516–525. https://doi.org/10.29408/edumatic.v8i2.27625

JISKA (Jurnal Informatika Sunan Kalijaga)
Vol. 11, No. 1, JANUARY, 2026, Pp. 127 – 142
ISSN: 2527 – 5836 (print) | 2528 – 0074 (online)

Comparative Analysis of Hybrid CNN-ViT and CNN for Brain Tumor Classification

Ahmad Fauzi [(1)*], Achmad Lutfi Fuadi [(2)], Agus Heri Yunial [(3)]
Departemen Teknik Informatika, Universitas Pamulang, Tangerang, Indonesia
e-mail : {dosen02621,dosen02524,dosen02525}@unpam.ac.id.
* Corresponding author.
This article was submitted on 5 December 2025, revised on 13 January 2026, accepted on 13 January 2026, and published on 25 January 2026.

Abstract

The automated categorization of brain cancers from MRI is essential for improving diagnostic precision. Traditional Convolutional Neural Networks (CNNs) are proficient in local feature extraction but are constrained in their ability to capture long-range spatial relationships, hence impairing performance on intricate malignancies. We propose a hybrid parallel architecture that merges a CNN with a Vision Transformer (ViT) to combine local and global feature modeling. We assessed our dual-branch model in comparison to a conventional CNN baseline using a curated dataset of 15,000 MRI images categorized into three classes: glioma, meningioma, and pituitary. The hybrid model exhibited enhanced performance, attaining 98.40% accuracy and 0.0783 loss, in contrast to the baseline's 97.40% accuracy and 0.1187 loss. The substantial decrease in misclassifications was validated by additional metrics, such as enhanced recall for the meningioma category. The integration of local and global variables produces a more precise, stable, and generalizable classification framework, demonstrating significant potential as a basis for dependable AI-driven Clinical Decision Support Systems (CDSS) in neuroradiology.

Keywords: Artificial Intelligence, Convolutional Neural Network, Machine Learning, Medical Image Analysis, Vision Transformer

Abstrak

Kategorisasi otomatis kanker otak dari citra MRI sangat penting untuk meningkatkan ketepatan diagnosis. Jaringan Syaraf Konvolusional (CNN) tradisional memiliki kemampuan yang baik dalam mengekstraksi fitur lokal, tetapi terbatas dalam menangkap hubungan spasial jangka panjang, sehingga mengurangi kinerjanya pada kasus keganasan yang kompleks. Kami mengusulkan arsitektur hibrida paralel yang menggabungkan CNN dengan Vision Transformer (ViT) untuk memadukan pemodelan fitur lokal dan global. Model dua cabang ini dievaluasi dan dibandingkan dengan model CNN konvensional menggunakan dataset terkurasi yang berisi 15.000 citra MRI yang diklasifikasikan ke dalam tiga kelas: glioma, meningioma, dan pituitary. Model hibrida menunjukkan peningkatan kinerja yang signifikan, mencapai akurasi 98,40% dan loss 0,0783, dibandingkan dengan model dasar (baseline) yang memiliki akurasi 97,40% dan loss 0,1187. Penurunan besar dalam kesalahan klasifikasi ini divalidasi melalui metrik tambahan, termasuk peningkatan recall untuk kategori meningioma. Integrasi antara variabel lokal dan global menghasilkan kerangka klasifikasi yang lebih akurat, stabil, dan dapat digeneralisasi dengan baik, menunjukkan potensi besar sebagai dasar bagi Sistem Pendukung Keputusan Klinis (Clinical Decision Support Systems/CDSS) berbasis AI yang andal di bidang neuroradiologi.Abstrak dalam bahasa Indonesia ditulis dengan pola yang sama dengan abstrak dalam Bahasa Inggris hanya tidak perlu dimiringkan.

Kata Kunci: Kecerdasan Buatan, Convolutional Neural Network, Pembelajaran Mesin, Analisis Citra Medis

1. INTRODUCTION

The classification of brain tumors using Magnetic Resonance Imaging (MRI) is a crucial process in the development of efficient diagnostic and treatment strategies in the field of neuroradiology (Aggarwal et al., 2023; Fujima et al., 2023a). MRI is the preferred method for visualizing brain

tissue due to its excellent soft tissue contrast, allowing for precise diagnosis of disease disorders (Senan et al., 2022). Manual interpretation of high-dimensional MRI scans is sometimes hampered by their labor-intensive nature and considerable inter-observer variability, leading to diagnostic ambiguity (Alanazi et al., 2022; Omer, 2024). Therefore, advances in automation techniques for image processing are becoming increasingly important to improve current clinical practice (Dai et al., 2025).

The development of deep learning technology has triggered a paradigm shift in medical image analysis by presenting an innovative methodology for image classification (Balamurugan & Gnanamanoharan, 2023; Xie, 2023). Convolutional Neural Network (CNN) is a fundamental architecture in this field, highly adept at hierarchical feature extraction and capable of mimicking the diagnostic accuracy of humans (Jamali et al., 2024; Liu et al., 2024; Parulian, 2025). Nevertheless, CNN faces significant obstacles related to intrinsic local bias. This limits its ability to understand long-term spatial relationships, which is something important for distinguishing tumor subtypes that are visually similar but histologically different (Hasan et al., 2025; Touvron et al., 2022).

To overcome these limitations, Vision Transformer (ViT) emerged as a new method that breaks down images into a series of patches and uses self-attention mechanisms to efficiently represent global spatial dependencies (Ruthven et al., 2023; Wibowo et al., 2025). ViT shows potential in better interpretability and prediction consistency between patches than classic CNN designs; nevertheless, ViT is characterized by enormous data needs and weaker local bias (Khatun et al., 2025).

Given the strengths and weaknesses of both architectures, CNN–ViT hybrid research has been on the rise. This hybrid model combines CNN's advantages in local feature extraction and ViT's capabilities in modeling global spatial relationships (Alayón et al., 2023; Emara et al., 2025; Zhang et al., 2023). This combination has been shown to improve classification accuracy compared to using CNN or ViT separately (Touvron et al., 2022). In the context of brain tumor classification, recent research shows that the CNN–ViT hybrid model provides more precise, more reliable, and more clinically interpretable predictions (Bukhari, 2024; Liu et al., 2023; Murugesan et al., 2025; Yu et al., 2024). However, there is still a significant research gap. Most hybrid studies today use sequential pipelines or partial integrations that have not fully leveraged the synergies between local CNN extraction and the global context of ViT. In addition, comparative evaluation of a robust CNN baseline with a uniform experimental setting is still very limited.

Based on the above observations, this study does not aim to introduce a new CNN-ViT architecture. Instead, it focuses on a controlled comparative evaluation between conventional CNNs and parallel CNN-ViT hybrid models under identical experimental conditions. Although the dual-flow CNN-ViT architecture has been explored in previous studies, there is still limited empirical evidence regarding its actual performance gains when compared to robust CNN baselines using the same dataset, preprocessing pipeline, and training protocols. Therefore, this work emphasizes performance comparisons, resiliency analysis, and computational trade-offs rather than architectural novelty.

2. METHODS

2.1 Dataset Acquisition and Characterization

This study uses the public brain MRI dataset from the Multi Cancer Dataset in the Kaggle repository (Naren, 2024). From this dataset, a subset of 15,000 T1-weighted axial images was selected with a balanced distribution among three categories of histologically confirmed tumors: glioma, meningioma, and pituitary tumors, each of 5,000 images. The selection of this subset is based on the need to ensure class balance, which is critical for the stability of the training and evaluation of multi-class classification models.

The dataset was then stratified into three parts: training (n = 9,600; 64%), validation (n = 2,400; 16%), and testing (n = 3,000; 20%). These divisions do not overlap to guarantee that the model is tested on data that has not been seen at all, thus realistically measuring generalization capabilities. This sharing strategy also follows best practices in the evaluation of deep learning models on medical data to avoid overfitting bias.

It is important to note that the Multi Cancer Dataset does not provide explicit patient-level identifiers. As a result, the dataset separation strategy is carried out at the image level using cascading random separation. As a result, the study could not fully guarantee that slices originating from the same patient did not appear in different subsets (training, validation, and testing). These limitations are inherent in the structure of the dataset and are recognized as a potential source of data leaks, which have also been reported in other studies using the same dataset. A visualization of the morphological characteristics of the three tumor types (glioma, meningioma, pituitary) is presented in Figure 1, which provides visual context and helps the reader understand the anatomical differences between the classes.

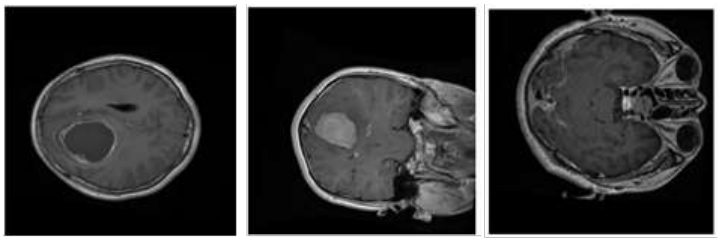


Figure 1 Characteristics of brain tumors: (a) glioma; (b) meningioma; (c) pituitary tumor

2.2 Image Data Pre-processing

A systematic picture pre-processing pipeline was established to guarantee that all model inputs met adequate quality and uniformity standards, thus facilitating an efficient and effective training process (Goyal et al., 2025). Initially, all photos were subjected to pixel intensity equalization with the use of a rescaling factor of 1/255. This approach normalizes pixel values to a range of 0 to 1 (Fujima et al., 2023b), hence improving data uniformity and diminishing model complexity (Goyal et al., 2025). This normalization is essential for expediting model convergence and averting high-intensity pixel values from disrupting the learning process (Goyal et al., 2025). Normalization is formulated as Eq. (1).

$$I_{norm}(x, y) = \frac{I(x, y)}{255} \tag{1}$$

Additionally, all images were scaled to a consistent dimension of 224 × 224 pixels to conform to the input specifications of the CNN architectures, a common procedure in modern image processing applications (Goyal et al., 2025). The data was processed in groups of 32 pictures. In the multi-class classification challenge, labels were one-hot encoded utilizing the 'categorical' mode, a commonly employed method that guarantees a suitable numerical representation for the loss function (Goyal et al., 2025). The division of training and validation data was managed automatically by configuring a validation_split option, which designated 20% of the training data as the validation set. This method enhances data usage and enables more precise evaluation of models during training (Goyal et al., 2025).

It is essential to emphasize that no data augmentation was implemented on the testing set; the photos underwent just normalization. This methodology guarantees that the ultimate model assessment is conducted on unblemished, undistorted data, yielding an accurate evaluation of its performance on novel instances (Goyal et al., 2025). To ensure evaluation consistency, data shuffling was deactivated during the test set processing. This process is crucial for preserving the

stability of evaluation metrics and guaranteeing reproducible, reliable outcomes (Ali et al., 2025; Goyal et al., 2025).

2.3 Data Augmentation Strategy

A systematic data augmentation method was exclusively applied to the training images to boost the models' generalization capabilities and extend the distributional representation of the training data (Krichen, 2023). The augmentation process comprised various geometric modifications: random rotations of up to 20 degrees, random horizontal and vertical shifts of up to 20% of the image dimensions, shear transformations, and random zooming (Keng & Merz, 2024; Krichen, 2023). Furthermore, horizontal flipping was utilized to create variations in object orientation, a method recognized for enhancing model robustness against input variations (Dragan et al., 2023). The augmentations were executed in real-time during the training phase via the ImageDataGenerator class. This method eliminates the necessity for explicit storage of augmented images, therefore guaranteeing memory and computational efficiency within the data processing pipeline (Checcucci et al., 2023).

2.4 Architectural Design

This study encompassed the design, execution, and comparative assessment of two separate deep learning systems. The first model is a standard Convolutional Neural Network (CNN), functioning as the experimental baseline to set a performance benchmark. The second is our suggested parallel hybrid CNN–ViT architecture, designed to address the intrinsic constraints of the baseline model.

Both architectures underwent comparable pre-processing pipelines, data partitioning systems, and training hyperparameters to guarantee a fair and rigorous comparison. This method isolates architectural design as the principal variable affecting performance outcomes. The comprehensive methodological architecture of this investigation, encompassing data collection to final model evaluation, is visually encapsulated in Figure 2.

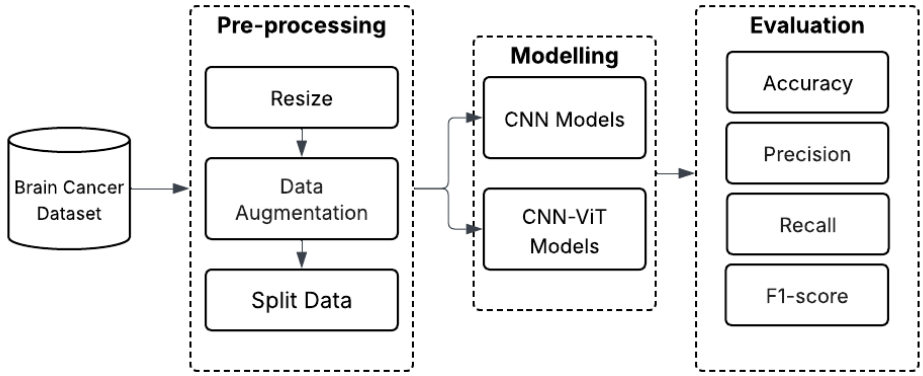


Figure 2 Research Methodology

2.5 Baseline CNN Architecture

The foundational Convolutional Neural Network (CNN) model was developed with the Keras Sequential API, a framework chosen for its efficacy and resilience in prototyping and constructing deep learning models (Pumperla & Cahall, 2022). The architecture, specified in Table 1, consists of two main components: a feature extraction backbone and a classification head. The feature extraction backbone is engineered to handle RGB picture inputs measuring 224 × 224 pixels and comprises five consecutive convolutional blocks. Each block consists of a Conv2D layer for feature extraction, succeeded by BatchNormalization to enhance learning stability and expedite convergence, and a MaxPooling2D layer for spatial down-sampling. The quantity of filters escalates systematically over these blocks (e.g., 32, 64, 128), facilitating the model's capacity for

hierarchical feature extraction, hence capturing patterns of escalating complexity from basic edges to elaborate textures.

Subsequent to the convolutional backbone, the resultant feature maps are flattened into a vector and transmitted to the classification head. This skull consists of three fully connected (Dense) layers. A Dropout layer with a rate of 0.5 is employed as a regularization technique to mitigate overfitting. The architecture concludes with a Dense layer employing a Softmax activation function, producing a probability distribution among the three tumor types, which is calculated using the formula in Eq. (2).

$$P(y = i \mid x) = \frac{e^{z_i}}{\sum_{j=1}^{C} e^{z_j}} \quad (2)$$

Table 1 Architectural Specifications of the Baseline CNN Model.

Layer	Output Size	Number of Filters/Unit	Kernel Size	Pooling
Conv2D + Batch Normalization	224×224	32	3×3	–
MaxPooling2D	112×112	–	–	2×2
Conv2D + Batch Normalization	112×112	64	3×3	–
MaxPooling2D	56×56	–	–	2×2
Conv2D + Batch Normalization	56×56	128	3×3	–
MaxPooling2D	28×28	–	–	2×2
Conv2D + Batch Normalization	28×28	256	3×3	–
MaxPooling2D	14×14	–	–	2×2
Conv2D + Batch Normalization	14×14	512	3×3	–
MaxPooling2D	7×7	–	–	2×2

2.6 Hybrid CNN–ViT Architecture

The suggested hybrid model was developed utilizing the Keras Functional API, a framework that offers the necessary flexibility for designing intricate, non-linear network topologies, including a parallel dual-stream architecture (Ali et al., 2025). Figure 3 demonstrates that this architecture is engineered to concurrently process an input image via two separate yet parallel pathways: a CNN-based feature extractor for local patterns and a Vision Transformer (ViT) backbone for global context modeling. This synergistic methodology, demonstrated to be beneficial in medical image analysis (Emara et al., 2025; Yu et al., 2024), seeks to develop a more nuanced and comprehensive feature representation for the classification of brain tumors (Kim et al., 2025; Tummala et al., 2022).

The initial stream, the CNN component, operates as a specialized local feature extractor. It utilizes the identical foundational architecture outlined in the preceding section, with five consecutive convolutional blocks (Conv2D, BatchNormalization, MaxPooling2D). This branch processes the 224×224 RGB input image and is chiefly tasked with capturing intricate spatial hierarchies, including textures and morphological characteristics, essential for local tumor classification. The second stream operates concurrently, employing a pre-trained Vision Transformer model, namely the ViT-Base-Patch16-224 version from the Hugging Face library. This ViT backbone analyzes the identical normalized input by initially segmenting it into a sequence of 16×16 patches. The patches are subsequently linearly embedded and integrated using positional encodings (Wu et al., 2020) to preserve spatial information, as shown in Eq. (3). In this formulation, E is an embedding matrix, E_{pos} is positional encoding, and N is the number of patches. The generated sequence of tokens is processed by the Transformer's self-attention layers using the relation described in Eq. (4), enabling the model to grasp long-range dependencies and overarching contextual linkages throughout the entire image (Deng et al., 2009). The model was initialized with weights pre-trained on the ImageNet dataset to utilize transferred information and enhance training stability.

$$z_0 = [x_p^1 E; x_p^2 E; \ldots; x_p^N E] + E_{pos} \quad (3)$$

$$Attention(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V \quad (4)$$

$$F_{fusion} = [F_{CNN} \parallel F_{ViT}] \quad (5)$$

The feature vectors obtained from the terminal layers of both the CNN and ViT branches are subsequently amalgamated. The fusion is accomplished by a Concatenation layer that integrates the local and global feature representations into a singular, cohesive vector, as expressed in Eq. (5). The concatenated vector is then processed by a classification head consisting of many Dense layers, which are regularized by L2 regularization and Dropout. The architecture concludes with a Dense layer utilizing a Softmax activation function to generate probability scores for the three tumor classifications: glioma, meningioma, and pituitary, in accordance with known methodologies (Touvron et al., 2022).

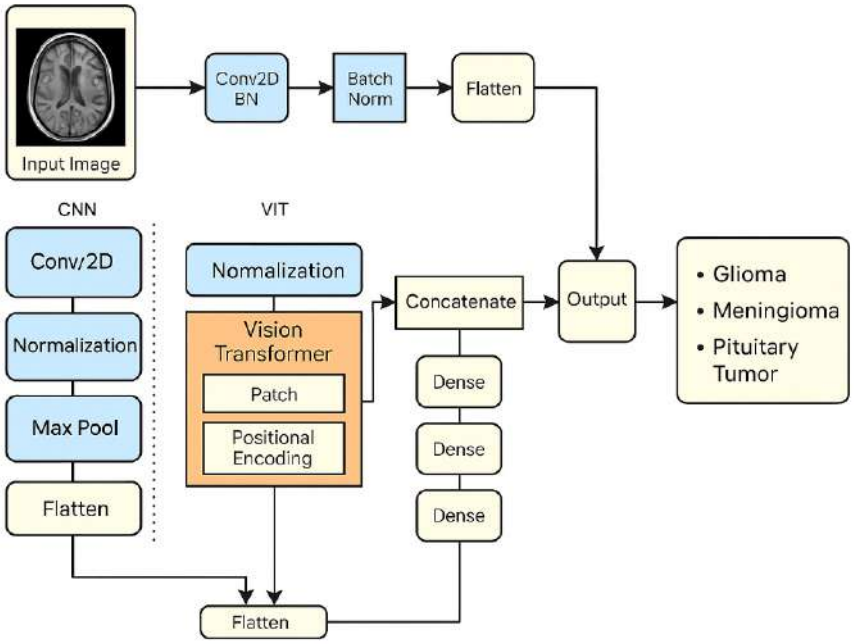


Figure 3 Visualization of the CNN–ViT hybrid architecture

Figure 3 is the architectural diagram of the proposed parallel dual-stream hybrid CNN-ViT model. The workflow commences with a 224×224 pixel MRI input, concurrently processed by two parallel streams. The CNN component focuses on extracting local spatial characteristics, whereas the ViT backbone encompasses global contextual links. The feature vectors generated by each stream are subsequently concatenated and forwarded via a classification head composed of dense layers. The final softmax output layer produces the categorization probabilities for the three tumor categories: glioma, meningioma, and pituitary.

2.7 Vision Transformer Fine-Tuning Strategy

In this study, the Vision Transformer backbone (ViT-Base-Patch16-224) was fully fine-tuned without freezing any of its layers. The model was initialized using ImageNet-21k pretrained weights and trained jointly with the CNN branch in an end-to-end manner. Feature extraction from the ViT branch was performed using the [CLS] token obtained from the last hidden state, which

represents the global image-level representation. The same learning rate was applied to both CNN and ViT components to maintain a unified optimization strategy.

2.8 Training and Evaluation Protocol

The evaluation process was created to objectively analyze and compare the efficacy of the two proposed architectures in classifying three types of brain cancers using MRI scans (Murugesan et al., 2025). The traditional CNN architecture functioned as the performance baseline for evaluating the effectiveness of the suggested hybrid CNN–ViT model (Ullah et al., 2023). A range of callback mechanisms was utilized to enhance the training process and mitigate overfitting. This incorporated EarlyStopping with a patience of five epochs, which automatically terminated training if the validation loss did not improve. The ModelCheckpoint callback was set up to preserve solely the model weights that achieved optimal performance on the validation set. Furthermore, ReduceLROnPlateau was employed to dynamically modify the learning rate, decreasing it when progress in learning plateaued. This amalgamation of callbacks is an established technique for augmenting training stability and promoting model generalization (Murugesan et al., 2025). Both models underwent training for a maximum of 50 epochs, with performance evaluated across the training, validation, and testing datasets (Ullah et al., 2023).

A confusion matrix (Eq. (10)) was produced for a detailed performance evaluation on the hold-out test set. Key classification metrics, specifically precision (Eq. (6)), recall (Eq. (7)), and the F1-score (Eq. (8)), were derived for each tumor type from this matrix (Yohannes & Al Rivan, 2022). The assessment findings from the baseline CNN established a vital benchmark for statistically assessing the performance improvements attained through the synergistic integration of CNN and Vision Transformer components in the proposed hybrid architecture (Ullah et al., 2023). Accuracy is mathematically defined as in Eq. (9).

$$Precision = \frac{TP}{TP + FP} \tag{6}$$

$$Recall = \frac{TP}{TP + FN} \tag{7}$$

$$F1Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{8}$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{9}$$

$$Confusion\ Matrix = \begin{bmatrix} TP & FP \\ FN & TN \end{bmatrix} \tag{10}$$

2.9 Statistical Significance Analysis

To assess whether the performance difference between the baseline CNN and the hybrid CNN–ViT model was statistically significant, a McNemar test was conducted using paired predictions on the same test set. This test is suitable for paired classification comparisons, as it focuses on prediction disagreements rather than overall accuracy values. The McNemar statistic is computed using Eq. (11).

$$X^2 = \frac{(|b - c| - 1)^2}{b + c} \tag{11}$$

3. RESULTS AND DISCUSSION

Two model architectures, a baseline Convolutional Neural Network (CNN) and the proposed hybrid CNN–ViT, were trained and evaluated for a three-class brain tumor classification task (glioma, meningioma, and pituitary tumor). Each model underwent training for a maximum of 50 epochs using 224×224 pixel RGB-formatted MRI scans. Performance was measured on three distinct data subsets: training (9,600 images), validation (2,400 images), and testing (3,000 images). All experiments were executed on the Kaggle Notebook platform, employing dual T4 GPU accelerators to ensure computational efficiency and a uniform training environment.

3.1 Performance of the Baseline CNN Model

The baseline CNN model attained a training accuracy of 99.53%, a validation accuracy of 98.75%, and a final test accuracy of 97.40%. The loss on the test set was noted at 0.1187. Figure 4 displays the training and validation accuracy and loss curves. These figures demonstrate a consistent convergence tendency, despite slight oscillations noted during the initial training epochs. Subsequent examination of the classification metrics indicates a macro-averaged precision of 0.98, a recall of 0.97, and an F1-score of 0.97. The confusion matrix, depicted in Figure 5, reveals that misclassifications primarily transpired between the meningioma and pituitary tumor categories. A total of 78 out of 3,000 test photos were inaccurately classified.

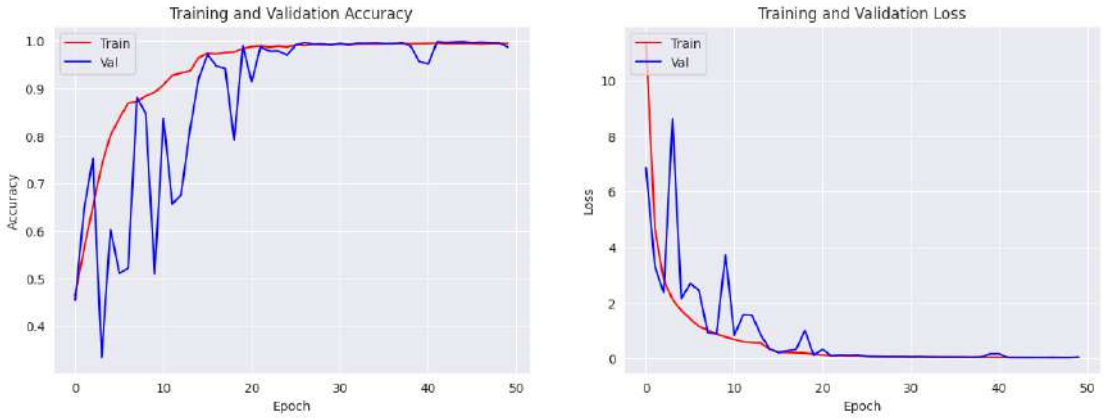


Figure 4 Training & Validation Accuracy

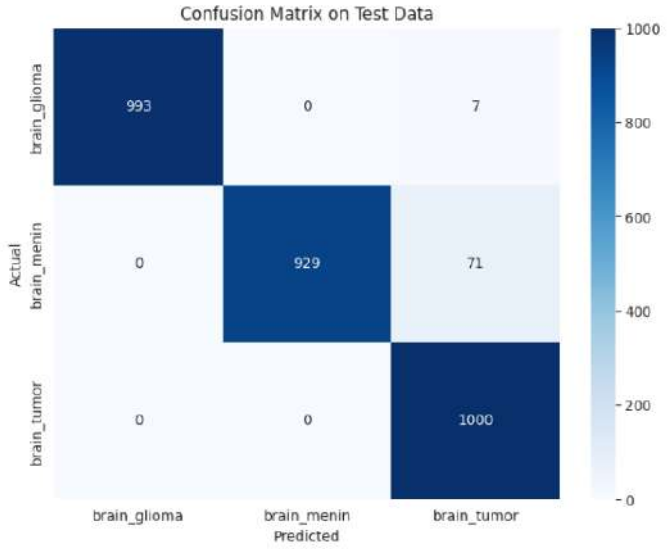


Figure 5 Confusion Matrix Results Baseline CNN Model

The baseline CNN features around 14.7 million trainable parameters and an average training duration of about 126 seconds per epoch. Although these data indicate significant computational efficiency, the model's previously acknowledged shortcoming in modeling global spatial interdependence is demonstrated by the observed discrepancies in per-class recall rates. Table 2 presents a detailed breakdown of the model's complexity and performance measures.

Table 2 Summary of Complexity and Training Performance for the Baseline CNN Model.

Epoch	Akurasi Train	Loss Train	Akurasi Val	Loss Val	Learning Rate	Waktu/Epoch (s)
1	0.4396	14.6324	0.3333	8.6493	5.0×10^{-4}	211
5	0.7988	1.8159	0.6988	1.9758	5.0×10^{-4}	173
10	0.9269	0.6411	0.8929	0.6110	1.5×10^{-4}	173
14	0.9559	0.3708	0.9717	0.2795	4.5×10^{-5}	172
21	0.9803	0.1865	0.9879	0.1548	4.5×10^{-5}	174
28	0.9833	0.1316	0.9925	0.1038	1.35×10^{-5}	174
34	0.9893	0.0887	0.9917	0.0783	1.35×10^{-5}	173
40	0.9923	0.0718	0.9937	0.0661	1.0×10^{-5}	173
47	0.9941	0.0616	0.9962	0.0518	1.0×10^{-5}	173
50	0.9941	0.0550	0.9971	0.0458	1.0×10^{-5}	173

3.2 Performance of the Hybrid CNN–ViT Model

The hybrid CNN–ViT architecture was developed to integrate the local feature extraction capabilities of a CNN with the global context modeling of a Vision Transformer. This was accomplished by incorporating a pre-trained ViT-Base-Patch16-224 model from the Hugging Face library into a parallel stream, markedly improving the model's feature representation capability. The hybrid CNN–ViT model, trained in the same configuration as the baseline, attained a training accuracy of 99.41%, a validation accuracy of 99.71%, and a final test accuracy of 98.40%. The test loss was 0.0783, indicating a significant 34% decrease relative to the baseline CNN. The training curves in Figure 7 illustrate that the hybrid model demonstrated a significantly steadier convergence pattern compared to the baseline. The slight variations in the validation curve specifically indicate enhanced generalization ability.

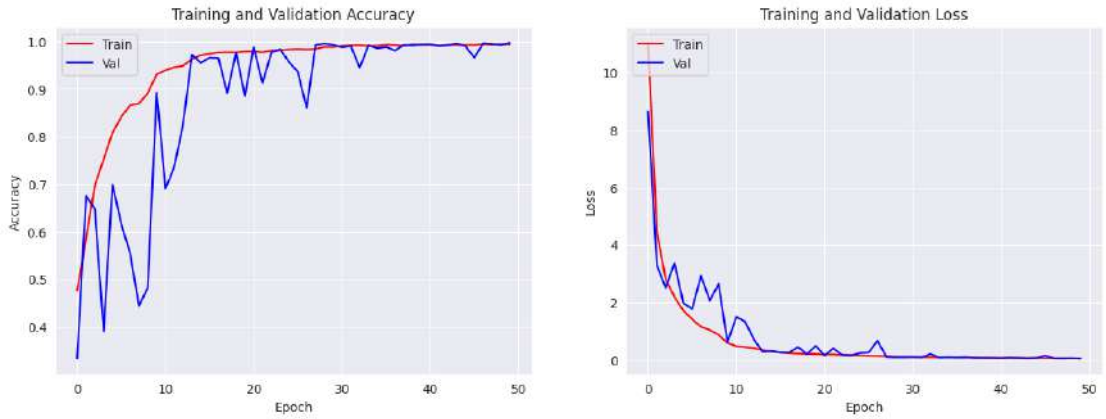


Figure 6 Training & Validation Loss Hybrid CNN-VIT

The hybrid model attained a consistent macro-averaged precision, recall, and F1-score of 0.98, as determined by the assessment of classification measures. The confusion matrix, illustrated in Figure 8, indicates a significant drop in classification errors, with merely 40 misclassified instances out of 3,000 test images, or a 48.7% reduction relative to the baseline. Notable enhancements in accuracy were evident in distinguishing between the meningioma and pituitary tumor categories.

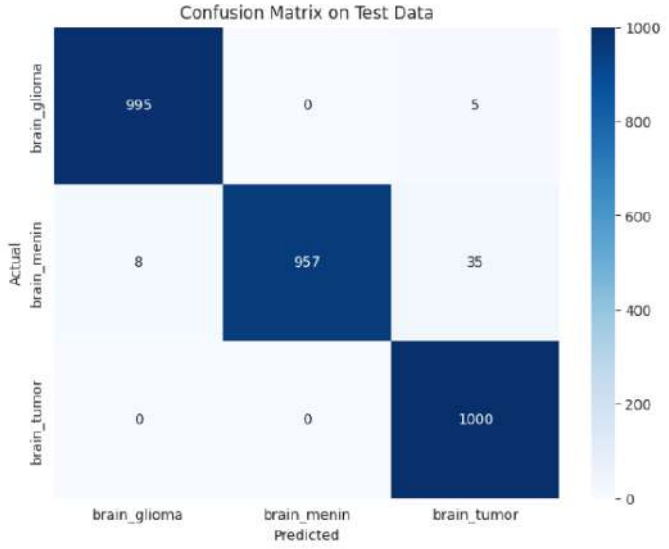


Figure 7 Confusion Matrix Results Hybrid CNN-VIT

Nonetheless, this enhancement in efficiency is coupled with a considerable rise in model complexity. The hybrid CNN–ViT architecture contains around 85 million trainable parameters, primarily attributable to the incorporation of the extensive ViT backbone. The average training duration per epoch thus rose to around 172 seconds, about 36% higher than that of the baseline CNN. Nonetheless, the significant enhancements in accuracy and predictive stability are contended to warrant this computational burden, especially in clinical applications where diagnostic dependability is crucial. Table 3 delineates a summary of the hybrid model's complexity and performance characteristics.

Table 3 Summary of Complexity and Training Performance for the Hybrid CNN–ViT Model.

Epoch	Akurasi Train	Loss Train	Akurasi Val	Loss Val	Learning Rate	Waktu/Epoch (s)
1	0.4396	14.6324	0.3333	8.6493	5.0×10^{-4}	211
5	0.7988	1.8159	0.6988	1.9758	5.0×10^{-4}	173
10	0.9269	0.6411	0.8929	0.6110	1.5×10^{-4}	173
14	0.9559	0.3708	0.9717	0.2795	4.5×10^{-5}	172
21	0.9803	0.1865	0.9879	0.1548	4.5×10^{-5}	174
28	0.9833	0.1316	0.9925	0.1038	1.35×10^{-5}	174
34	0.9893	0.0887	0.9917	0.0783	1.35×10^{-5}	173
40	0.9923	0.0718	0.9937	0.0661	1.0×10^{-5}	173
47	0.9941	0.0616	0.9962	0.0518	1.0×10^{-5}	173
50	0.9941	0.0550	0.9971	0.0458	1.0×10^{-5}	173

3.3 Statistical Significance Analysis

To evaluate the statistical significance of the performance differences between the CNN model and the CNN–ViT hybrid, a McNemar test was performed based on prediction pairs on the same test data with reference to Equation (9). Based on the confusion matrix, the CNN model produced a total of 78 misclassifications, dominated by meningioma errors that were misclassified as tumors (71 cases), while the CNN–ViT hybrid model resulted in 48 errors with a lower error distribution in the same class. Prediction pair analysis showed that the CNN–ViT hybrid model managed to correct 38 errors previously made by CNN (b = 38), while generating only 8 new errors that did not appear in the CNN model (c = 8). By substituting the values b and c into Equation (9), the statistical value χ^2 is obtained as 18.28 with a degree of freedom of one, which results in a p-

value < 0.001. These results show that the performance improvements achieved by the CNN–ViT hybrid model are statistically significant and are not caused by mere random variation.

3.4 Comparative Analysis Baseline CNN and Hybrid CNN–ViT

A thorough comparative analysis was done to evaluate the efficacy of the proposed hybrid architecture between the baseline CNN and the hybrid CNN–ViT models. The comparison included a thorough array of performance parameters, including accuracy, loss, precision, recall, and the F1-score, evaluated throughout the training, validation, and hold-out test sets. In addition to predicted accuracy, the investigation examined critical technical factors, including the overall count of trainable parameters and the average training duration per session. These factors offer a comprehensive assessment of the trade-offs among model efficacy, computing efficiency, and scalability. Table 4 summarizes the findings of this head-to-head comparison, offering a quantitative basis for the ensuing debate.

Table 4 Comparative Summary of Performance and Complexity for Baseline and Hybrid Models.

Evaluation Aspect	Baseline CNN	Hybrid CNN–ViT
Training Accuracy	99.53%	99.41%
Validation Accuracy	98.75%	99,71%
Test Accuracy	97.40%	98.40%
Test Loss	0.1187	0.0783
Precision (average)	0.98	0.98
Recall (average)	0.97	0.98
F1-Score (average)	0.97	0.98
Number of Prediction Errors	78	40
Number of Parameters	±14,7 million	±85 million
Training Time/Epoch	±126 seconds	±172 seconds

This paper presents a thorough assessment of an innovative hybrid CNN–ViT architecture aimed at overcoming ongoing difficulties in the automated categorization of brain tumors using MRI data. The results validate that the suggested hybrid model exhibits a substantial and measurable performance superiority compared to a traditional CNN baseline. This result provides robust empirical support for the theoretical assertion that synergistically integrating the local feature extraction abilities of CNNs with the global context modeling of Vision Transformers constitutes a very effective approach (Ishrak et al., 2025). The subsequent sections will analyze these findings, addressing their theoretical and practical consequences, and will conclude with a summary of the study's shortcomings and potential directions for further research.

Table 5 Comparison with Related Studies on Brain Tumor Classification

No.	Study	Model Architecture	Dataset	Accuracy
1	Tummala et al. (2022)	ViT Ensemble (B/16, B/32, L/16, L/32)	Figshare Brain MRI	98.7%
2	Ullah et al. (2023)	Enhanced CNN (VGG16, VGG19, ResNet101, InceptionV3)	Public Brain MRI	97.0%
3	Ali et al. (2025)	ResNet50	Kaggle Brain MRI	99.88%
4	Emara et al. (2025)	Unified CNN–ViT (HViT-CNN)	Multi-domain MRI (Brain)	98.4%
5	This Study	Parallel CNN–ViT (Dual-Stream)	Multi Cancer MRI	98.4%

The selected works encompass CNN-based, Vision Transformer-based, and hybrid CNN–Transformer architectures to provide a balanced contextual comparison with the proposed

method. It is important to note that variations in reported performance may arise from differences in dataset characteristics, network design, and evaluation protocols. Accordingly, this comparison is intended to highlight architectural trends and relative performance rather than to assert absolute superiority.

Based on the comparisons presented in Table 5, it can be observed that both the Vision Transformer, CNN-based approaches, and the CNN–Transformer hybrid architecture have shown competitive performance in the classification of brain tumors based on MRI images. The study by Tummala et al. (2022) emphasizes the power of global representation through the ViT ensemble, while Ali et al. (2025) and Ullah et al. (2023) show that an optimized CNN is still capable of achieving high accuracy on certain datasets. On the other hand, a hybrid approach such as that proposed by Emara et al. (2025) indicates the potential for the integration of local and global features within a single unified framework. In line with these findings, the results of this study show that the parallel CNN–ViT architecture is able to achieve performance comparable to the current approach, while maintaining classification stability and error reduction between classes. The difference in performance achievement between studies is influenced by the variation in the dataset, the number of classes, and the evaluation protocol used, so this comparison is intended to provide an empirical context for the position of this research in the existing research landscape.

3.5 Theoretical Implications

This study's primary conclusion offers strong empirical support for a fundamental theoretical principle in computer vision: the combined integration of local and global feature extractors produces a more effective and comprehensive data representation. Our hybrid model, which combines a CNN's ability to capture intricate local information with a ViT's capability to model extensive spatial dependencies, surpassed the performance of the standalone CNN architecture. The model's capacity to distinguish between physically identical tumor types, including meningioma and pituitary, is particularly clear, as indicated by a nearly 50% decrease in misclassification errors. This outcome provides robust empirical validation for the assertion made by Ishrak et al. (2025), which contends that hybrid models thrive specifically due to their integration of these two complementary feature extraction paradigms. Moreover, the efficacy of this parallel integration substantiates the findings of other scholars that the amalgamation of Transformer-based and convolutional techniques represents a highly promising and theoretically robust avenue for the progression of medical image analysis (Avcı, 2025; Ullah et al., 2023).

3.6 Practical Implications

This research illustrates the feasibility of a high-performance model that reconciles accuracy with deployability. The proposed hybrid CNN-ViT, exhibiting a test accuracy of 98.40% and a consistently elevated F1-score of 0.98, serves as a more dependable instrument for prospective clinical application than the baseline. Although this performance enhancement entails greater computing complexity, the trade-off is probably warranted in a clinical setting where diagnostic reliability is essential, a perspective aligned with the findings of (Tabassum & Nunavath, 2024).

Furthermore, in comparison to previous high-precision models, our methodology presents a unique benefit. It circumvents the intricate, resource-demanding ensembling methods necessitated by certain models (Ma et al., 2025) and the operational difficulties associated with multi-stage systems (So-yun Park et al., 2025), offering a more efficient, end-to-end solution. Thus, the architecture established in this research can provide a solid basis for the forthcoming generation of AI-enhanced Clinical Decision Support Systems (CDSS). This corroborates the assertion by (Hossain et al., 2025) that these hybrid models are positioned to catalyze substantial, concrete advancements in healthcare and medical diagnostics.

3.7 Limitations and Future Directions

Notwithstanding the encouraging findings, this study possesses multiple limitations that delineate explicit opportunities for subsequent research. A significant limitation is the model's considerable

computational burden, a challenge observed in other high-performance hybrid methodologies (Park et al., 2025). Consequently. Consequently, future research should prioritize model optimization strategies (e.g., pruning, quantization) and validation across varied, multi-institutional datasets to guarantee clinical reliability. Moreover, the model's therapeutic value could be substantially enhanced by broadening its application to more specific tasks, such as detailed glioma grading or semantic segmentation. Simultaneously, the implementation of explainability (XAI) methodologies is essential for clarifying the model's decision-making process. This will enhance its credibility for clinical adoption, a vital element for implementing such models in practice, as indicated by (Chibuike & Yang, 2024). Addressing these issues will be essential for actualizing the complete potential of hybrid models in practical diagnostic applications (Hossain et al., 2025).

4. CONCLUSIONS

This study introduced and validated an advanced computational method for brain tumor classification by a comparative examination of a conventional Convolutional Neural Network (CNN) and an innovative parallel hybrid CNN–Vision Transformer (ViT) architecture. The main goal was to improve classification accuracy by combining the local feature extraction abilities of CNNs with the global spatial representation strengths of ViTs. The experimental findings indicated that the suggested hybrid CNN–ViT architecture consistently surpassed the traditional CNN baseline across all primary assessment parameters. The hybrid model attained a final test accuracy of 98.40%, exceeding the baseline CNN's 97.40%, and demonstrated enhanced validation stability along with a significant decrease in inter-class confusion.

This research demonstrates that incorporating a ViT into a parallel CNN framework markedly boosts spatial modeling and augments the model's generalization skills on novel data. This method increases computational complexity, although the significant improvements in predicted accuracy and reliability provide a strong rationale for its use in critical clinical settings. This study significantly contributes to the advancement of deep learning-based diagnostic assistance systems in oncological radiology. It also establishes a novel trajectory for the investigation of parallel architectures that adeptly integrate local and global domains for enhanced medical image representation.

REFERENCES

Aggarwal, K., Manso Jimeno, M., Ravi, K. S., Gonzalez, G., & Geethanath, S. (2023). Developing and Deploying Deep Learning Models in Brain Magnetic Resonance Imaging: A Review. *NMR in Biomedicine*, *36*(12), Article ID: e5014. https://doi.org/10.1002/nbm.5014

Alanazi, M. F., Ali, M. U., Hussain, S. J., Zafar, A., Mohatram, M., Irfan, M., AlRuwaili, R., Alruwaili, M., Ali, N. H., & Albarrak, A. M. (2022). Brain Tumor/Mass Classification Framework Using Magnetic-Resonance-Imaging-Based Isolated and Developed Transfer Deep-Learning Model. *Sensors*, *22*(1), Article ID: 372. https://doi.org/10.3390/s22010372

Alayón, S., Hernández, J., Fumero, F. J., Sigut, J. F., & Díaz-Alemán, T. (2023). Comparison of the Performance of Convolutional Neural Networks and Vision Transformer-Based Systems for Automated Glaucoma Detection with Eye Fundus Images. *Applied Sciences*, *13*(23), Article ID: 12722. https://doi.org/10.3390/app132312722

Ali, R. R., Yaacob, N. M., Alqaryouti, M. H., Sadeq, A. E., Doheir, M., Iqtait, M., Rachmawanto, E. H., Sari, C. A., & Yaacob, S. S. (2025). Learning Architecture for Brain Tumor Classification Based on Deep Convolutional Neural Network: Classic and ResNet50. *Diagnostics*, *15*(5), Article ID: 624. https://doi.org/10.3390/diagnostics15050624

Avcı, D. (2025). A New Pes Planus Automatic Diagnosis Method: ViT-OELM Hybrid Modeling. *Diagnostics*, *15*(7), Article ID: 867. https://doi.org/10.3390/diagnostics15070867

Balamurugan, T., & Gnanamanoharan, E. (2023). Brain Tumor Segmentation and Classification Using Hybrid Deep CNN with LuNetClassifier. *Neural Computing and Applications*, *35*(6), 4739–4753. https://doi.org/10.1007/s00521-022-07934-7

Bukhari, M. T. (2024). Efficacy of Lightweight Vision Transformers in Diagnosis of Pneumonia. *International Journal of Clinical Nephrology*, *3*(1). https://doi.org/10.37579/2834-5142/020

Checcucci, E., Piazzolla, P., Marullo, G., Innocente, C., Salerno, F., Ulrich, L., Moos, S., Quarà, A., Volpi, G., Amparore, D., Piramide, F., Turcan, A., Garzena, V., Garino, D., De Cillis, S., Sica, M., Verri, P., Piana, A., Castellino, L., … Porpiglia, F. (2023). Development of Bleeding Artificial Intelligence Detector (BLAIR) System for Robotic Radical Prostatectomy. *Journal of Clinical Medicine*, *12*(23), Article ID: 7355. https://doi.org/10.3390/jcm12237355

Chibuike, O., & Yang, X. (2024). Convolutional Neural Network–Vision Transformer Architecture with Gated Control Mechanism and Multi-Scale Fusion for Enhanced Pulmonary Disease Classification. *Diagnostics*, *14*(24), Article ID: 2790. https://doi.org/10.3390/diagnostics14242790

Dai, Y., Lian, C., Zhang, Z., Gao, J., Lin, F., Li, Z., Wang, Q., Chu, T., Aishanjiang, D., Chen, M., Wang, X., Cheng, G., Huang, R., Dong, J., Zhang, H., & Mao, N. (2025). Development and Validation of a Deep Learning System to Differentiate HER2-Zero, HER2-Low, and HER2-Positive Breast Cancer Based on Dynamic Contrast-Enhanced MRI. *Journal of Magnetic Resonance Imaging*, *61*(5), 2212–2220. https://doi.org/10.1002/jmri.29670

Deng, J., Dong, W., Socher, R., Li, L.-J., Kai Li, & Li Fei-Fei. (2009). ImageNet: A Large-Scale Hierarchical Image Database. *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 248–255. https://doi.org/10.1109/CVPR.2009.5206848

Dragan, P., Merski, M., Wiśniewski, S., Sanmukh, S. G., & Latek, D. (2023). Chemokine Receptors—Structure-Based Virtual Screening Assisted by Machine Learning. *Pharmaceutics*, *15*(2), Article ID: 516. https://doi.org/10.3390/pharmaceutics15020516

Emara, H. M., El-Shafai, W., Soliman, N. F., Algarni, A. D., El-Samie, F. E. A., & Mahmoud, A. A. (2025). A Unified Vision Transformer and Convolutional Neural Network Framework for Multi-Domain Cancer Classification. In *Research Square*. https://doi.org/10.21203/rs.3.rs-6633290/v1

Fujima, N., Kamagata, K., Ueda, D., Fujita, S., Fushimi, Y., Yanagawa, M., Ito, R., Tsuboyama, T., Kawamura, M., Nakaura, T., Yamada, A., Nozaki, T., Fujioka, T., Matsui, Y., Hirata, K., Tatsugami, F., & Naganawa, S. (2023a). Current State of Artificial Intelligence in Clinical Applications for Head and Neck MR Imaging. *Magnetic Resonance in Medical Sciences*, *22*(4), 401–414. https://doi.org/10.2463/mrms.rev.2023-0047

Fujima, N., Kamagata, K., Ueda, D., Fujita, S., Fushimi, Y., Yanagawa, M., Ito, R., Tsuboyama, T., Kawamura, M., Nakaura, T., Yamada, A., Nozaki, T., Fujioka, T., Matsui, Y., Hirata, K., Tatsugami, F., & Naganawa, S. (2023b). Current State of Artificial Intelligence in Clinical Applications for Head and Neck MR Imaging. *Magnetic Resonance in Medical Sciences*, *22*(4), 401–414. https://doi.org/10.2463/mrms.rev.2023-0047

Goyal, A. K., Pandey, M., Singh, D. P., Choudhary, J., & Haripriya, R. (2025). FedContrast: A Contrastive Learning Framework to Mitigate Client Drift Under Statistical Heterogeneity in Federated Multi-Class Soil Classification. In *Research Square*. https://doi.org/10.21203/rs.3.rs-6774369/v1

Hasan, M. A., Haque, F., Sarker, H., Abdullah, R., Roy, T., Taaha, N., Arafat, Y., Patwary, A. K., Ahsan, M., & Haider, J. (2025). Mulberry Leaf Disease Detection by CNN-ViT with XAI Integration. *PLOS One*, *20*(6), Article ID: e0325188. https://doi.org/10.1371/journal.pone.0325188

Hossain, Z., Hossain, M. E., Ahmed, N., Kabir, M. F., & Hossain, I. S. (2025). Evaluating the Performance of Vision Transformers and Convolutional Neural Networks for Hostile Image Detection. *Indonesian Journal of Advanced Research*, *4*(1), 111–130. https://doi.org/10.55927/ijar.v4i1.13681

Ishrak, M. F., Rahman, M. M., Joy, M. I. K., Tamuly, A., Akter, S., Tanim, D. M., Jawar, S., Ahmed, N., & Rahman, M. S. (2025). Vision Transformer Embedded Feature Fusion Model with Pre-Trained Transformers for Keratoconus Disease Classification. *Emerging Science Journal*, *9*(2), 1037–1075. https://doi.org/10.28991/ESJ-2025-09-02-027

Jamali, A., Roy, S. K., Hong, D., Lu, B., & Ghamisi, P. (2024). How to Learn More? Exploring Kolmogorov–Arnold Networks for Hyperspectral Image Classification. *Remote Sensing*, *16*(21), Article ID: 4015. https://doi.org/10.3390/rs16214015

Keng, M., & Merz, K. M. (2024). Eliminating the Deadwood: A Machine Learning Model for CCS Knowledge-Based Conformational Focusing for Lipids. *Journal of Chemical Information and Modeling*, *64*(20), 7864–7872. https://doi.org/10.1021/acs.jcim.4c01051

Khatun, R., Chatterjee, S., Bert, C., Wadepohl, M., Ott, O. J., Semrau, S., Fietkau, R., Nürnberger, A., Gaipl, U. S., & Frey, B. (2025). Complex-Valued Neural Networks to Speed-Up MR Thermometry During Hyperthermia Using Fourier PD and PDUNet. *Scientific Reports*, *15*(1), Article ID: 11765. https://doi.org/10.1038/s41598-025-96071-x

Kim, J. W., Khan, A. U., & Banerjee, I. (2025). Systematic Review of Hybrid Vision Transformer Architectures for Radiological Image Analysis. *Journal of Imaging Informatics in Medicine*, *38*(5), 3248–3262. https://doi.org/10.1007/s10278-024-01322-4

Krichen, M. (2023). Convolutional Neural Networks: A Survey. *Computers*, *12*(8), Article ID: 151. https://doi.org/10.3390/computers12080151

Liu, S., Yue, W., Guo, Z., & Wang, L. (2024). Multi-Branch CNN and Grouping Cascade Attention for Medical Image Classification. *Scientific Reports*, *14*(1), Article ID: 15013. https://doi.org/10.1038/s41598-024-64982-w

Liu, Z., Tong, L., Chen, L., Jiang, Z., Zhou, F., Zhang, Q., Zhang, X., Jin, Y., & Zhou, H. (2023). Deep Learning Based Brain Tumor Segmentation: A Survey. *Complex & Intelligent Systems*, *9*(1), 1001–1026. https://doi.org/10.1007/s40747-022-00815-5

Ma, W., Chen, R., Zhang, K., Wu, S., & Ding, S. (2025). Instruct Where the Model Fails: Generative Data Augmentation via Guided Self-Contrastive Fine-Tuning. *Proceedings of the AAAI Conference on Artificial Intelligence*, *39*(6), 5991–5999. https://doi.org/10.1609/aaai.v39i6.32640

Murugesan, G., Nagendran, P., & Natarajan, J. (2025). Advancing Brain Tumor Diagnosis: Deep Siamese Convolutional Neural Network as a Superior Model for MRI Classification. *Brain-X*, *3*(2). https://doi.org/10.1002/brx2.70028

Naren, O. S. (2024). *Multi Cancer Dataset*. Kaggle. https://doi.org/10.34740/KAGGLE/DSV/3415848

Omer, A. A. M. (2024). Image Classification Based on Vision Transformer. *Journal of Computer and Communications*, *12*(04), 49–59. https://doi.org/10.4236/jcc.2024.124005

Park, Saeran, Ayana, G., Wako, B. D., Jeong, K. C., Yoon, S., & Choe, S. (2025). Vision Transformers for Low-Quality Histopathological Images: A Case Study on Squamous Cell Carcinoma Margin Classification. *Diagnostics*, *15*(3), 260. https://doi.org/10.3390/diagnostics15030260

Park, So-yun, Ayana, G., Wako, B. D., Jeong, K. C., Yoon, S.-D., & Choe, S. (2025). Vision Transformers for Low-Quality Histopathological Images: A Case Study on Squamous Cell Carcinoma Margin Classification. *Diagnostics*, *15*(3), Article ID: 260. https://doi.org/10.3390/diagnostics15030260

Parulian, O. S. (2025). Efficient Design and Compression of CNN Models for Rapid Character Recognition. *Jurnal Ilmu Komputer dan Informasi*, *18*(1), 127–140. https://doi.org/10.21609/jiki.v18i1.1443

Pumperla, M., & Cahall, D. (2022). Elephas: Distributed Deep Learning with Keras & Spark. *Journal of Open Source Software*, *7*(80), Article ID: 4073. https://doi.org/10.21105/joss.04073

Ruthven, M., Peplinski, A. M., Adams, D. M., King, A. P., & Miquel, M. E. (2023). Real-Time Speech MRI Datasets with Corresponding Articulator Ground-Truth Segmentations. *Scientific Data*, *10*(1), Article ID: 860. https://doi.org/10.1038/s41597-023-02766-z

Senan, E. M., Jadhav, M. E., Rassem, T. H., Aljaloud, A. S., Mohammed, B. A., & Al-Mekhlafi, Z. G. (2022). Early Diagnosis of Brain Tumour MRI Images Using Hybrid Techniques between Deep and Machine Learning. *Computational and Mathematical Methods in Medicine*, *2022*, 1–17. https://doi.org/10.1155/2022/8330833

Tabassum, I., & Nunavath, V. (2024). A Hybrid Deep Learning Approach for Multi-Class Cyberbullying Classification Using Multi-Modal Social Media Data. *Applied Sciences*, *14*(24), Article ID: 12007. https://doi.org/10.3390/app142412007

Touvron, H., Cord, M., & Jégou, H. (2022). DeiT III: Revenge of the ViT. In S. Avidan, G. Brostow, M. Cissé, G. M. Farinella, & T. Hassner (Eds.), *Computer Vision – ECCV 2022* (pp. 516–533). Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-20053-3_30

Tummala, S., Kadry, S., Bukhari, S. A. C., & Rauf, H. T. (2022). Classification of Brain Tumor from Magnetic Resonance Imaging Using Vision Transformers Ensembling. *Current Oncology*, *29*(10), 7498–7511. https://doi.org/10.3390/curroncol29100590

Ullah, Z., Odeh, A., Khattak, I., & Al Hasan, M. (2023). Enhancement of Pre-Trained Deep Learning Models to Improve Brain Tumor Classification. *Informatica*, *47*(6), 165. https://doi.org/10.31449/inf.v47i6.4645

Wibowo, F. A., Yulianto, T., Malun, N. O., Rionaldy, R., Yasin, V., & Siagian, R. C. (2025). Application of Machine Learning Methods for Classification of Gamma and Hadron Signals in High Energy Particle Detection. *Jurnal Ilmu Komputer dan Informasi*, *18*(2), 181–205. https://doi.org/10.21609/jiki.v18i2.1489

Wu, B., Xu, C., Dai, X., Wan, A., Zhang, P., Yan, Z., Tomizuka, M., Gonzalez, J., Keutzer, K., & Vajda, P. (2020). *Visual Transformers: Token-based Image Representation and Processing for Computer Vision*. http://arxiv.org/abs/2006.03677

Xie, X. (2023). Deep Learning-Based Image Classification of MRI Brain Image. *Theoretical and Natural Science*, *27*(1), 46–51. https://doi.org/10.54254/2753-8818/27/20240670

Yohannes, R., & Al Rivan, M. E. (2022). Klasifikasi Jenis Kanker Kulit Menggunakan CNN-SVM. *Jurnal Algoritme*, *2*(2), 133–144. https://doi.org/10.35957/algoritme.v2i2.2363

Yu, Z., Liu, F., & Li, Y. (2024). scTCA: A Hybrid Transformer-CNN Architecture for Imputation and Denoising of scDNA-seq Data. *Briefings in Bioinformatics*, *25*(6), Article ID: bbae577. https://doi.org/10.1093/bib/bbae577

Zhang, Y., Li, Z., Nan, N., & Wang, X. (2023). TranSegNet: Hybrid CNN-Vision Transformers Encoder for Retina Segmentation of Optical Coherence Tomography. *Life*, *13*(4), Article ID: 976. https://doi.org/10.3390/life13040976

9 772527 583007