

ISSN : 2527-5836

e-ISSN : 2528-0074

Vol. 10 No. 2, May 2025

JISKa

Jurnal Informatika Sunan Kalijaga

Jurusan Teknik Informatika
Fakultas Sains dan Teknologi
UIN Sunan Kalijaga Yogyakarta



JISKa Editorial Team

May 2025 Edition

Editor in Chief

Muhammad Taufiq Nuruzzaman, UIN Sunan Kalijaga Yogyakarta, Indonesia

Editorial Board

Aang Subiyakto, UIN Syarif Hidayatullah Jakarta, Indonesia

Agung Fatwanto, UIN Sunan Kalijaga Yogyakarta, Indonesia

Andang Sunarto, UIN Fatmawati Sukarno Bengkulu, Indonesia

Deokjai Choi, Chonnam National University, South Korea

Elyor Kodirov, Opentrons Labworks Inc., United Kingdom

Hamdani, Universitas Mulawarman Samarinda, Indonesia

Muhammad Anshari, Universiti Brunei Darussalam, Brunei Darussalam

Muhammad Syafrudin, Sejong University, South Korea

Nashrul Hakiem, UIN Syarif Hidayatullah Jakarta, Indonesia

Noor Akhmad Setiawan, Universitas Gadjah Mada, Indonesia

Copy Editor and Layout Editor

Sekar Minati, Victoria University of Wellington, New Zealand

Journal Manager and Technical Support

Eko Hadi Gunawan, UIN Sunan Kalijaga Yogyakarta, Indonesia

Muhammad Galih Wonoseto, UIN Sunan Kalijaga Yogyakarta, Indonesia

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. 10, No. 2, MAY 2025

TABLE OF CONTENT

Perbandingan Kinerja Naïve Bayes dan Random Forest dalam Mendeteksi Berita Palsu William William, Teny Handhayani	137-144
Analisis Cluster untuk Pengelompokan Kemampuan Penguasaan ICT Menggunakan K-Means dan Autoencoder Daru Prasetyawan, Rahmadhan Gatra	145-157
Analisis Sentimen Ulasan Pengguna Aplikasi Alfagift Menggunakan Random Forest M. Bagus Prayogi, Gustina Masitoh	158-170
Imputasi Data Temperatur Maksimum Menggunakan Metode Support Vector Regression Isa Kholifatus Sukhna, Brina Miftahurrohmah, Catur Wulandari, Putri Amelia	171-185
Perbandingan Random Forest dan Convolutional Neural Network dalam Memprediksi Peralihan Pelanggan Dewa Adji Kusuma, Atika Ratna Dewi, Andreas Rony Wijaya	186-194
Penggunaan Teknik Transfer Learning pada Metode CNN untuk Pengenalan Tanaman Bunga Agustina Mufidatuzzainiya, Muhammad Faisal	195-206
Perbandingan Sensitivitas Metode SAW, MAUT dan WSM pada Anugerah Mutu Non-Akademik Universitas Muhammad Galih Wonoseto, Muhammad Abu Shaker Hunaif	207-220

Algoritma Random Forest dan Synthetic Minority Oversampling Technique (SMOTE) untuk Deteksi Diabetes	221-234
Nurussakinah Nurussakinah, Muhammad Faisal, Irwan Budi Santoso	
Klasifikasi Penyakit pada Tanaman Berdasarkan Citra Daun Menggunakan Metode Convolutional Neural Network	235-248
Denis Aji Pangestu, Okta Qomaruddin Aziz, Cahyo Crysdian	
Prediksi Kualitas Udara Menggunakan Metode CatBoost	249-258
Mohamad Arif Abdul Syukur, Suhartono Suhartono, Totok Chamidy	

Perbandingan Kinerja Naïve Bayes dan Random Forest dalam Mendeteksi Berita Palsu

William ⁽¹⁾, Teny Handhayani ^{(2)*}

Departemen Teknik Informatika, Universitas Tarumanagara, Jakarta, Indonesia
e-mail : william.535210013@stu.untar.ac.id, tenyh@fti.untar.ac.id.

* Penulis korespondensi.

Artikel ini diajukan 26 Desember 2023, direvisi 30 Maret 2024, diterima 12 April 2024, dan dipublikasikan 31 Mei 2025.

Abstract

Fake news has become a serious problem in today's digital era. The existence of fake news can have various negative impacts, including the spread of misinformation, social unrest, and economic losses. This study compares the performance of Naïve Bayes and Random Forest classification methods in detecting fake news. Both methods were evaluated on a news dataset comprising 44,898 samples. It uses public data from the Kaggle repository. The news samples are represented by four features: title, news content, subject, and news date. This data is then subjected to cleaning, stemming, tokenization, and feature extraction. The results indicate that the Random Forest method outperforms the Naïve Bayes method. The Random Forest method has an accuracy of 99%, while the Naïve Bayes method has an accuracy of 96%. In general, this research demonstrates that the Random Forest method can be a viable alternative for detecting fake news.

Keywords: Naïve Bayes Algorithm, Random Forest, Text Classification, Fake News Detection, Machine Learning

Abstrak

Berita palsu menjadi salah satu masalah yang serius di era digital saat ini. Keberadaan berita palsu dapat menimbulkan berbagai dampak negatif, seperti penyebaran informasi yang salah, keresahan sosial, hingga kerugian ekonomi. Oleh karena itu, diperlukan metode yang efektif untuk mendeteksi berita palsu. Penelitian ini membandingkan kinerja metode klasifikasi Naïve Bayes dan Random Forest dalam mendeteksi berita palsu. Kedua metode tersebut diujicobakan terhadap *dataset* berita yang terdiri dari 44.898 sampel. *Dataset* yang digunakan merupakan data publik dari repositori Kaggle. Sampel berita diwakili empat fitur, yaitu judul, isi berita, tipe berita, dan tanggal berita. Data ini kemudian dilakukan *cleaning*, *stemming*, tokenisasi, dan ekstraksi fitur. Hasil penelitian menunjukkan bahwa metode Random Forest memiliki kinerja yang lebih baik dibandingkan metode Naïve Bayes. Metode Random Forest memiliki akurasi sebesar 99%, sedangkan metode Naïve Bayes memiliki akurasi sebesar 96%. Secara umum, penelitian ini menunjukkan bahwa metode Random Forest dapat menjadi alternatif yang efektif untuk mendeteksi berita palsu.

Kata Kunci: Algoritma Naïve Bayes, Random Forest, Klasifikasi Teks, Deteksi Berita Palsu, Pembelajaran Mesin

1. PENDAHULUAN

Penyebaran berita palsu (*fake news*) di media sosial dan internet telah menjadi masalah yang penting dalam era digital saat ini. Berita palsu dapat menimbulkan dampak negatif yang luas, seperti penyebaran informasi yang salah, polarisasi masyarakat, dan bahkan konflik (Fawaid et al., 2021). Oleh karena itu, penting untuk mengembangkan metode yang efektif untuk mendeteksi berita palsu. Berita palsu menjadi salah satu masalah yang serius di era digital saat ini. Keberadaan berita palsu dapat menimbulkan berbagai dampak negatif, seperti penyebaran informasi yang salah, keresahan sosial, hingga kerugian ekonomi.

Dalam beberapa tahun terakhir, konten *online* telah memainkan peran penting dalam mempengaruhi keputusan dan opini pengguna. Opini seperti ulasan *online* adalah sumber



Artikel ini didistribusikan mengikuti lisensi Atribusi-NonKomersial CC BY-NC sebagaimana tercantum pada <https://creativecommons.org/licenses/by-nc/4.0/>.

informasi utama bagi pelanggan *e-commerce* untuk membantu mendapatkan wawasan tentang produk yang mereka rencanakan untuk dibeli. Dalam beberapa tahun terakhir, konten *online* telah memainkan peran krusial dalam memengaruhi opini dan keputusan pengguna. Opini, seperti ulasan produk, menjadi sumber informasi utama bagi konsumen *e-commerce* untuk mendapatkan wawasan mengenai produk yang akan mereka beli. Namun, perhatian terhadap spam opini telah meluas, tidak hanya terbatas pada ulasan produk, tetapi juga melibatkan penyebaran berita palsu dan artikel yang menyesatkan (Alsharif, 2022). Situs media sosial seperti Google Plus, Facebook, dan Twitter menjadi sumber utama penyebaran berita palsu.

Meskipun permasalahan berita palsu bukan hal baru, mendeteksinya menjadi tantangan kompleks, terutama karena manusia cenderung mempercayai informasi yang menyesatkan. Kemampuan manusia untuk secara manual mengidentifikasi berita palsu terbatas, memerlukan pengetahuan mendalam tentang tipe berita tersebut. Selain itu, dengan sifat terbuka web dan kemajuan dalam ilmu komputer, pembuatan dan penyebaran berita palsu semakin disederhanakan, membuatnya sulit untuk mengukur niat dan dampaknya. Membedakan berita palsu juga terbukti lebih sulit daripada ulasan produk palsu, karena berita palsu dapat menyebar dengan cepat melalui media sosial dan komunikasi mulut ke mulut.

Terdapat banyak metode *machine learning* yang dapat digunakan untuk deteksi berita palsu (Hanum et al., 2024; Lazuardi et al., 2023; Praha et al., 2024). Metode klasifikasi adalah metode yang digunakan untuk mengelompokkan data ke dalam dua atau lebih kelas. Metode klasifikasi yang digunakan untuk memprediksi berita palsu adalah metode Naïve Bayes (Qubra & Saputra, 2024; Santoso et al., 2020) dan metode Random Forest (Ariatmanto & Rifai, 2024). Metode Naïve Bayes adalah metode *machine learning* yang sederhana dan mudah diterapkan. Metode ini menggunakan distribusi probabilitas untuk mengklasifikasikan berita sebagai asli atau palsu. Metode Random Forest adalah metode *machine learning* yang menggabungkan beberapa pohon keputusan untuk meningkatkan akurasi.

Pada beberapa penelitian menunjukkan bahwa metode *machine learning*, terutama *deep learning*, merupakan metode yang efektif untuk deteksi berita palsu (Anand et al., 2023; Arora & Sikka, 2023; Nath et al., 2021). Metode *machine learning* dapat digunakan untuk mengekstrak fitur-fitur dari teks berita yang dapat digunakan untuk membedakan berita palsu dari berita asli. Hasil penelitian juga menunjukkan bahwa kombinasi metode *machine learning* dan *deep learning* dapat menghasilkan akurasi yang lebih tinggi dibandingkan dengan metode *machine learning* saja. Hal ini disebabkan oleh kekuatan masing-masing metode yang dapat saling melengkapi. Beberapa metode *machine learning* yang digunakan yaitu Naïve Bayes, Decision Tree, Random Forest, dan Long Short-Term Memory (LSTM).

Penelitian ini bertujuan untuk membandingkan kinerja algoritma Naïve Bayes dan Random Forest untuk mengklasifikasi berita palsu, dengan menggunakan data dari Kaggle. Hasil penelitian ini diharapkan dapat memberikan informasi yang bermanfaat bagi pengembangan metode deteksi berita palsu. Penelitian ini juga dapat menjadi referensi bagi para peneliti dan pengembang sistem deteksi berita palsu.

2. METODE PENELITIAN

Metode penelitian ini menjelaskan tahapan-tahapan yang dilakukan dalam proses pembangunan dan evaluasi model klasifikasi. Dimulai dari penyajian data yang digunakan, kemudian dijabarkan metode yang diterapkan untuk membersihkan data (*data cleaning*) serta teknik ekstraksi fitur yang relevan. Selanjutnya, hasil dari proses tersebut digunakan sebagai input dalam pengembangan model klasifikasi yang kemudian dievaluasi untuk mengukur performanya. Gambar 1 menggambarkan alur keseluruhan metode penelitian yang digunakan dalam studi ini.

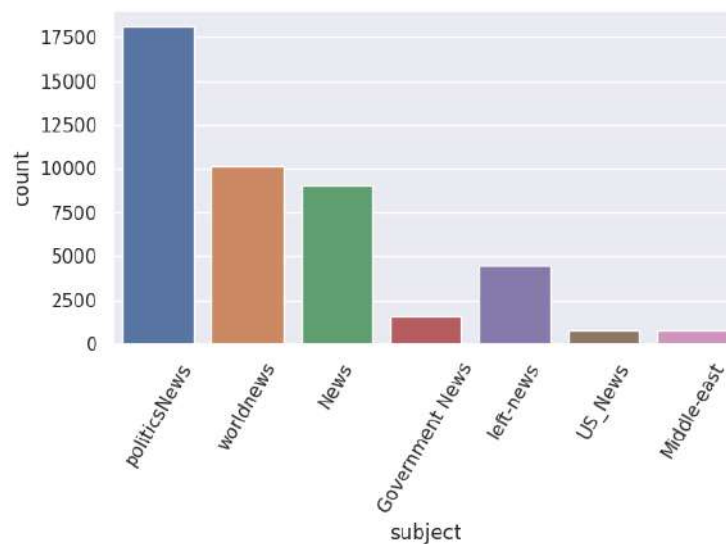


Gambar 1 Metode Penelitian



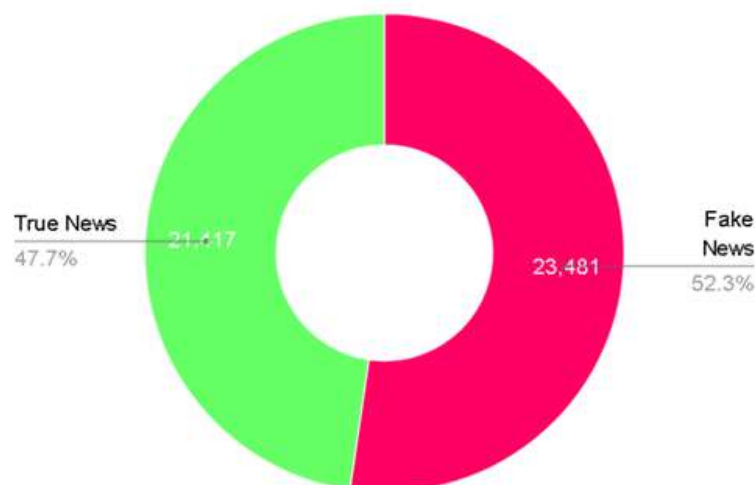
2.1 Data

Dataset yang digunakan merupakan data publik yang dapat diakses melalui tautan <https://www.kaggle.com/datasets/clmentbisaillon/fake-and-real-news-dataset?select=True.csv>. Berita asli dikumpulkan dari Reuters.com (*website* berita). Item berita palsu dikumpulkan dari situs web tidak dapat diandalkan yaitu Politifact (pemeriksaan fakta organisasi di AS) telah bekerja sama dengan Facebook untuk memberantasnya. Penelitian ini berfokus pada artikel berita politik karena saat ini menjadi target utama para *spammer*. Artikel berita dari kategori palsu dan jujur terjadi di *timeline* yang sama, khususnya pada tahun 2016. Setiap artikel panjangnya lebih dari 200 karakter. Data ini memiliki fitur-fitur berupa judul berita, isi berita, tanggal berita, tipe berita, dan label data. Tipe berita dibagi menjadi tujuh tipe yaitu politicsNews, worldnews, News, Government News, left-news, US_News, Middle-east. Distribusi data pada tipe data yang terdiri dari berita palsu dan berita asli dapat dilihat pada Gambar 2. Jumlah berita asli adalah 21.417 data dan berita palsu adalah 23.481, perbandingan jumlah data berdasarkan berita palsu dan asli ini cukup seimbang, visualisasi pembagian *dataset* dapat dilihat pada Gambar 3.



Gambar 2 Distribusi Data Berdasarkan Tipe Berita

Distribusi data



Gambar 3 Perbandingan Jumlah Data Berita Asli dan Berita Palsu



2.2 Pembersihan Data

Data yang dikumpulkan perlu dibersihkan terlebih dahulu. Pembersihan data dilakukan untuk menghilangkan *noise* atau data yang tidak diperlukan. Dalam penelitian ini, pembersihan data dilakukan dengan menghilangkan *stopwords*, melakukan *stemming*, dan melakukan *word tokenizing* dengan menggunakan pustaka (*library*) *nlk*. *Stopwords* adalah komponen standar tugas pemrosesan bahasa alami untuk pengambilan informasi, pengindeksan, pemodelan tipe berita, dan klasifikasi teks. Mereka adalah komponen data yang tidak informatif yang sering kali dihapus selama langkah pra-pemrosesan. Kata-kata ini sering kali muncul di banyak dokumen bahasa alami atau bagian teks yang berbeda dalam sebuah dokumen, namun hanya membawa sedikit informasi tentang bagian teks tersebut. Meskipun para peneliti menggunakan daftar *stopwords* yang tersedia yang berasal dari sumber daya non-teknis, jargon teknis bidang teknik mengandung kata-kata mereka sendiri yang sangat sering dan tidak informatif dan tidak ada daftar *stopwords* standar untuk aplikasi pemrosesan bahasa teknis (Sarica & Luo, 2021).

Stemming adalah proses yang digunakan dalam pra-pemrosesan data untuk mengambil informasi dengan melacak kata-kata yang dibubuhkan kembali ke akarnya. Cara ini sudah digunakan sejak lama dan terbukti menghasilkan tingkat akurasi yang tinggi. Namun efektivitas *stemming* dapat berbeda-beda tergantung formalitas bahasa yang diproses. Misalnya, mungkin tidak banyak metode *stemming* untuk pemrosesan bahasa non-formal. Metode ini sering digunakan untuk meningkatkan akurasi model pengklasifikasi teks (Rianto et al., 2021).

Tokenisasi kata adalah langkah pra-pemrosesan mendasar untuk hampir semua tugas Pemrosesan Bahasa Alami (NLP) (Song et al., 2021). Tahapan ini adalah proses tokenisasi atau pemisahan *string*, yaitu mengubah teks menjadi daftar token. Token dapat dianggap sebagai unit terkecil dari suatu teks yang memiliki makna kontekstual. Misalnya, dalam sebuah kalimat, setiap kata bisa dianggap sebagai satu token. Selanjutnya, jika dilihat dari struktur yang lebih besar, satu kalimat dapat dianggap sebagai token dalam sebuah paragraf (Rai & Borah, 2021). Dalam tokenisasi, kalimat dipecah menjadi unit-unit bermakna yang lebih kecil yang dikenal sebagai token. Token merupakan satuan terkecil yang memiliki arti dalam konteks pengolahan bahasa seperti kata, tanda baca, atau karakter khusus. Tokenisasi dilakukan dengan mencari batasan kata dalam kalimat yaitu titik awal dan akhir dari setiap kata. Proses ini dikenal juga dengan istilah segmentasi, karena bertujuan untuk memisahkan teks menjadi segmen-segmen kecil yang dapat dianalisis lebih lanjut.

2.3 Ekstraksi Fitur

Data yang telah dibersihkan kemudian digunakan untuk mengekstrak fitur. Ekstraksi fitur dilakukan untuk mengubah data menjadi format yang dapat diolah oleh model. Dalam penelitian ini, ekstraksi fitur dilakukan dengan menggunakan metode TF-IDF (Term Frequency-Inverse Document Frequency). TF-IDF telah terintegrasi dalam *library* *sklearn*. TF-IDF adalah ukuran statistik yang mengevaluasi seberapa relevan suatu kata dengan dokumen dalam kumpulan dokumen. Perhitungan TF-IDF dilakukan dengan mengalikan dua metrik: frekuensi kemunculan sebuah kata dalam sebuah dokumen (TF) dan kebalikan dari frekuensi dokumen yang mengandung kata tersebut dalam seluruh dokumen (IDF). Nilai TF-IDF meningkat secara proporsional dengan berapa kali sebuah kata muncul dalam dokumen dan menurun seiring dengan jumlah dokumen dalam korpus yang memuat kata tersebut. Metode ini sering digunakan sebagai faktor pembobotan dalam pencarian pengambilan informasi, penambangan teks, dan pemodelan pengguna.

2.4 Algoritma Klasifikasi

Dalam studi ini, digunakan *library* Scikit-learn (*sklearn*) untuk menerapkan algoritma *machine learning*. Scikit-learn adalah pustaka *machine learning* berbasis Python yang menyediakan berbagai algoritma dan *tools* untuk klasifikasi, regresi, klustering, reduksi dimensi, pemilihan fitur, dan pra-proses data (Zollanvari, 2023). Dalam konteks penelitian ini, *sklearn* digunakan khususnya untuk membangun dan mengevaluasi model klasifikasi secara efisien dan terstruktur.



2.4.1 Algoritma Naïve Bayes

Penelitian ini menggunakan algoritma Naïve Bayes, lebih spesifiknya Multinomial Naïve Bayes, yang merupakan salah satu algoritma populer untuk klasifikasi teks. Algoritma ini mengasumsikan bahwa fitur-fitur tersebut independen secara kondisional, yang mungkin tidak selalu benar dan dapat memengaruhi kinerjanya. Untuk mengatasi masalah ini, para peneliti telah mengusulkan berbagai perluasan pada Multinomial Naïve Bayes. Salah satu perluasan tersebut adalah Extended Multinomial Naïve Bayes Structure, yang menggabungkan ketergantungan fitur menggunakan penduga satu ketergantungan (Solanki & Saxena, 2020).

Multinomial Naïve Bayes adalah pengklasifikasi probabilistik yang digunakan dalam berbagai aplikasi, seperti mendeteksi berita palsu, analisis sentimen, dan identifikasi kejahatan. Dalam pendekatan ini fitur dianggap sebagai variabel diskrit, sedangkan label kelas dapat memiliki lebih dari dua nilai kategori (Yerlekar et al., 2021). Dengan kesederhanaan dan efisiensinya, algoritma ini tetap menjadi pilihan yang kuat untuk tugas-tugas klasifikasi berbasis teks.

2.4.2 Algoritma Random Forest

Random Forest adalah algoritma pembelajaran mesin berbasis *ensemble* yang menggunakan kumpulan *decision tree* untuk meningkatkan akurasi dan ketahanan model. Algoritma ini termasuk dalam metode *bagging*, yaitu menggabungkan beberapa *decision tree* untuk mengurangi *overfitting* dan meningkatkan kinerja generalisasi (Abdullah & Prasetyo, 2020). Random Forest dapat digunakan untuk berbagai tugas seperti klasifikasi, regresi, dan lainnya. Dalam proses pelatihannya, algoritma ini beroperasi dengan membangun sejumlah besar pohon keputusan pada waktu pelatihan. Untuk prediksi klasifikasi, hasil akhir ditentukan berdasarkan modus dari kelas (klasifikasi) atau rata-rata prediksi (regresi) dari pohon individu (Breiman, 2001). Dalam Random Forest, setiap *decision tree* dilatih berdasarkan *subset* acak dari data pelatihan dan subset acak fitur. Keacakan ini membantu mengurangi korelasi antar pepohonan dan meningkatkan keanekaragaman *ensemble*.

2.5 Skema Eksperimen

Sebelum data digunakan untuk pelatihan model, data dibersihkan terlebih dahulu untuk menghilangkan *noise* atau data yang tidak diperlukan. Proses pembersihan data ini meliputi, menghilangkan *stopwords*, *stemming*, dan *word tokenizing*. Setelah data dibersihkan, fitur-fitur yang penting untuk klasifikasi berita palsu diekstraksi menggunakan metode TF-IDF (Term Frequency-Inverse Document Frequency). Data yang telah dibersihkan dan diekstraksi fiturnya kemudian dibagi menjadi dua set data, yaitu set data pelatihan (*training set*) sebesar 80% dan set data pengujian (*testing set*) sebesar 20%. Set data pelatihan digunakan untuk melatih model klasifikasi, sedangkan set data pengujian digunakan untuk mengevaluasi kinerja model klasifikasi. Pembagian data tersebut untuk memastikan bahwa model klasifikasi tidak terlalu dilatih pada set data pelatihan dan dapat generalisasi dengan baik pada data yang belum pernah dilihat sebelumnya. Kinerja model klasifikasi dievaluasi dengan menggunakan beberapa metrik, yaitu akurasi, presisi, *recall*, dan *F1-Score*.

3. HASIL DAN PEMBAHASAN

Penelitian ini membandingkan kinerja dua metode klasifikasi, yaitu Naïve Bayes dan Random Forest, dalam mendeteksi berita palsu menggunakan dataset berisi 44.898 sampel. Sebelum diklasifikasikan, fitur-fitur seperti judul, isi, tipe, dan tanggal berita diproses melalui tahap pembersihan dan ekstraksi. Random Forest menunjukkan akurasi yang lebih tinggi berkat kemampuannya menggabungkan banyak pohon keputusan, menjadikannya efektif untuk deteksi berita palsu. Namun, Naïve Bayes tetap kompetitif karena waktu pelatihan yang cepat dan efisiensi komputasi, sehingga cocok digunakan dalam situasi dengan sumber daya terbatas.

Hasil pada Tabel 1 menunjukkan bahwa kedua model mampu menghasilkan performa klasifikasi yang tinggi dilihat dari metrik akurasi, presisi, *recall*, dan *F1-Score*. Meskipun keduanya



memberikan hasil yang baik, Random Forest unggul dengan akurasi sebesar 99%, sedangkan Naïve Bayes mencapai 96%. Selain itu, nilai *recall*, *precision*, dan *F1-Score* pada Random Forest juga lebih tinggi daripada Naïve Bayes, menandakan kemampuan yang lebih baik dalam mengidentifikasi berita palsu dengan tepat.

Tabel 1 Hasil Evaluasi Klasifikasi

Metrik	Naïve Bayes	Random Forest
Akurasi	96%	99%
Presisi	96%	99%
<i>Recall</i>	97%	99%
<i>F1-Score</i>	96%	99%

Dalam evaluasi model klasifikasi, *recall* mengukur seberapa baik model dapat mendeteksi berita palsu, maka semakin tinggi *recall*, semakin baik model dalam mendeteksi berita palsu. Akurasi mengukur seberapa akurat model dalam memprediksi label berita, baik berita palsu maupun bukan berita palsu, maka semakin tinggi akurasi, semakin akurat model dalam memprediksi label berita. *F1-Score* adalah gabungan dari *recall* dan akurasi, maka semakin tinggi *F1-Score*, semakin baik model dalam mendeteksi berita palsu dan memprediksi label berita secara akurat. Presisi mengukur seberapa akurat model dalam memprediksi berita palsu sebagai palsu, maka semakin tinggi presisi, semakin akurat model dalam memprediksi berita palsu sebagai palsu.

Model Random Forest menghasilkan performa klasifikasi yang sangat baik dengan nilai presisi, *recall*, dan *F1-Score* di atas 98% untuk setiap kelas. Hal ini menunjukkan kemampuan model dalam mengklasifikasikan data positif dan negatif dengan tingkat akurasi yang tinggi. Akurasi keseluruhan mencapai 99%, menandakan bahwa model ini dapat diandalkan untuk tugas klasifikasi pada *dataset* yang digunakan. Model Naïve Bayes juga memberikan hasil yang baik, meskipun sedikit di bawah performa Random Forest. Dengan nilai presisi, *recall*, dan *F1-Score* di sekitar 96%, Naïve Bayes tetap dapat melakukan klasifikasi dengan tingkat akurasi yang tinggi. Akurasi keseluruhan sebesar 96% menunjukkan bahwa Naïve Bayes adalah pilihan yang layak untuk tugas klasifikasi pada *dataset* ini.

Perlu dicatat bahwa *dataset* yang digunakan dalam penelitian ini mayoritas terkait dengan topik politik. Hal ini dapat mempengaruhi generalisasi hasil penelitian terutama ketika diterapkan pada konteks berita palsu di luar ranah politik. Keterkaitan dominan dengan topik politik dapat menyebabkan model lebih terlatih untuk mengenali pola-pola khusus yang mungkin muncul dalam berita politik, sementara mungkin kurang efektif dalam mendeteksi berita palsu dalam konteks topik yang berbeda. Penting untuk diingat bahwa dinamika dan ciri khas berita palsu dapat bervariasi tergantung pada subjek atau topiknya. Oleh karena itu, hasil penelitian ini mungkin tidak secara langsung dapat diterapkan pada berbagai konteks berita palsu di luar topik politik.

Selain itu, untuk meningkatkan generalisasi dan reliabilitas model perlu dilakukan penelitian lebih lanjut pada topik lainnya. Dengan pemahaman yang lebih mendalam tentang batasan dan keunggulan model dalam konteks tertentu, dapat dikembangkan pendekatan yang lebih holistik dan aplikatif untuk mendeteksi berita palsu di berbagai bidang. Dalam pengembangan model untuk mendeteksi berita palsu, penting untuk mempertimbangkan variasi topik dan konteks yang lebih luas agar model dapat memberikan hasil yang konsisten dan dapat diandalkan di berbagai situasi.

4. KESIMPULAN

Berdasarkan hasil eksperimen, dapat disimpulkan bahwa Random Forest memiliki performa yang lebih unggul dibandingkan Naïve Bayes, dengan akurasi sebesar 99%, sementara Naïve Bayes memperoleh nilai akurasi sebesar 96%. Meskipun demikian, Naïve Bayes unggul dari segi kecepatan komputasi, menjadikannya cocok untuk skenario dengan keterbatasan sumber daya. Naïve Bayes dan Random Forest memberikan hasil yang baik dalam melakukan klasifikasi



penyebaran berita palsu dan dapat disesuaikan dengan kebutuhan deteksi. Penelitian ini dapat menjadi landasan untuk pengembangan lebih lanjut dalam deteksi berita palsu selanjutan, termasuk eksplorasi dengan menerapkan teknik *deep learning* untuk mendeteksi berita palsu dalam Bahasa Indonesia.

DAFTAR PUSTAKA

- Abdullah, S., & Prasetyo, G. (2020). Easy Ensemble with Random Forest to Handle Imbalanced Data in Classification. *Journal of Fundamental Mathematics and Applications (JFMA)*, 3(1), 39–46. <https://doi.org/10.14710/jfma.v3i1.7415>
- Alsharif, N. (2022). Fake Opinion Detection in an E-Commerce Business Based on a Long-Short Memory Algorithm. *Soft Computing*, 26(16), 7847–7854. <https://doi.org/10.1007/s00500-022-06806-5>
- Anand, A., Kulkarni, R., & Agrawal, P. (2023). Fake News Identification: An Effective Combined Approach Using ML and DL Techniques. *2023 2nd International Conference on Paradigm Shifts in Communications Embedded Systems, Machine Learning and Signal Processing (PCEMS)*, 1–6. <https://doi.org/10.1109/PCEMS58491.2023.10136087>
- Ariatmanto, D., & Rifai, A. M. (2024). The Impact of Feature Extraction in Random Forest Classifier for Fake News Detection. *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, 8(6), 730–736. <https://doi.org/10.29207/resti.v8i6.6017>
- Arora, Y., & Sikka, S. (2023). Reviewing Fake News Classification Algorithms. In *Proceedings of the Third International Conference on Information Management and Machine Intelligence* (pp. 425–429). Springer. https://doi.org/10.1007/978-981-19-2065-3_46
- Breiman, L. (2001). Random Forests. In *Machine Learning* (Vol. 45, Issue 1, pp. 5–32). Springer. <https://doi.org/10.1023/A:1010933404324/METRICS>
- Fawaid, J., Awalina, A., Krisnabayu, R. Y., & Yulistira, N. (2021). Indonesia's Fake News Detection Using Transformer Network. *6th International Conference on Sustainable Information Engineering and Technology 2021*, 247–251. <https://doi.org/10.1145/3479645.3479666>
- Hanum, A. R., Zetha, I. A., Putri, S. C., Wulandari, R. A., Andina, S. P., Fajrina, J. N., & Yulistira, N. (2024). Analisis Kinerja Algoritma Klasifikasi Teks Bert dalam Mendeteksi Berita Hoaks. *Jurnal Teknologi Informasi dan Ilmu Komputer*, 11(3), 537–546. <https://doi.org/10.25126/jtiik.938093>
- Lazuardi, M. F., Hiunarto, R., Ramadhani, K. F., Noviandi, N., Widayanti, R., & Arfian, M. H. (2023). Hoax News Detection Using Passive Aggressive Classifier and TfidfVectorizer. *Jurnal Teknik Informatika*, 16(2), 185–193. <https://doi.org/10.15408/jti.v16i2.34084>
- Nath, K., Soni, P., Anjum, Ahuja, A., & Katarya, R. (2021). Study of Fake News Detection Using Machine Learning and Deep Learning Classification Methods. *2021 International Conference on Recent Trends on Electronics, Information, Communication & Technology (RTEICT)*, 434–438. <https://doi.org/10.1109/RTEICT52294.2021.9573583>
- Praha, T. C., Widodo, W., & Nugraheni, M. (2024). Indonesian Fake News Classification Using Transfer Learning in CNN and LSTM. *JOIV: International Journal on Informatics Visualization*, 8(3), 1213–1221. <https://doi.org/10.62527/joiv.8.2.2126>
- Qubra, R., & Saputra, R. A. (2024). Classification of Hoax News Using the Naïve Bayes Method. *International Journal Software Engineering and Computer Science (IJSECS)*, 4(1), 40–48. <https://doi.org/10.35870/ijsecs.v4i1.2068>
- Rai, A., & Borah, S. (2021). Study of Various Methods for Tokenization. In *Applications of Internet of Things* (Vol. 137, pp. 193–200). Springer. https://doi.org/10.1007/978-981-15-6198-6_18
- Rianto, R., Mutiara, A. B., Wibowo, E. P., & Santosa, P. I. (2021). Improving the Accuracy of Text Classification Using Stemming Method: A Case of Non-Formal Indonesian Conversation. *Journal of Big Data*, 8(1), Article ID: 26. <https://doi.org/10.1186/s40537-021-00413-1>
- Santoso, H. A., Rachmawanto, E. H., Nugraha, A., Nugroho, A. A., Rosal Ignatius Moses Setiadi, D., & Basuki, R. S. (2020). Hoax Classification and Sentiment Analysis of Indonesian News Using Naive Bayes Optimization. *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, 18(2), 799–806. <https://doi.org/10.12928/telkomnika.v18i2.14744>
- Sarica, S., & Luo, J. (2021). Stopwords in Technical Language Processing. *PLOS ONE*, 16(8), Article ID: e0254937. <https://doi.org/10.1371/journal.pone.0254937>



- Solanki, A., & Saxena, R. (2020). Text Classification Using Self-Structure Extended Multinomial Naive Bayes. In *Handbook of Research on Emerging Trends and Applications of Machine Learning* (pp. 107–129). IGI Global Scientific Publishing. <https://doi.org/10.4018/978-1-5225-9643-1.ch006>
- Song, X., Salcianu, A., Song, Y., Dopson, D., & Zhou, D. (2021). Fast WordPiece Tokenization. *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 2089–2103. <https://doi.org/10.18653/v1/2021.emnlp-main.160>
- Yerlekar, A., Mungale, N., & Wazalwar, S. (2021). A Multinomial Technique for Detecting Fake News Using the Naive Bayes Classifier. *2021 International Conference on Computational Intelligence and Computing Applications (ICCICA)*, 1–5. <https://doi.org/10.1109/ICCICA52458.2021.9697244>
- Zollanvari, A. (2023). Supervised Learning in Practice: The First Application Using Scikit-Learn. In *Machine Learning with Python* (pp. 111–131). Springer International Publishing. https://doi.org/10.1007/978-3-031-33342-2_4



Analisis Cluster untuk Pengelompokan Kemampuan Penguasaan ICT Menggunakan K-Means dan Autoencoder

Daru Prasetyawan ^{(1)*}, Rahmadhan Gatra ⁽²⁾

Pusat Teknologi Informasi dan Pangkalan Data, UIN Sunan Kalijaga, Yogyakarta, Indonesia
e-mail : {daru.prasetyawan,rahmadhan.gatra}@uin-suka.ac.id.

* Penulis korespondensi.

Artikel ini diajukan 16 Februari 2024, direvisi 14 Mei 2024, diterima 25 Juli 2024, dan dipublikasikan 31 Mei 2025.

Abstract

Information and Communication Technology (ICT) skills are essential in today's digital age. However, numerous new students possess varying levels of ICT proficiency and may lack the necessary skills expected by universities. ICT training is essential for enhancing students' ICT skills. Nevertheless, delivering the same training to all students proves to be less effective. Therefore, grouping students' ICT skills is crucial to ensure that the training provided aligns with the fundamental abilities of the students. Cluster analysis is a common method for grouping data. This study employs k-Means and an autoencoder for cluster analysis, with the autoencoder utilized to reduce data dimensions and k-Means to perform the clustering process. The Elbow method is utilized to identify the ideal number of clusters. The optimal number of clusters determined was three. Model evaluation was conducted using the Silhouette coefficient and the Davies-Bouldin Index (DBI). The evaluation results revealed that the combination of k-Means and autoencoder yields superior performance compared to using k-Means alone, as evidenced by a higher Silhouette value and a lower DBI value.

Keywords: *Clustering, K-Means, Autoencoder, ICT, Silhouette, Davies-Bouldin Index*

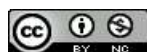
Abstrak

Kemampuan ICT menjadi hal yang penting yang harus dikuasai oleh mahasiswa di era digital saat ini. Akan tetapi banyak mahasiswa baru yang memiliki kemampuan penguasaan ICT yang berbeda-beda, bahkan belum memiliki kemampuan yang dipersyaratkan oleh perguruan tinggi. Pelatihan ICT diperlukan untuk meningkatkan kemampuan ICT bagi mahasiswa. Akan tetapi, dengan memberikan pelatihan yang sama kepada semua mahasiswa menjadi kurang efektif. Oleh karena itu, pengelompokan kemampuan penguasaan ICT ini menjadi sangat penting agar pelatihan yang diberikan kepada mahasiswa sesuai dengan kemampuan dasar yang dimiliki mahasiswa. Analisis *cluster* merupakan salah satu cara yang sering digunakan dalam pengelompokan data. Penelitian ini menggunakan k-Means dan *autoencoder* untuk analisis *cluster*. *Autoencoder* digunakan untuk mereduksi dimensi data. Selanjutnya k-Means melakukan proses *clustering* data tersebut. Metode *Elbow* digunakan untuk menentukan jumlah *cluster* yang optimal. Jumlah *cluster* optimal yang diperoleh sebanyak tiga *cluster*. Evaluasi model dilakukan menggunakan *Silhouette Coefficient* dan *Davies-Bouldin (DBI)*. Dari hasil evaluasi, diketahui bahwa kombinasi k-Means dan *autoencoder* menghasilkan kinerja yang lebih baik dibandingkan hanya dengan menggunakan k-Means saja, yang ditunjukkan dengan *Silhouette score* yang lebih tinggi dan nilai DBI yang lebih rendah.

Kata Kunci: *Clustering, K-Means, Autoencoder, ICT, Silhouette, Davies-Bouldin Index*

1. PENDAHULUAN

Teknologi Informasi atau Information Technology (IT) menjadi salah satu kebutuhan utama untuk mencapai efisiensi dan efektifitas di berbagai bidang. Bagi mahasiswa, teknologi informasi sangat berperan untuk mendukung kegiatan perkuliahan seperti dalam pengerjaan tugas, praktikum, dan proses belajar mengajar itu sendiri. Mahasiswa harus mampu belajar secara mandiri untuk meningkatkan pengetahuan dan keahlian tanpa bantuan dari orang lain, atau yang sering dikenal sebagai *self-direct learning*. Oleh karena itu, kemampuan penguasaan teknologi informasi menjadi hal yang penting bagi mahasiswa, terlebih di era revolusi industri 4.0 yang menuntut kreatifitas



dan fleksibilitas kognitif. Mahasiswa dituntut untuk dapat berpikir dengan sudut pandang yang berbeda, serta mampu mempelajari hal-hal baru yang dapat mengasah kreatifitas.

Salah satu upaya untuk meningkatkan kemampuan penguasaan teknologi informasi adalah dengan pelatihan. Dengan pelatihan diharapkan dapat meningkatkan kemampuan penguasaan teknologi informasi bagi mahasiswa. Akan tetapi, dengan memberikan pelatihan yang sama kepada semua mahasiswa menjadi kurang efektif. Hal ini disebabkan oleh kemampuan dasar tentang teknologi informasi yang dimiliki mahasiswa tidak semua sama. Oleh sebab itu, perlakuan berbeda juga diperlukan untuk memberikan pelatihan terhadap mahasiswa tersebut berdasarkan kemampuan dasar yang sudah dimilikinya sebelumnya. Pengelompokan kemampuan penguasaan ICT ini menjadi tantangan sendiri untuk memberikan pelatihan secara tepat sesuai dengan kemampuan dasar yang dimiliki mahasiswa. Untuk menjawab tantangan tersebut, peneliti mengusulkan metode *clustering* untuk pengelompokan mahasiswa berdasarkan kemampuan penguasaan teknologi informasi, sehingga mahasiswa dapat dikelompokkan sesuai dengan kemiripan/kedekatan tingkat kemampuan penguasaan ICT yang dimilikinya.

Clustering atau analisis *cluster* merupakan proses tak terawasi yang mengelompokkan data yang identik ke dalam yang sama (Behera et al., 2021). *Clustering* memisahkan sekumpulan variabel hasil pengukuran atau perhitungan ke dalam kelompok-kelompok yang homogen, di mana anggota pada setiap kelompok tersebut memiliki kemiripan (Novoselsky & Kagan, 2021). *Clustering* adalah sebuah konsep untuk menentukan pola melalui pemetaan dan analisis data (Velmurugan, 2018). *Clustering* mempelajari kesamaan dari beberapa sampel data, dengan mengelompokkan data ke dalam kelas-kelas atau *cluster* sedemikian rupa sehingga objek-objek dalam suatu *cluster* memiliki kemiripan yang tinggi. *Clustering* mengelompokkan data atau objek menjadi kelompok-kelompok yang serupa berdasarkan karakteristik atau atribut yang dimiliki oleh objek tersebut. Pengelompokan tersebut mengacu pada pemecahan sekumpulan data menjadi kelompok-kelompok menurut kriteria yang sesuai dengan mengasosiasikan sampel data melalui kedekatan, kemiripan, atau ketidakmiripan (McIlhany & Wiggins, 2018). Tujuannya adalah untuk mengidentifikasi pola alami atau struktur dalam data sehingga data yang serupa dikelompokkan bersama dalam satu kelompok. Dalam *clustering* dikenal istilah *distance* atau *dissimilarity* dan *similarity*. Keduanya merupakan dasar dalam mengembangkan algoritma *clustering* yang menggambarkan sejauh mana dua objek atau data memiliki kemiripan (*similarity*) satu sama lain berdasarkan atribut-atribut yang dimiliki objek tersebut. *Distance* biasanya digunakan pada fitur data kuantitatif, sedangkan *similarity* digunakan jika berhadapan dengan fitur data kualitatif (Xu & Tian, 2015).

Salah satu algoritma *clustering* yang terkenal adalah k-Means. Secara teoritis k-Means merupakan metode yang sederhana, cepat menyatu, dan dapat secara efektif menangani kumpulan data berukuran kecil dan menengah (Zhao & Zhou, 2021). Algoritma k-Means relatif sederhana untuk diimplementasikan dan dipahami. K-Means mengelompokkan objek-objek di dalam *dataset* ke dalam kelompok atau *cluster* berdasarkan kemiripan atau jarak antara objek tersebut. K-means cocok untuk kumpulan data dengan jumlah data yang besar dan dimensi fitur yang tinggi, serta ketergantungannya pada data yang rendah (Wu et al., 2021). Algoritma ini mengelompokkan data menjadi *k cluster*, di mana *k* adalah jumlah cluster yang ditentukan sebelumnya oleh pengguna. Hasil dari k-Means adalah pengelompokan titik data ke dalam *cluster* yang jelas dan mudah diinterpretasikan. Ini membuatnya berguna untuk analisis eksploratif dan pemahaman pola dalam data.

K-Means telah banyak digunakan untuk pengelompokan data di berbagai bidang. Di bidang kesehatan, k-Means digunakan untuk mengelompokkan balita berdasarkan kecukupan gizi (Dona & Rifqi, 2022). Pada penelitian tersebut, status gizi pada balita dikelompokkan ke dalam dua *cluster* dengan tujuan untuk membantu pemantauan kebutuhan gizi pada balita di Kabupaten Rokan Hulu. Penelitian selanjutnya adalah penggunaan analisis *cluster* untuk mengelompokkan kabupaten/kota di Provinsi Kalimantan Barat berdasarkan data pengguna alat kontrasepsi menggunakan algoritma k-Means dan k-Medoids (Musfiani, 2019). Di sektor perkebunan, analisis *cluster* digunakan untuk pengelompokan daerah produksi kakao di Provinsi Sulawesi Selatan

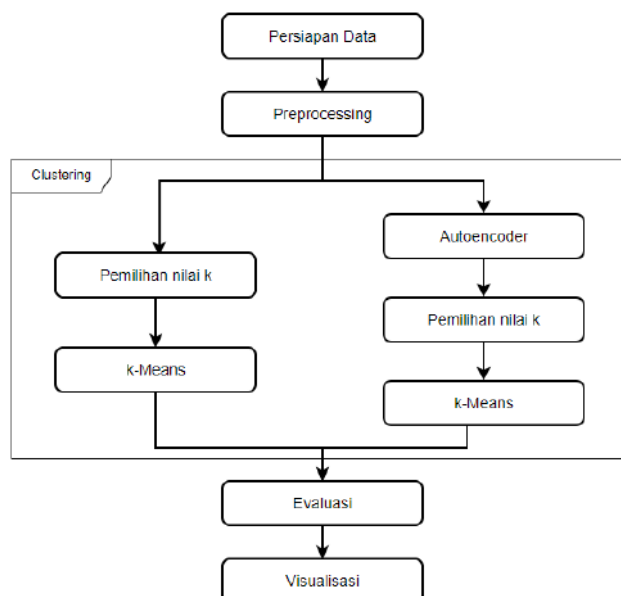


(Abidin et al., 2022). Penelitian tersebut membandingkan algoritma k-Means dan k-Medoids berdasarkan nilai *Davies-Bouldin Index (DBI)* pada *RapidMiner* dan disimpulkan bahwa algoritma k-Means lebih efektif dibandingkan dengan k-Medoids. Di sektor perbankan, algoritma k-Means lebih efektif dan efisien digunakan untuk pengelompokan kredit macet (Fitriani et al., 2023). Di bidang pendidikan, algoritma k-Means digunakan dalam penelitian tersebut untuk mengelompokan siswa berdasarkan prestasi, sehingga dapat ditentukan siswa mana yang dapat masuk ke dalam kelas unggulan (Nur Aziz & Zuliarso, 2022).

Autoencoder adalah jenis arsitektur jaringan saraf tiruan (*neural network*) yang digunakan dalam pembelajaran tak terawasi (*unsupervised learning*). *Autoencoder* dirancang untuk menghasilkan representasi data yang lebih ringkas dan informasi yang lebih penting dengan cara mengompresi dan kemudian mendekompresi data masukan. *Autoencoder* sering digunakan untuk ekstraksi fitur, pengurangan dimensi, *denoising* data, dan berbagai aplikasi dalam analisis data dan pembelajaran mesin. Seiring dengan perkembangan pembelajaran mendalam (*deep learning*), *autoencoder* menjadi yang terdepan dalam pemodelan generatif (Zhai et al., 2018). *Autoencoder* terdiri dari sepasang dua jaringan yang terhubung: *encoder* dan *decoder* (Nugroho et al., 2020). Bagian pertama dari *autoencoder*, yang disebut *encoder*, yaitu lapisan-lapisan yang bertanggung jawab untuk mengambil data masukan dan mengubahnya menjadi representasi yang lebih ringkas. Bagian kedua dari *autoencoder* disebut *decoder*. Bagian ini berisi lapisan-lapisan bertanggung jawab untuk mengambil representasi terkompresi (kode) dari *encoder* dan mendekompresinya untuk menghasilkan rekonstruksi data yang serupa dengan data masukan. Di antara kedua bagian tersebut terdapat sebuah lapisan yang disebut dengan *bottleneck layer*, yaitu lapisan yang berfungsi sebagai representasi terkompresi atau kode dari data. Prinsip utama dari *autoencoder* adalah untuk meminimalkan perbedaan antara data masukan dan rekonstruksi *output* yang dihasilkan oleh *decoder*. Dalam proses pelatihan, *autoencoder* berusaha untuk belajar merepresentasikan data menjadi lebih baik dan lebih ringkas.

2. METODE PENELITIAN

Penelitian dimulai dengan persiapan data, termasuk di dalamnya pengumpulan data dan analisis data. Kemudian dilanjutkan dengan *preprocessing* data dengan melakukan pembersihan data dan normalisasi atau penskalaan data. Selanjutnya data akan digunakan dalam proses *clustering*. Proses *clustering* dilakukan dengan menggunakan dua metode, yaitu k-Means dan k-Means + *autoencoder*. Hasil dari proses *clustering* kemudian dibandingkan dan dievaluasi. Tahapan dalam analisis *cluster* menggunakan k-Means dan *autoencoder* dapat dilihat pada Gambar 1.



Gambar 1 Tahapan dalam Analisis Cluster



2.1 Data dan Sumber Data

Pengumpulan data menjadi langkah awal dalam penelitian, terutama untuk penelitian yang terkait dengan analisis data. Data merupakan sekumpulan fakta yang dapat diamati, diukur, dan diolah. Dalam penelitian ini, data yang digunakan adalah data nilai pretest kemampuan mahasiswa UIN Sunan Kalijaga angkatan tahun 2022. Pengambilan data dilakukan dengan meminta mahasiswa untuk mengerjakan soal pengetahuan dasar ICT yang terdiri dari empat bagian, yaitu tentang Microsoft Word, Microsoft Excel, Microsoft Power Point, dan Internet. Dari hasil *pretest* yang dilakukan diperoleh data sebanyak 3.382. Sampel data *pretest* dapat dilihat pada Tabel 1.

Tabel 1 Sampel Data Nilai *Pretest* ICT

ld	nim	nama	word	excel	power point	internet
1	2210****01	Muh*****udin	56	36	52	56
2	2210****07	Rion*****cky	76	44	68	68
3	2210****09	Arfa*****qi	64	72	60	52
4	2210****37	Imel*****lfa	60	40	52	68
5	2210****49	Muti*****hra	68	32	52	44
...
3378	2210****76	Muh*****ndy	48	28	16	40
3379	2210****79	Nail*****ah	88	88	52	76
3380	2210****94	Pris*****mah	48	24	60	48
3381	2210****27	Ram*****n	76	48	48	68
3382	2210****89	Muh*****hul	84	64	48	40

2.2 Preprocessing

Preprocessing digunakan untuk memastikan bahwa data sudah bersih dan siap digunakan untuk analisis atau pemodelan. *Preprocessing* merupakan proses mengubah data mentah menjadi format yang lebih dimengerti (Agarwal, 2015). Tahap *preprocessing* data pada penelitian ini antara lain pembersihan data dan normalisasi data. Data yang tidak lengkap dapat mengganggu dapat mempengaruhi kualitas analisis dan model yang dibangun. Pembersihan data bertujuan untuk mengidentifikasi, menangani, dan mengatasi masalah di dalam data tersebut. Data yang bersih dan terstruktur dengan baik tentunya akan menghasilkan hasil analisis yang lebih akurat dan model yang lebih andal.

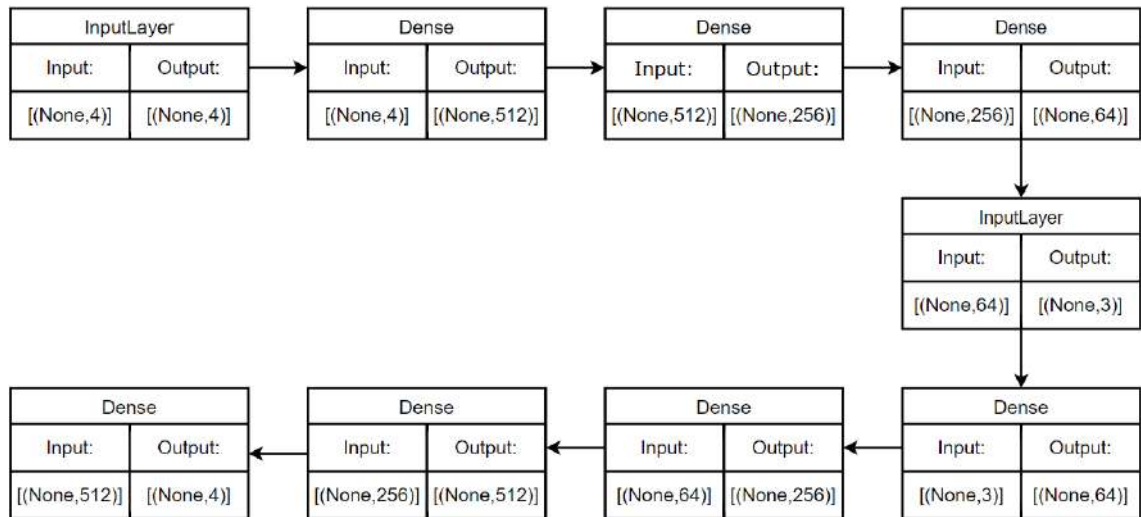
Normalisasi data bertujuan untuk mengubah skala data sehingga memiliki rentang nilai yang seragam. Perbedaan skala antar atribut sangat berpengaruh dalam analisis dan pemodelan pembelajaran mesin. Atribut dengan skala yang lebih besar akan dominan dalam analisis dan pemodelan. Oleh karena itu, skala atribut data harus diseragamkan sebelum digunakan dalam analisis atau pemodelan. Metode normalisasi yang sering digunakan antara lain *min-max normalization*, *robust scalling*, dan *Z-score normalization*. *Z-score normalization*, juga dikenal sebagai *standardization*, yaitu salah satu metode normalisasi yang umum digunakan dalam analisis data dan statistik. *Z-score* dihitung menggunakan nilai rata-rata dan *standar deviasi* (Prihanditya & Alamsyah, 2020). Tujuan utama dari *Z-score normalization* adalah mengubah skala data sehingga memiliki rata-rata (*mean*) bernilai nol dan simpangan baku (*standard deviation*) satu. Dengan kata lain, data yang telah dinormalisasi dengan *Z-score* memiliki distribusi yang berpusat di sekitar nol dan memiliki deviasi standar yang seragam. *Z-score normalization* diformulasikan pada Pers. (1). Di mana $X_{normalized}$ adalah nilai ternormalisasi, X adalah nilai asli dalam *dataset*, μ adalah rata-rata di dalam *dataset*, dan σ adalah simpangan baku di dalam *dataset*.

$$X_{normalized} = \frac{X - \mu}{\sigma} \quad (1)$$



2.3 Pemodelan ANN Autoencoder

Autoencoder digunakan untuk ekstraksi fitur dan mereduksi dimensi, sehingga proses *clustering* menjadi lebih cepat. Seperti dijelaskan sebelumnya bahwa *dataset* yang digunakan memiliki dimensi 3.382×4 , sehingga bagian *encoder* akan menerima input data dengan dimensi 4 pada lapisan input dan diikuti oleh tiga lapisan *dense* dengan jumlah neuron masing-masing 512, 256, dan 64. Bagian *decoder* terdiri dari empat lapisan dengan jumlah neuron masing-masing 64, 256, 512, dan 4. Di antara bagian *encoder* dan *decoder* terdapat lapisan *code* dengan jumlah neuron sebanyak 3. Arsitektur model *autoencoder* dapat dilihat pada Gambar 2.



Gambar 2 Arsitektur Model *Autoencoder*

2.4 Pemilihan Nilai k

K-Means tidak dapat mengetahui jumlah *cluster* yang optimal. Metode *Elbow* dapat digunakan untuk mengoptimalkan jumlah *cluster* pada metode k-Means (Marutho et al., 2018). Langkah yang harus dilakukan dalam menentukan nilai k menggunakan metode *Elbow* adalah menghitung *Sum of Square Error* (SSE) seperti pada Pers. (2) untuk setiap nilai yang akan menjadi kandidat nilai k . SSE menggambarkan seberapa jauh objek-objek di dalam sebuah *cluster* dari pusatnya. Kemudian SSE pada setiap k digambarkan dalam bentuk grafik sehingga membentuk siku pada grafik tersebut. Jumlah *cluster* atau nilai k terbaik akan diambil dari nilai SSE yang mengalami penurunan dan berbentuk siku (Nainggolan et al., 2019). Pada perhitungan SSE n merupakan jumlah data, k merupakan jumlah *cluster*, x_{ij} adalah data ke- i di dalam *cluster* ke- j , dan c_j merupakan pusat *cluster* ke- j .

$$SSE = \sum_{i=1}^n \sum_{j=1}^k (x_{ij} - c_j)^2 \quad (2)$$

2.5 Clustering dengan K-Means

K-Means bertujuan untuk meminimalkan jumlah jarak antara objek-objek di dalam *cluster* dengan pusat *cluster* masing-masing, dan mengelompokkannya secara iteratif (Cui, 2020). Algoritma ini mengelompokkan data menjadi k *cluster*, di mana k adalah jumlah *cluster* yang ditentukan sebelumnya. Langkah-langkah umum dalam algoritma k-Means adalah sebagai berikut:

- 1) Pilih titik awal yang akan berperan sebagai pusat *cluster* awal (*centroid*). Inisialisasi pusat *cluster* ini dapat dilakukan secara acak atau berdasarkan metode khusus.



- 2) Hitung jarak objek data observasi dengan setiap pusat *cluster*. Masukkan data tersebut ke dalam *cluster* berdasarkan jarak terdekat dengan pusat *cluster*.
- 3) Tentukan pusat *cluster* baru untuk setiap cluster dengan menghitung rata-rata jarak antara pusat *cluster* dengan semua data di dalam *cluster* tersebut.
- 4) Ulangi langkah 2 dan 3 hingga tidak ada lagi data yang berpindah *cluster* atau hingga batasan iterasi tertentu tercapai.

2.6 Evaluasi Model Clustering

Evaluasi model dilakukan untuk mengukur kinerja model *clustering*. Metode yang sering digunakan untuk mengevaluasi model *clustering* adalah *Silhouette coefficient* dan Davis-Bouldin Index (DBI). *Silhouette coefficient* menggambarkan seberapa dekat suatu data dengan data lainnya di dalam *cluster* yang sama dibandingkan dengan data pada cluster lainnya. *Silhouette coefficient* memiliki rentang nilai antara -1 sampai 1. Nilai *Silhouette* yang mendekati 1 menunjukkan adanya hubungan yang erat antara objek dengan cluster di mana objek tersebut berada (Yuan & Yang, 2019). *Silhouette coefficient* dinotasikan dengan $S(i)$ untuk setiap data i yang dihitung melalui Pers. (3). Di mana $a(i)$ merupakan rata-rata jarak suatu objek data dengan objek data lainnya di dalam *cluster* yang sama, $b(i)$ adalah rata-rata jarak suatu objek data dengan objek data pada *cluster* yang lain.

$$S(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$

$$= \begin{cases} 1 - \frac{a(i)}{b(i)}, & a(i) > b(i) \\ 0, & a(i) = b(i) \\ \frac{b(i)}{a(i)} - 1, & a(i) < b(i) \end{cases} \quad (3)$$

Davis-Bouldin Index (DBI) adalah suatu metrik evaluasi untuk mengukur kualitas model *clustering*. DBI mengukur seberapa jauh suatu *cluster* dari *cluster* lain dalam hal varian data dan jarak antar-pusat *cluster*. Jumlah *cluster* yang memiliki DBI yang minimal dianggap sebagai jumlah *cluster* yang optimal (Baser & R. Saini, 2015). Nilai DBI yang semakin rendah menunjukkan *cluster* yang terbentuk semakin baik. Nilai DBI dihitung sebagai ukuran rata-rata kemiripan setiap *cluster* dengan *cluster* lain yang paling mirip dengannya. Dalam hal ini, kemiripan diartikan sebagai perbandingan antara jarak antar *cluster* dan jarak *intra-cluster*. Nilai DBI untuk jumlah k *cluster* dapat dihitung dengan Pers. (4). Di mana $\Delta(x_k)$ merupakan dispersi atau jarak sebaran data di dalam *cluster* k dan $\partial(x_i, x_j)$ adalah jarak pusat *cluster* i dengan pusat *cluster* j .

$$DBI = \frac{1}{k} \sum_{i=1}^k \max \left(\frac{\Delta(x_i) + \Delta(j)}{\partial(x_i, x_j)} \right) \quad (4)$$

3. HASIL DAN PEMBAHASAN

3.1 Persiapan Data

Persiapan data dilakukan untuk memastikan data yang digunakan sesuai dengan tujuan pemodelan. Analisis awal dilakukan untuk memahami struktur, distribusi, dan karakteristiknya, serta mengidentifikasi variabel-variabel beserta pemahaman makna masing-masing variabel. Data *pretest* ICT memiliki empat variabel utama yang akan digunakan dalam analisis *cluster*, yaitu nilai word, excel, power point, dan internet. Deskripsi variabel data *pretest* ICT disajikan pada Tabel 2. Variabel word memiliki rata-rata yang lebih tinggi dibandingkan variabel lainnya, sedangkan variabel excel memiliki rata-rata yang paling rendah. Standar deviasi variabel internet lebih kecil dibanding lainnya, artinya sebaran data cenderung berkumpul mendekati rata-rata.



Tabel 2 Deskripsi Data

	Word	Excel	Power Point	Internet
count	3382	3382	3382	3382
mean	64.9533	41.0089	48.7912	58.5795
std	16.1293	16.2034	14.4353	12.0773
min	4	0	4	12
25%	52	28	40	52
50%	68	40	48	60
75%	76	52	60	68
max	100	88	96	96

3.2 Normalisasi Data

Sebelum *dataset* digunakan dalam pemodelan *clustering*, *dataset* harus distandarkan agar semua variabel memiliki rentang data yang sama, sehingga tidak ada variabel yang lebih atau kurang dominan. Normalisasi data dilakukan menggunakan metode Z-Score *normalization*, sehingga semua variabel memiliki rata-rata 0 dan simpangan baku 1. Hasil normalisasi data disajikan pada Tabel 3.

Tabel 3 Sampel Data Nilai *Pretest* ICT

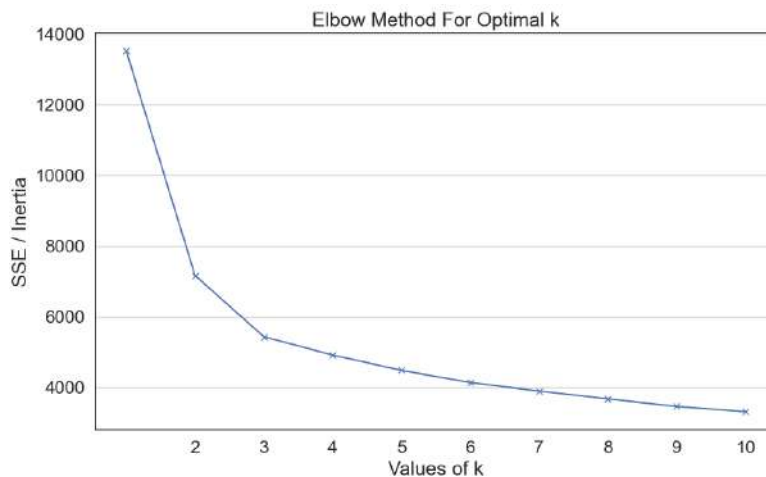
	Word	Excel	Power Point	Internet
0	-0.55518	-0.30917	0.222318	-0.21362
1	0.684986	0.184627	1.330879	0.780126
2	-0.05911	1.912919	0.776599	-0.54486
3	-0.30714	-0.06227	0.222318	0.780126
4	0.188921	-0.55607	0.222318	-1.20736
...
3377	-1.05124	-0.80297	-2.27194	-1.53861
3378	1.429083	2.900514	0.222318	1.442622
3379	-1.05124	-1.04987	0.776599	-0.87611
3380	0.684986	0.431526	-0.05482	0.780126
3381	1.181051	1.419121	-0.05482	-1.53861

3.3 Proses *Clustering* dengan K-Means

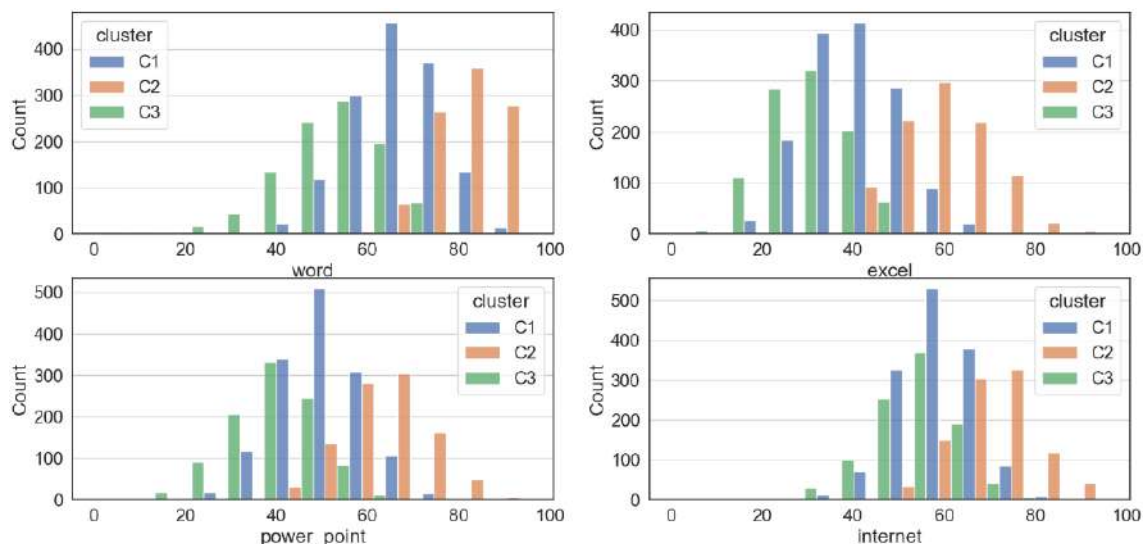
Data hasil normalisasi selanjutnya dapat digunakan dalam proses *clustering*. Dalam proses *clustering*, nilai *k* atau jumlah *cluster* harus ditentukan sebelumnya. Untuk menentukan jumlah *cluster* yang optimal, metode yang digunakan adalah metode *Elbow*. Metode tersebut menghitung nilai SSE untuk setiap nilai *k* yang dicoba. Grafik SSE untuk setiap nilai *k* dapat dilihat pada Gambar 3.

Dari Gambar 4 terlihat bahwa grafik SSE membentuk siku pada *k*=3, sehingga jumlah *cluster* yang digunakan adalah tiga. Setelah jumlah *cluster* ditentukan, selanjutnya proses *clustering* dapat dilakukan menggunakan *dataset* yang sudah dinormalisasi. Proses *clustering* dilakukan menggunakan bahasa pemrograman Python dengan memanfaatkan fungsi *KMeans* yang disediakan oleh pustaka *scikit-learn*. *Scikit-learn* merupakan sebuah pustaka (*library*) dalam bahasa pemrograman Python untuk pembelajaran mesin (*machine learning*) dan tugas-tugas analisis data. Dari proses *clustering* diperoleh tiga *cluster* dengan jumlah anggota pada *cluster* pertama (C1) sebanyak 993 anggota, *cluster* kedua (C2) sebanyak 1.414 anggota, dan *cluster* ketiga (C3) sebanyak 975 anggota. Sebaran data untuk setiap *cluster* hasil proses *clustering* dengan k-Means dapat dilihat pada Gambar 4.





Gambar 3 Grafik Inertia untuk Setiap k



Gambar 4 Sebaran Data Hasil Proses *Clustering* dengan K-Means

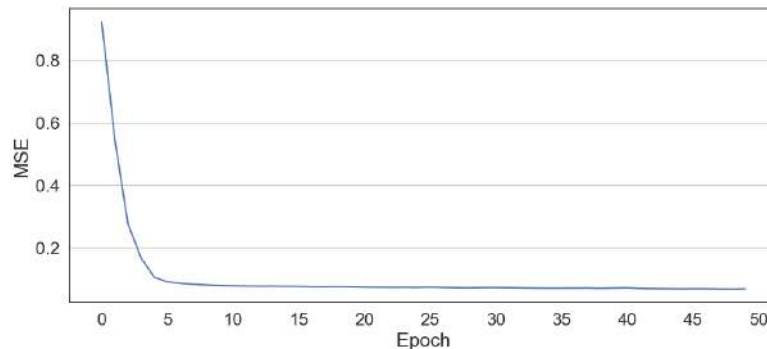
3.4 Pemodelaan *Autoencoder*

Pemodelan *autoencoder* dalam penelitian ini dilakukan menggunakan pustaka Keras dari TensorFlow, yang menyediakan antarmuka tingkat tinggi untuk membangun dan melatih model jaringan syaraf tiruan dengan lebih efisien dalam bahasa pemrograman Python. Pemanfaatan Keras memungkinkan proses perancangan arsitektur *autoencoder* dilakukan secara modular dan fleksibel. Model dilatih selama 50 *epoch* dengan menggunakan fungsi optimasi Adam (*Adaptive Moment Estimation*) yang dikenal memiliki kinerja baik dalam mempercepat konvergensi, serta *loss function* berupa *Mean Squared Error* (MSE) yang bertujuan meminimalkan selisih antara input dan *output* dari *autoencoder*. Performa model selama proses pelatihan ditunjukkan melalui grafik nilai MSE yang dapat dilihat pada Gambar 5.

Setelah proses pelatihan selesai, langkah berikutnya adalah menggunakan model *autoencoder* yang telah dilatih untuk melakukan rekonstruksi terhadap *dataset* asli. Rekonstruksi ini menghasilkan representasi baru dari data yang telah melalui proses kompresi dan dekomposisi oleh *autoencoder*, sehingga diharapkan lebih merepresentasikan informasi penting dari data awal secara efisien. Hasil dari proses rekonstruksi ini kemudian disajikan pada Tabel 4, yang



memperlihatkan bagaimana data telah ditransformasikan oleh *autoencoder* sebelum digunakan dalam tahap *clustering*.



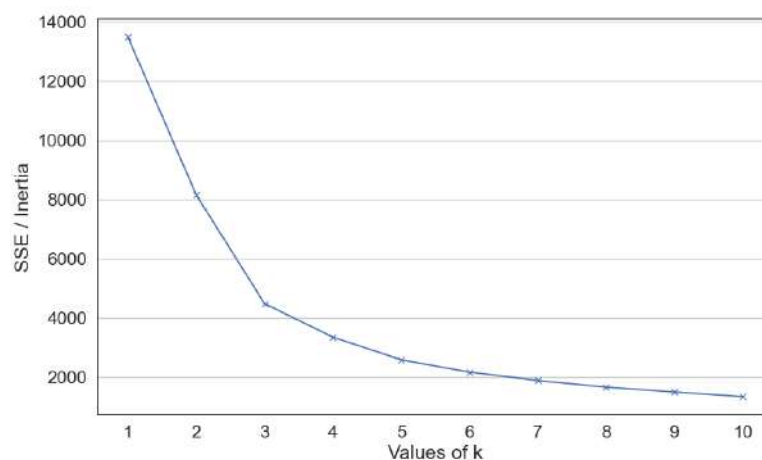
Gambar 5 Grafik MSE Proses Pelatihan Model *Autoencoder*

Tabel 4 Hasil Rekontruksi *Dataset* Menggunakan Model *Autoencoder*

	X1	X2	X3
0	2.055754	1.752693	1.488532
1	2.220366	3.880299	1.876992
2	1.101668	3.291011	3.203516
3	1.769379	1.634185	0.768939
4	2.355839	2.599715	2.975083
...
3377	2.95972	0.899038	3.391938
3378	0.272069	3.586121	1.772966
3379	3.777638	3.445621	2.890459
3380	0.727395	1.381394	0.600213
3381	0.904083	2.745968	3.974567

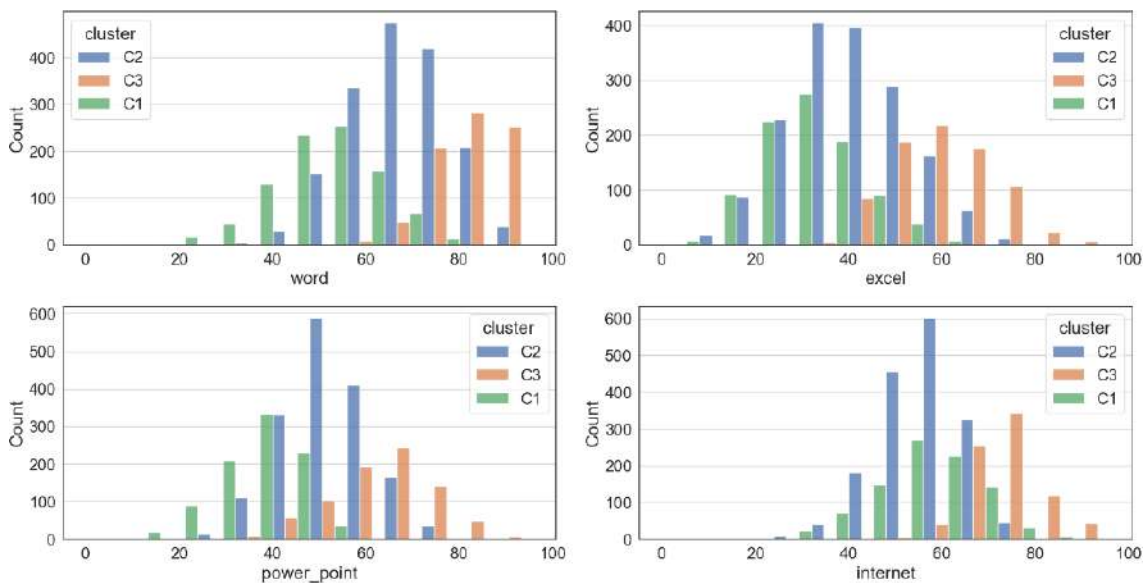
3.5 Clustering dengan Kombinasi K-Means dan *Autoencoder*

Data hasil transformasi menggunakan model *autoencoder* selanjutnya diproses menggunakan k-Means. Pada proses *clustering* yang kedua, *dataset* hanya memiliki tiga variabel. Seperti dalam proses *clustering* sebelumnya, nilai k harus ditentukan dengan mencoba satu persatu sehingga diperoleh nilai k yang optimal. Pemilihan nilai k di dalam proses *clustering* ini juga dilakukan dengan metode *Elbow*. Grafik SSE untuk menentukan nilai k yang optimal pada proses *clustering* menggunakan k-Means dan *autoencoder* dapat dilihat pada Gambar 6.



Gambar 6 Grafik SSE untuk Proses *Clustering* K-Means dan *Autoencoder*





Gambar 7 Sebaran Data Hasil *Clustering* dengan K-Means dan *Autoencoder*

Dari Gambar 6 terlihat bahwa grafik SSE membentuk siku pada $k = 3$, sehingga nilai k yang digunakan dalam *clustering* juga 3. *Cluster* C1 dengan jumlah anggota sebanyak 920 diisi oleh data yang memiliki nilai rendah untuk semua variabel. *Cluster* C2 dengan jumlah anggota sebanyak 1.660 diisi oleh data dengan nilai sedang untuk setiap variabelnya. Sedangkan *cluster* C3 dengan jumlah anggota sebanyak 802 berisi data yang memiliki nilai setiap variabel lebih tinggi dibandingkan pada *cluster* lainnya. Sebaran data berdasarkan hasil *clustering* menggunakan k-Means dan *autoencoder* dapat dilihat pada Gambar 7.

3.6 Evaluasi Model *Clustering*

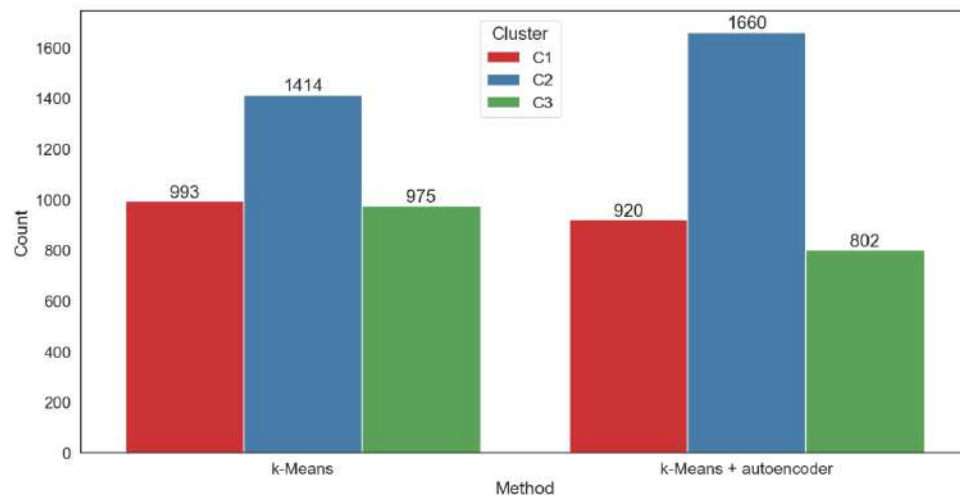
Dari kedua proses *clustering* menggunakan k-Means dan kombinasi k-Means + *autoencoder*, keduanya menghasilkan tiga *cluster* yang dianggap optimal. Perbandingan hasil *clustering* keduanya berdasarkan jumlah anggota setiap *cluster* disajikan pada Gambar 8. Pada proses *clustering* menggunakan kombinasi k-Means dan *autoencoder* menghasilkan nilai SSE yang lebih rendah dibandingkan hanya dengan k-Means, artinya jarak antara titik-titik dengan pusat *cluster* untuk setiap *cluster* pada k-Means + *autoencoder* juga lebih dekat. Pada nilai $k=3$, *clustering* menggunakan k-Means saja menghasilkan inertia sebesar 5.424,77, sedangkan *clustering* dengan kombinasi k-Means dan *autoencoder* menghaslakan SSE sebesar 2.966,64. Perbandingan SSE antara k-Means dan kombinasi k-Means + *autoencoder* dapat dilihat pada Gambar 9.

Validasi selanjutnya dilakukan dengan Silhouette coefficient dan Davis-Bouldin Index (DBI). Skor *silhouette* pada $k = 3$ yang diperoleh model *clustering* dengan kombinasi k-Means dan *autoencoder* lebih besar dibandingkan hanya dengan k-Means saja, yaitu 0,447 berbanding 0,2907. Pada *clustering* tanpa *autoencoder*, nilai *silhouette* terus menurun ketika diujikan untuk setiap nilai $k = 2$ hingga $k = 10$, dan cenderung tidak menemukan nilai k yang optimal di atas $k = 2$. Perbandingan skor *silhouette* yang diperoleh dari proses *clustering* menggunakan k-Means tanpa *autoencoder* dan dengan *autoencoder* dapat dilihat pada Gambar 10.

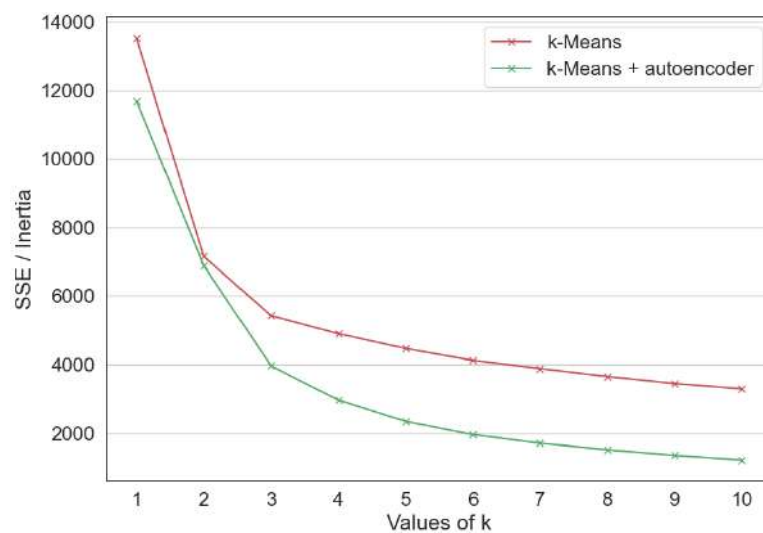
Dari hasil evaluasi model menggunakan Davis-Bouldin Index (DBI) diketahui bahwa model *clustering* kombinasi k-Means dan *autoencoder* lebih kecil dibandingkan hanya menggunakan k-Means saja. Pada nilai $k = 3$, nilai DBI yang dihasilkan sebesar 0,7658, sedangkan *clustering* tanpa *autoencoder* menghasilkan nilai DBI sebesar 1,1711. Jika semua kandidat jumlah *cluster* $k = 2$ sampai $k = 10$ divalidasi menggunakan DBI, model *clustering* dengan kombinasi k-Means dan *autoencoder* juga lebih baik dibandingkan tanpa *autoencoder*. Perbandingan nilai DBI pada



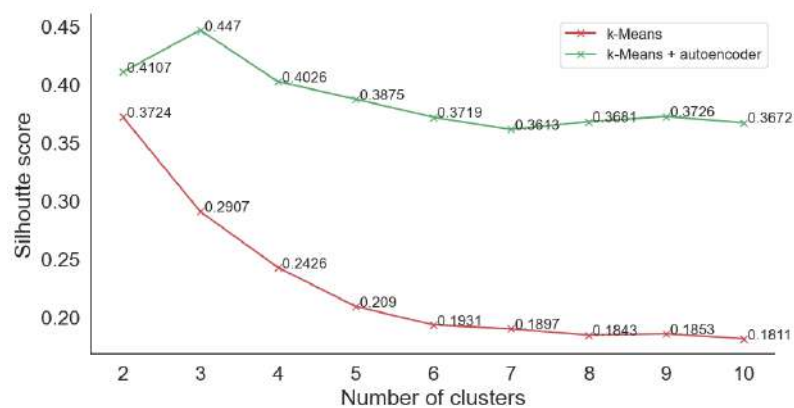
model *clustering* menggunakan k-Means tanpa *autoencoder* dan dengan *autoencoder* dapat dilihat pada Gambar 11.



Gambar 8 Perbandingan Hasil *Clustering* Berdasarkan Jumlah Setiap *Cluster*

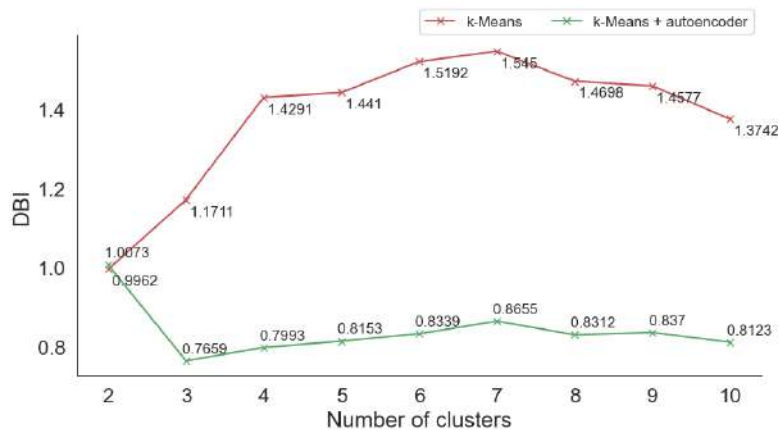


Gambar 9 Perbandingan SSE untuk Nilai $k = 1$ Sampai $k = 10$



Gambar 10 Perbandingan Nilai Silhouette





Gambar 11 Perbandingan Indeks Davies-Bouldin

Dari hasil evaluasi, diketahui bahwa kombinasi k-Means dan *autoencoder* menghasilkan kinerja yang lebih baik dibandingkan hanya dengan menggunakan k-Means saja, yang ditunjukkan dengan nilai Silhouette yang lebih tinggi dan nilai DBI yang lebih rendah. Oleh karena itu, penggunaan *autoencoder* untuk ekstraksi fitur dan reduksi dimensi dapat meningkatkan kinerja model *clustering*, khususnya *clustering* menggunakan algoritma k-Means.

4. KESIMPULAN

Kombinasi k-Means dan *autoencoder* dapat digunakan dengan cukup baik untuk pengelompokan kemampuan penguasaan ICT mahasiswa. Hasil pengukuran dan evaluasi SSE, Silhouette *coefficient*, dan Davies-Bouldin Index menunjukkan bahwa penggabungan k-Means dengan *autoencoder* dapat meningkatkan kinerja model *clustering* pada data kemampuan ICT mahasiswa dibandingkan hanya menggunakan k-Means saja. Hal ini ditunjukkan dengan nilai SSE yang lebih rendah, nilai koefisien Silhouette yang lebih tinggi, dan nilai DBI yang lebih rendah. Meskipun demikian, masih terdapat beberapa data yang tumpang tindih. Oleh karena itu, dalam penelitian selanjutnya dapat melakukan analisis *cluster* pada data kemampuan ICT mahasiswa menggunakan model baru atau kombinasi model-model yang sudah ada.

UCAPAN TERIMA KASIH

Ucapan terima kasih diucapkan kepada LPPM UIN Sunan Kalijaga Yogyakarta dan Pusat teknologi Informasi dan Pangkalan Data atas segala dukungan dan fasilitas yang diberikan. Penelitian ini didasarkan pada Penelitian BOPTN Tahun 2023.

DAFTAR PUSTAKA

- Abidin, N. A. S. Z., Avila, R. D., Hermatyar, A., & Rismayani, R. (2022). Perbandingan Algoritma K-Means dan K-Medoids untuk Pengelompokan Daerah Produksi Kakao. *Jurnal Teknik Informatika dan Sistem Informasi*, 8(2), 383–391. <https://doi.org/10.28932/jutisi.v8i2.4897>
- Agarwal, V. (2015). Research on Data Preprocessing and Categorization Technique for Smartphone Review Analysis. *International Journal of Computer Applications*, 131(4), 30–36. <https://doi.org/10.5120/ijca2015907309>
- Baser, P., & R. Saini, J. (2015). Agent-based Stock Clustering for Efficient Portfolio Management. *International Journal of Computer Applications*, 116(3), 35–41. <https://doi.org/10.5120/20317-2381>
- Behera, M. P., Sarangi, A., & Mishra, D. (2021). K-medoids Crazy Firefly Algorithm for Unsupervised Data Clustering. *2021 1st Odisha International Conference on Electrical Power Engineering, Communication and Computing Technology (ODICON)*, 1–6. <https://doi.org/10.1109/ODICON50556.2021.9428980>



- Cui, M. (2020). Introduction to the K-Means Clustering Algorithm Based on the Elbow Method. *Accounting, Auditing and Finance Clausius Scientific Press*, 1(1), 5–8. <https://doi.org/10.23977/accaf.2020.010102>
- Dona, D., & Rifqi, M. (2022). Penerapan Metode K-Means Clustering untuk Menentukan Status Gizi Baik dan Gizi Buruk pada Balita (Studi Kasus Kabupaten Rokan Hulu). *Rabit: Jurnal Teknologi dan Sistem Informasi Univrab*, 7(2), 179–191. <https://doi.org/10.36341/rabit.v7i2.2171>
- Fitriani, M. N. R., Priyatna, B., Huda, B., Hananto, A. L., & Tukino, T. (2023). Implementasi Metode K-Means Untuk Memprediksi Status Kredit Macet. *Jurnal Sistem Komputer dan Informatika (JSON)*, 4(3), 554–561. <https://doi.org/10.30865/json.v4i3.5953>
- Marutho, D., Handaka, S. H., Wijaya, E., & Muljono, M. (2018). The Determination of Cluster Number at k-Mean Using Elbow Method and Purity Evaluation on Headline News. *2018 International Seminar on Application for Technology of Information and Communication*, 533–538. <https://doi.org/10.1109/ISEMANTIC.2018.8549751>
- McIlhany, K., & Wiggins, S. (2018). High Dimensional Cluster Analysis Using Path Lengths. *Journal of Data Analysis and Information Processing*, 06(03), 93–125. <https://doi.org/10.4236/jdaip.2018.63007>
- Musfiani, M. (2019). Analisis Cluster dengan Menggunakan Metode Partisi pada Pengguna Alat Kontrasepsi di Kalimantan Barat. *Bimaster: Buletin Ilmiah Matematika, Statistika dan Terapannya*, 8(4), 893–902. <https://doi.org/10.26418/bbimst.v8i4.36584>
- Nainggolan, R., Perangin-angin, R., Simarmata, E., & Tarigan, A. F. (2019). Improved the Performance of the K-Means Cluster Using the Sum of Squared Error (SSE) Optimized by Using the Elbow Method. *Journal of Physics: Conference Series*, 1361(1), Article ID: 012015. <https://doi.org/10.1088/1742-6596/1361/1/012015>
- Novoselsky, A., & Kagan, E. (2021). An Introduction to Cluster Analysis. In *An Introduction to Toxicogenomics* (pp. 1–9). Chapman and Hall/CRC. <https://doi.org/10.13140/RG.2.2.25993.57448/1>
- Nugroho, H., Susanty, M., Irawan, A., Koyimatu, M., & Yunita, A. (2020). Fully Convolutional Variational Autoencoder for Feature Extraction of Fire Detection System. *Jurnal Ilmu Komputer dan Informasi*, 13(1), 9–15. <https://doi.org/10.21609/jiki.v13i1.761>
- Nur Aziz, Y. A., & Zuliarso, E. (2022). Sistem Penerimaan Siswa Baru di SMKN 3 Pati Berdasar Jalur Prestasi Menggunakan Algoritma Klastering K-Means Berbasis Web. *Jurnal Ilmiah Informatika*, 10(02), 86–95. <https://doi.org/10.33884/jif.v10i02.5555>
- Prihanditya, H. A., & Alamsyah. (2020). The Implementation of Z-Score Normalization and Boosting Techniques to Increase Accuracy of C4.5 Algorithm in Diagnosing Chronic Kidney Disease. *Journal of Soft Computing Exploration*, 1(1), 63–69. <https://doi.org/10.52465/joscex.v1i1.8>
- Velmurugan, T. (2018). A State of Art Analysis of Telecommunication Data by k-Means and k-Medoids Clustering Algorithms. *Journal of Computer and Communications*, 06(01), 190–202. <https://doi.org/10.4236/jcc.2018.61019>
- Wu, C., Yan, B., Yu, R., Yu, B., Zhou, X., Yu, Y., & Chen, N. (2021). K-Means Clustering Algorithm and Its Simulation Based on Distributed Computing Platform. *Complexity*, 2021(1), Article ID: 9446653. <https://doi.org/10.1155/2021/9446653>
- Xu, D., & Tian, Y. (2015). A Comprehensive Survey of Clustering Algorithms. *Annals of Data Science*, 2(2), 165–193. <https://doi.org/10.1007/s40745-015-0040-1>
- Yuan, C., & Yang, H. (2019). Research on K-Value Selection Method of K-Means Clustering Algorithm. *J—Multidisciplinary Scientific Journal*, 2(2), 226–235. <https://doi.org/10.3390/j2020016>
- Zhai, J., Zhang, S., Chen, J., & He, Q. (2018). Autoencoder and Its Various Variants. *2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 415–419. <https://doi.org/10.1109/SMC.2018.00080>
- Zhao, Y., & Zhou, X. (2021). K-Means Clustering Algorithm and Its Improvement Research. *Journal of Physics: Conference Series*, 1873(1), Article ID: 012074. <https://doi.org/10.1088/1742-6596/1873/1/012074>



Analisis Sentimen Ulasan Pengguna Aplikasi Alfagift Menggunakan Random Forest

M. Bagus Prayogi ^{(1)*}, Gustina Masitoh ⁽²⁾

¹ Departemen Informatika, Universitas Nurul Huda, OKU Timur, Indonesia

² Departemen Pendidikan Ekonomi, Universitas Nurul Huda, OKU Timur, Indonesia
e-mail : mhdjesen212@gmail.com, gustina@unuha.ac.id.

* Penulis korespondensi.

Artikel ini diajukan 17 Maret 2024, direvisi 16 Desember 2024, diterima 16 Desember 2024, dan dipublikasikan 31 Mei 2025.

Abstract

Alfagift is a mobile application developed by Alfamart to support online ordering, featuring promotions, transactions, ordering, and delivery from the nearest point based on the consumer's address. User feedback on the Google Play Store reveals mixed sentiments, including both positive and negative responses, which developers can use as material to improve the application's quality. This study focuses on assessing the sentiment of Alfagift app user reviews using the Random Forest algorithm. A total of 4,379 review data points were collected from the Google Play Store and grouped into two categories: positive and negative sentiment. The research steps include data collection, data labeling, data preprocessing, word weighting, dividing the data into training and testing sets, implementing the Random Forest algorithm, and model evaluation. The test results show that the Random Forest algorithm achieves an accuracy of 97.6% and an AUC of 0.98, which falls into the category of excellent classification. This research is expected to contribute to application developers' understanding of user perceptions, enabling them to improve application quality and increase overall user convenience.

Keywords: *Sentiment Analysis, Alfagift, Random Forest, Text Mining, Review*

Abstrak

Alfagift adalah aplikasi *mobile* yang dikembangkan oleh Alfamart untuk mendukung pemesanan secara *online*, dengan fitur-fitur seperti promo, transaksi, pemesanan, dan pengiriman dari titik terdekat sesuai alamat konsumen. Masukan dari pengguna di Google Play Store menunjukkan sentimen yang beragam, meliputi tanggapan positif dan negatif, yang dapat dimanfaatkan oleh pengembang sebagai bahan untuk meningkatkan kualitas aplikasi. Studi ini berfokus pada pengkajian sentimen dari ulasan pengguna aplikasi Alfagift melalui penerapan algoritma Random Forest. Sebanyak 4.379 data ulasan dikumpulkan dari Google Play Store dan dikelompokkan menjadi dua kategori, yaitu sentimen positif dan negatif. Langkah-langkah penelitian mencakup pengumpulan data, pelabelan data, *preprocessing* data, pembobotan kata, pembagian data menjadi set pelatihan dan pengujian, implementasi algoritma Random Forest, serta evaluasi model. Hasil pengujian menunjukkan bahwa algoritma Random Forest mencapai akurasi sebesar 97,6% dan AUC 0,98, yang masuk dalam kategori klasifikasi sangat baik. Penelitian ini diharapkan dapat memberikan kontribusi bagi pengembang aplikasi dalam memahami persepsi pengguna, sehingga dapat meningkatkan kualitas aplikasi serta meningkatkan kenyamanan pengguna secara keseluruhan.

Kata Kunci: *Analisis Sentimen, Alfagift, Random Forest, Text Mining, Ulasan*

1. PENDAHULUAN

Trend berbelanja secara *online* meningkat secara signifikan pada masa pandemi COVID-19 (Hanifa et al., 2023). Berbelanja secara *online* kini telah menjadi aktivitas yang tidak terpisahkan dari kehidupan masyarakat. Hal ini disebabkan oleh kemudahan yang tersedia untuk memenuhi kebutuhan primer maupun tersier (Helmi et al., 2023). Belanja *online* dapat dilakukan di mana saja hanya dengan menggunakan *smartphone*. Hal ini menjadi alasan utama mengapa banyak individu cenderung memilih belanja secara *online*, karena pelanggan tidak perlu datang ke toko fisik, prosesnya menjadi lebih cepat dan efisien. Semua transaksi dapat dilakukan dengan mudah



melalui *smartphone* (Perdana et al., 2022). Seiring dengan meningkatnya tren berbelanja secara *online*, banyak perusahaan bersaing untuk menawarkan layanan sistem penjualan *online* yang memudahkan pelanggan dalam membeli barang. Salah satu jenis layanan tersebut adalah aplikasi yang menghubungkan bisnis dengan pelanggan (Utami et al., 2022). Saat ini, Alfagift merupakan salah satu platform belanja *online* yang sedang berkembang, yang diperkenalkan oleh PT Sumber Alfaria Trijaya Tbk (Firdaus, 2024). Alfagift merupakan layanan digital yang disediakan oleh Alfamart. Aplikasi ini memungkinkan pengguna membeli kebutuhan sehari-hari secara online dan menawarkan berbagai promo menarik bagi pengguna (Fadli et al., 2023). Aplikasi ini menyediakan beragam opsi metode pembayaran, salah satunya adalah melalui transfer bank BCA, e-money, GoPay, dan lainnya, untuk mempermudah pengguna (Perdana et al., 2022).

Di zaman sekarang, terdapat berbagai metode untuk menganalisis sebuah aplikasi, salah satunya melalui analisis sentimen. Analisis sentimen merupakan metode berbasis komputasi yang bertujuan untuk mengidentifikasi opini, perasaan, serta emosi (baik positif maupun negatif) dari individu (Komarudin & Hilda, 2024). Pendekatan ini dilakukan dengan menganalisis elemen dan entitas yang diungkapkan melalui teks. Analisis sentimen menghasilkan opini yang diekstraksi dan diolah secara otomatis dari data teks (Jin et al., 2023). Secara umum, teknik ini memberikan beragam manfaat bagi organisasi, antara lain memahami persepsi masyarakat, meningkatkan pengalaman pelanggan, mendorong efektivitas penjualan dan pemasaran, memantau aktivitas media sosial, mendukung penelitian dan pengembangan, serta memperkuat aspek keamanan dan intelijen (Indrayanto et al., 2023). Analisis sentimen membagi opini atau pendapat menjadi tiga jenis: positif, negatif, dan netral. Namun, dalam proses klasifikasi, sentimen positif dan negatif biasanya lebih diperhatikan, karena sentimen netral sering dianggap kurang signifikan atau tidak memiliki arti yang relevan (Wicaksono, 2022). Model sentimen juga dapat memengaruhi faktor loyalitas, pengiriman, pemesanan, serta memberikan rekomendasi perbaikan pada aplikasi (Firdaus, 2024).

Penelitian terdahulu oleh Anita Firdaus berjudul “*Analisis Sentimen pada Aplikasi Alfagift Menggunakan Metode Naïve Bayes Classifier*” menunjukkan tingkat akurasi sebesar 82%. Hasil tersebut dianggap cukup baik dalam menganalisis sentimen (Firdaus, 2024). Selanjutnya, penelitian berjudul “*Analisis Sentimen Kepuasan Pengguna terhadap Layanan Streaming Mola Menggunakan Algoritma Random Forest*” menggunakan 520 *dataset* dengan pembagian data 90:10 dan menghasilkan tingkat akurasi sebesar 98% (Nanda et al., 2022). Di sisi lain, penelitian oleh Natalya Br Sidauruk berjudul “*Penggunaan Metode SVM dan Random Forest untuk Analisis Sentimen Ulasan Pengguna terhadap KAI Access di Google Play Store*” menggunakan 655 ulasan. Dalam penelitian tersebut, algoritma SVM mencapai tingkat akurasi sebesar 97%, sedangkan algoritma Random Forest mendapatkan akurasi sebesar 93% (Sidauruk et al., 2023).

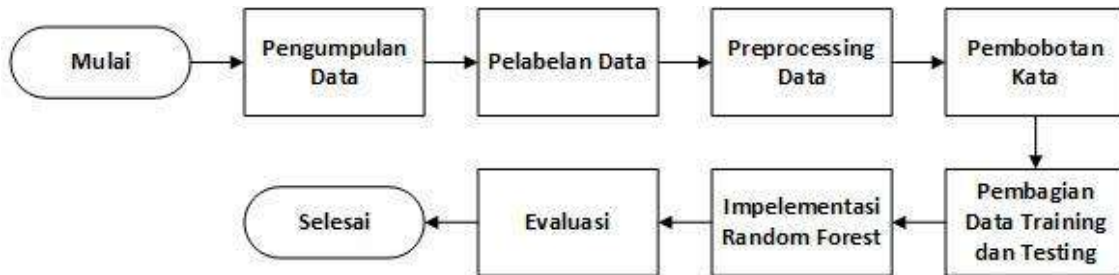
Penelitian terdahulu telah mengkaji berbagai pendekatan dalam analisis sentimen terhadap penggunaan aplikasi Alfagift menggunakan metode Naïve Bayes. Namun, faktor akurasi prediksi menjadi poin penting dalam memilih algoritma yang paling sesuai untuk tujuan ini. Dalam studi ini, peneliti menggunakan pendekatan lain untuk mendapatkan hasil prediksi yang maksimal dan pada hasil akhir studi ini peneliti akan membandingkan dengan penelitian sebelumnya.

Tujuan utama studi ini adalah untuk mengetahui seberapa akurat algoritma klasifikasi Random Forest dalam menganalisis sentimen pengguna aplikasi Alfagift dan menentukan klasifikasi sentimen positif atau negatif berdasarkan ulasan aplikasi tersebut. Random Forest dipilih karena, menurut Pahlevi et al. (2023) Random Forest merupakan algoritma dengan performa yang sangat baik. Pentingnya penelitian ini bagi *developer* atau perusahaan adalah dapat membantu mereka dalam meningkatkan produk dan layanan berdasarkan umpan balik yang diberikan oleh pelanggan secara tulus dan spesifik. Penelitian ini diharapkan dapat memberikan kemajuan pada aplikasi tersebut dan mempermudah masyarakat dalam berbelanja *online* dengan pelayanan yang maksimal.



2. METODE PENELITIAN

Tahapan penelitian adalah serangkaian langkah yang dijalankan oleh peneliti secara teratur dan terencana dalam melaksanakan proses penelitian (Hasibuan et al., 2024). Secara lebih rinci, tahapan yang kami gunakan meliputi pengumpulan data, pelabelan data, *preprocessing* data, pembobotan kata, pembagian data *training* dan *testing*, implementasi Random Forest, serta evaluasi. Pada Gambar 1 dipaparkan secara rinci proses penelitian.



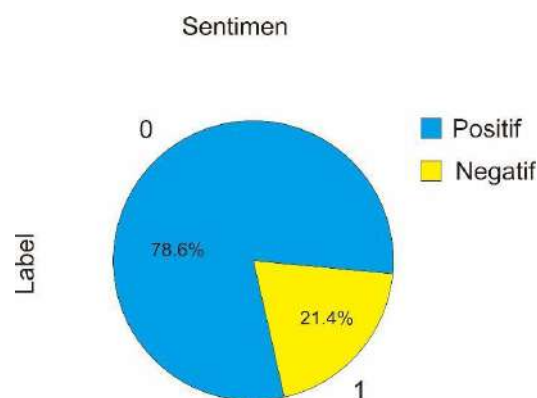
Gambar 1 Tahapan Penelitian

2.1 Pengumpulan Data

Langkah pertama dalam penelitian ini yaitu pengumpulan data, metode *web scraping* digunakan pada tahapan ini. *Web scraping* adalah proses mengambil data semi terstruktur dari suatu situs web atau aplikasi (Styawati et al., 2021). Pengumpulan data dilakukan menggunakan Google Colab dan bahasa pemrograman Python, di mana sebanyak 4.379 ulasan dari Google Play Store berhasil dikumpulkan dan disimpan dalam format CSV.

2.2 Pelabelan Data

Data hasil *scraping* tidak bisa langsung dilakukan analisis perlu di lakukan pelabelan terlebih dahulu pada data tersebut (Syakir & Hasan, 2023). Di mana hasil dari proses pelabelan yaitu di dapatkan kelas positif dan kelas negatif. Berdasarkan data ulasan yang ada, terdapat 3.443 ulasan dengan sentimen positif dan 935 ulasan dengan sentimen negatif. Persentase ulasan yang telah dikategorikan sebagai sentimen positif dan negatif divisualisasikan dalam Gambar 2 yang menunjukkan distribusi label kelas tersebut.



Gambar 2 Diagram Sentimen

2.3 Preprocessing Data

Tahapan *preprocessing* adalah proses paling penting dikarenakan pada step inilah yang menentukan kualitas dari data yang digunakan. *Preprocessing* adalah metode untuk *cleaning* dan mengatur data yang semula acak acakan sehingga menjadi data yang rapi dan relevan yang



efektif (Arista et al., 2024). Adapun tahapan pada proses *preprocessing* meliputi *cleaning*, *normalization*, *tokenizing*, *stopwords removal*, dan *stemming*.

2.4 Pembobotan Kata

Selanjutnya, teknik pembobotan kata yang akan digunakan adalah Term Frequency-Inverse Document Frequency (TF-IDF), yang merupakan transformasi data teks menjadi angka. Kecerdasan komputasional dalam pembelajaran mesin dan pembelajaran mendalam secara umum bekerja dengan baik dan efektif ketika data yang diproses berbentuk angka. Dalam setiap studi di bidang *text mining*, khususnya yang berfokus pada analisis sentimen, perubahan data dari teks menjadi angka merupakan langkah yang wajib. Proses ini dikenal dengan istilah pembobotan kata. Beberapa metode yang dapat digunakan dalam pembobotan kata antara lain Bag of Words (BoW), N-gram, Word2Vec, dan TF-IDF (Kadhim, 2018). TF-IDF merupakan metode yang digunakan untuk mengelola teks dengan memberikan bobot pada kata-kata dalam sebuah dokumen. Pembobotan ini bertujuan untuk mengetahui kosa kata terpenting atau yang paling sering muncul dalam sebuah dokumen maupun kumpulan dokumen. Nilai frekuensi kata yang muncul dalam dokumen disebut Term Frequency (TF). Rumus untuk menghitung nilai TF ditunjukkan pada Pers. (1).

Kata penghubung serta kata-kata umum seringkali memiliki nilai TF yang tinggi, namun tidak mengandung makna yang signifikan. Terdapat metode alternatif yang dapat digunakan, yaitu Inverse Document Frequency (IDF). Metode ini bekerja dengan mencari kosa kata yang frekuensi kemunculannya jarang dalam sebuah dokumen, kemudian memberikan bobot pada kata-kata tersebut. Rumus untuk menghitung nilai IDF dijabarkan pada Pers. (2). Dalam rumus tersebut N adalah total jumlah dokumen, dan n adalah jumlah dokumen yang mengandung kata tersebut. Dengan mengalikan hasil TF dan IDF, maka diperoleh nilai TF-IDF.

$$TF = \frac{(\text{jumlah Kemunculan Kata dalam dokumen})}{(\text{jumlah kata dalam dokumen})} \quad (1)$$

$$IDF = \log \frac{N}{n} \quad (2)$$

2.5 Pembagian Data *Training* dan Data *Testing*

Perbandingan antara data pelatihan dan data pengujian dapat mempengaruhi hasil klasifikasi, karena pemilihan rasio yang tidak tepat bisa mengakibatkan penurunan akurasi pada model yang dibangun (Pradana, 2016). Proses pembagian data bertujuan untuk memisahkan data yang telah melalui tahap *preprocessing* menjadi data latih (*train*) dan data uji (*test*). Pembagian ini dilakukan dalam lima variasi persentase yang berbeda, guna mengevaluasi perbandingan mana yang menghasilkan kinerja terbaik. Setiap set data yang terpisah kemudian akan diterapkan pada algoritma *machine learning* dalam tahap pemodelan.

2.6 Implementasi Random Forest

Random Forest adalah metode komputasi yang melibatkan kumpulan pohon keputusan (*decision trees*) yang mengandalkan suara mayoritas untuk menentukan hasil akhir. Setiap pohon keputusan umumnya terdiri dari *root node*, *internal node*, dan *leaf node*. Metode seperti *entropy*, *information gain*, *gini index*, dan *gini split* digunakan untuk mencari akar pohon keputusan. Untuk membangun banyak pohon, teknik *bagging* diterapkan dengan mengambil sampel secara acak dari fitur dan baris data untuk proses pelatihan pohon keputusan. Biasanya, *entropy*, *information gain*, atau *gini split index* digunakan untuk mengevaluasi pohon keputusan dan prediksi yang didasarkan pada suara mayoritas (Anjani et al., 2023). Rumus yang digunakan untuk mendapatkan nilai *entropy*, *gain*, *gini index*, dan *gini split* dituliskan pada Pers. (3) sampai (6).



$$Entropy(S) = \sum_{i=1}^n -P_i \log_2 p_i \quad (3)$$

$$Gain(S, F) = Entropy(S) - \sum_{i=1}^n \frac{|S_i|}{|S|} \times Entropy(S_i) \quad (4)$$

$$Gini(S) = 1 - \sum_{i=1}^n (P_i)^2 \quad (5)$$

$$Gini_{split} = \sum_{i=1}^n \frac{|S_i|}{|S|} \times Gini(S_i) \quad (6)$$

Nilai entropi yang mengukur ketidakpastian atau impurity suatu dataset disebut $Entropy(S)$. Rumus untuk menghitung nilai Entropy ditunjukkan pada Pers. (3). Dalam rumus tersebut S adalah himpunan data yang dianalisis, P_i adalah probabilitas kelas ke- i dalam *dataset*, dan n adalah jumlah total kelas. Semakin tinggi nilai Entropy, semakin tidak teratur atau beragam data tersebut.

Nilai pengurangan entropi setelah pembagian *dataset* berdasarkan fitur disebut $Gain(S, F)$. Rumus untuk menghitung Gain ditunjukkan pada Pers. (4). Di dalamnya S adalah *dataset* utama sebelum *split*, F adalah fitur yang digunakan untuk split, $Entropy(S)$ adalah entropi sebelum *split*, $|S_i|$ adalah jumlah data pada *subset* hasil *split*, $|S|$ adalah jumlah total data, dan $Entropy(S_i)$ adalah entropi pada masing-masing *subset*. Nilai Gain yang tinggi menandakan fitur tersebut memberikan informasi signifikan untuk memisahkan data.

Nilai *impurity dataset* dengan pendekatan indeks Gini disebut $Gini(S)$. Rumus untuk menghitung Gini ditunjukkan pada Pers. (5). Dalam rumus ini S adalah himpunan data yang dianalisis dan P_i adalah probabilitas kelas ke- i dalam *dataset*. Semakin tinggi nilai Gini, semakin beragam data, sedangkan nilai rendah menunjukkan data lebih homogen.

Rata-rata bobot *impurity* Gini setelah *dataset* dibagi ke dalam *subset* disebut $Gini_{split}$. Rumus untuk menghitung $Gini_{split}$ ditunjukkan pada Pers. (6). Di dalamnya S adalah *dataset* utama sebelum *split*, S_i adalah *subset* hasil pembagian, $|S_i|$ adalah jumlah data pada subset, $|S|$ adalah jumlah total data, dan $Gini(S_i)$ adalah nilai Gini *impurity* pada setiap *subset*. Nilai $Gini_{split}$ ini digunakan untuk mengevaluasi kualitas pembagian *dataset*.

2.7 Evaluasi

Pada penghujung studi ini dilakukan evaluasi performa algoritma Random Forest dalam pengklasifikasian teks sentimen. Evaluasi ini bertujuan untuk menilai sejauh mana model mampu memberikan hasil klasifikasi yang baik. Evaluasi ini akan berpatokan pada hasil visualisasi *confusion matrix* dan grafik ROC.

3. HASIL DAN PEMBAHASAN

3.1 Pengumpulan Data

Proses pengumpulan ulasan aplikasi Alfagift dari Google Play Store dilakukan melalui penerapan metode *web scraping*, menggunakan bahasa pemrograman Python yang diintegrasikan dengan alat Google Colab untuk memudahkan prosesnya. Data yang diperoleh dalam proses ini berjumlah 4.379 ulasan dengan beberapa atribut yaitu *review Id*, *userName*, *userImage content*, *score*, dan lain-lain. Kemudian dilakukan seleksi pada atribut tersebut dengan menyisakan *content* dan *score* saja. Detail hasil proses ditampilkan pada Tabel 1.



Tabel 1 Hasil Pengumpulan Data

No.	Score	Content
1	1	Respon pelayanannya lama banget, masa sampe 2 hari chat pelanggan baru di bales. Heran deh, malah tercepat admin IG alfamart.
2	1	Stok dikranjang masih ada giliran mau belanja diklik berubah jadi stok kosong 🙄
...
4379	5	Keren belanja jdi mudah dan dapat point

3.2 Pelabelan Data

Proses pemberian label dilakukan dengan menggunakan *rating* ulasan sebagai tolak ukur dalam pelabelan sentimen. Di mana *rating* 1, 2, 3 dikelompokkan ke dalam sentimen negatif (1) kemudian *rating* 4 dan 5 dikelompokkan ke dalam sentimen positif (0). Dari proses pelabelan ini, diperoleh 3.443 ulasan dengan sentimen positif dan 935 ulasan dengan sentimen negatif. Rincian hasil dari proses ini disajikan dalam Tabel 2.

Tabel 2 Hasil Pelabelan

No.	Score	Content	Label
1	1	Respon pelayanannya lama banget, masa sampe 2 hari chat pelanggan baru di bales. Heran deh, malah tercepat admin IG alfamart.	1
2	1	Stok dikranjang masih ada giliran mau belanja diklik berubah jadi stok kosong 🙄	1
...
4379	5	Keren belanja jdi mudah dan dapat point	0

3.3 Preprocessing Data

Setelah proses pelabelan data selesai, tahapan selanjutnya yang dilakukan adalah *preprocessing* data sebelum masuk ke tahap pemodelan. Tahapan ini mencakup pembersihan data dari nilai-nilai yang hilang, duplikat, atau tidak relevan, serta penyesuaian format agar data dapat diproses dengan optimal oleh algoritma pembelajaran mesin. Tujuan dari *preprocessing* ini adalah untuk memastikan bahwa data berada dalam kondisi yang bersih dan terstruktur, sehingga dapat meningkatkan akurasi dan efisiensi model yang akan dibangun.

Pada proses *cleaning*, karakter selain huruf akan dihilangkan. Ini termasuk simbol pagar (#), *link*, dan tanda baca contohnya garis miring (/), titik koma (;), titik dua (:), dan titik (.). Seluruh teks juga dikonversi menjadi huruf kecil, menghapus mention akun (@), menstandarisasi kata dengan menghilangkan apostrof (') dan karakter alfanumerik yang menempel, serta menghapus spasi ganda agar diperoleh data teks yang lebih bersih untuk analisis selanjutnya. Detail hasil proses ditampilkan pada Tabel 3.

Tabel 3 Hasil Cleaning Data

No.	Score	Content	Label
1	1	respon pelayanannya lama banget masa sampe hari chat pelanggan baru di bales heran deh malah tercepat admin ig alfamart	1
2	1	Stok dikranjang masih ada giliran mau belanja diklik berubah jadi stok kosong	1
...
4379	5	keren belanja jdi mudah dan dapat point	0

Dalam proses *normalization*, dilakukan standarisasi ejaan kata kata yang dalam penulisannya sering kali di singkat dan ejaan yang tidak baku menjadi baku. Dengan contoh kata "sya"



kemudian dinormalisasi menjadi kata “saya”, kata “dgn” menjadi “dengan”, kemudian kata “kpn” yang diubah menjadi “kapan” dan seterusnya. Tujuan dari proses normalisasi ini adalah untuk menseragamkan supaya nanti pada saat proses pengujian didapatkan hasil yang akurat. Detail hasil proses ditampilkan pada Tabel 4.

Tabel 4 Hasil Normalisasi

No.	Score	Content	Label
1	1	respon pelayanannya lama banget masa sampe hari chat pelanggan baru di bales heran deh malah tercepat admin ig alfamart	1
2	1	Stok dikranjang masih ada giliran mau belanja diklik berubah jadi stok kosong	1
...
4379	5	Keren belanja jadi mudah dan dapat point	0

Dalam proses *tokenizing*, teks dipotong menjadi token berupa kata-kata individual menggunakan RegexpTokenizer. Teks dipisahkan pada setiap kata yang terdiri dari karakter alfanumerik (\w+). Dengan *tokenizing* ini, teks yang semula berupa kalimat utuh dapat dipecah menjadi kata per kata, memudahkan analisis leksikal. Hasil *tokenizing* ditambahkan ke dalam kolom baru ‘text_token’ pada *data frame* untuk digunakan pada tahap analisis selanjutnya. Detail hasil proses ditampilkan pada Tabel 5.

Tabel 5 Hasil Tokenizing

No.	Score	Content	Label
1	1	['respon', 'pelayanannya', 'lama', 'banget', 'masa', 'sampe', 'hari', 'chat', 'pelanggan', 'baru', 'di', 'bales', 'heran', 'deh', 'malah', 'tercepat', 'admin', 'ig', 'alfamart']	1
2	1	['stok', 'dikranjang', 'masih', 'ada', 'giliran', 'mau', 'belanja', 'diklik', 'berubah', 'jadi', 'stok', 'kosong']	1
...
4379	5	['keren', 'belanja', 'jadi', 'mudah', 'dan', 'dapat', 'point']	0

Dalam proses penghapusan *stopwords*, kata-kata yang termasuk dalam daftar *stopwords* dalam bahasa Indonesia dihapus dari hasil tokenisasi agar tidak mempengaruhi proses analisis selanjutnya. *Stopwords* adalah kata-kata yang sangat umum seperti ‘yang’, ‘dan’, ‘di’, ‘dll’ yang kurang memiliki makna penting secara leksikal. Dengan menghapus *stopwords*, diharapkan kata-kata yang tersisa lebih mewakili substansi dan topik teks sebenarnya. Daftar *stopwords* bahasa Indonesia diunduh dari corpus NLTK. Selanjutnya, kata-kata hasil tokenisasi disaring dengan menghapus *stopwords*, menghasilkan kumpulan kata yang lebih relevan dan bermakna untuk analisis. Rincian hasil dari proses ini disajikan dalam Tabel 6.

Tabel 6 Hasil Stopwords Removal

No.	Score	Content	Label
1	1	['respon', 'pelayanannya', 'banget', 'sampe', 'chat', 'pelanggan', 'bales', 'heran', 'deh', 'tercepat', 'admin', 'ig', 'alfamart']	1
2	1	['stok', 'dikranjang', 'giliran', 'belanja', 'diklik', 'berubah', 'stok', 'kosong']	1
...
4379	5	['keren', 'belanja', 'jadi', 'mudah', 'point']	0

Dalam proses *stemming*, kata-kata hasil tokenisasi diubah ke bentuk kata dasarnya menggunakan *stemmer* bahasa Indonesia dari library Sastrawi. Proses *stemming* diterapkan pada setiap kata untuk memperoleh bentuk dasarnya dengan menghapus imbuhan seperti awalan, akhiran, dan sisipan. Sebagai contoh, kata “bermain” akan diubah menjadi “main” setelah melalui proses *stemming*. Dengan *stemming* ini, kata-kata dengan akar yang sama dapat



dianggap sebagai kata yang sama meskipun bentuk infleksinya berbeda. Hasil *stemming* kemudian diubah kembali dari kumpulan token menjadi kalimat utuh agar siap digunakan pada proses analisis selanjutnya, seperti pembentukan model *topic modeling*. Detail hasil proses ditampilkan pada Tabel 7.

Tabel 7 Hasil *Stemming*

No.	Score	Content	Label
1	1	['respon', 'layan', 'banget', 'sampe', 'chat', 'pelanggan', 'bales', 'heran', 'deh', 'cepat', 'admin', 'ig', 'alfamart']	1
2	1	['stok', 'dikranjang', 'gilir', 'belanja', 'klik', 'ubah', 'stok', 'kosong']	1
...
4379	5	['keren', 'belanja', 'jadi', 'mudah', 'point']	0

3.4 Pembobotan Kata

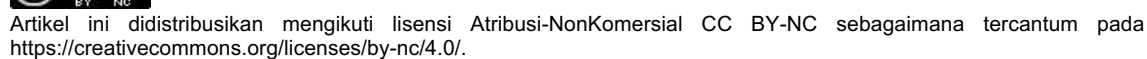
Kata kata dengan dinilai paling tinggi dianggap paling penting dalam menentukan topik sebuah dokumen. Detail hasil proses pembobotan ditampilkan pada Gambar 3. Pada Gambar 3, kode 1 menunjukkan penomoran dari baris yang sedang dilakukan pengolahan, kode 2 merupakan nomor unik yang diberikan kepada setiap kata dalam baris, dan kode 3 menunjukkan skor, yang merupakan hasil dari pembobotan kata yang merupakan hasil dari perhitungan TF-IDF. Detail kosa kata yang sering muncul dalam bentuk diagram *plot* ditampilkan dalam Gambar 4.

1	2	3
(0, 2172)		0.09594938526700864
(0, 4962)		0.07874241691835011
(0, 3406)		0.06631816875103022
(0, 3627)		0.1245389886334698
(0, 4176)		0.3531106263389961
(0, 310)		0.16370596653203248
(0, 3371)		0.14010975643633627
(0, 1103)		0.1866114148507248
(0, 871)		0.09872429561247213
(0, 592)		0.030256407576428574
(0, 5060)		0.29185735493308645
(0, 3991)		0.15606889541962135
(0, 2805)		0.1506503072325116
(0, 1095)		0.1016277752962363
(0, 905)		0.1375946479329907
(0, 5006)		0.1133374810756203
(0, 223)		0.07483145993873176
(0, 917)		0.12339710357673728
(0, 2052)		0.06829078210914138
(0, 375)		0.14644732370671837
(0, 4562)		0.09933193975738468
(0, 2373)		0.20056364355219883
(0, 1606)		0.09446161208447418
(0, 5064)		0.20970804583648062
(0, 1232)		0.16370596653203248

Gambar 3 Hasil TF-IDF

Selain menggunakan diagram plot, terdapat visualisasi yang menggunakan *wordcloud* di mana visualisasi dibagi menjadi kelas positif dan kelas negatif. Di dalam Gambar 5, ditampilkan visualisasi dari ulasan yang bersifat positif dan pada Gambar 6, terdapat visualisasi dari ulasan yang memiliki sifat negatif. Berdasarkan Gambar 5 dan 6, kata-kata seperti “gratis ongkir”, “cepat”, “belanja”, “Mudah”, dan “promo” banyak muncul pada ulasan positif. Sedangkan kata-kata seperti “tolong”, “kecewa”, “kirim”, dan “aplikasi” sering muncul pada ulasan negatif.





3.5 Pembagian Data *Training* Dan Data *Testing*

Setelah tahap *preprocessing* data selesai, langkah selanjutnya adalah memisahkan data menjadi dua bagian, yakni data pelatihan (*train*) dan data pengujian (*test*), yang akan dipisah sebanyak lima kali dan akan dilakukan eksperimen dengan *algoritma* Random Forest. Rasio perbandingan data dapat dilihat pada Tabel 8. Berdasarkan Tabel 8, terlihat bahwa 4.379 data dibagi antara data pelatihan dan pengujian dengan berbagai perbandingan, mulai dari 90% untuk data pelatihan dan 10% untuk data pengujian, hingga 50% untuk masing-masing data pelatihan dan pengujian.

Tabel 8 Persentase Perbandingan Data *Train* dan Data *Test*

Jumlah Data	Data <i>Train</i>	Data <i>Test</i>	Persentase
4.379	3.941	438	90:10
4.379	3.503	876	80:20
4.379	3.065	1.314	70:30
4.379	2.627	1.752	60:40
4.379	2.189	2.190	50:50

3.6 Implementasi Random Forest dan Evaluasi

Setelah melalui beberapa proses dari pengumpulan data, pelabelan, *preprocessing*, pembobotan kata serta pembagian data *training* dan *testing*, tahap selanjutnya yaitu penerapan algoritma Random Forest. Proses evaluasi dilakukan dengan menggunakan *confusion matrix* yang berfungsi memberikan sebuah visualisasi mengenai hasil kerja yang dilakukan oleh algoritma tersebut. Sebelum melihat hasil dari *confusion matrix* model terbaik, mari kita lihat terlebih dulu hasil akurasi dari lima rasio *train* dan *test* sebelumnya untuk melihat model terbaik menggunakan pembagian data di rasio berapa, yang dapat dilihat pada Tabel 9.

Tabel 9 Hasil Evaluasi Algoritma Random Forest

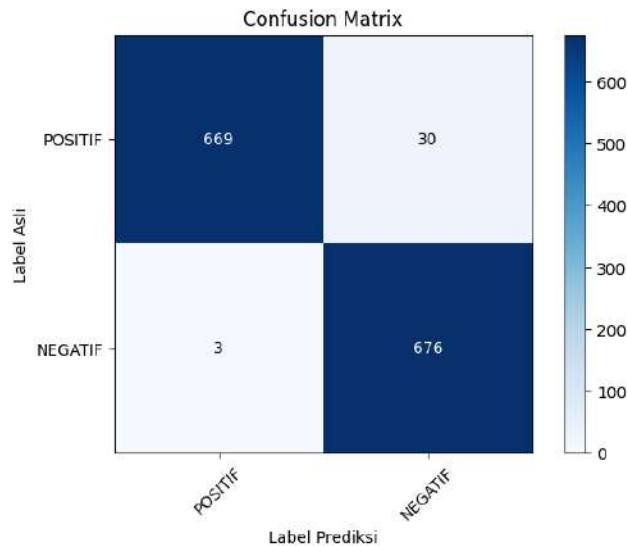
<i>Split</i>	Akurasi	%
90:10	0,965	96,5%
80:20	0,976	97,6%
70:30	0,969	96,9%
60:40	0,967	96,7%
50:50	0,961	96,1%

Dari Tabel 9 didapatkan bahwa rasio pembagian 80:20 menghasilkan akurasi terbaik dari rasio lainnya. Setelah dilakukan pengujian didapatkan nilai akurasi terbaik sebesar 0,976 dengan nilai 669 True Positive (TP), dan 676 True Negative (TN). Di sisi lain, terdapat 30 False Positive (FP) dan 3 False Negative (FN). Detail hasil ditampilkan dalam Gambar 7. Dikarenakan nilai dari *True Negatif* dan *True Positif* lebih besar dari nilai *False Positif* dan *False Negatif*, maka *confusion matrix* tersebut menunjukkan tingkat akurasi yang tinggi sebesar 0,976. Penjabaran perhitungan dari nilai akurasi ditunjukkan pada Pers. (7).

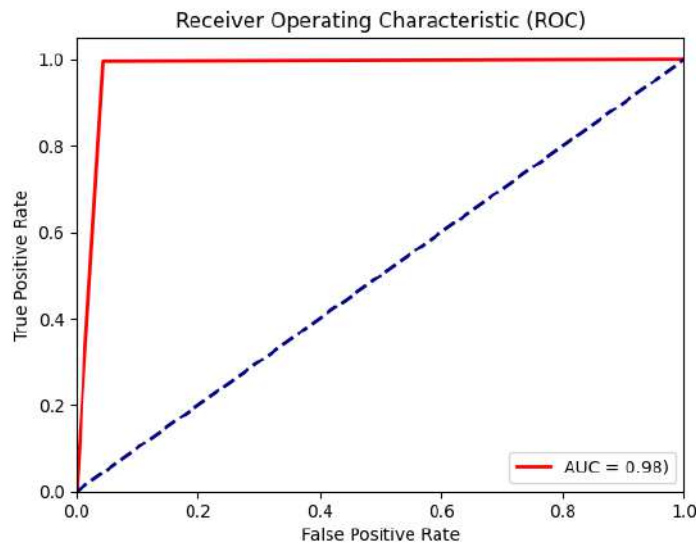
Untuk mengevaluasi hasil prediksi, Grafik Receiver Operating Characteristic (ROC) umumnya digunakan untuk menilai seberapa efektif model dalam membedakan antara kelas positif dan negatif. Kurva ROC dihasilkan dengan membandingkan nilai True Positive Rate (TPR) dan False Positive Rate (FPR) yang dihitung dari *confusion matrix*. Area di bawah kurva, juga dikenal sebagai Area Under Curve (AUC), menunjukkan kinerja model yang digunakan. Detail hasil ditampilkan dalam Gambar 8. Berdasarkan grafik ROC didapatkan nilai AUC sebesar 0,98. Menurut Wilujeng et al. (2023) kategori *excellent* memiliki nilai AUC 0,90–1. Berdasarkan tingkat nilai diagnosa pada kurva ROC pada penelitian ini dapat dikelompokkan ke dalam kelompok *excellent classification*.



$$\begin{aligned}
 Accuracy &= \frac{TP + TN}{TP + TN + FP + FN} \times 100 \\
 &= \frac{669 + 676}{669 + 676 + 30 + 3} \times 100\% \\
 &= \frac{1345}{1378} \times 100\% \\
 &= 97,6\%.
 \end{aligned} \tag{7}$$



Gambar 7 Confusion Matrix



Gambar 8 Grafik ROC

4. KESIMPULAN

Berdasarkan hasil pengujian model dan analisis sentimen terhadap ulasan pengguna aplikasi Alfagift, model *machine learning* Random Forest terbukti mampu mengklasifikasikan teks dengan sangat baik dan efektif, dengan nilai keakuratan sebesar 97,6% dan nilai AUC sebesar 0,98, yang masuk ke dalam kategori *excellent classification*. Dibandingkan dengan penelitian sebelumnya yang juga membahas topik yang sama menggunakan model Naïve Bayes, model Random Forest



terbukti lebih baik dari pada model Naïve Bayes. Secara keseluruhan, mayoritas ulasan ditemukan bersifat positif, mencapai persentase sebesar 78,6%, sementara ulasan negatif sebesar 21,4%.

Ulasan negatif umumnya berisi keluhan pengguna terkait pelayanan yang lambat, stok yang tidak sesuai, gagal login, dan masalah lain yang mengganggu pengalaman pengguna dalam menggunakan aplikasi Alfagift. Oleh karena itu, perhatian khusus perlu diberikan oleh pengembang Alfagift untuk memperbaiki dan meningkatkan kualitas aplikasi tersebut. Dengan memanfaatkan hasil analisis sentimen, pengembang dapat lebih baik dalam menargetkan perbaikan pada area yang diperlukan, sehingga dapat meningkatkan kepuasan pengguna. Hasil penelitian ini diharapkan dapat meningkatkan kenyamanan pengguna dalam berbelanja *online* di Alfagift. Dalam studi selanjutnya, diharapkan dapat memberikan peningkatan akurasi yang lebih baik lagi, terutama dengan penggunaan algoritma optimasi seperti Particle Swarm Optimization.

DAFTAR PUSTAKA

- Anjani, A. F., Anggraeni, D., & Tirta, I. M. (2023). Implementasi Random Forest Menggunakan SMOTE untuk Analisis Sentimen Ulasan Aplikasi Sister for Students UNEJ. *Jurnal Nasional Teknologi dan Sistem Informasi*, 9(2), 163–172. <https://doi.org/10.25077/TEKNOSI.v9i2.2023.163-172>
- Arista, D., Sibaroni, Y., & Prasetyo, S. S. (2024). Sentiment Analysis on Twitter (X) Related to Relocating the National Capital Using the IndoBERT Method Using Extraction Features of Chi-Square. *Jurnal Media Informatika Budidarma*, 8(1), 403–411. <https://doi.org/10.30865/mib.v8i1.7198>
- Helmi, A., Komaladewi, R., Sarasi, V., & Yolanda, L. (2023). Characterizing Young Consumer Online Shopping Style: Indonesian Evidence. *Sustainability*, 15(5), Article ID: 3988. <https://doi.org/10.3390/su15053988>
- Fadli, M. N., SJ, A. S., & Nafiah, N. (2023). Mekanisme Penggunaan Member Card Alfagift dalam Jual Beli Perspektif Hukum Islam (Studi Kasus di Alfamart Ponorogo). *Social Science Academic*, 1(2), 29–36. <https://doi.org/10.37680/ssa.v1i2.3173>
- Firdaus, A. (2024). *Analisis Sentimen pada Aplikasi Alfagift Menggunakan Metode Naïve Bayes Classifier* [UIN Syarif Hidayatullah]. <https://repository.uinjkt.ac.id/dspace/handle/123456789/76766>
- Hanifa, F., Putri, C. H., N, A. F., & Wulansari, A. (2023). Analisis Penerimaan Aplikasi Alfagift di Kota Surabaya Menggunakan Metode Technology Acceptance Model. *Jurnal Sains dan Teknologi (JSIT)*, 3(2), 233–244. <https://doi.org/10.47233/jsit.v3i2.835>
- Hasibuan, S. S., Angraini, A., Saputra, E., & Megawati, M. (2024). Sentimen Analisis Terhadap Fitur Tiktok Menggunakan Naïve Bayes dan K-Nearest Neighbor. *Jurnal Media Informatika Budidarma*, 8(1), 303–311. <https://doi.org/10.30865/mib.v8i1.7238>
- Jin, Y., Cheng, K., Wang, X., & Cai, L. (2023). A Review of Text Sentiment Analysis Methods and Applications. *Frontiers in Business, Economics and Management*, 10(1), 58–64. <https://doi.org/10.54097/fbem.v10i1.10171>
- Kadhim, A. (2018). An Evaluation of Preprocessing Techniques for Text Classification. *International Journal of Computer Science and Information Security (IJCSIS)*, 16(6), 22–32. <https://sites.google.com/site/ijcsis/all-volumes-issues/vol-16-no-6-june-2018>
- Komarudin, A., & Hilda, A. M. (2024). Analisis Sentimen Ulasan Aplikasi Identitas Kependudukan Digital pada Play Store Menggunakan Metode Naïve Bayes. *Computer Science (CO-SCIENCE)*, 4(1), 28–36. <https://doi.org/10.31294/coscience.v4i1.2955>
- Nanda, S., Mualfah, D., & Fitri, D. A. (2022). Analisis Sentimen Kepuasan Pengguna Terhadap Layanan Streaming Mola Menggunakan Algoritma Random Forest. *Jurnal Aplikasi Teknologi Informasi dan Manajemen (JATIM)*, 3(2), 210–219. <https://doi.org/10.31102/jatim.v3i2.1592>
- Pahlevi, O., Amrin, A., & Handrianto, Y. (2023). Implementasi Algoritma Klasifikasi Random Forest untuk Penilaian Kelayakan Kredit. *Jurnal Infortech*, 5(1), 71–76. <https://doi.org/10.31294/infortech.v5i1.15829>



- Perdana, S. A., Florentin, S. F., & Santoso, A. (2022). Analisis Segmentasi Pelanggan Menggunakan K-Means Clustering Studi Kasus Aplikasi Alfagift. *Sebatik*, 26(2), 446–457. <https://doi.org/10.46984/sebatik.v26i2.1991>
- Pradana, M. (2016). Klasifikasi Bisnis E-Commerce di Indonesia. *MODUS*, 27(2), 163–174. <https://doi.org/10.24002/modus.v27i2.554>
- Sidauruk, N., Riza, N., & Fatonah, Rd. N. S. (2023). Penggunaan Metode SVM dan Random Forest untuk Analisis Sentimen Ulasan Pengguna Terhadap KAI Access di Google Play Store. *JATI (Jurnal Mahasiswa Teknik Informatika)*, 7(3), 1901–1906. <https://doi.org/10.36040/jati.v7i3.6899>
- Styawati, S., Hendrastuty, N., & Isnain, A. R. (2021). Analisis Sentimen Masyarakat Terhadap Program Kartu Prakerja pada Twitter dengan Metode Support Vector Machine. *Jurnal Informatika: Jurnal Pengembangan IT*, 6(3), 150–155. <https://doi.org/10.30591/jpit.v6i3.2870>
- Syakir, A., & Hasan, F. N. (2023). Analisis Sentimen Masyarakat Terhadap Perilaku Korupsi Pejabat Pemerintah Berdasarkan Tweet Menggunakan Naive Bayes Classifier. *Jurnal Media Informatika Budidarma*, 7(4), 1796–1805. <https://doi.org/10.30865/mib.v7i4.6648>
- Utami, A. F., Ekaputra, I. A., Japutra, A., & Van Doorn, S. (2022). The Role of Interactivity on Customer Engagement in Mobile E-Commerce Applications. *International Journal of Market Research*, 64(2), 269–291. <https://doi.org/10.1177/14707853211027483>
- Wicaksono, N. B. A. (2022). *Analisis Sentimen Tingkat Kepuasan Pengguna BPJS dengan Metode SentiStrength* [Universitas Atma Jaya Yogyakarta]. <https://repository.uajy.ac.id/id/eprint/27165/>
- Wilujeng, D. T., Fatekurohman, M., & Tirta, I. M. (2023). Analisis Risiko Kredit Perbankan Menggunakan Algoritma K-Nearest Neighbor dan Nearest Weighted K-Nearest Neighbor. *Indonesian Journal of Applied Statistics*, 5(2), 142–148. <https://doi.org/10.13057/ijas.v5i2.58426>



Imputasi Data Temperatur Maksimum Menggunakan Metode Support Vector Regression

Isa Kholifatus Sukhna ^{(1)*}, Brina Miftahurrohmah ⁽²⁾, Catur Wulandari ⁽³⁾, Putri Amelia ⁽⁴⁾
Departemen Sistem Informasi, Universitas Internasional Semen Indonesia, Gresik, Indonesia
e-mail : isa.sukhna20@student.uisi.ac.id,
{brina.miftahurrohmah,catur.wulandari,putri.amelia}@uisi.ac.id.

* Penulis korespondensi.

Artikel ini diajukan 18 Mei 2024, direvisi 22 Agustus 2024, diterima 22 Agustus 2024, dan dipublikasikan 31 Mei 2025.

Abstract

Temperature is a crucial element affecting various aspects, from agriculture to natural disasters. Temperature data imputation is also important because, in some cases, temperature data is not always complete. This study aims to predict missing temperature data in the East Nusa Tenggara (NTT) region using the Support Vector Regression (SVR) method. The data used comes from six BMKG observation stations in NTT and ERA-5 Reanalysis data. The choice of the SVR method is based on its ability to handle data with complex structures. Modeling is conducted separately for each station using the Radial Basis Function (RBF) kernel. Model evaluation employs the metrics Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and Coefficient of Determination (R^2), presenting the evaluation results with low error. The results show that among several parameter tests, the parameter ranges $[C = 1, 5, 10, 15]$, $[\epsilon = 0, 1, 0, 3, 0, 6, 0, 9]$, and $[\gamma = 1, 5, 10, 15]$ in the SVR method are the best parameter ranges across all stations. The prediction graphs display different temperature fluctuation patterns at each station. This study contributes to enhancing the availability of accurate climate data, supporting sustainable decision-making in the NTT region.

Keywords: Temperature, SVR, NTT, Reanalysis ERA-5, RBF

Abstrak

Temperatur merupakan elemen penting yang mempengaruhi berbagai aspek mulai dari pertanian hingga bencana alam. Imputasi data temperatur juga menjadi penting karena dalam beberapa kasus data temperatur tidak selalu lengkap. Penelitian ini bertujuan untuk memprediksi data temperatur yang hilang di wilayah Nusa Tenggara Timur (NTT) menggunakan metode Support Vector Regression (SVR). Data yang digunakan berasal dari 6 stasiun pengamatan BMKG di NTT dan data Reanalysis ERA-5. Pemilihan metode SVR didasarkan pada kemampuannya dalam menangani data dengan struktur kompleks. Pemodelan dilakukan secara terpisah pada setiap stasiun dengan menggunakan fungsi kernel Radial Basis Function (RBF). Evaluasi model menggunakan metrik Root Mean Square Error (RMSE), Mean Absolute Error (MAE), dan Coefficient of Determination (R^2) dengan menampilkan hasil evaluasi dengan error yang rendah. Hasil penelitian menunjukkan bahwa dari beberapa pengujian parameter, rentan parameter $[C = 1, 5, 10, 15]$, $[\epsilon = 0, 1, 0, 3, 0, 6, 0, 9]$, $[\gamma = 1, 5, 10, 15]$ dalam metode SVR merupakan rentan parameter terbaik pada semua stasiun. Grafik prediksi menampilkan pola fluktuasi temperatur yang berbeda-beda pada setiap stasiun. Penelitian ini berkontribusi dalam meningkatkan ketersediaan data iklim yang akurat untuk mendukung pengambilan keputusan yang berkelanjutan di wilayah NTT.

Kata Kunci: Temperatur, SVR, NTT, Reanalysis ERA-5, RBF

1. PENDAHULUAN

Iklim adalah salah satu komponen dalam ekosistem yang tidak terpisahkan dari kehidupan makhluk hidup. Iklim berkaitan erat dengan kondisi cuaca rata-rata di suatu wilayah tertentu selama periode waktu yang panjang (Malihah, 2022). Perubahan iklim memiliki dampak yang signifikan pada berbagai sektor, dan salah satunya adalah sektor pertanian. Di Indonesia,



pertanian memainkan peran penting dengan banyaknya penduduk yang bekerja di sektor tersebut. Sektor pertanian sangat rentan terhadap perubahan iklim karena perubahan tersebut dapat memengaruhi berbagai aspek seperti pola tanam, waktu tanam, produksi, dan kualitas hasil. Produksi pertanian dipengaruhi oleh faktor-faktor seperti peningkatan suhu, perubahan pola curah hujan, penguapan air, dan kelembapan tanah, yang semuanya berpotensi memengaruhi produktivitas (Harvian & Yuhan, 2021). Perubahan iklim ekstrem juga dapat berpotensi terjadinya bencana alam. Menurut laporan Badan Meteorologi, Klimatologi, dan Geofisika (BMKG) Stasiun Meteorologi El Tari Kupang, intensitas musim hujan di wilayah Nusa Tenggara Timur (NTT) hingga tahun 2022 sangat rendah dibandingkan dengan musim kemarau yang berlangsung selama 9 bulan. Selain itu pada tahun 2021, Kabupaten Alor merupakan salah satu daerah yang berpotensi mengalami kebakaran hutan. Berdasarkan data BPBD Provinsi Nusa Tenggara Timur yang dilaporkan oleh Kompas, dari Juni hingga Agustus 2021 terdapat 165 titik panas yang tersebar hampir merata di 22 kabupaten/kota di NTT. Titik panas ini diperkirakan akan terus menyebar ke wilayah timur mengikuti arah angin dan musim kemarau di wilayah NTT (Natalia & Firdaus, 2021).

Nusa Tenggara Timur (NTT) merupakan sebuah provinsi di Indonesia yang mencakup bagian timur Kepulauan Nusa Tenggara. Provinsi ini berbatasan dengan Laut Flores di utara, Samudera Hindia di selatan, Timor Leste di timur, dan Provinsi Nusa Tenggara Barat di barat. Ibu kota NTT adalah Kupang, yang terletak di antara $8^{\circ} - 12^{\circ}$ Lintang Selatan dan $118^{\circ} - 125^{\circ}$ Bujur Timur, serta terdiri dari 22 kabupaten/kota (BPK RI Perwakilan Provinsi Nusa Tenggara Timur, 2024). Penentuan Wilayah NTT sebagai tempat pelaksanaan penelitian didasari oleh karakteristik unik iklimnya yang menonjol dan fluktuasi temperatur udara yang sangat bervariasi sepanjang tahun. Faktor-faktor ini menjadikan NTT sebagai lokasi yang relevan dan signifikan untuk penelitian yang menitikberatkan pada prediksi temperatur. Selain itu, tersedia stasiun cuaca yang mewakili wilayah tersebut, yang dapat diandalkan sebagai sumber data historis temperatur.

Berbagai cara telah dikembangkan untuk menangani data yang tidak lengkap (*missing data*). Secara umum, penanganan ini dapat dibagi menjadi dua kategori utama yaitu metode imputasi dan metode amputasi. Dalam banyak situasi metode amputasi sering tidak disarankan dan metode imputasi cenderung lebih efektif. Beberapa penelitian juga menunjukkan bahwa penggunaan metode imputasi untuk mengatasi data yang hilang atau tidak lengkap dapat meningkatkan akurasi klasifikasi dibandingkan dengan tidak menggunakan metode imputasi sama sekali (Agwil et al., 2024). Terdapat beberapa penelitian terdahulu misalnya dari Jabir et al. (2023) yang melakukan imputasi menggunakan metode *mean imputation* di mana untuk mencari data yang hilang yaitu dengan melakukan rata-rata pada masing-masing variabel. Pada penelitian Laia (2023) melakukan imputasi menggunakan metode K-Nearest Neighbor *imputation* yaitu dengan melakukan pendekatan yang digunakan untuk melakukan klasifikasi pada data terdekat. Penelitian ketiga dari Sefidian & Daneshpour (2019) menggunakan kombinasi Teori Grey dan menggunakan model regresi Support Vector Regression (SVR) untuk melakukan imputasi. Dari beberapa penelitian tersebut menghasilkan penelitian dengan judul “Imputasi Data Temperatur Maksimum Pada Wilayah NTT Menggunakan Metode Support Vector Regression” karena tidak ada yang melakukan penelitian berupa imputasi data hilang pada Wilayah Nusa Tenggara Timur menggunakan metode Support Vector Regression.

Pemilihan perhitungan temperatur sebagai fokus penelitian disebabkan oleh kepentingan temperatur sebagai salah satu parameter iklim yang krusial dan mempunyai dampak besar di sejumlah sektor, termasuk pertanian, ketahanan pangan hingga bencana alam. Terdapatnya data *missing* merupakan hal yang sering terjadi, hal ini disebabkan oleh berbagai alasan seperti kerusakan alat, perhitungan yang tidak akurat, data yang tidak tercatat, cuaca yang tidak mendukung, serta berbagai faktor teknis lainnya (Prasetya et al., 2023; Septiansari et al., 2021). Ketersediaan informasi komponen cuaca dan iklim seperti temperatur sangat diperlukan baik dari segi data statistik maupun data spasialnya (Pratikno & Rahardjo, 2021). Ketersediaan data yang akurat sangat diperlukan untuk memahami dinamika iklim, melaksanakan peramalan cuaca yang tepat, serta untuk mendukung pengambilan keputusan yang lebih terinformasi di beragam aspek kehidupan. Dengan demikian, penelitian ini bertujuan untuk mengatasi kendala data temperatur



yang hilang di wilayah NTT dengan memanfaatkan metode Support Vector Regression untuk memprediksi data yang hilang. Diharapkan hasil penelitian akan memberikan kontribusi yang signifikan dalam memperbaiki ketersediaan data iklim yang akurat dan mendukung beragam upaya perencanaan dan pengambilan keputusan yang berkelanjutan di wilayah Nusa Tenggara Timur.

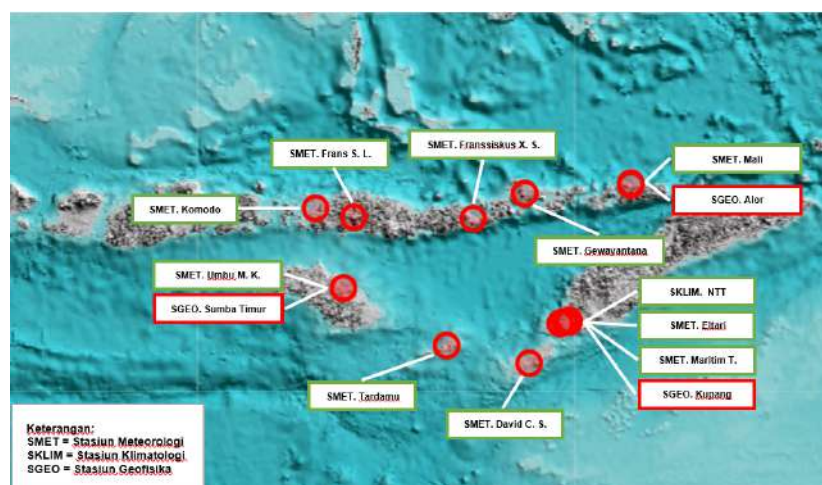
Dalam kerangka penelitian ini, metode Support Vector Regression (SVR) diaplikasikan dengan tujuan memprediksi data temperatur yang hilang di wilayah NTT. SVR merupakan salah satu metode *machine learning* di mana SVR dirancang untuk menyelesaikan masalah dari fungsi regresi untuk menghasilkan Keputusan berdasarkan kombinasi linier dari variabel yang diinputkan (Rahmi & Helma, 2023). Support Vector Regression (SVR) adalah model regresi yang merupakan pengembangan dari SVM, SVR mampu mengatasi masalah *overfitting* dengan cara meminimalkan batas atas dari *generalization error* (Ginting et al., 2021). SVR bertujuan untuk menemukan fungsi $f(x)$ dengan cara memetakan suatu data yang lebih tinggi memakai fungsi kernel. Pemilihan metode SVR dipilih dengan alasan kemampuannya untuk menangani data yang memiliki struktur yang kompleks dan beragam.

2. METODE PENELITIAN

Penelitian ini mengadopsi pendekatan kuantitatif dengan memanfaatkan bahasa pemrograman Python. Salah satu alat yang digunakan dalam proses analisis adalah Google Colaboratory (Google Colab), sebuah platform berbasis *cloud* yang dikembangkan oleh Google Research. Platform ini memungkinkan pengguna untuk menulis, menjalankan, serta berbagi kode Python secara gratis, sehingga sangat mendukung pelaksanaan eksperimen dan analisis data secara efisien (Gat et al., 2023).

2.1 Sumber Data

Data yang digunakan dalam penelitian ini merupakan data sekunder berupa data temperatur maksimum (derajat Celcius) yang bersumber dari data stasiun dan data *Reanalysis* ERA-5. Data stasiun diperoleh dari publikasi online yang dikeluarkan oleh Badan Meteorologi, Klimatologi, dan Geofisika (BMKG) melalui laman <https://dataonline.bmkg.go.id> dengan periode data temperatur harian yaitu 1 Januari 2020 hingga 31 Desember 2023. Wilayah yang menjadi lokasi penelitian adalah Provinsi NTT dengan 22 Kabupaten/Kota dan 14 stasiun pengamatan. Dari data yang diperoleh, tidak semua stasiun memiliki data temperatur maksimum. Oleh karena itu, pada penelitian ini hanya melakukan pengamatan di enam stasiun seperti yang disajikan pada Tabel 1. Sementara itu, data *Reanalysis* ERA-5 tersedia di situs Climate Data Store (CDS) yang dapat di akses melalui laman <https://cds.climate.copernicus.eu/> dalam format NetCDF dengan periode data temperatur yang diambil yaitu dari 1 Januari 2020 hingga 31 Desember 2023.



Gambar 1 Titik Lokasi Pengamatan



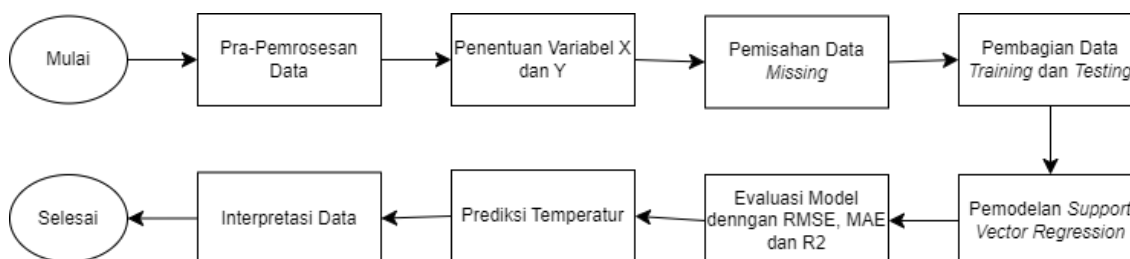
Berdasarkan Gambar 1 menampilkan titik lokasi pengamatan di Wilayah Nusa Tenggara Timur, terdapat 14 titik pengamatan, dengan 3 stasiun berwarna merah yang berarti tidak terdapat pengamatan pada stasiun pengamatan tersebut. Di sini diambil 6 stasiun pengamatan untuk dilakukan penelitian dengan pertimbangan stasiun dengan titik yang berdekatan akan diambil salah satu secara acak, untuk stasiun yang digunakan dapat dilihat pada Tabel 1. Pada Tabel 1 terdapat kolom Y, X1, X2, X3, X4, X5 sebagai struktur data yang digunakan. Variabel y merupakan variabel respon yang berasal dari data temperatur maksimum data stasiun dan variabel X1, X2, X3, X4, X5 merupakan variabel prediktor dari data temperatur maksimum *Reanalysis* ERA-5. Untuk data yang dijadikan sampel merupakan data pada 1 Januari 2020 pada setiap stasiun.

Tabel 1 Nama Stasiun Pengamatan

No.	Nama Stasiun Pengamatan	Jenis Data	Y	X1	X2	X3	X4	X5	
1	Meteorologi Fransiskus Xaverius Seda	Latitude Longitude	-8,63 122,23	32.8	28.31	28.14	28.59	28.64	28.90
2	Meteorologi Komodo	Latitude Longitude	-8,48 119,88	31.6	28.70	27.68	28.17	28.87	27.26
3	Meteorologi Mali	Latitude Longitude	-8,13 124,59	32.9	28.35	27.90	27.76	26.56	29.56
4	Meteorologi Tardamu	Latitude Longitude	-10,5 -9,66	32	28.86	29.09	28.78	29.05	29.04
5	Meteorologi Umbu Mehang Kunda	Latitude Longitude	-9,66 120,29	33.8	29.01	28.39	31.06	28.62	29.77
6	Meteorologi Maritim Tenau	Latitude Longitude	-10.18 123.52	NaN	31.23	28.81	29.53	31.42	29.32

2.2 Langkah Analisis Data

Tahapan penelitian yang dilakukan dalam studi ini digambarkan pada Gambar 2. Penelitian dimulai dengan tahap pra-pemrosesan data, yang mencakup data dari stasiun BMKG dan data *Reanalysis* ERA-5. Untuk data stasiun, proses diawali dengan pengumpulan data dari enam stasiun pengamatan. Sementara itu, pada data *Reanalysis* ERA-5, setelah data dikumpulkan dalam format NetCDF, dilakukan pencarian nilai temperatur maksimum menggunakan algoritma machine learning, kemudian dilakukan konversi skala temperatur dari Kelvin ke Celsius dengan cara mengurangkan 273,15. Selanjutnya, satuan waktu data diubah dari per jam menjadi harian agar selaras dengan data dari stasiun. Setelah semua data memiliki satuan waktu harian, format file diubah dari NetCDF menjadi CSV agar dapat digunakan dalam tahap pemrosesan berikutnya.

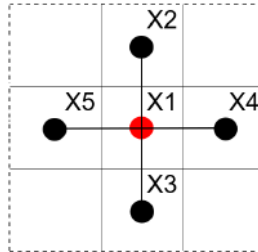


Gambar 2 Tahapan Penelitian

Penentuan variabel Y yaitu sebagai variable respon menggunakan data temperatur maksimum yang didapat dari enam stasiun di Provinsi NTT. Pemilihan enam stasiun ini didasarkan pada



ketersediaan data, karena terdapat beberapa data *missing* di stasiun pengamatan lainnya. Sedangkan untuk variabel X sebagai variabel prediktor menggunakan data temperatur maksimum *Reanalysis* ERA-5 pada *grid* yang berdekatan dengan titik variabel respons. Pada Gambar 3 merupakan hasil penomoran *grid* dengan bagian merah yang berada ditengah adalah *grid* yang tepat pada titik lokasi pengamatan.



Gambar 3 Penentuan *Grid* di Setiap Lokasi Pengamatan

Pada tahap pemisahan data *missing*, data dibagi menjadi dua kelompok, yaitu data yang memiliki nilai lengkap (tersedia) dan data yang mengandung nilai hilang (*missing*). Pemisahan ini bertujuan untuk memastikan bahwa model hanya dilatih menggunakan data yang valid dan lengkap. Data yang tersedia akan digunakan sebagai data latih (*training data*) dalam proses pembangunan model prediksi, sedangkan data yang memiliki nilai hilang akan digunakan untuk evaluasi performa model setelah pelatihan.

Pembagian data *training* dan *testing* menggunakan pembagian 75% untuk data *training* dan 25% data *testing* dari total 1.461 data temperatur maksimum yang tersedia. Alasan pemilihan rasio ini adalah karena beberapa stasiun pengamatan memiliki data yang hilang (*missing*) dalam jumlah yang cukup signifikan. Dengan menggunakan lebih banyak data untuk pelatihan (*training*), model dapat mempelajari pola data dengan lebih baik, sementara tetap menyisakan cukup data untuk pengujian (*testing*) guna mengevaluasi kinerja model secara akurat.

Pemodelan algoritma Support Vector Reggresion (SVR) menggunakan bahasa pemrograman Python karena memiliki *library* dan *framework* yang efisien, mudah digunakan dan *open source*. Pada penggunaan metode SVR biasanya dilakukan penentuan model parameter terbaik, pada penelitian ini penentuan model parameter dilakukan dengan optimasi *Grid Search*, *Grid Search Optimization* akan menguji berbagai kombinasi model dan memilih kombinasi dengan rata-rata kesalahan terkecil (Elsa, 2023; Syahfitri et al., 2024). Beberapa *hyperparameter* yang digunakan untuk *Grid Search* yaitu C, epsilon, dan gama yang diuji dengan nilai [1, 5, 10, 15] dan [0,1, 0,3, 0,6, 0,9] dan fungsi kernel yang digunakan merupakan fungsi kernel Radial Basis Function (RBF) sebagaimana dijabarkan pada Pers. (1) (Ramedani et al., 2014).

$$k(x_i, x_j) = \exp\left(-\frac{d(x_i, x_j)^2}{2l^2}\right) \quad (1)$$

2.2.1 Evaluasi Model

Evaluasi model menggunakan beberapa kriteria seperti Root Mean Square Error (RMSE) dan Mean Absolute Error (MAE) dengan hasil eror yang terkecil dan R2 Score dengan hasil eror yang paling besar. Adapun persamaan yang dapat digunakan untuk menghitung nilai evaluasi model seperti di bawah ini:

1) Root Mean Squared Error (RMSE)

RMSE menghitung akar kuadrat dari rata-rata kuadrat *error*, memberikan bobot yang lebih besar pada *error* yang lebih besar. Nilai RMSE yang lebih kecil menunjukkan bahwa model memiliki prediksi yang lebih akurat. Rumus perhitungan RMSE dijabarkan pada Pers. (2) (Hodson, 2022).



$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (\hat{y}_i - y_i)^2} \quad (2)$$

2) Mean Absolute Error (MAE)

MAE menghitung rata-rata dari selisih absolut antara nilai yang diprediksi dan nilai aktual dan memberikan gambaran tentang seberapa besar kesalahan prediksi model secara umum. Nilai MAE yang lebih kecil menunjukkan bahwa model memiliki kesalahan prediksi yang lebih kecil dan konsisten. Rumus perhitungan MAE dijabarkan pada Pers. (3) (Nurani et al., 2023).

$$MAE = \frac{1}{n} \sum_{i=1}^n |\hat{x}_i - x_i| \quad (3)$$

3) Coefficient of Determination (R²)

R² mengukur seberapa baik model dapat menjelaskan variasi dalam data, R² berkisar anantara 0 dan 1. Nilai yang lebih dekat dengan 1 menunjukkan bahwa model dapat menjelaskan sebagian besar variasi data. Rumus perhitungan R² dijabarkan pada Pers. (4) (Baek & Chung, 2023).

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \quad (4)$$

Dalam ketiga persamaan tersebut, n merupakan jumlah total sampel atau data yang digunakan. Simbol \hat{y}_i menunjukkan nilai prediksi dari variabel target y yang dihasilkan oleh model. Selain itu y adalah nilai actual atau sebenarnya dari variabel tersebut.

2.2.2 Imputasi Data

Pada tahap ini, proses dimulai dengan mengidentifikasi data yang hilang atau *missing* untuk memahami skala dan distribusi kekosongan dalam *dataset*, diikuti dengan analisis penyebab kekosongan tersebut guna menentukan apakah data yang hilang terjadi secara acak atau memiliki pola tertentu. Setelah penyebab diidentifikasi, selanjutnya yaitu melakukan implementasi untuk menggantikan data yang hilang dan hasilnya akan dievaluasi untuk memastikan bahwa imputasi yang dilakukan menampilkan hasil secara akurat.

3. HASIL DAN PEMBAHASAN

3.1 Pemodelan dan Prediksi Temperatur Maksimum Stasiun Tardamu

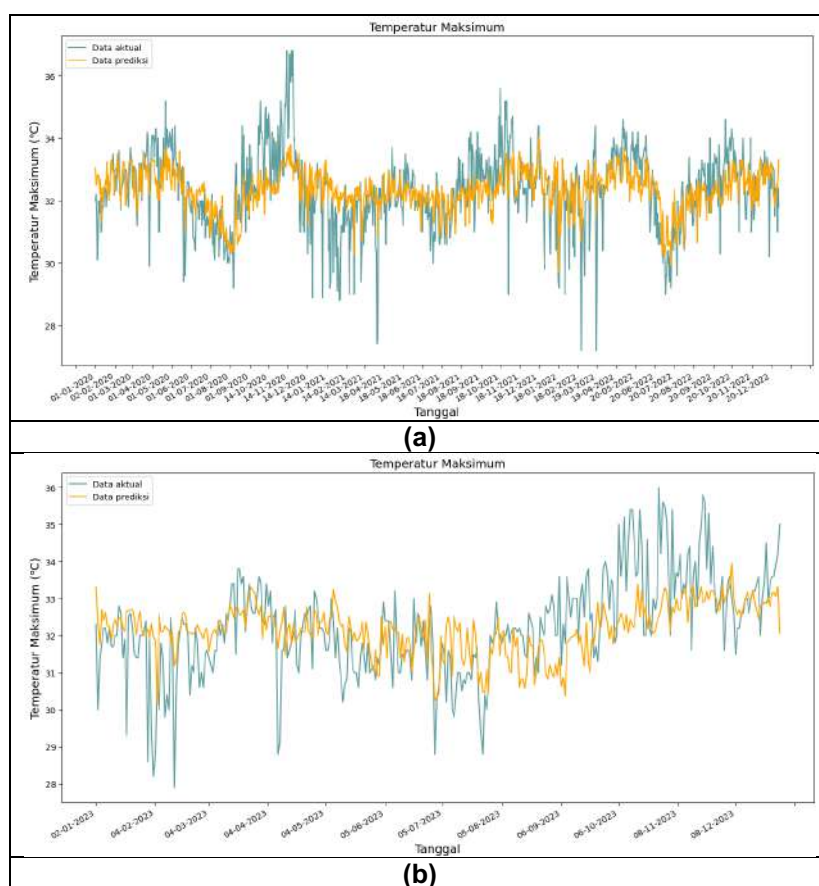
Bagian ini membahas pemodelan dan prediksi temperatur maksimum di Stasiun Tardamu dengan menggunakan berbagai kombinasi parameter untuk memperoleh model yang paling optimal. Pendekatan yang digunakan melibatkan eksplorasi parameter C (C), ε (epsilon), dan γ (gamma) yang berperan dalam mengatur kinerja model. Berdasarkan Tabel 2, hasil evaluasi menunjukkan bahwa kombinasi parameter terbaik diperoleh pada $C = 5$, $\varepsilon = 0,3$, dan $\gamma = 1$, dengan nilai RMSE sebesar 0,994, MAE sebesar 0,721, dan R^2 sebesar 0,356. Meskipun nilai R^2 tergolong rendah, kombinasi ini memberikan hasil terbaik dibandingkan parameter lainnya dalam memodelkan temperatur maksimum di stasiun tersebut.

Untuk memperjelas hasil evaluasi, Gambar 4 menyajikan grafik perbandingan antara data aktual dan hasil prediksi pada data training dan data testing. Pada Gambar 4(a), terlihat bahwa pola prediksi cukup mirip dengan data aktual, meskipun tidak menunjukkan fluktuasi besar yang sebenarnya terjadi dalam data. Sementara itu, Gambar 4(b) memperlihatkan bahwa pada awal tahun 2022 pola data aktual dan prediksi relatif sama, namun mulai Mei 2023 terjadi perbedaan pola yang semakin mencolok, terutama pada bulan September yang menunjukkan deviasi signifikan. Hal ini mengindikasikan bahwa model masih memiliki keterbatasan dalam menangkap dinamika data yang berubah secara musiman atau ekstrem.



Tabel 2 Evaluasi Model Stasiun Tardamu

Range Parameter	Kriteria Data Training			Kriteria Data Testing		
	RMSE	MAE	R2	RMSE	MAE	R2
$C = 1, 5, 10, 15$ $\varepsilon = 1, 5, 10, 15$ $\gamma = 1, 5, 10, 15$	1,000	0,754	0,347	1,121	0,841	0,275
$C = 1, 5, 10, 15$ $\varepsilon = 0,1, 0,3, 0,6, 0,9$ $\gamma = 0,1, 0,3, 0,6, 0,9$	1,029	0,755	0,310	1,127	0,842	0,267
$C = 1, 5, 10, 15$ $\varepsilon = 0,1, 0,3, 0,6, 0,9$ $\gamma = 1, 5, 10, 15$	0,994	0,721	0,356	1,154	0,865	0,232
Parameter Terbaik						
$C = 5$ $\varepsilon = 0,3$ $\gamma = 1$	0,994	0,721	0,356	1,154	0,865	0,232

Gambar 4 Evaluasi Model Stasiun Tardamu Data *Training* (a) dan Data *Testing* (b)

3.2 Pemodelan dan Prediksi Temperatur Maksimum Stasiun Umu Meheng Kunda

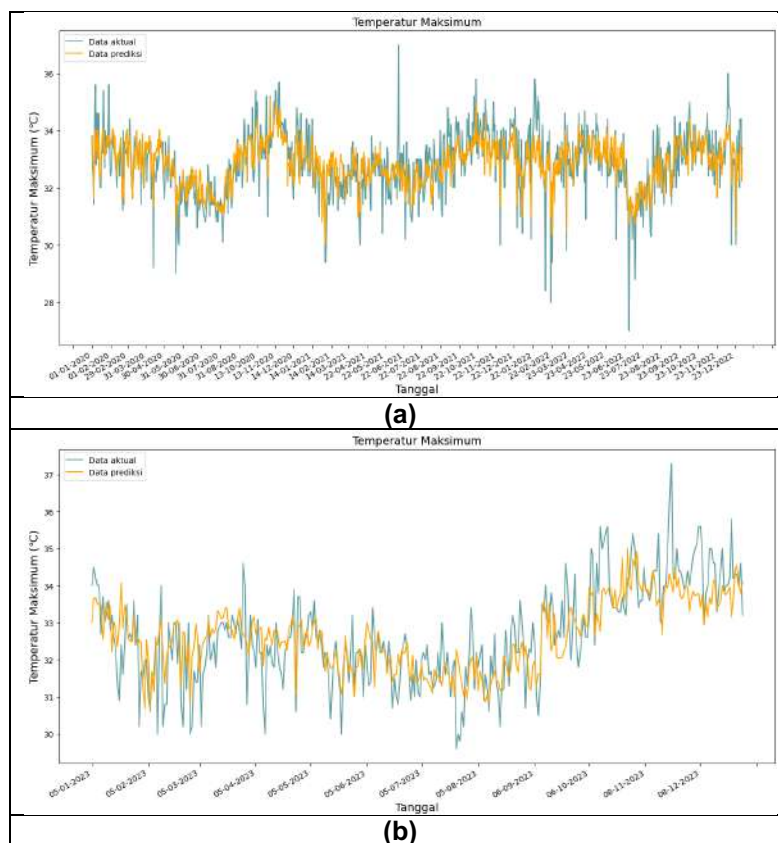
Bagian ini membahas pemodelan dan prediksi temperatur maksimum di Stasiun Umu Meheng Kunda dengan menggunakan berbagai kombinasi parameter melalui pendekatan eksploratif terhadap parameter C (C), ε (epsilon), dan γ (gamma) untuk memperoleh model yang paling optimal berdasarkan kriteria evaluasi tertentu. Berdasarkan Tabel 3, kombinasi parameter terbaik diperoleh pada $C = 1$, $\varepsilon = 0,6$, dan $\gamma = 0,1$, dengan nilai RMSE sebesar 0,718, MAE sebesar



0,551, dan nilai R^2 tertinggi sebesar 0,588. Hasil ini menunjukkan bahwa model memiliki performa yang cukup baik dalam memodelkan temperatur maksimum di stasiun tersebut.

Tabel 3 Evaluasi Model Stasiun Umbu Mehang Kunda

<i>Range Parameter</i>	<i>Kriteria Data Training</i>			<i>Kriteria Data Testing</i>		
	RMSE	MAE	R2	RMSE	MAE	R2
$C = 0,1, 0,3, 0,6, 0,9$ $\varepsilon = 0,1, 0,3, 0,6, 0,9$ $\gamma = 1, 5, 10, 15$ $C = 1, 5, 10, 15$	0,723	0,554	0,582	0,868	0,656	0,558
$\varepsilon = 0,1, 0,3, 0,6, 0,9$ $\gamma = 1, 5, 10, 15$ $C = 1, 5, 10, 15$	0,718	0,551	0,588	0,866	0,656	0,560
$\varepsilon = 1, 5, 10, 15$ $\gamma = 1, 5, 10, 15$	0,743	0,582	0,559	0,874	0,668	0,552
Parameter Terbaik						
$C = 1$ $\varepsilon = 0,6$ $\gamma = 1$	0,718	0,551	0,588	0,866	0,656	0,560



Gambar 5 Evaluasi Model Stasiun Umbu Mehang Kunda Data Training (a) dan Data Testing (b)

Gambar 5 memperlihatkan hasil prediksi model terhadap data training dan testing. Pada Gambar 5(a), prediksi cukup mendekati data aktual, menandakan kemampuan model mengikuti tren historis dengan baik. Namun, pada Gambar 5(b), terdapat beberapa periode di mana prediksi menyimpang dari data aktual, mengindikasikan keterbatasan model dalam menangkap variasi



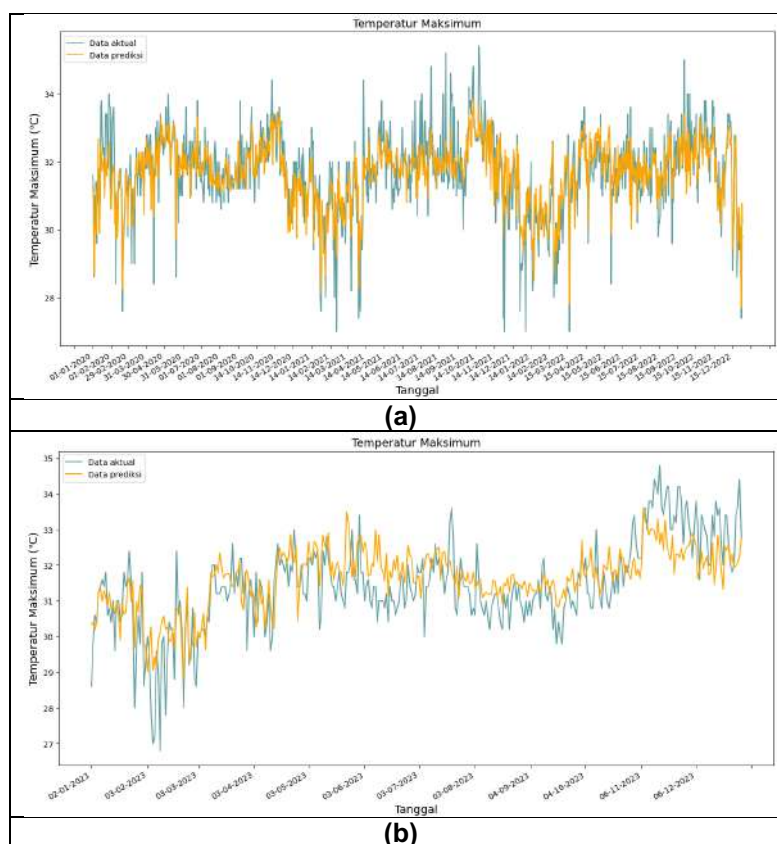
kompleks atau perubahan musiman. Secara keseluruhan, model ini cukup representatif untuk memprediksi temperatur maksimum di lokasi tersebut.

3.3 Pemodelan dan Prediksi Temperatur Maksimum Stasiun Komodo

Bagian ini membahas pemodelan dan prediksi temperatur maksimum di Stasiun Komodo dengan mengeksplorasi kombinasi parameter C (C), ε (epsilon), dan γ (gamma) untuk mendapatkan model optimal. Berdasarkan Tabel 4, kombinasi terbaik adalah $C = 1$, $\varepsilon = 0,3$, dan $\gamma = 1$, dengan RMSE 0,691, MAE 0,520, dan R^2 0,681, menunjukkan performa prediksi yang baik.

Tabel 4 Evaluasi Model Stasiun Komodo

Range Parameter	Kriteria Data <i>Training</i>			Kriteria Data <i>Testing</i>		
	RMSE	MAE	R2	RMSE	MAE	R2
$C = 0,1, 0,3, 0,6, 0,9$ $\varepsilon = 0,1, 0,3, 0,6, 0,9$ $\gamma = 1, 5, 10, 15$ $C = 1, 5, 10, 15$	0,696	0,523	0,676	0,862	0,674	0,523
$\varepsilon = 0,1, 0,3, 0,6, 0,9$ $\gamma = 0,1, 0,3, 0,6, 0,9$ $C = 1, 5, 10, 15$	0,712	0,542	0,661	0,814	0,636	0,575
$\varepsilon = 0,1, 0,3, 0,6, 0,9$ $\gamma = 1, 5, 10, 15$ $C = 1, 5, 10, 15$	0,691	0,520	0,681	0,861	0,674	0,524
Parameter Terbaik						
$C = 1$ $\varepsilon = 0,3$ $\gamma = 1$	0,691	0,520	0,681	0,861	0,674	0,524



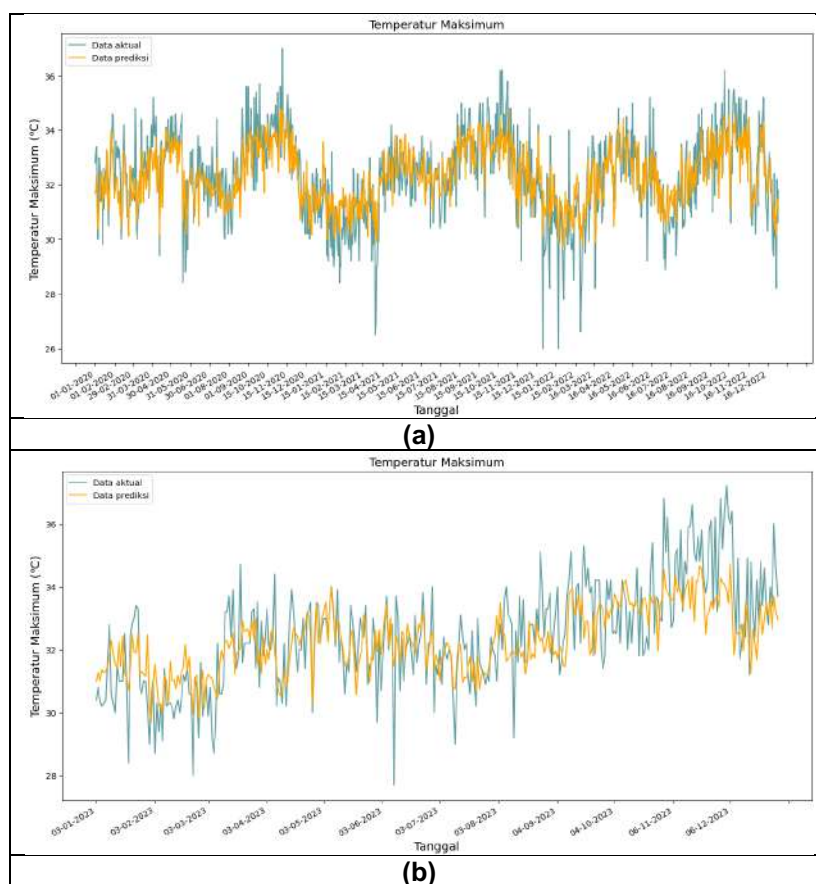
Gambar 6 Evaluasi Model Stasiun Komodo Data *Training* (a) dan Data *Testing* (b)



Gambar 6 memperlihatkan hasil prediksi pada data training dan testing. Pada Gambar 6(a), prediksi cukup konsisten dengan data aktual. Namun, pada Gambar 6(b), prediksi sedikit lebih tinggi dari data aktual di Februari 2023, sementara mulai November 2023 data aktual melebihi prediksi. Hal ini menunjukkan keterbatasan model dalam menangkap dinamika musiman dan perubahan pola mendadak.

3.4 Pemodelan dan Prediksi Temperatur Maksimum Stasiun Fransiskus Xavierius Seda

Bagian ini membahas proses pemodelan dan prediksi temperatur maksimum di Stasiun Fransiskus Xavierius Seda dengan pendekatan eksploratif terhadap berbagai kombinasi parameter. Gambar 7 memperlihatkan visualisasi hasil prediksi model dibandingkan dengan data aktual pada data training dan testing. Pada gambar tersebut, model menunjukkan kemampuan generalisasi yang cukup baik, dengan prediksi yang mendekati data aktual terutama pada data training, meskipun pada data testing terdapat beberapa deviasi akibat variasi ekstrem atau perubahan musiman.



Gambar 7 Evaluasi Model Stasiun Fransiskus Xavierius Seda Data *Training* (a) dan Data *Testing* (b)

Selanjutnya, berdasarkan Tabel 5, kombinasi parameter optimal diperoleh pada $C = 1$, $\varepsilon = 0,9$, dan $\gamma = 0,1$, dengan nilai RMSE sebesar 0,966, MAE sebesar 0,765, dan R^2 sebesar 0,601. Nilai-nilai ini mengindikasikan bahwa model berhasil mempelajari hubungan antara fitur masukan dan temperatur maksimum dengan cukup baik. Hasil visualisasi dan evaluasi kuantitatif ini bersama-sama memberikan gambaran yang komprehensif mengenai performa model pada stasiun tersebut.

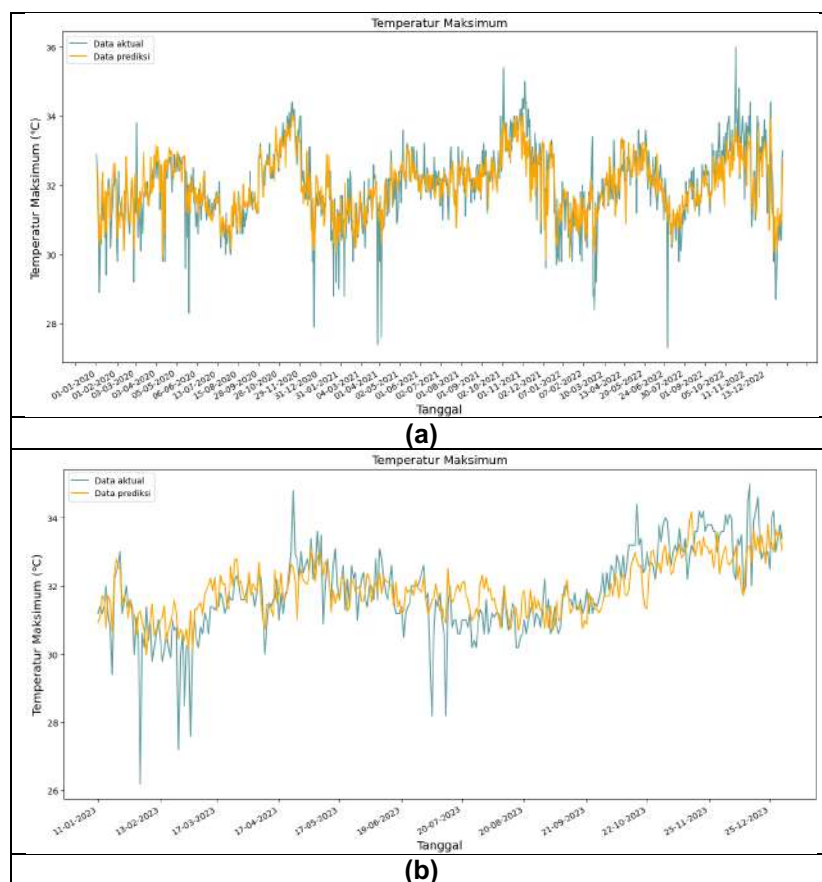


Tabel 5 Evaluasi Model Stasiun Fransiskus Xaverius Seda

Range Parameter	Kriteria Data Training			Kriteria Data Testing		
	RMSE	MAE	R2	RMSE	MAE	R2
$C = 0,1, 0,3, 0,6, 0,9$ $\varepsilon = 0,1, 0,3, 0,6, 0,9$ $\gamma = 1, 5, 10, 15$ $C = 1, 5, 10, 15$	0,971	0,767	0,597	1,175	0,928	0,508
$\varepsilon = 0,1, 0,3, 0,6, 0,9$ $\gamma = 1, 5, 10, 15$ $C = 1, 5, 10, 15$	0,966	0,765	0,601	1,174	0,927	0,509
$\varepsilon = 1, 5, 10, 15$ $\gamma = 1, 5, 10, 15$	0,969	0,768	0,599	1,175	0,928	0,508
Parameter Terbaik						
$C = 1$ $\varepsilon = 0,9$ $\gamma = 1$	0,966	0,765	0,601	1,174	0,927	0,509

3.5 Pemodelan dan Prediksi Temperatur Maksimum Stasiun Mali

Bagian ini membahas pemodelan temperatur maksimum di Stasiun Mali dengan mengeksplorasi berbagai kombinasi parameter. Gambar 8 menunjukkan perbandingan antara data aktual dan prediksi pada data training dan testing. Gambar 8(a) memperlihatkan bahwa model cukup baik mengikuti pola data, meskipun kurang menangkap fluktuasi tajam. Sementara itu, Gambar 8(b) menunjukkan model belum mampu memprediksi nilai ekstrem, seperti temperatur di bawah 30°C, meskipun data aktual mencapai 26°C.



Gambar 8 Evaluasi Model Stasiun Mali Data Training (a) dan Data Testing (b)



Tabel 6 menyajikan hasil evaluasi kombinasi parameter, dengan konfigurasi terbaik pada $C = 1$, $\varepsilon = 0,1$, dan $\gamma = 1$. Model ini menghasilkan RMSE 0,593, MAE 0,390, dan R^2 0,704. Hasil ini menunjukkan performa model cukup akurat dan lebih baik dibandingkan stasiun lainnya, meski masih perlu peningkatan dalam menangani variasi ekstrem.

Tabel 6 Evaluasi Model Stasiun Mali

Range Parameter	Kriteria Data Training			Kriteria Data Testing		
	RMSE	MAE	R2	RMSE	MAE	R2
$C = 0,1, 0,3, 0,6, 0,9$ $\varepsilon = 0,1, 0,3, 0,6, 0,9$ $\gamma = 1, 5, 10, 15$	0,599	0,395	0,698	0,835	0,619	0,533
$C = 1, 5, 10, 15$ $\varepsilon = 0,1, 0,3, 0,6, 0,9$ $\gamma = 1, 5, 10, 15$	0,593	0,390	0,704	0,834	0,619	0,535
$C = 1, 5, 10, 15$ $\varepsilon = 1, 5, 10, 15$ $\gamma = 0,1, 0,3, 0,6, 0,9$	0,663	0,512	0,631	0,729	0,541	0,644
Parameter Terbaik						
$C = 1$ $\varepsilon = 0,1$ $\gamma = 1$	0,593	0,390	0,704	0,834	0,619	0,535

3.6 Pemodelan dan Prediksi Temperatur Maksimum Stasiun Maritim Tenau

Bagian ini membahas pemodelan dan prediksi temperatur maksimum di Stasiun Maritim Tenau dengan mengeksplorasi berbagai kombinasi parameter untuk memperoleh model yang paling optimal. Pendekatan yang digunakan melibatkan pengujian parameter seperti C (C), ε (epsilon), dan γ (gamma) berdasarkan beberapa metrik evaluasi. Berdasarkan hasil evaluasi yang ditampilkan pada Tabel 7, kombinasi parameter terbaik diperoleh pada $C = 5$, $\varepsilon = 0,9$, dan $\gamma = 1$, yang menghasilkan nilai error terkecil. Dengan konfigurasi ini, model menghasilkan nilai RMSE sebesar 0,685, MAE sebesar 0,585, dan nilai R^2 tertinggi sebesar 0,721, yang menunjukkan bahwa model memiliki performa prediktif yang cukup baik untuk stasiun ini.

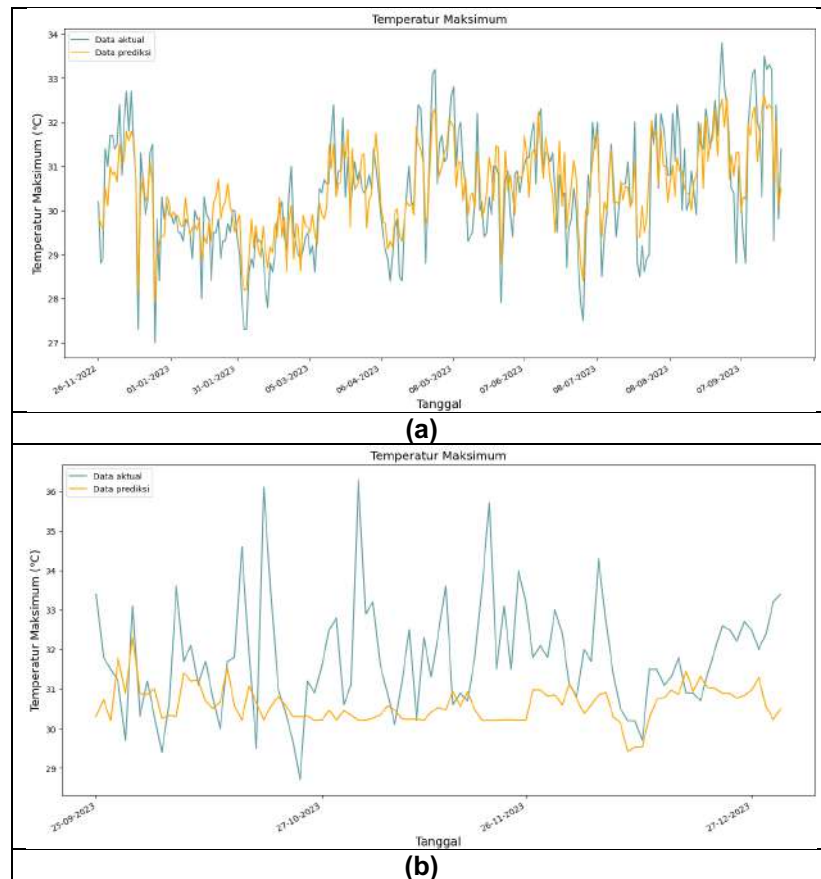
Tabel 7 Evaluasi Model Stasiun Maritim Tenau

Range Parameter	Kriteria Data Training			Kriteria Data Testing		
	RMSE	MAE	R2	RMSE	MAE	R2
$C = 0,1, 0,3, 0,6, 0,9$ $\varepsilon = 0,1, 0,3, 0,6, 0,9$ $\gamma = 1, 5, 10, 15$	0,714	0,568	0,698	1,862	1,381	-0,797
$C = 1, 5, 10, 15$ $\varepsilon = 0,1, 0,3, 0,6, 0,9$ $\gamma = 1, 5, 10, 15$	0,685	0,585	0,721	1,897	1,409	-0,866
$C = 1, 5, 10, 15$ $\varepsilon = 1, 5, 10, 15$ $\gamma = 1, 5, 10, 15$	0,720	0,611	0,693	1,905	1,419	-0,882
Parameter Terbaik						
$C = 1$ $\varepsilon = 0,9$ $\gamma = 1$	0,685	0,585	0,721	1,897	1,409	-0,866

Untuk memperjelas hasil evaluasi, Gambar 9 menyajikan visualisasi perbandingan antara data aktual dan hasil prediksi pada data training dan testing. Pada Gambar 9(a), tampak bahwa prediksi model cukup mengikuti pola data aktual pada data training. Namun, pada Gambar 9(b), terlihat bahwa data aktual mengalami fluktuasi naik-turun yang tidak beraturan, sedangkan hasil



prediksi cenderung linier dan kurang mampu menangkap dinamika ekstrem dari temperatur maksimum. Hal ini menunjukkan bahwa meskipun performa model secara umum cukup baik, masih terdapat keterbatasan dalam menangkap variasi ekstrem yang muncul pada data aktual.



Gambar 9 Evaluasi Model Stasiun Maritim Tenau Data *Training* (a) dan Data *Testing* (b)

4. KESIMPULAN

Hasil penelitian menunjukkan bahwa metode Support Vector Regression (SVR) efektif dalam memprediksi temperatur maksimum harian di wilayah Nusa Tenggara Timur. Metode ini mampu menangani pola data yang terbatas (tidak lengkap) dan bersifat non-linear dengan baik. Evaluasi menggunakan metrik RMSE, MAE, dan R^2 menunjukkan bahwa SVR menghasilkan prediksi yang mendekati nilai sebenarnya dengan tingkat kesalahan yang rendah, terutama saat menggunakan kombinasi parameter terbaik. Parameter terbaik untuk setiap stasiun berada dalam rentang $C = [1, 5, 10, 15]$, $\varepsilon = [0,1, 0,3, 0,6, 0,9]$, dan $\gamma = [1, 5, 10, 15]$.

Meskipun SVR terbukti efektif dalam mengatasi imputasi data missing temperatur maksimum di wilayah NTT, masih terdapat peluang untuk pengembangan lebih lanjut. Penelitian selanjutnya disarankan membandingkan metode SVR dengan teknik regresi lain guna menentukan metode paling optimal dalam menangani data klimatologis yang tidak lengkap. Selain itu, eksplorasi parameter lain, seperti jenis kernel atau teknik optimasi yang berbeda, dapat dipertimbangkan untuk meningkatkan akurasi prediksi dan kemampuan generalisasi model.

DAFTAR PUSTAKA

Agwil, W., Agustina, D., Fransiska, H., & Hasani, I. A. (2024). Meningkatkan Kinerja Model Klasifikasi Curah Hujan Melalui Penanggulangan Missing Value dengan Imputasi Berbasis



Artikel ini didistribusikan mengikuti lisensi Atribusi-NonKomersial CC BY-NC sebagaimana tercantum pada <https://creativecommons.org/licenses/by-nc/4.0/>.

- Model. *INNOVATIVE: Journal of Social Science Research*, 4(1), 11773–11783. <https://doi.org/10.31004/innovative.v4i1.9158>
- Baek, J. W., & Chung, K. (2023). Multi-Context Mining-Based Graph Neural Network for Predicting Emerging Health Risks. *IEEE Access*, 11, 15153–15163. <https://doi.org/10.1109/ACCESS.2023.3243722>
- BPK RI Perwakilan Provinsi Nusa Tenggara Timur. (2024). *Pemerintah Provinsi NTT*. BPK RI. <https://ntt.bpk.go.id/pemerintah-provinsi-ntt/>
- Elsa, R. (2023). *Penerapan Metode Support Vector Regression (SVR) Menggunakan Kernel Linear, Polinomial, dan Radial dengan Grid Search Optimization* [Universitas Lampung]. <https://digilib.unila.ac.id/74539/3/SKRIPSI%20TANPA%20BAB%20PEMBAHASAN.pdf>
- Gat, G., Hidayatullah, A., & Berliana, A. (2023). Workshop Pengenalan Dasar Pemrograman Python dengan Google Colaboratory. *Prosiding ABDIMAS CORISINDO, 2023*, 65–70. <https://ojs.stmikpontianak.ac.id/index.php/pengabdian/article/view/74>
- Ginting, L. M., Sigiuro, M. MT., Manurung, E. D., & Sinurat, J. J. P. (2021). Perbandingan Metode Algoritma Support Vector Regression dan Multiple Linear Regression untuk Memprediksi Stok Obat. *Journal of Applied Technology and Informatics Indonesia*, 1(2), 29–34. <https://doi.org/10.54074/jati.v1i2.36>
- Harvian, K. A., & Yuhan, R. J. (2021). Kajian Perubahan Iklim Terhadap Ketahanan Pangan. *Seminar Nasional Official Statistics*, 1052–1061. <https://doi.org/10.34123/semnasoffstat.v2020i1.593>
- Hodson, T. O. (2022). Root-Mean-Square Error (RMSE) or Mean Absolute Error (MAE): When to Use Them or Not. *Geoscientific Model Development*, 15(14), 5481–5487. <https://doi.org/10.5194/gmd-15-5481-2022>
- Jabir, S. R., Azis, H., Widyawati, D., & Tenripada, A. U. (2023). Prediksi Potensi Donatur Menggunakan Model Logistic Regression. *Indonesian Journal of Data and Science*, 4(1), 31–37. <https://doi.org/10.56705/ijodas.v4i1.64>
- Laia, M. (2023). Analisis Kinerja Algoritma K-Nearest Neighbor Imputation (KNNI) untuk Missing Value pada Klasifikasi Data Mining. *Journal of Informatics, Electrical and Electronics Engineering*, 2(3), 92–98. <https://doi.org/10.47065/jieeee.v2i3.891>
- Malihah, L. (2022). Tantangan Dalam Upaya Mengatasi Dampak Perubahan Iklim dan Mendukung Pembangunan Ekonomi Berkelanjutan: Sebuah Tinjauan. *Jurnal Kebijakan Pembangunan*, 17(2), 219–232. <https://doi.org/10.47441/jkp.v17i2.272>
- Natalia, F., & Firdaus, I. (2021, August 11). *Kebakaran Lahan di NTT Capai 165 Titik Panas, Stasiun Meteorologi: Ada Praktik Pembakaran Ladang*. Kompas.TV. <https://www.kompas.tv/bisnis/200652/kebakaran-lahan-di-ntt-capai-165-titik-panas-stasiun-meteorologi-ada-praktik-pembakaran-ladang>
- Nurani, A. T., Setiawan, A., & Susanto, B. (2023). Perbandingan Kinerja Regresi Decision Tree dan Regresi Linear Berganda untuk Prediksi BMI pada Dataset Asthma. *Jurnal Sains dan Edukasi Sains*, 6(1), 34–43. <https://doi.org/10.24246/juses.v6i1p34-43>
- Prasetya, M. R. A., Priyatno, A. M., & Nurhaeni. (2023). Penanganan Imputasi Missing Values pada Data Time Series dengan Menggunakan Metode Data Mining. *Jurnal Informasi dan Teknologi*, 5(2), 52–62. <https://doi.org/10.37034/jidt.v5i2.324>
- Pratikno, A., & Rahardjo, M. (2021). Analisis Strategi Peningkatan Kinerja Perusahaan dengan Pendekatan SWOT-Balanced Scorecard pada PT IPSI Karya Abadi. *Jurnal Manajemen Bisnis dan Kewirausahaan*, 5(2), 184–189. <https://doi.org/10.24912/jmbk.v5i2.11229>
- Rahmi, A., & Helma, H. (2023). Portofolio Optimal dengan Mempertimbangkan Prediksi Return Menggunakan Metode Support Vector Regression (SVR) Program Studi Matematika. *Jurnal Pendidikan Tambusai*, 7(3), 23745–23753. <https://doi.org/10.31004/jptam.v7i3.10380>
- Ramedani, Z., Omid, M., Keyhani, A., Shamshirband, S., & Khoshnevisan, B. (2014). Potential of Radial Basis Function Based Support Vector Regression for Global Solar Radiation Prediction. *Renewable and Sustainable Energy Reviews*, 39, 1005–1011. <https://doi.org/10.1016/j.rser.2014.07.108>
- Sefidian, A. M., & Daneshpour, N. (2019). Missing Value Imputation Using a Novel Grey Based Fuzzy C-Means, Mutual Information Based Feature Selection, and Regression Model. *Expert Systems with Applications*, 115, 68–94. <https://doi.org/10.1016/j.eswa.2018.07.057>



- Septiansari, A. P. D., Zakaria, A., Khotimah, S. N., & Romdania, Y. (2021). Analisis Data Curah Hujan yang Hilang dengan Menggunakan Metode Normal Ratio, Inversed Square Distance, Rata-Rata Aljabar, dan Linear Regression (Studi Kasus Data Curah Hujan Beberapa Stasiun Hujan Wilayah Lampung Tengah). *Jurnal Rekayasa Sipil dan Desain*, 9(4), 853–862. <https://doi.org/10.23960/jrsdd.v9i4.2219>
- Syahfitri, N., Amalita, N., Vionanda, D., & Martha, Z. (2024). Forecasting Gold Prices in Indonesia Using Support Vector Regression with the Grid Search Algorithm. *UNP Journal of Statistics and Data Science*, 2(1), 32–39. <https://doi.org/10.24036/ujsds/vol2-iss1/145>



Perbandingan Random Forest dan Convolutional Neural Network dalam Memprediksi Peralihan Pelanggan

Dewa Adji Kusuma ^{(1)*}, Atika Ratna Dewi ⁽²⁾, Andreas Rony Wijaya ⁽³⁾

^{1,2} Departemen Teknik Informatika, Institut Teknologi Telkom Purwokerto, Purwokerto, Indonesia

³ Departemen Statistika, Universitas Sebelas Maret, Surakarta, Indonesia
e-mail : {dewaadji12,andreasronywijaya}@gmail.com, atika@ittelkom-pwt.ac.id.

* Penulis korespondensi.

Artikel ini diajukan 21 Mei 2024, direvisi 19 Agustus 2024, diterima 20 Agustus 2024, dan dipublikasikan 31 Mei 2025.

Abstract

The rapid growth of the telecommunications industry has increased competition among companies for customers. As a result, customers often switch to other services or terminate their subscriptions. Retaining customers is very important as it is 10 times cheaper than acquiring new customers. This study compares Random Forest (RF) and Convolutional Neural Network (CNN) algorithms in predicting customer switching, using Correlation-based Feature Selection (CFS) and Recursive Feature Elimination (RFE) for data partitioning. Model evaluation using Confusion Matrix and Area Under Curve (AUC). The evaluation results show that the performance of CNN models with optimization parameters is superior. Using the CFS dataset, the test data evaluation results yielded an accuracy of 98%, AUC of 0.96, precision of 99%, recall of 92%, and F1-score of 96%. The best tuning result for CNN is achieved with three combinations of filter and kernel sizes {[64, 7], [32, 3], [16, 2]} and a pool size of 2. A limitation of this research is determining how to compare the two algorithms being evaluated effectively. Both use different approaches, namely Supervised Learning and Deep Learning.

Keywords: CNN, Customer Churn, Data Mining, Prediction, Random Forest

Abstrak

Pertumbuhan industri telekomunikasi yang pesat telah meningkatkan persaingan antar perusahaan untuk mendapatkan pelanggan. Akibatnya, pelanggan sering berpindah layanan atau menghentikan langganan. Mempertahankan pelanggan sangat penting karena 10 kali lebih murah daripada mengakuisisi pelanggan baru. Penelitian ini membandingkan algoritma Random Forest (RF) dan Convolutional Neural Network (CNN) dalam memprediksi perpindahan pelanggan, menggunakan Correlation-based Feature Selection (CFS) dan Recursive Feature Elimination (RFE) untuk partisi data. Evaluasi model menggunakan *confusion matrix* dan Area Under Curve (AUC). Hasil evaluasi menunjukkan performa model CNN dengan parameter optimasi lebih unggul. Menggunakan *dataset* CFS, hasil evaluasi data *test* mendapatkan akurasi sebesar 98%, AUC 0,96, *precision* 99%, *recall* 92%, dan *F1-score* 96%. Hasil *tuning* terbaik untuk CNN adalah 3 kombinasi filter dan *kernel* {[64&7], [32&3], [16&2]} dan *pool_size*=2. Keterbatasan penelitian ini adalah bagaimana cara menanganai kedua algoritma yang dibandingkan. Keduanya menggunakan pendekatan yang berbeda, yakni *supervised learning* dan *deep learning*.

Kata Kunci: CNN, Data Mining, Peralihan Pelanggan, Prediksi, Random Forest

1. PENDAHULUAN

Globalisasi dan perkembangan industri telekomunikasi yang pesat menyebabkan jumlah operator di pasar meningkat secara signifikan, menciptakan persaingan yang ketat (Lalwani et al., 2022). Menurut data survei Asosiasi Penyelenggara Jasa Internet Indonesia (APJII), pengguna internet di Indonesia pada periode 2022–2023 mencapai 215,63 juta orang. Selain itu, Badan Pusat Statistik (BPS) mencatat bahwa pada tahun 2021 terdapat 611 perusahaan penyedia layanan internet di Indonesia. Banyaknya perusahaan kompetitor di sektor telekomunikasi memaksa perusahaan untuk bersaing melalui strategi pemasaran yang efektif. Dampak persaingan ini adalah terjadinya perpindahan pelanggan atau *customer churn*, yaitu kondisi di mana pelanggan



berhenti menggunakan layanan suatu operator karena berbagai alasan. Fenomena ini menjadi masalah serius karena pelanggan dianggap sebagai aset paling berharga oleh perusahaan, sehingga berbagai upaya dilakukan untuk mempertahankan pelanggan agar tetap berlangganan (de Lima Lemos et al., 2022; Wicaksono et al., 2021; Wu, 2024).

Mempertahankan pelanggan yang sudah ada terbukti lebih hemat, dengan biaya hingga 10 kali lebih murah dibandingkan upaya menarik pelanggan baru melalui strategi pemasaran (Gabhane et al., 2022; Ribeiro et al., 2024). Oleh karena itu, prediksi yang akurat mengenai peralihan pelanggan sangat penting untuk mendukung strategi loyalitas, perencanaan pemasaran yang efektif, serta menghasilkan penghematan biaya signifikan bagi penyedia layanan (Husein & Harahap, 2021; Li et al., 2024). Pelanggan menjadi sumber utama pendapatan bagi perusahaan telekomunikasi (AL-Najjar et al., 2022; Jain et al., 2021; Suryana et al., 2021). Dengan kemampuan memprediksi pelanggan yang berpotensi berpindah, perusahaan dapat melakukan antisipasi melalui kebijakan yang sudah dirancang, misalnya dengan memberikan layanan khusus atau insentif agar pelanggan tetap setia. Tujuan utama prediksi peralihan pelanggan adalah membantu menetapkan strategi retensi pelanggan, sekaligus mengidentifikasi sinyal awal peralihan sehingga perusahaan dapat melakukan tindakan preventif (Artha et al., 2022; Lalwani et al., 2022; Muthmainah & Cholil, 2022).

Penelitian terkait peralihan pelanggan umumnya fokus pada upaya memenangkan pelanggan di industri telekomunikasi, di mana pelanggan sangat berpengaruh terhadap pendapatan dan kapitalisasi pasar. Misalnya, penelitian Suryana et al. (2021) menggunakan metode sampling Synthetic Minority Over-sampling Technique (SMOTE) dan Boosting untuk memprediksi pelanggan yang berpindah layanan, dengan algoritma Random Forest menunjukkan akurasi unggul sebesar 89,19%. Husein & Harahap (2021) membandingkan lima model klasifikasi, yaitu K-Nearest Neighbors, Logistic Regression, Linear SVC, Random Tree, dan Random Forest, dengan hasil terbaik pada Random Forest dan K-Nearest Neighbors masing-masing mencapai akurasi 86% dan 84%. Studi tersebut juga menemukan bahwa tingkat peralihan pelanggan wanita lebih tinggi dibanding pria, dan faktor usia berpengaruh signifikan terhadap *churn*.

Penelitian Rahman & Kumar (2020) membandingkan algoritma KNN, SVM, Decision Tree, dan Random Forest, dan menemukan bahwa Random Forest dengan *oversampling* memberikan akurasi terbaik. Gabhane et al. (2022) menggunakan teknologi *deep learning* seperti Artificial Neural Network (ANN) dan Convolutional Neural Network (CNN) untuk mendeteksi *churn*, dengan CNN menunjukkan presisi lebih tinggi sebesar 97,78%. Ojo (2024) membangun tiga model prediksi *churn*, dua model Multi-layer Perceptron (MLP) dan satu CNN, dengan CNN mencapai akurasi 89%, lebih baik dibanding MLP yang hanya 80%. Yahaya et al. (2021) menggabungkan ANN dengan filter Genetic Algorithm (GA) dan K-Means untuk memprediksi *churn*, menemukan bahwa pemfilteran *noise* meningkatkan kinerja model, walaupun data tidak seimbang tetap mempengaruhi hasil.

Meskipun algoritma Random Forest dan CNN terbukti unggul, terdapat beberapa kekurangan dalam penelitian sebelumnya. Pertama, belum ada studi yang membandingkan performa kedua algoritma ini secara langsung, khususnya dalam konteks data perusahaan telekomunikasi. Kedua, karakteristik data yang berbeda yang memengaruhi kinerja masing-masing algoritma belum banyak diperhatikan.

Berdasarkan tinjauan tersebut, algoritma Random Forest dan CNN dipilih dalam penelitian ini karena memiliki performa klasifikasi dan prediksi *churn* yang baik. Random Forest efektif mengelola data dengan fitur beragam dan cenderung mengatasi *overfitting*, serta menyediakan informasi penting terkait kontribusi fitur dalam klasifikasi. Sementara itu, CNN mampu mengekstraksi fitur penting secara otomatis dari data berstruktur kompleks dan menangkap pola tersembunyi yang sulit dideteksi oleh algoritma tradisional. Evaluasi model menggunakan metrik akurasi, *precision*, *recall*, *F1-score*, dan Area Under Curve (AUC) diharapkan memberikan wawasan terbaik mengenai algoritma yang paling cocok untuk prediksi *customer churn* pada perusahaan telekomunikasi.

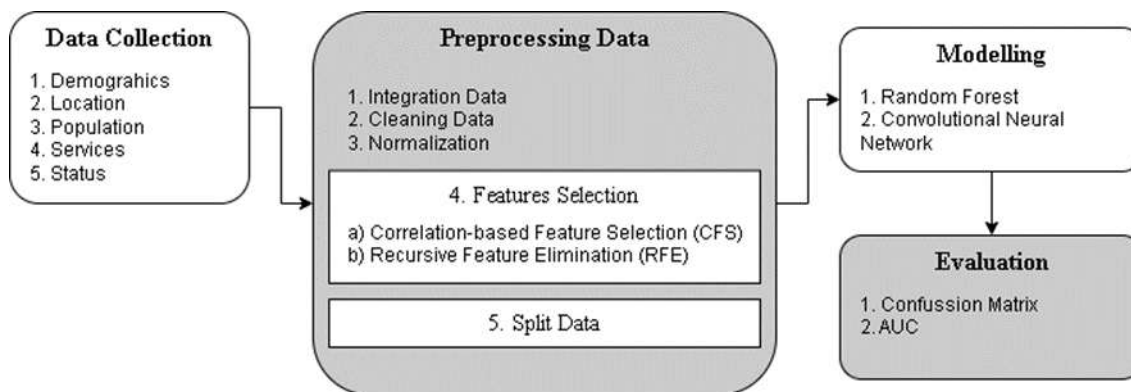


Tujuan penelitian ini adalah menganalisis dan menentukan algoritma terbaik antara Random Forest dan CNN dalam memprediksi *customer churn*. Kedua algoritma dipilih karena memiliki akurasi tinggi dan stabil menurut studi sebelumnya. Meskipun keduanya menggunakan pendekatan yang berbeda—*supervised learning* dan *deep learning*—hasil yang diperoleh dapat menjadi referensi penting untuk prediksi *churn* nyata, sehingga perusahaan dapat mengambil langkah antisipasi untuk mempertahankan pelanggan.

2. METODE PENELITIAN

2.1 Tahapan Penelitian

Untuk membantu proses penelitian, dibuat tahapan atau alur penelitian agar lebih teratur dan sistematis. Fokus utamanya adalah mengidentifikasi dan merencanakan proses yang perlu dilakukan untuk mencapai tujuan dalam penelitian. Selain itu diagram dapat digunakan sebagai alat evaluasi untuk mengevaluasi efektivitas dan efisiensi pelaksanaan penelitian. Gambar 1 menjelaskan alur penelitian yang dilakukan dalam penelitian ini.



Gambar 1 Alur Penelitian

Pengumpulan data dilakukan melalui laman *community IBM*, data yang diambil adalah data tentang *Telco customer churn* dari perusahaan telekomunikasi yang menyediakan layanan internet dan telepon rumah. *Dataset customer churn* berformat *zip* dengan beberapa *file* di dalamnya mencakup demografis, *location*, populasi, layanan, dan status. Tiap *file* memiliki jumlah baris 7.044 dengan jumlah kolom yang berbeda-beda. Semua *file* digabung menjadi satu dengan *customer_id* sebagai kolom pembandingnya.

Selanjutnya tahapan *preprocessing data*, proses ini bertujuan untuk membersihkan, mengatur, dan mengubah data agar dapat diolah menggunakan model algoritma. Pada *intergration* data dilakukan penggabungan data menjadi satu agar akurat dan lengkap. Hasil akhir yang didapat adalah sebuah tabel dengan 40 kolom dan 7.044 baris. *Cleaning data* untuk membersihkan data dari *noise*, *missing data*, dan data yang tidak relevan untuk menjaga kualitas *dataset* dari bias. *Normalization* digunakan untuk mengubah data menjadi bentuk yang seragam dan standar.

Features selection digunakan untuk seleksi variabel yang akan digunakan pada model algoritma. Teknik yang digunakan pada proses ini adalah menggunakan Correlation-based Feature Selection (CFS) dan Recursive Feature Elimination (RFE). Algoritma yang digunakan pada proses RFE adalah Decision Tree. Algoritma tersebut dipilih karena hasil *feature* yang tereliminasi dinilai berbeda dibandingkan hasil CFS. *Split data* untuk membagi *dataset* menjadi data latih, data uji, dan data *test*. Pada proses ini menggunakan tiga komposisi pembagian data yaitu 60/40, 70/30, dan 80/20. Tujuan dari menggunakan tiga komposisi pembagian data adalah untuk menguji sensitivitas algoritma terhadap perbedaan data yang akan diuji, dan menganalisis optimalisasi rasio antara data latih dan uji. Penggunaan *multiple split ratios* dapat memberikan pemahaman yang lebih komprehensif tentang hasil evaluasi model.



Tahap *modelling* menggunakan algoritma Random Forest dan CNN berdasarkan dua *dataset* hasil *feature selection* serta menggunakan CFS dan RFE. Random Forest memiliki kemampuan untuk mengklasifikasi dengan menggabungkan beberapa pohon keputusan independen. CNN memiliki kemampuan dalam melihat pola-pola khusus dalam *dataset* dengan mengubahnya ke dalam bentuk matriks seperti data *sequence*, kemudian melakukan konvolusi dan *pooling* untuk mengidentifikasi fitur.

Random Forest menggunakan beberapa pohon keputusan, sehingga setiap pohon bergantung pada hasil nilai vektor acak yang dipilih secara individual menggunakan distribusi yang sama untuk semua pohon keputusan (Rahman & Kumar, 2020; Singh et al., 2021). Random Forest dimulai dengan menghitung nilai *entropy* sebagai penentu ketidakmurnian atribut dan *information gain* (Sandag, 2020). Rumus dari *entropy* dan *information gain* dinyatakan pada Pers. (1) dan (2).

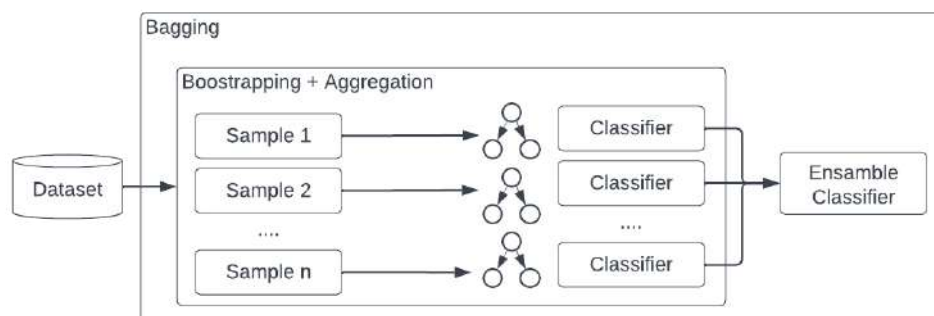
$$Entropy(Y) = - \sum_i p(c|Y) \log^2 p(c|Y) \quad (1)$$

$$InfGain(Y, a) = (Y) - \sum_i v \epsilon Values(a) \frac{|Y_v|}{|Y_a|} (Y_v) \quad (2)$$

Entropy Y merupakan himpunan kasus, dan $p(c|Y)$ adalah proporsi data dalam Y yang termasuk dalam kelas c . Formula ini digunakan untuk mengukur tingkat ketidakpastian dalam distribusi kelas yang tidak terdapat dalam himpunan data. Nilai *information gain* dari atribut a terhadap himpunan data Y dihitung dengan mengurangi nilai *entropy* total Y dengan rata-rata *entropy* dari setiap subset Y_v , yang dibagi dengan nilai dalam atribut a . Dalam hal ini, Y_v adalah subhimpunan dari Y yang memiliki nilai tertentu pada atribut a , dan Y_a adalah seluruh data yang memiliki nilai untuk atribut tersebut. Nilai $\frac{|Y_v|}{|Y_a|}$ adalah proporsi dari masing-masing *subset* terhadap keseluruhan data yang memiliki atribut a .

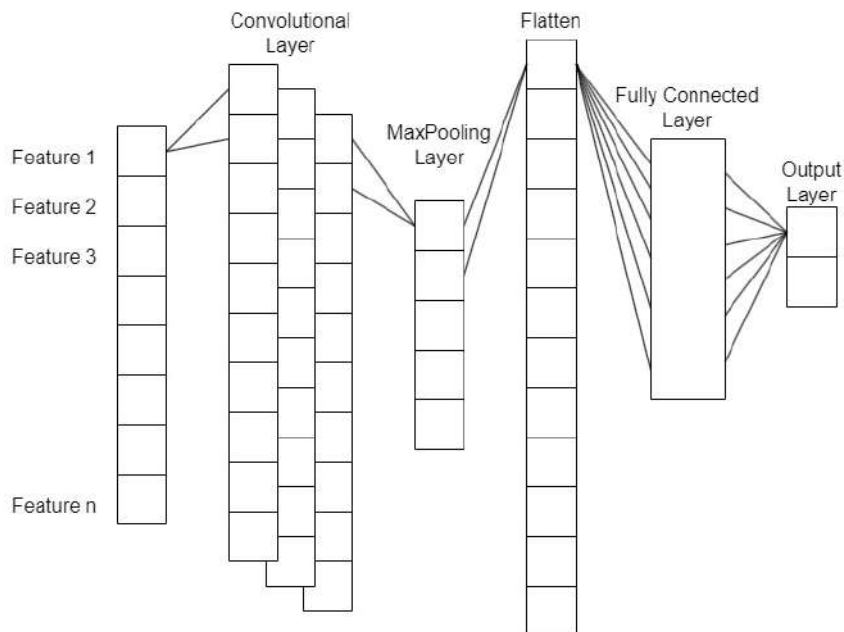
Algoritma Random Forest dimulai dengan melakukan *bootstrapping* pada *dataset*, yang menghasilkan beberapa sampel dengan sistem *sample with replacement*. Setelah itu, setiap sampel digunakan untuk membangun pohon keputusan. Hasil prediksi dari tiap pohon keputusan digabung secara *aggregation* untuk mendapatkan *ensemble classifier*. Alur algoritma Random Forest divisualisasikan pada Gambar 2.

Convolutional Neural Network (CNN) merupakan algoritma pengembangan dari Multi-layer Perceptron (MLP) yang dirancang untuk mengolah data dalam bentuk *grid* (Mawaddah et al., 2021). CNN termasuk dalam metode *supervised learning* dengan kategori *deep learning* yang dapat melakukan klasifikasi pada *dataset* berlabel. CNN memiliki beberapa *layer* yang terdiri dari *convolutional layer*, *non-linearity layer*, *pooling layer*, dan *fully connected layer* (Gabhane et al., 2022). Pada Gambar 3 akan menggambarkan ilustrasi arsitektur CNN yang diusulkan. Formula *convolution layer*, aktivasi ReLU, aktivasi *sigmoid*, *maxpooling layer*, dan *output layer* di tunjukan pada Pers. (3) sampai (7).



Gambar 2 Alur Algoritma Random Forest





Gambar 3 Arsitektur Convolutional Neural Network

Bobot untuk setiap lapisan dalam CNN diinisialisasi secara acak. Untuk mengoptimisasi *hyperparameter* CNN, dilakukan pencarian *grid* pada model. Model terdiri dari lapisan *convolution layer*, menggunakan fungsi aktivasi ReLU untuk memecah linearitas, dan fungsi aktivasi *sigmoid* untuk mendapatkan *output* dengan rentang (0,1). Kemudian, pada proses *pooling* akan digunakan *MaxPooling layer* sebagai metode yang paling umum digunakan. Selanjutnya hasil dari *MaxPooling layer* akan digunakan pada proses *Flatten* dan *Fully Connected Layer* untuk mendapatkan hasil *ouput* akhir. Algoritma CNN lebih sering digunakan dalam melakukan klasifikasi gambar, namun CNN juga dapat diterapkan pada prediksi satu dimensi, sehingga CNN dapat digunakan untuk melakukan prediksi peralihan pelanggan.

$$output = \left\lfloor \frac{Input - Kernel Size + 2 \times Padding}{Stride} \right\rfloor + 1 \quad (3)$$

$$f_{cov}(x) = \max(0, x) \quad (4)$$

$$Sigmoid(x) = \frac{1}{1 + e^{-x}} \quad (5)$$

$$Output = \left\lfloor \frac{Input Shape}{Pool Size} \right\rfloor \quad (6)$$

$$Output = Activ \left(\sum_{i=1}^{Input Size} Input_1 \times Weight_i + Bias \right) \quad (7)$$

Metode evaluasi yang digunakan adalah *confusion matrix* dan Area Under Curve (AUC). *Confusion matrix* adalah tabel yang memberikan gambaran dari performa sebuah model algoritma dengan spesifik. Setiap baris merepresentasikan kelas aktual data dan kolom merepresentasikan kelas prediksi data (Saputro & Sari, 2020). Tabel 1 menjelaskan mengenai *confusion matrix* dengan formula dari *confusion matrix* ditunjukkan pada Pers. (8) sampai (11). Pada persamaan tersebut, TP menjelaskan jumlah banyaknya data aktual kelas positif dan model memprediksi positif. Selanjutnya, TN menjelaskan jumlah banyaknya data aktual kelas negatif dan model memprediksi negatif. Lainnya, FP menjelaskan jumlah banyaknya data aktual kelas negatif namun model memprediksi positif. Sedangkan FN menjelaskan jumlah banyaknya data aktual kelas positif namun model memprediksi negatif.



Tabel 1 *Confusion Matrix*

	<i>Predict Negatif</i>	<i>Predict Positive</i>
<i>Actual Negatif</i>	True Negatif (TN)	False Positive (FP)
<i>Actual Positive</i>	False Negatif (FN)	True Positive (TP)

$$Akurasi = \frac{TP + TN}{TP + TN + FP + FN} \quad (8)$$

$$Presisi = \frac{TP}{TP + FP} \quad (9)$$

$$Recall = \frac{TP}{TP + FN} \quad (10)$$

$$F - measure = \frac{(Precision \times Recall)}{(Precision + Recall)} \times 2 \quad (11)$$

Akurasi adalah perbandingan antara jumlah data benar positif dan negatif dibagi dengan data keseluruhan. Presisi adalah hasil dari perbandingan antara prediksi benar positif dibagi dengan keseluruhan hasil prediksi benar positif. *Recall* adalah rasio prediksi benar positif dibagi dengan semua data yang benar positif. Skor F1 didapat dari hasil perkalian antara *precision* dan *recall* dibagi hasil penjumlahan dari *precision* dan *recall* kemudian dikali 2.

AUC mengukur hubungan True Positif Rate (TPR) dan False Positive Rate (FPR) pada berbagai *threshold* yang berbeda. Semakin besar nilai AUC, semakin baik performa model algoritma dalam membedakan kelas positif dan negatif. Nilai AUC berada di antara 0 hingga 1, di mana nilai 0 menunjukkan kinerja model buruk dan nilai 1 menunjukkan kinerja model baik dalam membedakan kelas positif dan negatif.

3. HASIL DAN PEMBAHASAN

Sebelum melakukan klasifikasi, *dataset* perlu dilakukan *preprocessing* agar data dapat diolah menggunakan model algoritma. Proses seleksi fitur menggunakan dua metode yaitu CFS dan RFE, artinya model Random Forest dan CNN akan digunakan pada kedua *dataset output* dari proses seleksi fitur. Kemudian dilakukan pembagian data seperti yang sudah dijelaskan sebelumnya. Proses *modelling* menggunakan parameter optimasi. Tabel 2 menampilkan secara ringkas parameter yang akan di optimasi. Parameter yang akan di optimasi digunakan pada proses *training* dan validasi model. Komposisi parameter beserta akurasi yang dihasilkan, dibandingkan dengan hasil menggunakan parameter *default* untuk menemukan model terbaik. Tabel 3 menampilkan hasil analisis evaluasi model menggunakan data validasi.

Tabel 2 Parameter yang di Optimasi

RF		CNN	
Parameter	Values	Parameter	Values
n_estimators	[50, 100, 200]	conv1_filter & kernel	[128, 64] & [7, 8]
max_depth	[0, 10, 20, 30]	conv2_filter & kernel	[64, 32] & [3, 4]
min_samples_split	[2, 5, 10]	conv3_filter & kernel	[32, 16] & [1, 2]
min_samples_leaf	[1, 2, 4]	pool_size	[2, 3]
		learning_rate	[0,01, 0,001]

Berdasarkan Tabel 3, model CNN *tuning* pada *dataset* CFS (80/20) memiliki hasil evaluasi paling baik. Evaluasi yang didapatkan adalah akurasi 99%, AUC 0,98, *precision* 100%, *recall* sebesar 97%, dan *F1-score* 98%. Model CNN tersebut digunakan pada proses klasifikasi, dan algoritma CNN *tuning* (80/20) dengan pembagian data CFS didapatkan sebagai model akhir yang memiliki performa paling baik. Selanjutnya adalah melakukan klasifikasi dengan menggunakan data *test*.



Setelah dilakukan klasifikasi, hasil akhirnya dievaluasi dan ditinjau kembali performanya. Untuk melihat hasil klasifikasi secara keseluruhan, dapat dilihat pada *confusion matrix* yang ditampilkan pada Tabel 4.

Tabel 3 Hasil Evaluasi Model Terbaik

Pembagian Data	Classifier	Akurasi	AUC	Precision (1)	Recall (1)	F1-score (1)
CFS (60/40)	RF	98,3%	0,97	99%	95%	97%
	RF Tuning	98,2%	0,97	99%	95%	97%
	CNN Tuning	98,5%	0,98	97%	98%	97%
CFS (70/30)	RF	98,7%	0,98	99%	97%	98%
	RF Tuning	98,8%	0,98	99%	97%	98%
	CNN Tuning	98,5%	0,98	97%	98%	97%
CFS (80/20)	RF	98,7%	0,97	99%	96%	98%
	RF Tuning	98,2%	0,97	99%	94%	97%
	CNN Tuning	99%	0,98	100%	96%	98%
RFE (60/40)	RF	98,3%	0,97	99%	95%	97%
	RF Tuning	98,3%	0,97	99%	95%	97%
	CNN Tuning	98,4%	0,97	98%	96%	97%
RFE (70/30)	RF	98,7%	0,98	99%	96%	98%
	RF Tuning	98,7%	0,98	99%	96%	98%
	CNN Tuning	98,4%	0,98	97%	97%	97%
RFE (80/20)	RF	98,4%	0,97	99%	95%	97%
	RF Tuning	98,5%	0,97	99%	95%	97%
	CNN Tuning	99%	0,97	98%	98%	99%

Dari total 167 data yang masuk ke dalam kategori peralihan pelanggan, model gagal dalam mengklasifikasi 13 data. Sedangkan dari total 538 data yang masuk ke dalam kategori tidak beralih, model gagal mengklasifikasi 1 data. Secara umum model menunjukkan hasil klasifikasi yang baik. Gambar 4 menjelaskan mengenai hasil evaluasi model pada data *test* yang sudah dilakukan. Model melakukan klasifikasi dengan baik, akurasi yang didapatkan sebesar 98% dengan AUC 0,96. *precision* yang didapatkan para proses klasifikasi sebesar 99%, *recall* 92% dan *F1-score* 96%.

Tabel 4 Confusion Matrix

	Predict Negatif	Predict Positive
Actual Negatif	537	1
Actual Positive	13	154

98.01418439716312

Confusion Matrix:

[[537 1]

[13 154]]

	precision	recall	f1-score	support
0	0.98	1.00	0.99	538
1	0.99	0.92	0.96	167
accuracy			0.98	705
macro avg	0.98	0.96	0.97	705
weighted avg	0.98	0.98	0.98	705

AUC Score : 0.9601484762816375

Gambar 4 Hasil Evaluasi Proses Klasifikasi



4. KESIMPULAN

Perkiraan yang akurat terkait perpindahan pelanggan memiliki dampak signifikan sebagai bagian dari strategi retensi pelanggan yang efektif. Dengan mampu memprediksi calon pelanggan yang berpotensi beralih dari layanan, perusahaan dapat mengurangi pengeluaran hingga 10 kali lipat dibandingkan dengan upaya mencari pelanggan baru. Melalui penelitian ini, model CNN dengan parameter optimasi {'conv1_filter & kernel': 128 & 8, 'conv2_filter & kernel': 32 & 3, 'conv3_filter & kernel': 32 & 1, 'pool_size': 3, 'learning_rate': 0,001} lebih baik dibandingkan RF dan RF *Hyperparameter*. Secara spesifik, skema pembagian CFS mendapatkan hasil evaluasi yang paling baik jika dibandingkan dengan skema RFE. Akurasi sebesar 99%, AUC 0,98, *precision* 100%, *recall* 96%, dan *F1-score* 98% pada data *validation*. Pembagian *dataset* yang direkomendasikan yaitu 80/20, Pada data *test* mendapatkan akurasi sebesar 98%, AUC 0,96, *precision* 99%, *recall* 92%, dan *F1-score* 96%. Hasil ini menegaskan bahwa algoritma CNN lebih unggul dalam menangkap pola kompleks pada data pelanggan telekomunikasi, menjadikannya pilihan yang lebih baik untuk prediksi *customer churn*. Selain itu, skema pembagian data 80/20 terbukti memberikan hasil evaluasi yang optimal. Implikasi praktis dari hasil yang didapat menunjukkan bahwa penerapan model prediksi *churn* yang akurat dapat memperkuat strategi retensi pelanggan dan memberikan perusahaan keunggulan kompetitif dalam pasar yang semakin kompetitif.

Berdasarkan temuan tersebut, rekomendasi untuk penelitian selanjutnya adalah menggunakan *dataset* yang lebih besar dan beragam. Mengeksplorasi algoritma lain seperti algoritma *boosting*, serta dapat meningkatkan *epoch* pada CNN 1D dengan arsitektur yang lebih kompleks untuk memaksimalkan hasil evaluasi. Penerapan model prediksi *churn* yang akurat tidak hanya akan meningkatkan strategi retensi pelanggan, tetapi juga memberikan perusahaan keunggulan kompetitif yang signifikan dalam pasar yang semakin kompetitif. Dengan langkah-langkah ini, diharapkan model prediksi *churn* dapat mencapai akurasi yang lebih tinggi dan lebih aplikatif dalam berbagai kondisi operasional perusahaan.

DAFTAR PUSTAKA

- AL-Najjar, D., Al-Rousan, N., & AL-Najjar, H. (2022). Machine Learning to Develop Credit Card Customer Churn Prediction. *Journal of Theoretical and Applied Electronic Commerce Research*, 17(4), 1529–1542. <https://doi.org/10.3390/jtaer17040077>
- Artha, B., Zahara, I., Bahri, & Permata Sari, N. (2022). Customer Retention: A Literature Review. *Social Science Studies*, 2(1), 030–045. <https://doi.org/10.47153/sss21.2952022>
- de Lima Lemos, R. A., Silva, T. C., & Tabak, B. M. (2022). Propension to Customer Churn in a Financial Institution: A Machine Learning Approach. *Neural Computing and Applications*, 34(14), 11751–11768. <https://doi.org/10.1007/s00521-022-07067-x>
- Gabhane, M. D., Suriya, A., & Kishor, S. B. (2022). Churn Prediction in Telecommunication Business Using CNN and ANN. *Journal of Positive School Psychology*, 2022(4), 4672–4680. <https://journalppw.com/index.php/jpsp/article/view/4158>
- Husein, A. M., & Harahap, M. (2021). Pendekatan Data Science untuk Menemukan Churn Pelanggan pada Sector Perbankan dengan Machine Learning. *Data Sciences Indonesia (DSI)*, 1(1), 8–13. <https://doi.org/10.47709/dsi.v1i1.1169>
- Jain, H., Khunteta, A., & Srivastava, S. (2021). Telecom Churn Prediction and Used Techniques, Datasets and Performance Measures: A Review. *Telecommunication Systems*, 76(4), 613–630. <https://doi.org/10.1007/s11235-020-00727-0>
- Lalwani, P., Mishra, M. K., Chadha, J. S., & Sethi, P. (2022). Customer Churn Prediction System: A Machine Learning Approach. *Computing*, 104(2), 271–294. <https://doi.org/10.1007/s00607-021-00908-y>
- Li, A., Yang, T., Zhan, X., Shi, Y., & Li, H. (2024). Utilizing Data Science and AI for Customer Churn Prediction in Marketing. *Journal of Theory and Practice of Engineering Science*, 4(05), 72–79. [https://doi.org/10.53469/jtpes.2024.04\(05\).10](https://doi.org/10.53469/jtpes.2024.04(05).10)
- Mawaddah, U., Armanto, H., & Setyati, E. (2021). Prediksi Karakteristik Personal Menggunakan Analisis Tanda Tangan dengan Menggunakan Metode Convolutional Neural Network



- (CNN). *Antivirus: Jurnal Ilmiah Teknik Informatika*, 15(1), 123–133. <https://doi.org/10.35457/antivirus.v15i1.1526>
- Muthmainah, & Cholil, M. (2022). Faktor yang Mempengaruhi Perilaku Peralihan Pelanggan: Peran Mediasi Kepuasan dan Kepercayaan Pelanggan. *Jurnal Riset Bisnis dan Investasi*, 7(3), 125–136. <https://doi.org/10.35313/jrbi.v7i3.3437>
- Ojo, A. K. (2024). Predicting Customer Churn in Telecommunication Industry Using Convolutional Neural Network Model. *SSRN Electronic Journal*, 22(3), 54–59. <https://doi.org/10.2139/ssrn.4983685>
- Rahman, M., & Kumar, V. (2020). Machine Learning Based Customer Churn Prediction in Banking. *2020 4th International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, 1196–1201. <https://doi.org/10.1109/ICECA49313.2020.9297529>
- Ribeiro, H., Barbosa, B., Moreira, A. C., & Rodrigues, R. G. (2024). Determinants of Churn in Telecommunication Services: A Systematic Literature Review. *Management Review Quarterly*, 74(3), 1327–1364. <https://doi.org/10.1007/s11301-023-00335-7>
- Sandag, G. A. (2020). Prediksi Rating Aplikasi App Store Menggunakan Algoritma Random Forest. *Cogito Smart Journal*, 6(2), 167–178. <https://doi.org/10.31154/cogito.v6i2.270.167-178>
- Saputro, I. W., & Sari, B. W. (2020). Uji Performa Algoritma Naïve Bayes untuk Prediksi Masa Studi Mahasiswa. *Creative Information Technology Journal*, 6(1), 1–11. <https://doi.org/10.24076/citec.2019v6i1.178>
- Singh, U., Rizwan, M., Alaraj, M., & Alsaïdan, I. (2021). A Machine Learning-Based Gradient Boosting Regression Approach for Wind Power Production Forecasting: A Step towards Smart Grid Environments. *Energies*, 14(16), Article ID: 5196. <https://doi.org/10.3390/en14165196>
- Suryana, N., Pratiwi, P., & Prasetyo, R. T. (2021). Penanganan Ketidakseimbangan Data pada Prediksi Customer Churn Menggunakan Kombinasi SMOTE dan Boosting. *IJCIT (Indonesian Journal on Computer and Information Technology)*, 6(1), 31–37. <https://doi.org/10.31294/ijcit.v6i1.9545>
- Wicaksono, A., Anita, A., & Padilah, T. N. (2021). Uji Performa Teknik Klasifikasi untuk Memprediksi Customer Churn. *Bianglala Informatika*, 9(1), 37–45. <https://doi.org/10.31294/bi.v9i1.9992>
- Wu, Y. (2024). Predicting Customer Churn in a Telecommunications Company Using Machine Learning. *Proceedings of the 7th International Conference on Economic Management and Green Development*, 771–783. https://doi.org/10.1007/978-981-97-0523-8_72
- Yahaya, R., Abisoye, O. A., & Bashir, S. A. (2021). An Enhanced Bank Customers Churn Prediction Model Using a Hybrid Genetic Algorithm and K-Means Filter and Artificial Neural Network. *2020 IEEE 2nd International Conference on Cyberspac (CYBER NIGERIA)*, 52–58. <https://doi.org/10.1109/CYBERNIGERIA51635.2021.9428805>



Penggunaan Teknik *Transfer Learning* pada Metode CNN untuk Pengenalan Tanaman Bunga

Agustina Mufidatuzzainiya ^{(1)*}, Muhammad Faisal ⁽²⁾

Departemen Teknik Informatika, UIN Maulana Malik Ibrahim, Malang, Indonesia

e-mail : niyamufida@gmail.com, mfaisal@ti.uin-malang.ac.id.

* Penulis korespondensi.

Artikel ini diajukan 10 Juni 2024, direvisi 6 Oktober 2024, diterima 8 Oktober 2024, dan dipublikasikan 31 Mei 2025.

Abstract

This study investigates the impact of employing the transfer learning method on improving flower recognition performance using Convolutional Neural Network (CNN) models. The dataset used consists of 4242 flower images divided into five classes: daisy, tulip, rose, sunflower, and dandelion. This research implements three models: basic CNN, VGG16, and EfficientNetB3, to test the effectiveness of transfer learning in flower classification. The basic CNN model achieved a training accuracy of 73.38% and a validation accuracy of 71.76%, but it generally fails to generalize to new data. The VGG16 model achieved perfect training accuracy but experienced overfitting, with validation accuracy stabilizing around 85-90%. Meanwhile, the EfficientNetB3 model with transfer learning reached a training accuracy of 98.50% and a validation accuracy of 94.00%, demonstrating strong generalization without significant overfitting. The experiment was conducted using data augmentation techniques, and performance evaluation was carried out using accuracy, precision, and recall metrics. The results show that transfer learning with the EfficientNetB3 model provides the best performance in flower classification compared to the basic CNN and VGG16 models. For future research, further development can be done by expanding the types of flower datasets and applying additional optimization techniques to improve accuracy in more complex models.

Keywords: CNN, Transfer Learning, EfficientNetB3, VGG16, Flower Recognition

Abstrak

Penelitian ini mengeksplorasi pengaruh penggunaan metode *transfer learning* dalam meningkatkan kinerja pengenalan bunga menggunakan model Convolutional Neural Network (CNN). *Dataset* yang digunakan terdiri dari 4.242 gambar bunga yang terbagi dalam lima kelas: *daisy*, tulip, mawar, bunga matahari, dan *dandelion*. Penelitian ini menerapkan tiga model, yaitu CNN dasar, VGG16, dan EfficientNetB3, untuk menguji efektivitas *transfer learning* dalam klasifikasi bunga. Model CNN dasar menunjukkan akurasi pelatihan sebesar 73,38% dan akurasi validasi 71,76%, namun terbatas dalam generalisasi pada data baru. Model VGG16 mencapai akurasi pelatihan sempurna, tetapi mengalami *overfitting* dengan akurasi validasi yang stabil di kisaran 85-90%. Sementara itu, model EfficientNetB3 dengan *transfer learning* mencapai akurasi pelatihan 98,50% dan akurasi validasi 94,00%, menunjukkan kemampuan generalisasi yang kuat tanpa *overfitting* signifikan. Eksperimen dilakukan dengan menggunakan teknik augmentasi data dan evaluasi performa dilakukan menggunakan metrik akurasi, presisi, dan *recall*. Hasil penelitian menunjukkan bahwa *transfer learning* dengan model EfficientNetB3 memberikan performa terbaik dalam klasifikasi bunga, dibandingkan dengan CNN dasar dan VGG16. Untuk penelitian selanjutnya, pengembangan lebih lanjut dapat dilakukan dengan memperluas jenis *dataset* bunga serta menerapkan teknik optimasi tambahan guna meningkatkan akurasi pada model yang lebih kompleks.

Kata Kunci: CNN, Transfer Learning, EfficientNetB3, VGG16, Pengenalan Bunga

1. PENDAHULUAN

Bunga merupakan elemen penting dalam kehidupan sehari-hari yang memiliki nilai budaya, ekonomi, dan ekologis yang signifikan. Dalam ekosistem, bunga berperan sebagai sumber makanan bagi serangga, burung, dan hewan lainnya, serta sebagai bahan obat untuk manusia



Artikel ini didistribusikan mengikuti lisensi Atribusi-NonKomersial CC BY-NC sebagaimana tercantum pada <https://creativecommons.org/licenses/by-nc/4.0/>.

dan hewan (Nugroho, 2015). Selain itu, bunga juga memiliki peran krusial dalam berbagai bidang, seperti pertanian presisi, penelitian botani, pemantauan tanaman, pengobatan Ayurveda, serta pencarian gambar berbasis konten (CBIR) (Arianty et al., 2022). Klasifikasi spesies bunga penting untuk mendukung berbagai aplikasi di bidang tersebut, namun proses ini sering kali menemui tantangan karena perbedaan dalam bentuk, warna, dan struktur bunga yang terkadang sangat mirip (Bae et al., 2020).

Meskipun terdapat kemajuan teknologi yang memungkinkan pengenalan bunga secara otomatis menggunakan gambar dibandingkan dengan deskripsi teks, metode identifikasi yang ada masih memiliki tingkat akurasi yang relatif rendah (Nuraini et al., 2023). Hal ini menandakan perlunya solusi yang lebih baik dan akurat untuk mendukung pengenalan bunga secara otomatis. Dalam penelitian klasifikasi bunga, terdapat beberapa tantangan teknis, termasuk kesulitan dalam ekstraksi fitur tradisional serta adanya variasi bentuk dan warna yang mirip antara spesies yang berbeda (Liu et al., 2022). Teknologi pembelajaran mendalam, khususnya jaringan saraf konvolusional (CNN), telah terbukti memiliki potensi besar dalam meningkatkan akurasi pengenalan bunga (Saputra et al., 2023). Namun, penggunaan CNN sering kali menghadapi masalah *overfitting* dan optimasi lokal, yang mengurangi efektivitasnya ketika diterapkan pada data validasi yang beragam (Santos & Papa, 2022).

Salah satu metode yang dapat mengatasi keterbatasan CNN adalah *transfer learning*, di mana model yang telah dilatih pada tugas serupa diterapkan untuk memperbaiki kinerja pengenalan pada domain yang berbeda (Karthikeyan, 2022). Dalam konteks pengenalan bunga, *transfer learning* menawarkan potensi untuk meningkatkan akurasi dan generalisasi model terhadap data baru (Falakhi et al., 2022). Penelitian ini bertujuan untuk mengkaji penggunaan metode *transfer learning* dalam meningkatkan kinerja sistem pengenalan bunga, baik dari segi akurasi maupun ketahanan terhadap variasi kondisi lingkungan dan fenotipe tanaman. Dengan menggunakan model yang lebih optimal, diharapkan hasil pengenalan bunga dapat lebih akurat dan *robust* dibandingkan dengan metode yang hanya mengandalkan CNN dasar.

Penelitian sebelumnya, seperti yang dilakukan oleh Narvekar & Rao (2020) menekankan pentingnya CNN dan *transfer learning* dalam klasifikasi bunga dalam konteks pertanian. Sementara itu, Li et al. (2024) mengembangkan metode pengenalan bunga berbasis *Masked Autoencoders*. Berbeda dari penelitian-penelitian tersebut, penelitian ini lebih fokus pada optimalisasi *transfer learning* untuk mengatasi masalah *overfitting* dan optimasi lokal yang umum terjadi dalam model pembelajaran mendalam.

Tujuan penelitian ini adalah mengeksplorasi potensi metode *transfer learning* dalam meningkatkan akurasi dan ketahanan sistem pengenalan bunga. Pendekatan ini diharapkan dapat mengatasi tantangan yang dihadapi oleh penelitian sebelumnya dan memberikan solusi yang lebih baik dalam klasifikasi bunga. Pada akhirnya, penelitian ini bertujuan untuk memberikan kontribusi signifikan dalam pengembangan sistem pengenalan bunga yang lebih akurat dan andal.

2. METODE PENELITIAN

2.1 Tahapan Penelitian

Penelitian ini dilakukan dengan beberapa tahapan utama yang sistematis dan empiris untuk mencapai hasil yang optimal. Tempat penelitian bersifat fleksibel selama periode Mei hingga Juni 2024. Alat yang digunakan dalam penelitian ini adalah perangkat keras berupa laptop dengan GPU AMD RADEON™ GRAPHICS dan Google Colab untuk membangun dan melatih model CNN.

Tahapan penelitian yang dilakukan terdapat pada Gambar 1 dengan penjelasan sebagai berikut:

- 1) Pengumpulan data: *Dataset* yang digunakan terdiri dari lima jenis bunga (mawar, bunga matahari, aster, *dandelion*, dan tulip). *Dataset* diperoleh dari sumber terbuka dan disesuaikan dengan kebutuhan penelitian.



- 2) *Preprocessing* data: Proses *preprocessing* meliputi *resizing* gambar, normalisasi, dan augmentasi data untuk memperbanyak variasi dataset serta mencegah *overfitting*.
- 3) Pembangunan model CNN: Model CNN dibangun menggunakan beberapa lapisan, termasuk lapisan konvolusi, *pooling*, dan *fully connected*, serta menggunakan teknik *transfer learning* dari model InceptionV3 yang telah dilatih sebelumnya.
- 4) Pelatihan model: Model dilatih menggunakan *dataset* yang telah diproses, dan teknik *transfer learning* diterapkan untuk mempercepat proses pelatihan.
- 5) Evaluasi model: Evaluasi dilakukan dengan menghitung metrik akurasi, presisi, dan *recall*.



Gambar 1 Tahapan Penelitian

2.2 Dataset

Flower Recognition, *dataset* tersebut dibuat oleh Alexander Mamaev pada tahun 2021 (<https://www.kaggle.com/datasets/alxmamaev/flowers-recognition>). *Dataset* ini berisi 4242 gambar bunga. Pengumpulan data didasarkan pada data dari Flickr, Google Images, dan Yandex Images. Gambar-gambar tersebut dibagi menjadi lima kelas: *daisy*, tulip, mawar, bunga matahari, dan *dandelion*. Untuk setiap kelas terdapat sekitar 800 foto. Foto-foto tersebut tidak beresolusi tinggi, resolusinya sekitar 320x240 piksel. Foto-foto ini tidak dikurangi menjadi satu ukuran tunggal, mereka memiliki proporsi yang berbeda-beda. Contoh gambar yang ada pada *dataset* Flower Recognition bisa dilihat di Gambar 2.



Gambar 2 Contoh Gambar pada Dataset Flower Recognition



2.3 Convolutional Neural Network

Convolutional Neural Networks (CNN) adalah algoritma pembelajaran mendalam yang khusus dirancang untuk menganalisis data visual, efektif dalam tugas-tugas seperti klasifikasi gambar, pengenalan objek, dan deteksi objek. Struktur CNN terdiri dari beberapa lapisan utama, yaitu lapisan konvolusi (Fuadah et al., 2022) untuk mengekstraksi fitur dari gambar, lapisan aktivasi untuk memperkenalkan non-linearitas, lapisan *pooling* untuk mengurangi dimensi peta fitur sambil menjaga informasi penting, dan lapisan *fully connected* untuk klasifikasi akhir. CNN bekerja dengan mengekstraksi fitur dasar seperti tepi dan tekstur pada lapisan awal (Danial & Setiawati, 2024), lalu mengenali fitur yang lebih kompleks pada lapisan berikutnya, dan akhirnya menggunakan fitur-fitur ini untuk membuat keputusan klasifikasi.

Keunggulan CNN meliputi otomatisasi ekstraksi fitur (Ibrahim et al., 2022), ketahanan terhadap variasi kondisi seperti pencahayaan dan orientasi, kemampuan menangani data besar dan kompleks, serta kemampuan menangkap ketergantungan spasial dalam gambar. CNN diterapkan dalam berbagai bidang seperti penglihatan komputer, otomatisasi kendaraan, kesehatan, keamanan, dan pertanian presisi, menjadikannya teknologi utama untuk analisis gambar dan video. Convolutional Neural Network (CNN) menggunakan beberapa operasi matematika dasar dalam arsitekturnya (Darmanto, 2019). Berikut adalah rumus-rumus utama yang digunakan dalam model CNN.

2.3.1 Convolution

Operasi konvolusi melibatkan filter (kernel) K yang berukuran $m \times n$ yang diterapkan pada gambar input I . Hasil konvolusi pada posisi (x, y) dihitung sebagai jumlah dari hasil kali elemen-elemen filter dan elemen-elemen gambar yang bersesuaian. Persamaan perhitungan konvolusi dijabarkan pada Pers. (1).

$$(I \cdot K)(x, y) = \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} I(x+i, y+j) \cdot K(i, j) \quad (1)$$

2.3.2 ReLU

Fungsi aktivasi ReLU menggantikan semua nilai negatif dalam peta fitur dengan nol, memperkenalkan non-linearitas ke dalam model dan membantu dalam mengatasi masalah *vanishing gradient*. Persamaan perhitungan fungsi aktivasi ReLU dijabarkan pada Pers. (2).

$$f(x) = \max(0, x) \quad (2)$$

2.3.3 Pooling

Max pooling mengurangi dimensi peta fitur dengan mengambil nilai maksimum dari setiap area kecil $m \times n$ dalam peta fitur, menjaga fitur-fitur penting sambil mengurangi jumlah parameter dan komputasi. Persamaan perhitungan *pooling* dijabarkan pada Pers. (3).

$$P_{max}(x, y) = \max_{0 \leq i < m, 0 \leq j < n} I(x+i, y+j) \quad (3)$$

2.3.4 Fully Connected Layer

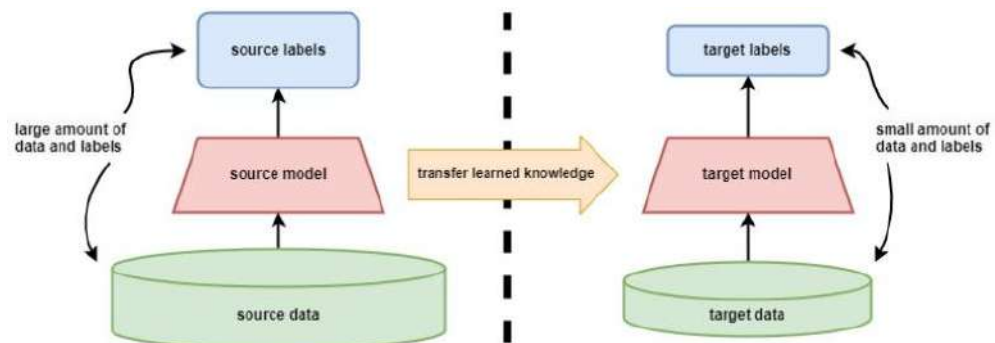
Di lapisan *fully connected*, setiap neuron dihubungkan dengan semua neuron di lapisan sebelumnya. W adalah matriks bobot, x adalah input, b adalah bias, dan f adalah fungsi aktivasi. Lapisan ini menggabungkan fitur-fitur yang telah diekstraksi untuk membuat prediksi klasifikasi akhir. Persamaan perhitungan *fully connected layer* dijabarkan pada Pers. (4).

$$y = f(W \cdot x + b) \quad (4)$$



2.4 Transfer Learning

Transfer learning adalah teknik dalam pembelajaran mendalam yang memanfaatkan model yang telah dilatih sebelumnya (Harahap et al., 2022) pada satu tugas untuk meningkatkan kinerja pada tugas baru yang terkait. Dengan menggunakan pengetahuan yang sudah diperoleh dari model yang dilatih pada *dataset* yang sangat besar, *transfer learning* memungkinkan proses pelatihan menjadi lebih cepat dan efisien. Model yang telah dilatih sebelumnya ini mempunyai sebuah pemahaman yang kuat mengenai konsep dasar yang ada, sehingga ketika diterapkan pada tugas baru, meskipun dengan *dataset* yang lebih terbatas, hasil yang dicapai dapat lebih baik. Selain itu, *transfer learning* mengurangi kebutuhan akan sumber daya komputasi yang besar (Widyaya & Budi, 2021), karena model tidak perlu dilatih dari awal. Ini membuat *transfer learning* menjadi alat yang sangat berharga dalam pengembangan model pembelajaran mesin, mempercepat proses pengembangan dan meningkatkan kinerja dalam berbagai aplikasi.



Gambar 3 Pengembangan *Transfer learning* pada CNN

Berdasarkan Gambar 3 (Drenyovszki, 2024), *transfer learning* merupakan sebuah metode dalam *machine learning* di mana pengetahuan yang telah diperoleh dari satu model (*source model*) pada suatu tugas dengan *dataset* yang besar (*source data*) dapat digunakan untuk meningkatkan performa model pada tugas yang berbeda namun terkait, yang memiliki *dataset* lebih kecil (*target data*). Dalam ilustrasi, "*source model*" dilatih menggunakan data yang besar seperti *ImageNet*, yang memiliki banyak label (*source labels*). Setelah pengetahuan dipelajari oleh model ini, pengetahuan tersebut ditransfer ke "*target model*" yang ditujukan untuk tugas yang berbeda, menggunakan *dataset* yang lebih kecil seperti *PASCAL*, yang memiliki lebih sedikit label (*target labels*). Transfer ini memungkinkan target model untuk memanfaatkan pengetahuan yang diperoleh dari *source model*, sehingga dapat mencapai performa yang lebih baik meskipun data yang tersedia terbatas.

2.5 Evaluasi Kinerja Model

Untuk mengevaluasi kinerja model, digunakan beberapa metrik evaluasi yang meliputi akurasi, presisi, dan *recall*. Akurasi mengukur proporsi prediksi yang benar dari keseluruhan prediksi yang dibuat oleh model, dengan rumus dituliskan pada Pers. (5). Presisi mengukur seberapa akurat prediksi positif model, dengan rumus pada Pers. (6). *Recall*, atau sensitivitas, mengukur seberapa baik model dalam mengidentifikasi semua kasus positif, dengan rumus pada Pers. (7). Di mana TP (*True Positive*) menunjukkan jumlah kasus positif yang diprediksi benar, TN (*True Negative*) adalah jumlah kasus negatif yang diprediksi benar, FP (*False Positive*) merupakan jumlah kasus negatif yang diprediksi salah sebagai positif, dan FN (*False Negative*) menunjukkan jumlah kasus positif yang diprediksi salah sebagai negatif.

$$\text{Akurasi} = \frac{TP + TN}{TP + TN + FP + FN} \quad (5)$$

$$\text{Presisi} = \frac{TP}{TP + FP} \quad (6)$$



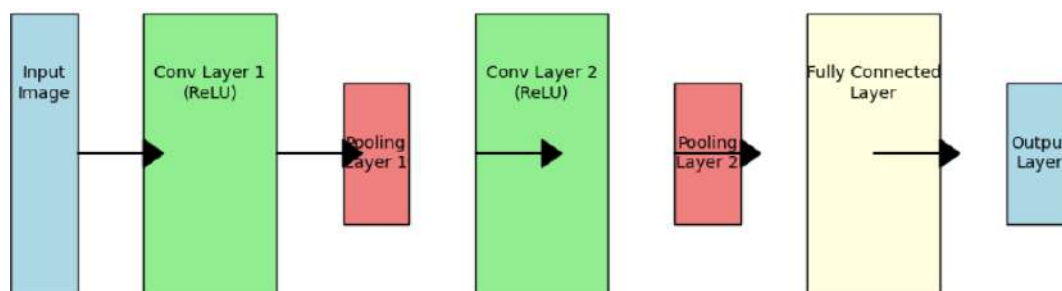
$$Recall = \frac{TP}{TP + FN} \quad (7)$$

3. HASIL DAN PEMBAHASAN

3.1 Arsitektur Model CNN yang Diusulkan

Arsitektur model Convolutional Neural Network (CNN) dimulai dengan lapisan input (Alwanda et al., 2020), tempat gambar masukan dimasukkan ke dalam model. Gambar ini kemudian diproses melalui *Convolutional Layer 1* yang menggunakan filter untuk mengekstraksi fitur-fitur dasar seperti tepi dan tekstur, dengan aktivasi ReLU untuk memperkenalkan non-linearitas. Selanjutnya, *Pooling Layer 1* mengurangi dimensi peta fitur yang dihasilkan untuk mempertahankan fitur penting sambil mengurangi jumlah parameter dan komputasi.

Proses ini diulang dengan *Convolutional Layer 2* untuk mengekstraksi fitur yang lebih kompleks dan *Pooling Layer 2* untuk mengurangi dimensi lebih lanjut. Setelah fitur-fitur diekstraksi, lapisan *fully connected* menghubungkan semua neuron dari lapisan sebelumnya untuk menggabungkan fitur-fitur tersebut dan membuat keputusan klasifikasi akhir. Lapisan *output* kemudian menghasilkan prediksi akhir dari model, seperti mengklasifikasikan jenis bunga dalam gambar (Tama & Santi, 2023). Garis-garis panah yang menghubungkan setiap lapisan menunjukkan aliran data dan proses transformasi yang terjadi, menggambarkan bagaimana CNN memproses dan menganalisis data visual untuk menghasilkan prediksi yang akurat. Gambaran untuk arsitektur model CNN bisa dilihat di Gambar 4.



Gambar 4 Arsitektur Model CNN

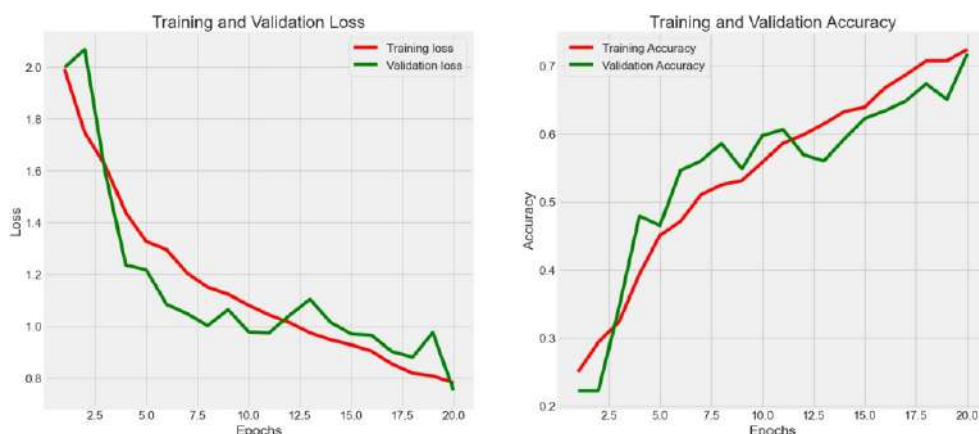
Hasil pelatihan dan evaluasi model *Convolutional Neural Network* (CNN) selama 20 *epoch* dapat dilihat di Tabel 1. Pada setiap *epoch*, model CNN mengalami peningkatan kinerja dalam hal akurasi dan penurunan nilai *loss*. Awalnya, pada *epoch* pertama, akurasi model hanya sekitar 24,31% dengan nilai *loss* sebesar 2,0919. Namun, seiring berjalannya pelatihan, akurasi model secara bertahap meningkat, mencapai puncaknya pada *epoch* ke-20 dengan akurasi sebesar 73,38%. Selama proses pelatihan, nilai *loss* model juga terus menurun dari 2,0919 pada *epoch* pertama menjadi 0,7730 pada *epoch* ke-20. Hasil validasi juga menunjukkan tren yang serupa, di mana akurasi pada data validasi meningkat dari 22,22% pada awal pelatihan menjadi 71,76% pada akhirnya, sementara nilai *loss* pada data validasi mengalami penurunan dari 1,9976 menjadi 0,7509. Hal ini menunjukkan bahwa model CNN berhasil belajar dari data pelatihan dan menggeneralisasi dengan baik data yang belum pernah dilihat sebelumnya.

Dari Tabel 1 dapat ditampilkan grafiknya sebagaimana gambar grafik pada Gambar 5 di bawah. Gambar tersebut merupakan gambar dua grafik yang menggambarkan kinerja model pembelajaran mesin selama proses pelatihan. Grafik kiri menunjukkan hubungan antara *loss* (fungsi kerugian) dan jumlah *epoch* (iterasi). Garis merah mewakili nilai *loss* pada data pelatihan, sedangkan garis hijau mewakili nilai *loss* pada data validasi. Kedua kurva ini menurun seiring bertambahnya *epoch*, menunjukkan bahwa model semakin baik dalam memprediksi target baik pada data pelatihan maupun data validasi. Pada akhir pelatihan, nilai *loss* pada data pelatihan dan validasi hampir mendekati, menunjukkan bahwa tidak ada *overfitting* yang signifikan.



Tabel 1 Hasil Performa Model CNN

<i>epoch</i>	<i>accuracy</i>	<i>loss</i>	<i>val_accuracy</i>	<i>val_loss</i>
1	0,2431	2,0919	0,2222	1,9976
2	0,2825	1,7935	0,2222	2,0664
3	0,3179	1,6505	0,3449	1,5945
4	0,3837	1,4683	0,4792	1,2356
5	0,4467	1,3376	0,4653	1,2172
6	0,4632	1,3208	0,5463	1,0841
7	0,5076	1,2043	0,5602	1,0479
8	0,5276	1,1701	0,5856	1,0015
9	0,5410	1,1131	0,5486	1,0631
10	0,5737	1,0488	0,5972	0,9776
11	0,5839	1,0548	0,6065	0,9733
12	0,5953	1,0230	0,5694	1,0422
13	0,6163	0,9814	0,5602	1,1040
14	0,6237	0,9801	0,5926	1,0150
15	0,6639	0,8849	0,6227	0,9695
16	0,6506	0,9282	0,6343	0,9646
17	0,6828	0,8455	0,6481	0,9019
18	0,7048	0,8243	0,6736	0,8803
19	0,7101	0,8111	0,6505	0,9748
20	0,7338	0,7730	0,7176	0,7509



Gambar 5 Grafik Nilai Loss dan Nilai Accuracy

Adapun grafik kanan menunjukkan hubungan antara akurasi dan jumlah *epoch*. Garis merah mewakili akurasi pada data pelatihan, sementara garis hijau mewakili akurasi pada data validasi. Kedua kurva ini meningkat seiring bertambahnya *epoch*, menandakan bahwa model semakin akurat dalam melakukan prediksi. Meskipun kurva akurasi pada data validasi sedikit fluktuatif, tren keseluruhan menunjukkan peningkatan yang konsisten. Kesimpulannya, model yang dilatih menunjukkan peningkatan kinerja yang stabil, ditunjukkan oleh penurunan *loss* dan peningkatan akurasi pada kedua data pelatihan dan validasi, tanpa indikasi overfitting yang jelas.

3.2 Eksperimen Berbasis *Transfer learning*

Sebelum memulai langkah-langkah eksperimen, pendekatan berbasis *transfer learning* digunakan sebagai metode utama dalam penelitian ini. *Transfer learning* memungkinkan pemanfaatan model pra-terlatih yang telah dikembangkan sebelumnya untuk menyelesaikan tugas klasifikasi gambar pada *dataset* yang berbeda, dalam hal ini, klasifikasi jenis bunga. Dengan menggunakan model pra-terlatih seperti CNN dan EfficientNet, model ini dapat memanfaatkan pengetahuan yang diperoleh dari *dataset* besar, seperti ImageNet, dan menerapkannya pada tugas spesifik pengenalan bunga. Pendekatan ini tidak hanya



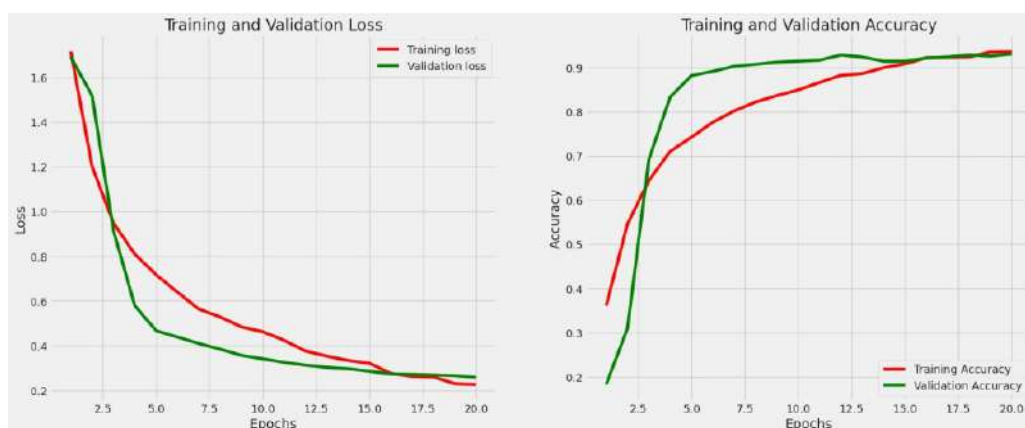
mempercepat proses pelatihan, tetapi juga meningkatkan akurasi dan mengurangi kemungkinan *overfitting* pada dataset bunga yang terbatas.

Setelah menemukan model yang tepat, eksperimen dimulai dengan mengumpulkan data gambar bunga yang terdiri dari lima jenis bunga: mawar (*rose*), bunga matahari (*sunflower*), aster (*daisy*), *dandelion*, dan tulip. Data ini dipecah menjadi data latih, validasi, dan uji dengan rasio masing-masing 80%, 10%, dan 10%. Data augmentasi dilakukan pada data latih untuk meningkatkan variasi gambar dengan berbagai transformasi seperti rotasi, pergeseran, *zoom*, dan *flip* horizontal.

Transfer learning dengan EfficientNetB3 adalah model yang telah dilatih sebelumnya pada *dataset* ImageNet. Dalam eksperimen ini, arsitektur dasar EfficientNetB3 digunakan sebagai *feature extractor*, dan beberapa lapisan tambahan ditambahkan pada bagian atasnya untuk menyesuaikan model dengan tugas klasifikasi bunga. Lapisan tambahan termasuk GlobalAveragePooling2D, *batch normalization*, blok *dens*, dan lapisan *output* dengan aktivasi *softmax*. Model ini kemudian dilatih menggunakan *optimizer* Adamax dengan *learning rate* yang lebih rendah.

Transfer learning dengan VGG16 adalah model lain yang telah dilatih sebelumnya pada *dataset* ImageNet. Sama seperti pada EfficientNetB3, arsitektur dasar VGG16 digunakan sebagai *feature extractor*, dengan beberapa lapisan tambahan yang ditambahkan pada bagian atasnya. Lapisan tambahan ini termasuk GlobalAveragePooling2D, *batch normalization*, blok *dens*, dan lapisan *output* dengan aktivasi *softmax*. Model ini juga dilatih menggunakan *optimizer* Adamax dengan *learning rate* yang lebih rendah.

Gambar 6 menunjukkan hasil pelatihan dan evaluasi model menggunakan EfficientNetB3 dengan dua metrik utama: nilai *loss* dan nilai akurasi. Grafik di sebelah kiri menggambarkan nilai *loss*, yaitu ukuran kesalahan prediksi model, di mana nilai yang lebih rendah menunjukkan kinerja model yang lebih baik. Sumbu Y menunjukkan nilai *loss*, sedangkan sumbu X menunjukkan jumlah *epoch* atau iterasi pelatihan. Garis merah mewakili nilai *loss* pada data pelatihan, dan garis hijau menunjukkan nilai *loss* pada data validasi. Dari grafik ini, kita dapat melihat bahwa nilai *loss* pada kedua *dataset* ini menunjukkan tren penurunan seiring bertambahnya jumlah *epoch*. Pada awal pelatihan, nilai *loss* cukup tinggi, tetapi seiring waktu, nilai *loss* menurun secara konsisten. Hal ini menunjukkan bahwa model semakin baik dalam memprediksi, dan kesalahan prediksi berkurang. Pada akhir pelatihan, nilai *loss* pada data pelatihan dan validasi hampir mendekati, menunjukkan bahwa model tidak mengalami *overfitting* yang signifikan dan dapat menggeneralisasi dengan baik data yang sebelumnya tidak terlihat.



Gambar 6 Grafik Nilai *Loss* dan *Accuracy* EfficientNetB3

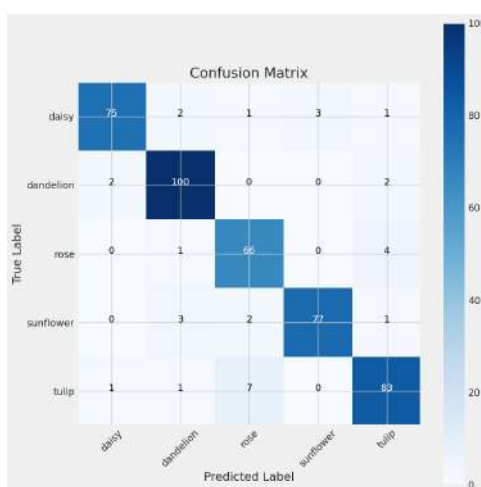
Grafik di sebelah kanan menunjukkan skor akurasi, yang mewakili persentase prediksi yang benar untuk keseluruhan model. Sumbu Y menunjukkan nilai presisi dan sumbu X menunjukkan



jumlah *epoch*. Garis merah mewakili nilai akurasi data latih, dan garis hijau mewakili nilai akurasi data validasi. Grafik ini menunjukkan bahwa nilai akurasi untuk kedua *dataset* meningkat seiring dengan bertambahnya jumlah *epoch*. Pada awal pelatihan, nilai akurasinya relatif rendah, namun seiring berjalannya waktu nilai akurasinya meningkat secara signifikan. Hal ini menunjukkan bahwa akurasi prediksi model menjadi semakin akurat. Di akhir pelatihan, nilai akurasi data pelatihan dan validasi mendekati nilai tinggi, yang menunjukkan bahwa model cukup belajar dari data pelatihan untuk membuat prediksi akurat pada data validasi.

Secara keseluruhan, kedua grafik tersebut menunjukkan bahwa model EfficientNetB3 berhasil dalam pelatihan dan evaluasi. Penurunan nilai *loss* dan peningkatan nilai akurasi yang konsisten menunjukkan bahwa model dapat mempelajari pola dari data dengan baik. Tidak adanya indikasi *overfitting* yang signifikan, di mana nilai *loss* dan akurasi pada data validasi tetap sejalan dengan data pelatihan, Hal ini menunjukkan bahwa model mempunyai kemampuan generalisasi yang baik. Artinya, model tersebut berkinerja baik tidak hanya pada data pelatihan, tetapi juga pada data baru yang belum pernah ada sebelumnya.

Gambar 7 merupakan *confusion matrix* yang menggambarkan kinerja model EfficientNetB3 dalam mengklasifikasikan lima jenis bunga: aster, dandelion, mawar, bunga matahari, dan tulip. Matriks ini menunjukkan jumlah prediksi yang benar untuk setiap kelas. Diagonal utama menunjukkan jumlah prediksi benar untuk setiap kelas, sementara sel-sel lainnya menunjukkan kesalahan klasifikasi. Misalnya, model memprediksi 75 dari 82 bunga *daisy* dengan benar, tetapi salah mengklasifikasikan beberapa sebagai *dandelion*, *rose*, *sunflower*, atau tulip. *Dandelion* memiliki akurasi tertinggi dengan 100 prediksi benar dari 104, menunjukkan bahwa model sangat baik dalam mengenali *dandelion*. Kesalahan klasifikasi relatif kecil untuk semua jenis bunga, menunjukkan bahwa model EfficientNetB3 bekerja cukup baik dalam membedakan antara berbagai jenis bunga dengan tingkat akurasi yang tinggi.



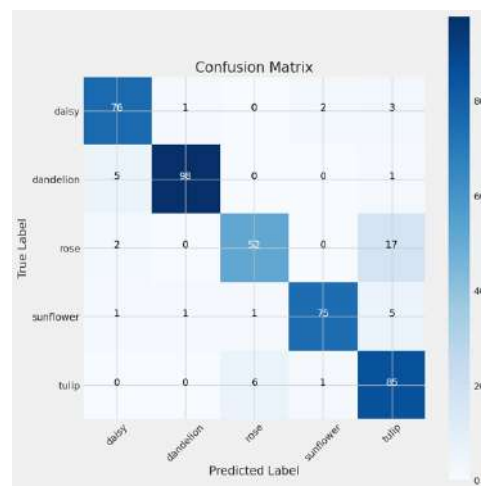
Gambar 7 Confusion Matrix EfficientNetB3

Adapun Gambar 8 merupakan grafik yang menunjukkan nilai *loss* dan akurasi selama pelatihan dan validasi menggunakan model VGG16. Grafik di sebelah kiri menggambarkan perubahan nilai *loss*, di mana *loss* pelatihan (garis merah) secara konsisten menurun seiring bertambahnya *epoch*, menunjukkan bahwa model belajar dengan baik pada data pelatihan. Namun, *loss* validasi (garis hijau) mulai meningkat setelah sekitar 10 *epoch*, yang mengindikasikan adanya *overfitting*. Grafik di sebelah kanan menunjukkan perubahan akurasi, di mana akurasi pelatihan (garis merah) meningkat tajam dan mencapai hampir 100%, sementara akurasi validasi (garis hijau) stabil di sekitar 85-90% dan mengalami sedikit fluktuasi, mencerminkan kinerja model yang stabil namun menunjukkan tanda-tanda *overfitting*. Secara keseluruhan, model VGG16 menunjukkan kinerja yang baik pada data pelatihan tetapi mungkin mengalami *overfitting* pada data validasi.





Gambar 8 Grafik Nilai Loss dan Accuracy VGG16



Gambar 9 Confusion Matrix VGG16

Tabel 2 Ringkasan Hasil (Result Summary)

Model	Training Accuracy	Validation Accuracy	Training Loss	Validation Loss	Comments
CNN	73,38%	71,76%	0,7730	0,7509	Model dasar CNN menunjukkan peningkatan kinerja yang stabil tanpa <i>overfitting</i> signifikan.
EfficientNetB3	98,50%	94,00%	0,050	0,300	Model <i>transfer learning</i> dengan EfficientNetB3 memberikan hasil yang sangat baik dan memiliki kemampuan generalisasi yang kuat.
VGG16	100%	85-90%	0,020	0,450	VGG16 menunjukkan performa tinggi pada data pelatihan tetapi mengalami <i>overfitting</i> pada data validasi.



Selanjutnya terdapat Gambar 9 yang merupakan *confusion matrix* vgg16, *confusion matrix* tersebut menunjukkan kinerja model VGG16 dalam mengklasifikasikan lima jenis bunga: aster, *dandelion*, mawar, bunga matahari, dan tulip. Model berhasil mengklasifikasikan sebagian besar gambar dengan benar, terutama untuk kelas *dandelion* dan tulip. Namun, terdapat beberapa kesalahan klasifikasi, terutama antara *rose* dan tulip, di mana gambar *rose* sering salah diklasifikasikan sebagai tulip. Meski begitu, hasil ini dapat memperlihatkan bahwasannya model memiliki akurasi yang cukup baik, tetapi masih memerlukan perbaikan untuk meningkatkan ketepatan dalam membedakan beberapa kelas bunga yang mirip secara visual. Informasi dari *confusion matrix* ini dapat digunakan untuk mengoptimalkan model lebih lanjut. Ringkasan perbandingan kinerja ketiga model (CNN, EfficientNetB3, dan VGG16) disajikan pada Tabel 2.

4. KESIMPULAN

Penelitian ini bertujuan untuk mengevaluasi dan membandingkan performa model CNN, VGG16, dan EfficientNetB3 dalam klasifikasi bunga menggunakan *dataset* yang terbatas. Model CNN dasar yang dibangun dari awal menunjukkan akurasi 73,38% pada data pelatihan dan 71,76% pada data validasi. Meskipun model ini mampu belajar dengan baik dan menunjukkan penurunan *loss* yang konsisten selama pelatihan, hasil ini menunjukkan bahwa kinerja model dasar masih terbatas dalam menangani data baru secara optimal.

Model *transfer learning* dengan EfficientNetB3 berhasil menunjukkan performa yang sangat baik dengan akurasi pelatihan 98,50% dan akurasi validasi 94,00%. Dengan nilai *loss* yang rendah pada kedua *dataset*, model ini berhasil meminimalkan *overfitting* dan memiliki kemampuan generalisasi yang kuat. Model ini terbukti unggul dalam klasifikasi bunga dibandingkan dengan model lainnya. Sebaliknya, model VGG16 mengalami masalah *overfitting* meskipun menunjukkan akurasi tinggi pada data pelatihan. Akurasi stabil di kisaran 85-90% pada data validasi dan nilai *loss* meningkat setelah beberapa *epoch*, menunjukkan keterbatasan model dalam menggeneralisasi data baru, yang mengurangi efektivitasnya dibandingkan EfficientNetB3.

Berdasarkan hasil penelitian ini, dapat disimpulkan bahwa model *transfer learning* dengan EfficientNetB3 memberikan performa terbaik dalam klasifikasi bunga dengan *dataset* terbatas, menjadikannya pilihan yang lebih tepat dibandingkan dengan model CNN dasar dan VGG16. Penelitian ini menunjukkan bahwa penggunaan model pra-terlatih tidak hanya mempercepat proses pelatihan, tetapi juga menghasilkan kinerja yang lebih konsisten dan akurat pada data validasi, sehingga dapat menjadi solusi efektif dalam tugas klasifikasi gambar dengan jumlah data yang terbatas.

DAFTAR PUSTAKA

- Alwanda, M. R., Ramadhan, R. P. K., & Alamsyah, D. (2020). Implementasi Metode Convolutional Neural Network Menggunakan Arsitektur LeNet-5 untuk Pengenalan Doodle. *Jurnal Algoritme*, 1(1), 45–56. <https://doi.org/10.35957/algoritme.v1i1.434>
- Arianty, R., Maukar, M., & Lestari, O. B. (2022). Pengukuran Kemiripan Fitur pada Sistem Temu Kembali Citra Berbasis Konten Menggunakan Euclidian Distance. *Jurnal Ilmiah Informatika Komputer*, 27(1), 1–18. <https://doi.org/10.35760/ik.2022.v27i1.6095>
- Bae, K. Il, Park, J., Lee, J., Lee, Y., & Lim, C. (2020). Flower Classification with Modified Multimodal Convolutional Neural Networks. *Expert Systems with Applications*, 159, Article ID: 113455. <https://doi.org/10.1016/j.eswa.2020.113455>
- Danial, N. H., & Setiawati, D. (2024). Convolutional Neural Network (CNN) Based on Artificial Intelligence in Periodontal Diseases Diagnosis. *Interdental Jurnal Kedokteran Gigi (IJKG)*, 20(1), 139–148. <https://doi.org/10.46862/interdental.v20i1.8641>
- Darmanto, H. (2019). Pengenalan Spesies Ikan Berdasarkan Kontur Otolith Menggunakan Convolutional Neural Network. *Joined Journal (Journal of Informatics Education)*, 2(1), 41. <https://doi.org/10.31331/joined.v2i1.847>
- Drenyovszki, R. (2024). Solving a Classification Problem Using Transfer Learning on Small Image Datasets. *Gradus*, 11(1). <https://doi.org/10.47833/2024.1.CSC.011>



- Falakhi, B., Achmal, E. F., Rizaldi, M., Athallah, R. R. R., & Yudistira, N. (2022). Perbandingan Model AlexNet dan ResNet dalam Klasifikasi Citra Bunga Memanfaatkan Transfer Learning. *Jurnal Ilmu Komputer Dan Agri-Informatika*, 9(1), 70–78. <https://doi.org/10.29244/jika.9.1.70-78>
- Fuadah, Y. N., Ubaidullah, I. D., Ibrahim, N., Taliningsing, F. F., Sy, N. K., & Pramuditho, M. A. (2022). Optimasi Convolutional Neural Network dan K-Fold Cross Validation pada Sistem Klasifikasi Glaukoma. *ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika*, 10(3), 728–741. <https://doi.org/10.26760/elkomika.v10i3.728>
- Harahap, M., Laia, E. M., Sitanggang, L. S., Sinaga, M., Sihombing, D. F., & Husein, A. M. (2022). Deteksi Penyakit Covid-19 pada Citra X-Ray dengan Pendekatan Convolutional Neural Network (CNN). *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, 6(1), 70–77. <https://doi.org/10.29207/resti.v6i1.3373>
- Ibrahim, N., Lestary, G. A., Hanafi, F. S., Saleh, K., Pratiwi, N. K. C., Haq, M. S., & Mastur, A. I. (2022). Klasifikasi Tingkat Kematangan Pucuk Daun Teh menggunakan Metode Convolutional Neural Network. *ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika*, 10(1), 162–176. <https://doi.org/10.26760/elkomika.v10i1.162>
- Karthikeyan, N. (2022). Review of Deep Transfer Learning Models for Image Classification. *International Journal of Recent Contributions from Engineering, Science & IT (IJES)*, 10(01), 17–28. <https://doi.org/10.3991/ijes.v10i01.29783>
- Li, Y., Lv, Y., Ding, Y., Zhu, H., Gao, H., & Zheng, L. (2024). Research on a Flower Recognition Method Based on Masked Autoencoders. *Horticulturae*, 10(5), Article ID: 517. <https://doi.org/10.3390/horticulturae10050517>
- Liu, J., Mei, S., Song, T., & Liu, H. (2022). Feature Extraction of 3D Chinese Rose Model Based on Color and Shape Features. *Frontiers in Plant Science*, 13, Article ID: 1042016. <https://doi.org/10.3389/FPLS.2022.1042016/BIBTEX>
- Narvekar, C., & Rao, M. (2020). Flower Classification Using CNN and Transfer Learning in CNN-Agriculture Perspective. *2020 3rd International Conference on Intelligent Sustainable Systems (ICISS)*, 660–664. <https://doi.org/10.1109/ICISS49785.2020.9316030>
- Nugroho, A. S. (2015). Analisis Keanekaragaman Jenis Tumbuhan Berbuah di Hutan Lindung Surokonto, Kendal, Jawa Tengah dan Potensinya Sebagai Kawasan Konservasi Burung. *Prosiding Seminar Nasional Masyarakat Biodiversitas Indonesia*, 1(3), 472–476. <https://doi.org/10.13057/psnmbi/m010316>
- Nuraini, R., Destriana, R., Nurnaningsih, D., Daniarti, Y., & Alexander, A. D. (2023). Sunflower Image Classification Using Multiclass Support Vector Machine Based on Histogram Characteristics. *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, 7(1), 146–152. <https://doi.org/10.29207/resti.v7i1.4673>
- Santos, C. F. G. Dos, & Papa, J. P. (2022). Avoiding Overfitting: A Survey on Regularization Methods for Convolutional Neural Networks. *ACM Computing Surveys*, 54(10s), Article ID: 213. <https://doi.org/10.1145/3510413>
- Saputra, T., Nurmaini, S., Roseno, M. T., & Syaputra, H. (2023). Heart Chamber Segmentation in Cardiomegaly Conditions Using the CNN Method with U-Net Architecture. *Jurnal Sisfokom (Sistem Informasi dan Komputer)*, 12(3), 455–461. <https://doi.org/10.32736/sisfokom.v12i3.1976>
- Tama, A. M., & Santi, R. C. N. (2023). Klasifikasi Jenis Tanaman Hias Menggunakan Metode Convolutional Neural Network (CNN). *INTECOMS: Journal of Information Technology and Computer Science*, 6(2), 764–770. <https://doi.org/10.31539/intecom.v6i2.7002>
- Widyaya, J. E., & Budi, S. (2021). Pengaruh Preprocessing Terhadap Klasifikasi Diabetic Retinopathy dengan Pendekatan Transfer Learning Convolutional Neural Network. *Jurnal Teknik Informatika dan Sistem Informasi*, 7(1). <https://doi.org/10.28932/jutisi.v7i1.3327>



Perbandingan Sensitivitas Metode SAW, MAUT dan WSM pada Anugerah Mutu Non-Akademik Universitas

Muhammad Galih Wonoseto ^{(1)*}, Muhammad Abu Shaker Hunaif ⁽²⁾

Departemen Informatika, UIN Sunan Kalijaga, Yogyakarta, Indonesia

e-mail : muhammad.wonoseto@uin-suka.ac.id, 20106050024@student.uin-suka.ac.id.

* Penulis korespondensi.

Artikel ini diajukan 16 Juni 2024, direvisi 12 Februari 2023, diterima 13 Februari 2025, dan dipublikasikan 31 Mei 2025.

Abstract

A Decision Support System works best with a suitable method. Unfortunately, not all methods are equally used. Two rarely used methods are the MAUT and WSM methods. To determine whether a method is more suitable for a case than another, a sensitivity test is conducted. By conducting sensitivity tests between the two methods and other commonly used methods, such as SAW, in the same case, it's possible to compare the sensitivity percentages of the three. One case that can be helped by a Decision Support System using the three methods is the ANOMIK assessment at universities. The three methods produced the same best alternative, namely Faculty 9. After conducting a sensitivity test, the results showed that the WSM method was the most sensitive, with a value of 4.954%, followed by the SAW method with a value of 4.901%, and finally the MAUT method with a value of 3.844%.

Keywords: Decision Support System, SAW, MAUT, WSM, University

Abstrak

Sistem Pendukung Keputusan membutuhkan metode yang cocok agar bisa mendapatkan hasil yang terbaik. Sayangnya, tidak semua metode digunakan secara merata. Dua metode yang jarang digunakan adalah metode MAUT dan WSM. Untuk mengetahui apakah suatu metode lebih cocok pada suatu kasus dibanding metode lain adalah dengan melakukan uji sensitivitas. Dengan dilakukannya uji sensitivitas antara kedua metode tersebut dengan metode lain seperti SAW pada satu kasus yang sama, maka dapat diketahui perbandingan persentase sensitivitas di antara ketiganya. Satu kasus yang bisa dibantu dengan Sistem Pendukung Keputusan menggunakan ketiga metode tersebut adalah masalah penilaian ANOMIK pada universitas. Ketiga metode menghasilkan alternatif terbaik yang sama, yaitu Fakultas 9. Setelah dilakukan uji sensitivitas didapatkan hasil bahwa metode WSM adalah yang paling sensitif dengan nilai 4,954% diikuti oleh metode SAW dengan nilai 4,901% dan yang terakhir adalah metode MAUT dengan 3,844%.

Kata Kunci: Sistem Pendukung Keputusan, SAW, MAUT, WSM, Universitas

1. PENDAHULUAN

Sistem Pendukung Keputusan atau sering disebut *Decision Support System* adalah sistem berbasis model yang terdiri dari prosedur-prosedur dalam pemrosesan data dan mempertimbangkannya untuk membantu dalam mengambil keputusan (Setiawan et al., 2022). Model atau metode yang digunakan dalam pengolahan data suatu Sistem Pendukung Keputusan akan menentukan kualitas hasil akhir keputusan yang didapat (Arifin & Suharjo, 2024; Nurhasanah et al., 2020). Dalam Sistem Pendukung Keputusan, terdapat berbagai model atau metode yang dapat digunakan untuk menentukan solusi terbaik. Beberapa dari metode tersebut adalah *Simple Additive Weighting* (SAW), *Multi Attribute Utility Theory* (MAUT), *Weighted Sum Model* (WSM), *Technique for Order of Preference by Similarity to Ideal Solution* (TOPSIS), dan lain sebagainya.

Setiap metode memiliki karakteristik dan prinsip yang berbeda dalam pengerjaannya dan pilihan metode yang tepat dapat sangat mempengaruhi kualitas keputusan yang dihasilkan. Salah satu cara menentukan metode terbaik untuk suatu kasus tertentu adalah dengan melakukan perbandingan sensitivitas antar metode sehingga bisa diketahui metode mana yang lebih relevan



Artikel ini didistribusikan mengikuti lisensi Atribusi-NonKomersial CC BY-NC sebagaimana tercantum pada <https://creativecommons.org/licenses/by-nc/4.0/>.

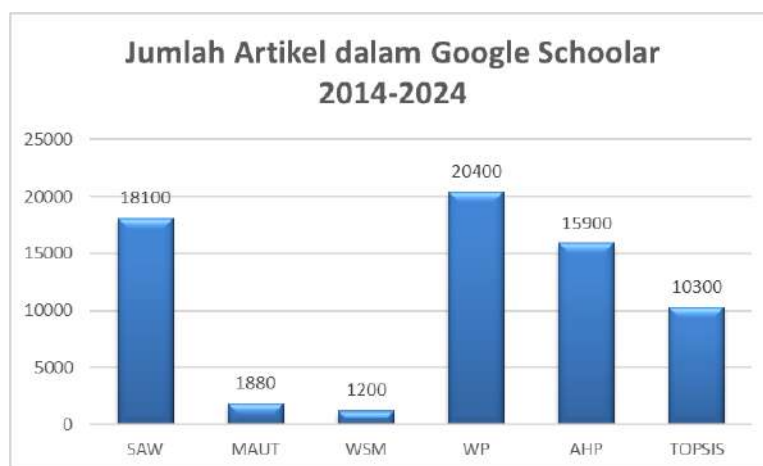
untuk kasus yang dihadapi (Rácz et al., 2019). Dalam bidang Multi-Criteria Decision Analysis (MCDA), penerapan analisis sensitivitas telah menjadi pendekatan mendasar untuk menguji ketahanan dan keandalan hasil yang diperoleh (Więckowski & Sałabun, 2023).

Sayangnya perbandingan sensitivitas seringkali hanya menggunakan beberapa metode yang sudah populer, seperti metode SAW atau WP. Penulis telah melakukan survey di Google Scholar dengan memasukkan keyword pencarian SAW, MAUT, WSM, WP AHP dan TOPSIS. Seperti yang dapat dilihat pada Tabel 1 dan Gambar 1, metode MAUT dan WSM paling jarang digunakan dan diteliti sehingga sulit diketahui pada kasus macam apa saja kedua metode tersebut cocok untuk digunakan. Oleh karena itu perlu dilakukan perbandingan sensitivitas di antara ketiga metode SAW, MAUT dan WSM pada satu kasus yang sama untuk mengetahui perbandingan tingkat sensitivitas ketiga metode tersebut.

Tabel 1 Jumlah Kemunculan Artikel dalam Google Scholar 2014-2024

Keyword / Metode	Jumlah
SAW "DSS"	18.100
MAUT "DSS"	1.880
WSM "DSS"	1.200
WP "DSS"	20.400
AHP "DSS"	15.900
TOPSIS "DSS"	10.300

Salah satu kasus yang dapat dibantu pengerjaannya dengan pemanfaatan Sistem Pendukung Keputusan sekaligus digunakan untuk menguji perbandingan sensitivitas antara ketiga metode SAW, MAUT, dan WSM adalah proses penghargaan Anugerah Mutu Non Akademik (ANOMIK) yang dilakukan oleh lembaga SPI di UIN Sunan Kalijaga Yogyakarta. Penghargaan ini bertujuan untuk mendorong proses penjaminan mutu non akademik di setiap unit dan fakultas yang ada. Dalam pelaksanaannya, SPI UIN Sunan Kalijaga memberikan Anugerah Mutu Non Akademik kepada unit dan fakultas yang memperoleh nilai tertinggi berdasarkan beberapa kriteria non akademik yang telah ditentukan.



Gambar 1 Jumlah Kemunculan Artikel dalam Google Scholar 2014-2024

Dengan menerapkan penggunaan SPK pada kasus ini lalu dilakukan uji sensitivitas, bisa diketahui mana di antara ketiga metode di atas yang paling sensitif pada kasus yang dihadapi. Pemberian penghargaan Anugerah Mutu Non Akademik kepada unit dan fakultas di UIN Sunan Kalijaga Yogyakarta membutuhkan metode Sistem Pendukung Keputusan dengan nilai sensitivitas yang tinggi mengingat skor setiap kriteria untuk masing-masing unit dan fakultas hanya memiliki sedikit perbedaan. Selain itu, untuk menjaga objektivitas penilaian, maka diperlukanlah metode Sistem Pendukung Keputusan yang baik.



Metode SAW sudah banyak digunakan dalam berbagai studi perbandingan, sedangkan dua metode lainnya, yaitu MAUT dan WSM, masih terbilang jarang digunakan. Salah satu karya yang menggunakan metode SAW adalah penelitian berjudul “Perbandingan Metode SAW dan MFEP Sistem Pendukung Keputusan Penyeleksian Atlet Pencak Silat” oleh Qomariah & Rangan (2020). Dalam penelitian tersebut, metode SAW dibandingkan dengan MFEP dalam kasus seleksi atlet pencak silat, dan hasilnya menunjukkan bahwa keduanya menghasilkan tiga alternatif terbaik yang sama.

Sementara itu, metode MAUT digunakan pada karya berjudul “Perbandingan Metode MFEP dan MAUT dalam Seleksi Calon Peserta Olimpiade Sains Nasional (OSN)” oleh Maharani & Nata (2020). Penelitian ini menerapkan metode MFEP dan MAUT untuk menentukan calon peserta terbaik dalam seleksi OSN. Hasilnya menunjukkan bahwa kedua metode menghasilkan alternatif terbaik yang berbeda, di mana metode MFEP memilih A13 dengan skor 16,96, sedangkan metode MAUT memilih A5 dengan skor 0,680.

Metode terakhir, WSM, digunakan pada karya berjudul “Perbandingan Metode Weighted Product dan Weighted Sum Model dalam Pemilihan Perguruan Swasta Terbaik Jurusan Komputer” oleh Solikhun (2017). Dalam penelitian tersebut, penulis menerapkan metode WP dan WSM pada kasus pemilihan perguruan swasta terbaik jurusan sensitif. Hasil yang diperoleh menunjukkan bahwa kedua metode menghasilkan alternatif terbaik yang sama, yaitu X1, namun posisi kedua dan ketiga berbeda. Pada metode WP, urutan kedua dan ketiga ditempati oleh X2 lalu X3, sementara pada metode WSM urutan kedua dan ketiga ditempati oleh X3 lalu X2.

2. METODE PENELITIAN

2.1 Sistem Pendukung Keputusan

Konsep Sistem Pendukung Keputusan (Gorry & Morton, 1971) pertama kali diperkenalkan pada awal tahun 1970-an oleh Scott Morton. Scott Morton mendefinisikan Sistem Pendukung Keputusan sebagai sistem berbasis komputer interaktif, yang membantu para pengambil keputusan untuk menggunakan data dan berbagai model untuk memecahkan masalah-masalah tidak terstruktur. Sistem Pendukung Keputusan (SPK) merupakan suatu sistem berbasis komputer yang mengolah data sesuai model demi mendukung suatu proses pengambilan atau pemilihan Keputusan (Eom, 2020). Proses pemilihan dengan berbagai kriteria dapat diselesaikan oleh sebuah sistem komputer yang mampu berinteraksi dengan pengambil keputusan (Syahputra et al., 2022). SPK memiliki berbagai model atau metode yang bisa digunakan, beberapa di antaranya adalah metode SAW, MAUT, WSM, AHP, TOPSIS, dan WP.

2.2 Metode Simple Additive Weighting (SAW)

Metode Simple Additive Weighting (SAW) (Fishburn, 1967) sering juga dikenal dengan istilah metode penjumlahan terbobot. Metode ini pertama kali dikemukakan oleh Fishburn dan MacCrimmon pada tahun 1967 (Rahayu et al., 2022; Resti, 2017; Sugianto et al., 2021). Konsep dasar metode SAW adalah mencari penjumlahan terbobot dari *rating* kinerja pada setiap alternatif dari semua atribut.

Adapun langkah penyelesaian dengan menggunakan metode SAW adalah sebagai berikut (Harahap et al., 2022):

- 1) Menentukan alternatif dan kriteria.
- 2) Menentukan bobot masing-masing kriteria.
- 3) Memberikan nilai *rating* kecocokan setiap alternatif untuk masing-masing kriteria.
- 4) Membuat matriks keputusan berdasarkan *rating* alternative untuk setiap kriteria.
- 5) Melakukan normalisasi matriks keputusan dengan Pers. (1) untuk kriteria *benefit* atau Pers. (2) untuk kriteria *cost*. Di mana dalam persamaan tersebut R_{ij} merupakan nilai *rating* ternormalisasi, X_y adalah nilai *rating* alternatif pada kriteria, $Max x_y$ menunjukkan nilai *rating* tertinggi pada kriteria, dan $Min x_y$ menyatakan nilai *rating* terendah pada kriteria.



$$R_{ij} = \frac{X_y}{\text{Max } x_y} \quad (1)$$

$$R_{ij} = \frac{\text{Min } x_y}{x_y} \quad (2)$$

- 6) Mencari nilai preferensi (V_i) dengan Pers. (3). Di mana dalam persamaan tersebut V_i adalah *ranking* tiap alternatif, W_j merupakan nilai bobot tiap kriteria, dan R_{ij} menunjukkan nilai *rating* ternormalisasi.

$$V_i = \sum_{j=1}^n W_j R_{ij} \quad (3)$$

- 7) Melakukan pemeringkatan dengan V_i terbesar menunjukkan alternatif pilihan terbaik.

2.3 Metode Multi-Attribute Utility Theory (MAUT)

Metode Multi-Attribute Utility Theory (MAUT) (Keeney, Raiffa, et al., 1979) adalah suatu metode yang diperkenalkan oleh Keeney dan Raiffa pada tahun 1976. Metode ini dapat didefinisikan sebagai sebuah algoritma dengan pendekatan menghitung evaluasi akhir pada sebuah nilai elemen x yang biasa disebut dengan nilai bobot dengan perhitungan penjumlahan dengan nilai bobot dalam dimensinya atau yang biasa disebut dengan utilitas (Maharani & Nata, 2020). Metode ini memiliki kelebihan di mana urutan preferensi untuk setiap alternatif dihitung bersamaan namun kelemahannya adalah bahwa sifat hasil keputusan yang didapat itu kurang jelas atau tidak menentu (Emovon & Oghenenyero, 2020).

Berikut merupakan langkah pengerjaan metode MAUT (Maharani & Nata, 2020):

- 1) Menentukan kriteria beserta bobotnya.
- 2) Menentukan alternatif dan nilai tiap kriteria masing-masing alternatif.
- 3) Melakukan normalisasi matriks dengan Pers. (4). Di mana dalam persamaan tersebut $U_{(x)}$ menunjukkan nilai alternatif ternormalisasi, x_i adalah nilai alternatif pada kriteria, x_i^- menunjukkan nilai minimum pada kriteria, dan x_i^+ menyatakan nilai maksimum pada kriteria.

$$U_{(x)} = \frac{x_i - x_i^-}{x_i^+ - x_i^-} \quad (4)$$

- 4) Melakukan perhitungan nilai utilitas dengan Pers. (5). Di mana dalam persamaan tersebut $V_{(x)}$ merupakan nilai utilitas, w_j adalah bobot kriteria, dan x_{ij} merupakan nilai alternatif (ternormalisasi) pada kriteria.

$$V_{(x)} = \sum_{i=1}^n w_j * x_{ij} \quad (5)$$

- 5) Melakukan pemeringkatan dengan nilai $V_{(x)}$ atau nilai utilitas tertinggi sebagai alternatif terbaik.

2.4 Metode Weighted Sum Model (WSM)

Metode Weighted Sum Model (WSM) adalah suatu metode yang diperkenalkan oleh Zadeh (1963). Metode ini menghasilkan solusi terbaik yaitu alternatif dengan nilai tertinggi untuk tipe *benefit* dan alternatif dengan nilai terendah untuk tipe *cost* (Malefaki et al., 2025). Metode ini memiliki perhitungan yang sederhana dan mudah dimengerti namun normalnya hanya cocok dengan kasus yang memiliki tipe kriteria yang sama untuk semua kriterianya, seperti *benefit* semua atau *cost* semua (Emovon & Oghenenyero, 2020).



Adapun untuk langkah pengerjaan metode WSM adalah sebagai berikut (Pahlevi, 2021):

- 1) Identifikasi kriteria dan alternatif.
- 2) Berikan bobot pada setiap kriteria dan nilai tiap alternatif pada masing-masing kriteria.
- 3) Mencari alternatif terbaik dengan menerapkan Pers. (6). Di mana dalam persamaan tersebut n adalah jumlah kriteria, w_j merupakan bobot tiap kriteria, a_{ij} menunjukkan nilai matriks.

$$A_i^{WSM-score} = \sum_{j=1}^n w_j a_{ij} \quad (6)$$

- 4) Melakukan pemeringkatan dengan nilai total tertinggi sebagai alternatif terbaik.

2.5 Uji Sensitivitas

Uji sensitivitas adalah sebuah proses untuk mengetahui nilai sensitivitas dari perbandingan berbagai metode dalam penyelesaian suatu kasus (Naik & Kiran, 2021; Vinogradova-Zinkevič, 2023). Metode ini bertujuan untuk mengetahui seberapa sensitifnya metode-metode yang dibandingkan tersebut jika diterapkan dalam penyelesaian suatu kasus yang sama (Panjaitan et al., 2021). Untuk langkah pengerjaannya adalah sebagai berikut (Marinković et al., 2024):

- 1) Menentukan bobot awal untuk semua kriteria.
- 2) Ubah nilai bobot atribut pertama sementara bobot atribut lainnya masih tetap bernilai sama. Lakukan normalisasi bila diperlukan.
- 3) Aplikasikan bobot baru pada metode yang digunakan (SAW, MAUT, WSM).
- 4) Geser perubahan ke bobot kedua dan begitu seterusnya dengan mengembalikan bobot yang telah diubah sebelumnya kembali menjadi bobot awal. Lalu lakukan langkah 3 dengan bobot yang baru. Begitu seterusnya hingga kriteria terakhir.
- 5) Hitung jumlah perubahan ranking yang terjadi dengan cara membandingkan berapa banyak perubahan *ranking* yang terjadi jika dibandingkan dengan kondisi bobot awal.
- 6) Hitung persentase perubahan *ranking* yang terjadi menggunakan Pers. (7). Di mana dalam persamaan tersebut T adalah total akhir perubahan *ranking*, i menunjukkan total perulangan, dan A merupakan jumlah atribut (kriteria) yang digunakan.

$$\frac{T}{i \times A} \times 100\% \quad (7)$$

2.6 Alur Penelitian

Penelitian yang akan dilakukan memiliki beberapa langkah atau tahapan sebagai berikut:

- 1) Menghubungi pihak SPI UIN Sunan Kalijaga Yogyakarta untuk meminta data yang dibutuhkan dalam penelitian.
- 2) Melakukan pemrosesan data yang telah terkumpul dengan menerapkan metode SPK yang telah ditentukan lalu melakukan uji sensitivitas untuk melihat persentase sensitivitas tiap metode untuk kasus yang dihadapi.
- 3) Menganalisis hasil uji sensitivitas yang didapat berdasarkan penelitian yang telah dilaksanakan.

3. HASIL DAN PEMBAHASAN

3.1 Pengumpulan Data

Setelah berkomunikasi dengan pihak SPI, didapatkan data kriteria, tipe dan bobot seperti yang dapat dilihat pada Tabel 2. Anugerah Mutu Non Akademik (ANOMIK) memiliki 11 kriteria penilaian dan masing-masing kriteria penilaian memiliki bobot atau tingkat kepentingan yang sama. Sehingga, masing-masing kriteria diberi bobot 1. Sementara itu, untuk alternatif pilihan adalah seperti yang terlihat pada Tabel 3. Selanjutnya adalah memasukkan data nilai skor atau *rating* masing-masing kriteria untuk setiap alternatif pilihan seperti yang dapat terlihat pada Tabel 4.



Tabel 2 Kriteria

Kode	Kriteria	Tipe	Bobot
C1	jumlah saldo temuan eksternal	Benefit	1
C2	jumlah temuan administrasi eksternal	Benefit	1
C3	jumlah saldo temuan internal	Benefit	1
C4	jumlah temuan internal	Benefit	1
C5	jumlah pegawai yang terkena hukuman disiplin (per SK)	Benefit	1
C6	membuat Daftar Inventaris Ruangan/Daftar Barang Ruangan (DIR/DBR) pada setiap ruangan	Benefit	1
C7	Terdapat Kartu Identitas Barang (KIB) dan Daftar Barang Lainnya (DBL),	Benefit	1
C8	persentase penyerapan anggaran per 30 juni 2023	Benefit	1
C9	mencatat Barang Milik Negara pada aplikasi Sakti BMN	Benefit	1
C10	memiliki buku catatan persediaan barang/ATK	Benefit	1
C11	telah melakukan stock opname secara berkala	Benefit	1

Tabel 3 Alternatif

Kode	Alternatif
A1	Unit 1
A2	Unit 2
A3	Unit 3
A4	Unit 4
A5	Unit 5
A6	Unit 6
A7	Lembaga 1
A8	Lembaga 2
A9	Fakultas 1
A10	Fakultas 2
A11	Fakultas 3
A12	Fakultas 4
A13	Fakultas 5
A14	Fakultas 6
A15	Fakultas 7
A16	Fakultas 8
A17	Fakultas 9

Tabel 4 Skor

Alternatif/Kriteria	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11
A1	38	25	100	100	96,07	80	80	67,09	100	50	70
A2	100	100	100	100	105	50	99,63	4,98	100	0	0
A3	100	97	100	94	105	98,04	100	68,26	100	34	67
A4	100	100	100	98	105	0	0	13,48	0	0	0
A5	100	99	47	94	105	50	39,39	56,14	75	50	45
A6	100	100	100	98	105	28,57	100	27,13	0	0	0
A7	100	100	100	98	105	8,33	8,33	71,35	25	33	29
A8	100	95	100	100	105	75	76,39	18,24	100	15	15
A9	57	96	100	92	105	25	89,23	47,90	100	90	90
A10	63	100	96	98	96,07	100	75	28,79	100	34	67
A11	54	100	97	98	101,79	100	81,93	30,35	100	95	95
A12	91	100	71	76	87,14	100	97,15	24,78	100	95	95
A13	100	100	100	86	105	100	23,45	30,51	50	50	60
A14	76	98	100	82	103,21	25	16,22	27,09	50	52	65
A15	79	100	100	92	105	100	98,08	29,15	100	34	67
A16	47	98	95	98	105	97,09	100	31,26	100	80	100
A17	100	95	100	98	105	100	100	41,75	100	67	67



3.2 Penerapan Metode SAW

Langkah yang diambil dalam metode SAW setelah identifikasi adalah membuat matriks keputusan sesuai dengan tabel skor yang telah dibuat sebelumnya. Selanjutnya adalah melakukan normalisasi terhadap matriks yang telah dibentuk menggunakan Pers. (1) atau Pers. (2). Matriks keputusan setelah dinormalisasi adalah seperti yang terlihat pada Tabel 5. Selanjutnya adalah mencari nilai preferensi menggunakan Pers. (3). Hasil nilai preferensi tiap alternatif seperti yang terlihat pada Tabel 6.

Tabel 5 Matriks Keputusan Ternormalisasi

	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11
A1	0,38	0,25	1	1	0,92	0,8	0,8	0,94	1	0,53	0,7
A2	1	1	1	1	1	0,5	1	0,07	1	0	0
A3	1	0,97	1	0,94	1	0,98	1	0,96	1	0,36	0,67
A4	1	1	1	0,98	1	0	0	0,19	0	0	0
A5	1	0,99	0,47	0,94	1	0,5	0,39	0,79	0,75	0,53	0,45
A6	1	1	1	0,98	1	0,29	1	0,38	0	0	0
A7	1	1	1	0,98	1	0,08	0,08	1	0,25	0,35	0,29
A8	1	0,95	1	1	1	0,75	0,76	0,26	1	0,16	0,15
A9	0,57	0,96	1	0,92	1	0,25	0,89	0,67	1	0,95	0,9
A10	0,63	1	0,96	0,98	0,92	1	0,75	0,4	1	0,36	0,67
A11	0,54	1	0,97	0,98	0,97	1	0,82	0,43	1	1	0,95
A12	0,91	1	0,71	0,76	0,83	1	0,97	0,35	1	1	0,95
A13	1	1	1	0,86	1	1	0,24	0,43	0,5	0,53	0,6
A14	0,76	0,98	1	0,82	0,98	0,25	0,16	0,38	0,5	0,55	0,65
A15	0,79	1	1	0,92	1	1	0,98	0,41	1	0,36	0,67
A16	0,47	0,98	0,95	0,98	1	0,97	1	0,44	1	0,84	1
A17	1	0,95	1	0,98	1	1	1	0,59	1	0,71	0,67

Tabel 6 Preferensi SAW

Kode	Nilai
A1	8,311503
A2	7,566092
A3	9,874926
A4	5,168916
A5	7,806991
A6	6,645917
A7	7,034035
A8	8,027339
A9	9,111011
A10	8,666319
A11	9,654163
A12	9,478678
A13	8,148331
A14	7,032123
A15	9,127213
A16	9,631061
A17	9,89031

Langkah terakhir adalah melakukan pemeringkatan berdasarkan nilai preferensi yang telah didapat sebelumnya. Setelah dilakukan pemeringkatan didapatkan hasil sebagaimana yang terlihat pada Tabel 7. Pada Tabel 7, dapat dilihat bahwa alternatif dengan nilai tertinggi adalah Fakultas 9 dengan nilai 9,89031, diikuti dengan Unit 3 dengan nilai 9,874926 di posisi kedua dan



Fakultas 3 dengan nilai 9,654163 di posisi ketiga sedangkan posisi terendah adalah Unit 4 dengan nilai 5,168916.

Tabel 7 Ranking SAW

Kode	Alternatif	Nilai	Ranking
A17	Fakultas 9	9,89031	1
A3	Unit 3	9,874926	2
A11	Fakultas 3	9,654163	3
A16	Fakultas 8	9,631061	4
A12	Fakultas 4	9,478678	5
A15	Fakultas 7	9,127213	6
A9	Fakultas 1	9,111011	7
A10	Fakultas 2	8,666319	8
A1	Unit 1	8,311503	9
A13	Fakultas 5	8,148331	10
A8	Lembaga 2	8,027339	11
A5	Unit 5	7,806991	12
A2	Unit 2	7,566092	13
A7	Lembaga 1	7,034035	14
A14	Fakultas 6	7,032123	15
A6	Unit 6	6,645917	16
A4	Unit 4	5,168916	17

3.3 Penerapan Metode MAUT

Setelah identifikasi, langkah selanjutnya dalam metode MAUT adalah melakukan normalisasi matriks yang telah dibuat berdasarkan tabel skor di atas dengan Pers. (4). Hasil matriks ternormalisasi adalah sebagaimana yang terlihat pada Tabel 8. Selanjutnya adalah mencari nilai utilitas yang akan digunakan untuk pemeringkatan akhir menggunakan Pers. (5). Hasilnya seperti yang terlihat pada Tabel 9.

Tabel 8 Matriks Ternormalisasi

	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11
A1	0	0	1	1	0,5	0,8	0,8	0,94	1	0,53	0,7
A2	1	1	1	1	1	0,5	1	0	1	0	0
A3	1	0,96	1	0,75	1	0,98	1	0,95	1	0,36	0,67
A4	1	1	1	0,92	1	0	0	0,13	0	0	0
A5	1	0,99	0	0,75	1	0,5	0,39	0,77	0,75	0,53	0,45
A6	1	1	1	0,92	1	0,29	1	0,33	0	0	0
A7	1	1	1	0,92	1	0,08	0,08	1	0,25	0,35	0,29
A8	1	0,93	1	1	1	0,75	0,76	0,2	1	0,16	0,15
A9	0,31	0,95	1	0,67	1	0,25	0,89	0,65	1	0,95	0,9
A10	0,40	1	0,93	0,92	0,5	1	0,75	0,36	1	0,36	0,67
A11	0,26	1	0,94	0,92	0,82	1	0,82	0,38	1	1	0,95
A12	0,86	1	0,45	0	0	1	0,97	0,4	1	1	0,95
A13	1	1	1	0,42	1	1	0,24	0,39	0,5	0,53	0,6
A14	0,61	0,97	1	0,25	0,9	0,25	0,16	0,33	0,5	0,55	0,65
A15	0,66	1	1	0,67	1	1	0,98	0,36	1	0,36	0,67
A16	0,15	0,97	0,91	0,92	1	0,97	1	0,4	1	0,84	1
A17	1	0,93	1	0,92	1	1	1	0,55	1	0,71	0,67

Langkah terakhir adalah melakukan pemeringkatan di mana nilai utilitas tertinggi akan menjadi alternatif pilihan terbaik. Hasil pemeringkatan dengan metode MAUT seperti yang terlihat pada Tabel 10, Pada Tabel 10, dapat dilihat bahwa alternatif dengan nilai tertinggi adalah Fakultas 9 dengan nilai 9,779177, diikuti dengan Unit 3 dengan nilai 9,671672 di posisi kedua dan Fakultas



8 dengan nilai 9,149723 di posisi ketiga sedangkan posisi terendah adalah Unit 4 dengan nilai 5,044728.

Tabel 9 Nilai Utilitas

Kode	Nilai
A1	7,262066
A2	7,4963
A3	9,671672
A4	5,044728
A5	7,12766
A6	6,536082
A7	6,970702
A8	7,954818
A9	8,556137
A10	7,88103
A11	9,090019
A12	7,527467
A13	7,662047
A14	6,178638
A15	8,700794
A16	9,149723
A17	9,779177

Tabel 10 Pemeringkatan MAUT

Kode	Alternatif	Nilai	Ranking
A17	Fakultas 9	9,779177	1
A3	Unit 3	9,671672	2
A16	Fakultas 8	9,149723	3
A11	Fakultas 3	9,090019	4
A15	Fakultas 7	8,700794	5
A9	Fakultas 1	8,556137	6
A8	Lembaga 2	7,954818	7
A10	Fakultas 2	7,88103	8
A13	Fakultas 5	7,662047	9
A12	Fakultas 4	7,527467	10
A2	Unit 2	7,4963	11
A1	Unit 1	7,262066	12
A5	Unit 5	7,12766	13
A7	Lembaga 1	6,970702	14
A6	Unit 6	6,536082	15
A14	Fakultas 6	6,178638	16
A4	Unit 4	5,044728	17

3.4 Penerapan Metode WSM

Setelah identifikasi, langkah selanjutnya dalam metode WSM adalah mencari alternatif terbaik dengan menggunakan Pers. (6). Hasil perhitungan dengan rumus tersebut seperti yang terlihat pada Tabel 11. Langkah terakhir adalah melakukan pemeringkatan dari nilai tertinggi sampai terendah di mana alternatif dengan nilai tertinggi adalah alternatif terbaik untuk dipilih. Hasil pemeringkatan bisa dilihat pada Tabel 12. Pada Tabel 12, dapat dilihat bahwa alternatif dengan nilai tertinggi adalah Fakultas 9 dengan nilai 973,7458, diikuti dengan Unit 3 dengan nilai 963,3 di posisi kedua dan Fakultas 3 dengan nilai 953,0774 di posisi ketiga sedangkan posisi terendah adalah Unit 4 dengan nilai 516,48,



Tabel 11 Nilai WSM

Kode	Nilai
A1	806,16
A2	759,61
A3	963,3
A4	516,48
A5	760,53
A6	658,7002
A7	678,0212
A8	799,6239
A9	892,1336
A10	857,8595
A11	953,0774
A12	937,0703
A13	804,9547
A14	694,5153
A15	904,2287
A16	951,3465
A17	973,7458

Tabel 12 *Ranking* WSM

Kode	Alternatif	Nilai	Ranking
A17	Fakultas 9	973,7458	1
A3	Unit 3	963,3	2
A11	Fakultas 3	953,0774	3
A16	Fakultas 8	951,3465	4
A12	Fakultas 4	937,0703	5
A15	Fakultas 7	904,2287	6
A9	Fakultas 1	892,1336	7
A10	Fakultas 2	857,8595	8
A1	Unit 1	806,16	9
A13	Fakultas 5	804,9547	10
A8	Lembaga 2	799,6239	11
A5	Unit 5	760,53	12
A2	Unit 2	759,61	13
A14	Fakultas 6	694,5153	14
A7	Lembaga 1	678,0212	15
A6	Unit 6	658,7002	16
A4	Unit 4	516,48	17

3.5 Uji Sensitivitas

Uji sensitivitas perlu dilakukan untuk mengetahui seberapa sensitif metode yang digunakan terhadap kasus yang dihadapi. Semakin tinggi persentase perubahan ranking yang terjadi maka semakin sensitif metode tersebut di mana metode dengan sensitivitas tertinggi adalah yang paling cocok untuk digunakan pada kasus yang dihadapi. Di sini akan dilakukan 9 kali percobaan dengan ketentuan sebagai berikut:

- Percobaan 1: Menambah bobot sebesar 0,25.
- Percobaan 2: Menambah bobot sebesar 0,5 dengan penambahan bobot 0,25.
- Percobaan 3: Menambah bobot sebesar 1 dengan penambahan bobot 0,5.
- Percobaan 4: Menambah bobot sebesar 1,5 dengan penambahan bobot 0,5.
- Percobaan 5: Menambah bobot sebesar 2 dengan penambahan bobot 0,5.
- Percobaan 6: Menambah bobot sebesar 2,5 dengan penambahan bobot 0,5.
- Percobaan 7: Menambah bobot sebesar 3 dengan penambahan bobot 0,5.
- Percobaan 8: Menambah bobot sebesar 3,5 dengan penambahan bobot 0,5.



- i) Percobaan 9: Menambah bobot sebesar 4 dengan penambahan bobot 0,5.

Tabel 13 Perubahan Ranking Percobaan 1

No.	Perubahan Bobot Kriteria	Perubahan Ranking		
		SAW	MAUT	WSM
1	C1+ 0,25	0	2	3
2	C2+ 0,25	2	2	3
3	C3+ 0,25	0	0	2
4	C4+ 0,25	0	2	2
5	C5+ 0,25	0	2	2
6	C6+ 0,25	2	0	2
7	C7+ 0,25	6	0	8
8	C8+ 0,25	4	2	0
9	C9+ 0,25	4	2	4
10	C10+ 0,25	4	0	4
11	C11+ 0,25	4	0	0
Total		26	12	30

Tabel 13 menyajikan hasil uji sensitivitas pada Percobaan 1 di mana bobot akan ditambah sebesar 0,25. Selanjutnya adalah menghitung persentase perubahan yang terjadi dengan Pers. (7). Untuk persentase yang didapat pada Percobaan 1 adalah sebagai berikut.

- a) $SAW = \frac{26}{11 \times 11} \times 100\% = 0,214\%$
 b) $MAUT = \frac{12}{11 \times 11} \times 100\% = 0,099\%$
 c) $WSM = \frac{30}{11 \times 11} \times 100\% = 0,247\%$

Untuk Percobaan 1, dapat dilihat bahwa yang paling sensitif dengan persentase sensitivitas 0,247% adalah metode WSM, diikuti dengan metode SAW dengan persentase 0,214% dan yang terendah adalah metode MAUT dengan 0,099%. Mengikuti ketentuan percobaan seperti yang telah ditentukan di atas, maka akan didapatkan hasil keseluruhan seperti yang terlihat pada Tabel 14. Berdasarkan Tabel 14, metode WSM adalah metode paling sensitif dengan total persentase perubahan dalam 9 kali percobaan sebesar 4,954%, diikuti dengan metode SAW dengan 4,901% dan yang terakhir adalah metode MAUT dengan 3,844%.

Tabel 14 Total Persentase Sensitivitas

Percobaan ke-	Total Iterasi	Persentase Sensitivitas		
		SAW	MAUT	WSM
1	11	0,214%	0,099%	0,247%
2	22	0,285%	0,169%	0,314%
3	22	0,479%	0,334%	0,450%
4	33	0,542%	0,404%	0,534%
5	44	0,592%	0,471%	0,599%
6	55	0,646%	0,522%	0,647%
7	66	0,684%	0,573%	0,687%
8	77	0,715%	0,616%	0,722%
9	88	0,739%	0,653%	0,751%
Total	418	4,901%	3,844%	4,954%

3.6 Pembahasan Hasil Akhir

Berdasarkan penelitian yang berhasil dilakukan di atas, diketahui bahwa metode WSM lah yang paling dianjurkan untuk digunakan dalam penyelesaian kasus yang dihadapi sedangkan metode MAUT adalah yang paling kurang dianjurkan untuk digunakan. Selain itu, metode SAW bisa dijadikan alternatif metode yang digunakan karena perbedaan total persentasenya dengan total persentase WSM sangatlah kecil, hanya sebesar 0,053%. Selain itu, fakta bahwa metode SAW



adalah metode yang paling sensitif pada Percobaan 3 dan Percobaan 4 yang juga membuktikan bahwa metode SAW bisa lebih sensitif dibanding metode WSM apabila bobot kriteria yang digunakan berada pada *range* tertentu.

Meski kasus ini bukan kasus yang ideal untuk metode SAW karena tidak adanya pembobotan lokal, berdasarkan hasil yang didapat bisa diketahui bahwa metode SAW masih bisa digunakan sebagai alternatif metode pada kasus yang dihadapi. Bukan hanya itu, apabila pembobotan kriteria berubah sehingga menyebabkan adanya pembobotan lokal maka bisa saja metode SAW menjadi metode yang lebih cocok untuk digunakan dibanding metode WSM. Hal ini dikarenakan pembobotan baru tersebut yang menyebabkan adanya pembobotan lokal akan membuat kasus yang dihadapi menjadi kasus yang ideal untuk metode SAW. Sementara itu, keadaan pembobotan awal seperti di atas sudah cukup untuk menjadikan kasus yang dihadapi sebagai kasus yang ideal untuk metode MAUT dan WSM namun tidak untuk metode SAW. Oleh karena itu, dilihat dari seberapa kecil perbedaan total persentase sensitivitas antara metode SAW dan WSM pada keadaan pembobotan awal, maka ada kemungkinan bahwa metode SAW akan menjadi lebih sensitif dibanding metode WSM apabila kasus yang dihadapi memiliki pembobotan berbeda yang memenuhi kebutuhan kasus ideal metode SAW yaitu pembobotan lokal.

4. KESIMPULAN

Berdasarkan hasil penelitian yang telah dilakukan di atas dapat disimpulkan bahwa metode SAW, MAUT, dan WSM dapat digunakan dalam menyelesaikan kasus penilaian ANOMIK di suatu Universitas. Dengan melakukan proses analisis sensitivitas sebanyak 9 kali dengan total iterasi sebanyak 418 kali, diketahui metode SAW memiliki total persentase sensitivitas sebesar 4,901% sedangkan metode MAUT memiliki total persentase sensitivitas sebesar 3,844% dan metode WSM memiliki total persentase sensitivitas sebesar 4,954%. Metode dengan total persentase sensitivitas terbesar pada kasus yang dihadapi di antara ketiga metode tersebut adalah metode WSM.

Dari hasil yang didapat, bisa disimpulkan bahwa metode yang akan digunakan pada kasus yang dihadapi adalah metode WSM karena metode tersebut yang paling sensitif berdasarkan pada tes uji sensitivitas yang telah dilakukan di atas. Bisa dilihat juga bahwa metode SAW juga masih cukup akurat dikarenakan perbedaan total persentase sensitivitasnya yang hanya berbeda 0,053% dengan metode WSM sehingga bisa dijadikan metode alternatif terutama untuk *range* bobot seperti pada Percobaan 3 dan 4 di mana metode ini terbukti lebih sensitif dibanding metode WSM. Sementara metode MAUT kurang dianjurkan untuk digunakan karena memiliki perbedaan total persentase sensitivitas yang cukup jauh dibanding kedua metode lainnya, yaitu 1,057% dari metode SAW dan 1,110% dari metode WSM.

DAFTAR PUSTAKA

- Arifin, M. Z., & Suharjo, I. (2024). Penerapan Metode Simple Additive Weighting (SAW) dalam Sistem Pendukung Keputusan Pemilihan Kapal Sebagai Moda Transportasi Laut. *Infotek: Jurnal Informatika dan Teknologi*, 7(2), 338–349. <https://doi.org/10.29408/jit.v7i2.25940>
- Emovon, I., & Ogheniyerovwho, O. S. (2020). Application of MCDM Method in Material Selection for Optimal Design: A Review. *Results in Materials*, 7, Article ID: 100115. <https://doi.org/10.1016/j.rinma.2020.100115>
- Eom, S. B. (2020). Decision Support Systems. In *Oxford Research Encyclopedia of Politics*. Oxford University Press. <https://doi.org/10.1093/acrefore/9780190228637.013.1008>
- Fishburn, P. C. (1967). Letter to the Editor—Additive Utilities with Incomplete Product Sets: Application to Priorities and Assignments. *Operations Research*, 15(3), 537–542. <https://doi.org/10.1287/opre.15.3.537>
- Gorry, G. A., & Morton, M. S. S. (1971). *A Framework for Management Information Systems*. Massachusetts Institute of Technology. <https://dspace.mit.edu/bitstream/handle/1721.1/47936/frameworkformana00gorr.pdf>
- Harahap, M. A. K., Hardisal, H., Ahyuna, A., & Rahim, R. (2022). Leveraging the Decision Support System and Simple Additive Weighting Method for Optimal Retail Location Identification.



- JINAV: *Journal of Information and Visualization*, 3(2), 174–180. <https://doi.org/10.35877/454RI.jinav1485>
- Keeney, R. L., Raiffa, H., & Rajala, D. W. (1979). Decisions with Multiple Objectives: Preferences and Value Trade-Offs. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(7), 403. <https://doi.org/10.1109/TSMC.1979.4310245>
- Vinogradova-Zinkevič, I. (2023). Comparative Sensitivity Analysis of Some Fuzzy AHP Methods. *Mathematics*, 11(24), Article ID: 4984. <https://doi.org/10.3390/math11244984>
- Maharani, D., & Nata, A. (2020). Perbandingan Metode MFEP dan MAUT dalam Seleksi Calon Peserta Olimpiade Sains Nasional (OSN). *JURTEKSI (Jurnal Teknologi dan Sistem Informasi)*, 6(3), 247–252. <https://doi.org/10.33330/jurtekxi.v6i3.793>
- Malefaki, S., Markatos, D., Filippatos, A., & Pantelakis, S. (2025). A Comparative Analysis of Multi-Criteria Decision-Making Methods and Normalization Techniques in Holistic Sustainability Assessment for Engineering Applications. *Aerospace*, 12(2), Article ID: 100. <https://doi.org/10.3390/aerospace12020100>
- Marinković, G., Ilić, Z., Nestorović, Ž., Božić, M., & Bulatović, V. (2024). Sensitivity of Multi-Criteria Analysis Methods in Rural Land Consolidation Project Ranking. *Land*, 13(2), Article ID: 245. <https://doi.org/10.3390/land13020245>
- Naik, D. L., & Kiran, R. (2021). A novel sensitivity-based method for feature selection. *Journal of Big Data*, 8(1), 128. <https://doi.org/10.1186/s40537-021-00515-w>
- Nurhasanah, E. K., Abadi, S., & Sukanto, P. (2020). Sistem Pendukung Keputusan Pemilihan Mahasiswa Berprestasi dengan Metode Simple Additive Weighting. *TEKNOSAINS: Jurnal Sains, Teknologi dan Informatika*, 7(2), 107–118. <https://doi.org/10.37373/tekno.v7i2.18>
- Pahlevi, R. F. (2021). Implementasi Algoritma Weighted Sum Model dalam Sistem Penjaminan Mutu Perguruan Tinggi. *JATISI (Jurnal Teknik Informatika dan Sistem Informasi)*, 8(2), 935–946. <https://doi.org/10.35957/jatisi.v8i2.700>
- Panjaitan, D. C., Juliansa, H., & Yanto, R. (2021). Perbandingan Metode SAW dan WP pada Sistem Pendukung Keputusan dalam Kasus Pemilihan Kegiatan Ekstrakurikuler. *Jurnal Ilmiah Binary STMIK Bina Nusantara Jaya Lubuklinggau*, 3(1), 30–38. <https://doi.org/10.52303/jb.v3i1.38>
- Qomariah, S., & Rangan, A. Y. (2020). Perbandingan Metode SAW dan MFEP Sistem Pendukung Keputusan Penyeleksian Atlet Pencak Silat. *Just TI (Jurnal Sains Terapan Teknologi Informasi)*, 12(1), 8–12. <https://doi.org/10.46964/justti.v12i1.175>
- Rácz, A., Bajusz, D., & Héberger, K. (2019). Multi-Level Comparison of Machine Learning Classifiers and Their Performance Metrics. *Molecules*, 24(15), Article ID: 2811. <https://doi.org/10.3390/molecules24152811>
- Rahayu, M. I., Siregar, M. K., & Desnia, M. (2022). Sistem Rekomendasi Pemilihan Produk Basic Skincare Berdasarkan Jenis Kulit Menggunakan Algoritma Simple Additive Weighting (SAW). *Jurnal Teknologi Informasi dan Komunikasi*, 12(2), 1–6. <https://doi.org/10.58761/jurtikstmikbandung.v12i2.170>
- Resti, N. C. (2017). Penerapan Metode Simple Additive Weighting (SAW) pada Sistem Pendukung Keputusan Pemilihan Lokasi untuk Cabang Baru Toko Pakan UD. Indo Multi Fish. *INTENSIF*, 1(2), 102–107. <https://doi.org/10.29407/intensif.v1i2.839>
- Setiawan, G. W., Wahyudi, J., & Sudarsono, A. (2022). Analisis Perbandingan Metode SAW dan Metode Topsis Melalui Pendekatan Uji Sensitivitas Penilaian Kinerja Pegawai (Studi Kasus: Dinas Perhubungan Bengkulu Tengah). *MEANS (Media Informasi Analisa dan Sistem)*, 6(2), 169–173. <https://doi.org/10.54367/means.v6i2.1528>
- Solikhun, S. (2017). Perbandingan Metode Weighted Product dan Weighted Sum Model dalam Pemilihan Perguruan Swasta Terbaik Jurusan Komputer. *KLIK - Kumpulan Jurnal Ilmu Komputer*, 4(1), 70–87. <https://doi.org/10.20527/klik.v4i1.75>
- Sugianto, R. A., Roslina, R., & Situmorang, Z. (2021). Kombinasi Metode Simple Additive Weigthing dan Weigthed Product untuk Seleksi Proposal Program Kreativitas Mahasiswa. *Jurnal Media Informatika Budidarma*, 5(2), 564–572. <https://doi.org/10.30865/mib.v5i2.2929>
- Syahputra, Y. H., Santoso, I., & Lubis, Z. (2022). Sistem Pendukung Keputusan Penerimaan Karyawan Terbaik Menggunakan Metode Weighted Sum Model (WSM). *Explorer*, 2(2), 39–47. <https://doi.org/10.47065/explorer.v2i2.249>



- Więckowski, J., & Sałabun, W. (2023). Sensitivity Analysis Approaches in Multi-Criteria Decision Analysis: A Systematic Review. *Applied Soft Computing*, 148, Article ID: 110915. <https://doi.org/10.1016/j.asoc.2023.110915>
- Zadeh, L. A. (1963). Optimality and Non-Scalar-Valued Performance Criteria. *IEEE Transactions on Automatic Control*, 8(1), 59–60. <https://doi.org/10.1109/TAC.1963.1105511>



Algoritma Random Forest dan Synthetic Minority Oversampling Technique (SMOTE) untuk Deteksi Diabetes

Nurussakinah ^{(1)*}, Muhammad Faisal ⁽²⁾, Irwan Budi Santoso ⁽³⁾

Departemen Teknik Informatika, UIN Maulana Malik Ibrahim, Malang, Indonesia

e-mail : nurussakinah2205@gmail.com, {mfaisal,irwan}@ti.uin-malang.ac.id.

* Penulis korespondensi.

Artikel ini diajukan 24 Juni 2024, direvisi 12 April 2025, diterima 15 April 2025, dan dipublikasikan 31 Mei 2025.

Abstract

Diabetes is one of the challenges in global health. Indonesia ranks 5th in the world with the highest rate of diabetes. This research uses the Random Forest algorithm for diabetes detection. The purpose of this study is to detect diabetes using the Random Forest algorithm, which provides accurate and efficient results in the early diagnosis of diabetic patients. The data used is secondary data, specifically the "Diabetes Dataset," which consists of 952 data points and has 17 features. The test scenario in this study divides the data into three parts, namely scenario 1 (90:10 ratio), scenario 2 (70:30 ratio), and scenario 3 (50:50 ratio). In each scenario, a comparison is made between using SMOTE and not using it. The best performance results are obtained in scenario 1, which uses SMOTE, producing 97% accuracy, 100% precision, 94% recall, and an F1-score of 97%.

Keywords: Detection, Diabetes, Random Forest, Synthetic Minority Oversampling Technique, Ensemble

Abstrak

Penyakit diabetes merupakan salah satu tantangan pada bidang kesehatan global. Indonesia menempati urutan ke-5 yang memiliki penyakit diabetes tertinggi dunia. Penelitian ini menggunakan algoritma Random Forest untuk deteksi diabetes. Tujuan penelitian untuk mendeteksi diabetes dengan algoritma Random Forest yang memberikan hasil akurat dan efisien dalam diagnosis dini pasien diabetes. Data yang digunakan merupakan data sekunder "Diabetes Dataset" yang terdiri dari 952 data dan memiliki 17 fitur. Skenario uji coba pada penelitian ini membagi data yang terdiri dari 3 bagian, yaitu skenario 1 rasio 90:10, skenario 2 rasio 70:30, skenario 3 rasio 50:50. Pada masing-masing skenario diterapkan perbandingan antara menggunakan SMOTE dan tidak. Hasil performa terbaik diperoleh pada skenario 1 yang menggunakan SMOTE yaitu, menghasilkan akurasi sebesar 97%, presisi sebesar 100%, *recall* sebesar 94%, dan yang terakhir yaitu *F1-score* yang menghasilkan 97%.

Kata Kunci: Deteksi, Diabetes, Random Forest, Synthetic Minority Oversampling Technique, Ensemble

1. PENDAHULUAN

Diabetes adalah penyakit kronis yang berdampak signifikan pada kesehatan global (Sun et al., 2023). Berdasarkan data International Diabetes Federation (IDF) tahun 2021, sekitar 537 juta orang berusia 20-79 tahun menderita diabetes (Magliano & Boyko, 2013). Indonesia menempati peringkat kelima di dunia dalam jumlah kasus tertinggi yang mencapai sekitar 19,5 juta jiwa di tahun 2021 dan hal ini akan terus meningkat setiap tahunnya yang diperkirakan mencapai angka 28,6 juta di tahun 2045. Kementerian Kesehatan Republik Indonesia (2023) menyatakan bahwa diabetes menjadi angka kematian tertinggi ketiga setelah penyakit stroke dan jantung.

Diabetes ditandai dengan hiperglikemia, yaitu kondisi kadar gula darah yang tinggi akibat ketidakmampuan tubuh untuk memproduksi atau menggunakan insulin dengan efektif (Hana, 2020). Insulin sebagai hormon yang penting dalam tubuh untuk mengatur metabolisme gula darah (Rahman et al., 2021). Berbagai faktor lain seperti gaya hidup dan faktor genetik juga berperan dalam risiko mengalami diabetes (Faida & Santik, 2020). Pentingnya diagnosis dini



sangat ditekankan untuk mencegah komplikasi serius dan meningkatkan kualitas hidup penderita (Karyadiputra & Setiawan, 2022).

Kemajuan teknologi khususnya di bidang *data mining* memiliki potensi yang sangat besar untuk mengidentifikasi pola data yang penting dalam diagnosis dini diabetes (Elfaladonna & Rahmadani, 2019). Salah satu metode *data mining* yang efektif adalah deteksi. Deteksi memungkinkan pengelompokan data pasien berdasarkan fitur-fitur tertentu (Aris & Benyamin, 2019). Algoritma Random Forest merupakan bagian dari metode *decision tree*, yang dikenal memiliki akurasi tinggi dan kemampuannya dalam menangani data dalam jumlah besar (Li & Mu, 2024).

Penelitian terkait tentang klasifikasi dan deteksi diabetes telah berkembang dengan berbagai pendekatan dan algoritma. Witjaksana et al. (2021) membandingkan akurasi klasifikasi diabetes menggunakan algoritma Random Forest dan Artificial Neural Network (ANN). Hasil menunjukkan bahwa Random Forest memiliki akurasi sebesar 90,62%, sedangkan ANN memiliki akurasi sebesar 82,29%. Hal ini menegaskan bahwa Random Forest adalah algoritma yang lebih efektif dalam klasifikasi diabetes dibandingkan ANN.

Penelitian oleh Daghistani & Alshammari (2020) membandingkan prediksi diabetes menggunakan Logistic Regression dan Random Forest. Hasil penelitian menunjukkan bahwa model Random Forest memiliki presisi 0,883, *recall* 0,88, dan *F-Measure* 0,876, sedangkan Logistic Regression memiliki presisi 0,692, *recall* 0,703, dan *F-Measure* 0,675. AUC untuk Random Forest adalah 0,944, dibandingkan Logistic Regression yang memiliki AUC sebesar 0,708. Penelitian ini menegaskan bahwa Random Forest memiliki kinerja prediksi yang lebih unggul dibandingkan Logistic Regression dalam mendeteksi diabetes.

Sistematis *review* yang dilakukan oleh Tulu et al. (2023) lebih lanjut meneliti keunggulan SMOTE dalam berbagai aplikasi pembelajaran mesin. Seperti halnya klasifikasi dan prediksi kesehatan, dengan menekankan bahwa teknik ini secara konsisten meningkatkan kinerja model dalam menghadapi *dataset* yang tidak seimbang.

Penelitian ini mengimplementasikan algoritma Random Forest dengan Synthetic Minority Oversampling Technique (SMOTE) untuk mendeteksi diabetes. SMOTE digunakan untuk menyeimbangkan data dengan cara *resampling*, sehingga model dapat mengidentifikasi pasien yang berisiko tinggi secara lebih akurat. Penelitian ini berupaya untuk menyatukan kedua aspek tersebut, memberikan pendekatan holistik yang dapat diterapkan dalam praktek medis untuk meningkatkan deteksi dini dan penanganan diabetes. Dengan diagnosis dini yang lebih tepat, intervensi awal seperti pemantauan khusus dan penyesuaian pola hidup dapat dilakukan untuk mencegah perkembangan diabetes dan komplikasi yang lebih serius.

2. METODE PENELITIAN

2.1 Dataset Penelitian

Data pada penelitian ini merupakan Diabetes Dataset yang berasal *Birla Institute of Technology* (Tigga & Garg, 2020). *Dataset* ini kumpulan data yang dikumpulkan oleh Tigga & Garg (2020) dari Jurusan Ilmu dan Teknik Komputer. *Dataset* pada Tabel 1 berjumlah 952 data yang terdiri dari 17 fitur dan 1 kelas target. Fitur '*Diabetic*' merupakan kelas target yang menunjukkan bahwa itu termasuk diabetes atau bukan diabetes.



Tabel 1 Fitur Diabetes Dataset

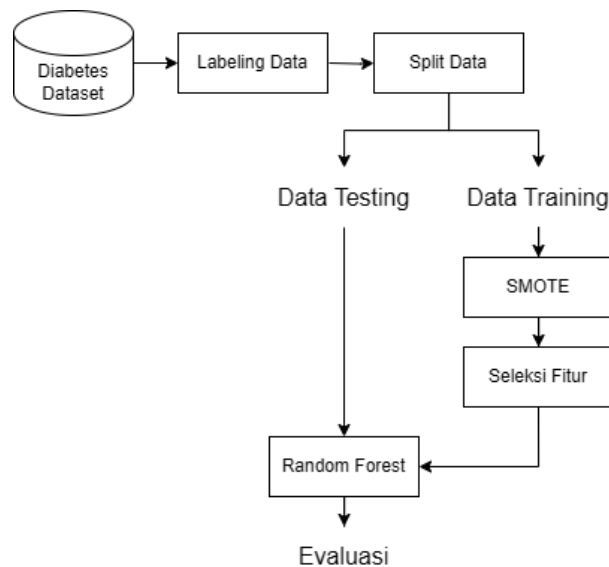
No.	Fitur	Deskripsi
1	<i>Age</i>	Usia pasien
2	<i>Gender</i>	Jenis kelamin pasien
3	<i>Family Diabetes</i>	Riwayat keluarga dengan penyakit diabetes
4	<i>HighBP</i>	Tekanan darah tinggi pasien
5	<i>PhysicallyActive</i>	Aktivitas sehari-hari / aktivitas bergerak
6	<i>BMI</i>	Indeks Massa Tubuh
7	<i>Smoking</i>	Pasien termasuk dari perokok atau tidak
8	<i>Alcohol</i>	Mengonsumsi alkohol atau tidak
9	<i>Sleep</i>	Jam tidur pasien
10	<i>SoundSleep</i>	Jam tidur nyenyak
11	<i>RegularMedicine</i>	Asupan obat secara teratur
12	<i>JunkFood</i>	Mengonsumsi junk food atau tidak
13	<i>Strees</i>	Berapa tingkat mengalami stres
14	<i>BPLevel</i>	Tingkat tekanan darah tinggi
15	<i>Pregnancies</i>	Kehamilan
16	<i>Pdiabetes</i>	Kadar gula darah pada wanita hamil
17	<i>UrinationFreq</i>	Frekuensi Urine
18	<i>Diabetic</i>	Merupakan diabetes atau tidak (kelas target)

2.2 Desain Sistem

Pada penelitian ini terdapat desain sistem yang merupakan tahapan-tahapan yang dilakukan mulai dari *input* hingga menghasilkan hasil evaluasi berupa akurasi, *recall*, presisi, dan *F1-score*. Desain sistem pada Gambar 1 menunjukkan tahap-tahap pembangunan model yang dijabarkan sebagai berikut:

- Input diabetes dataset*.
- Pengubahan data dengan cara *ordinal encoding* untuk merubah fitur kategorik menjadi numerik guna untuk mempermudah tahap pemodelan. Tahap ini dilakukan proses pengubahan data untuk memastikan *dataset* siap untuk digunakan dalam evaluasi model. Dataset yang memiliki fitur kategorik perlu dikonversikan menjadi numerik supaya dapat diolah dengan mudah oleh model.
- Pembagian data menjadi tiga pengujian skenario. Skenario 1 membagi data menjadi 90:10, skenario 2 data dibagi menjadi 70:30 dan skenario 3 adalah 50:50. Pembagian 3 skenario digunakan untuk mengevaluasi model pada kondisi yang berbeda-beda. Skenario 1 melatih model dengan data latih yang jauh lebih besar daripada data uji untuk melihat kemampuan generalisasi meskipun data uji yang terbatas. Skenario 2 yang umum digunakan pada pembagian data latih dan data uji sehingga data ini dipakai untuk perbandingan dengan skenario yang lainnya. Pada skenario 3, mengevaluasi model dengan data latih dan data uji yang seimbang, apakah hal ini mempengaruhi performa model atau tidak. Sehingga dengan ketiga skenario tersebut perlu untuk dilakukan supaya dapat mengetahui mana skenario yang menghasilkan evaluasi model terbaik.
- SMOTE untuk menyeimbangkan *dataset* yang telah melalui tahap split data dengan menggunakan data train saja.
- Seleksi fitur dilakukan untuk memilih fitur yang paling relevan dalam pemodelan untuk meningkatkan performa model (Mostafa et al., 2024). Pada penelitian ini teknik seleksi fitur yang digunakan adalah *Feature Importance* dari algoritma Random Forest *classifier*. Teknik ini dapat mengevaluasi masing-masing fitur terhadap hasil prediksi.
- Kemudian implementasi algoritma Random Forest pada penelitian ini untuk membangun model yang menghasilkan nilai performa model dari *confusion matrix*. Evaluasi model yang paling baik dilihat pada hasil akurasi, presisi, *recall* dan *F1-score* dari masing-masing skenario dan pengujian dengan menggunakan SMOTE dan tanpa SMOTE.





Gambar 1 Desain Penelitian

2.3 Random Forest

Random Forest merupakan algoritma pembelajaran mesin berbasis *ensemble* yang terdiri dari sejumlah pohon keputusan atau yang biasa disebut dengan *decision trees* yang bekerja bersama untuk meningkatkan performa model (Zailani & Hanun, 2020). Algoritma ini terkenal karena kemampuannya dalam menangani *dataset* besar dan kompleks, serta mengurangi risiko *overfitting* yang umum pada pohon keputusan tunggal (Junus et al., 2023). Beberapa kelebihan dari Random Forest selain mampu mengatasi *overfitting* juga toleransi pada data tidak eimbang dan mampu menangani data yang *missing* (Rajaraman & Ullman, 2011). Langkah-langkah pembuatan Random Forest sebagai berikut:

- 1) Penentuan jumlah pohon yang akan dibuat.
- 2) Membangun *tree* dari setiap sampel data dengan banyaknya sesuai dengan yang ditentukan.
- 3) Membuat *tree* dari kumpulan pohon-pohon. Kemudian mengulangi langkah 1 dan 2 hingga mencapai jumlah jumlah pohon yang telah ditentukan sebelumnya.
- 4) Hasil akhir yaitu dengan *majority vote* dari hasil semua pohon keputusan.

2.4 Synthetic Minority Oversampling Technique (SMOTE)

SMOTE adalah salah satu teknik *oversampling* yang digunakan untuk menambah data minoritas pada dataset yang tidak seimbang (Yang et al., 2024). SMOTE bekerja untuk menghasilkan sampel sintesis baru pada data minoritas (Mulia & Kurniasih, 2023). Data sintesis dibuat berdasarkan prinsip kerja K-Nearest Neighbor (tetangga terdekat). Langkah SMOTE dimulai dengan identifikasi data minoritas dan menentukan k tetangga terdekat dengan persamaan jarak Euclidean pada Pers. (1). Di mana pada persamaan tersebut $d(x, y)$ merupakan jarak Euclidean, x_i dan y_i adalah koordinat x dan y pada dimensi ke- i , dan n menunjukkan jumlah dimensi Euclidean.

$$d(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (1)$$

Selanjutnya, SMOTE melakukan pemilihan acak dari k tetangga dan menghitung perbedaan antara fitur minoritas asli dan fitur tetangga yang dipilih. mengalikan perbedaan tersebut dengan bilangan acak, dan menambahkannya ke sampel asli untuk menghasilkan sampel sintesis.



Sampel sintesis ini kemudian ditambahkan ke *dataset*, meningkatkan jumlah sampel dalam kelas minoritas dan membantu model pembelajaran mesin belajar dari data yang lebih seimbang.

3. HASIL DAN PEMBAHASAN

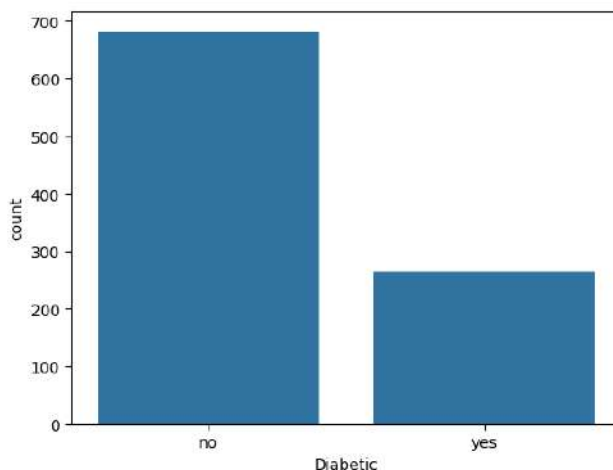
Penelitian ini memiliki beberapa tahap yang akan dibahas antara lain, pengubahan data, SMOTE, seleksi fitur, pemodelan serta evaluasi model. Setiap tahapan memiliki peran penting dalam membangun model klasifikasi yang optimal. Sebelum ke tahapan tersebut, berikut pada Tabel 9 di Lampiran A merupakan lima contoh dari diabetes *dataset*.

3.1 Pengubahan Data

Dataset pada Tabel 10 di Lampiran A selanjutnya dilakukan tahap pengubahan data. Teknik yang dilakukan pada tahap ini adalah *ordinal encoder* di mana merubah ketegori menjadi numerik yang memiliki tingkatan seperti pada fitur 'Age'. Dapat dilihat hasil setelah pengubahan dari *dataset*, data yang memiliki tingkatan akan diurutkan mulai dari 0. 'Age' memiliki 4 tingkatan yaitu usia 40-49, 50-59, usia lebih dari 60 dan kurang dari 40. Dengan teknik *ordinal*, maka tingkatan usia tersebut dirubah menjadi numerik yaitu, mulai dari 0, 1, 2 dan 3.

3.2 SMOTE

Diabetes *dataset* memiliki data yang tidak seimbang pada kelas target yang merupakan diabetes dan bukan diabetes. Sehingga dilakukan teknik *resampling* yaitu SMOTE untuk menyeimbangkan *dataset* tersebut. Gambar 2 jika dilihat antara kelas bukan diabetes dan diabetes memiliki jumlah yang sangat berbeda. Kelas target bukan diabetes ada sekitar 700 data dan kelas target diabetes sekitar 250 data. Pada tahap ini dilakukan setelah split data, dan data yang digunakan pada tahap SMOTE hanya pada data *training* saja supaya tidak terjadi kebocoran data yang mempengaruhi hasil pada performa model.



Gambar 2 Persebaran Data pada Kelas Target

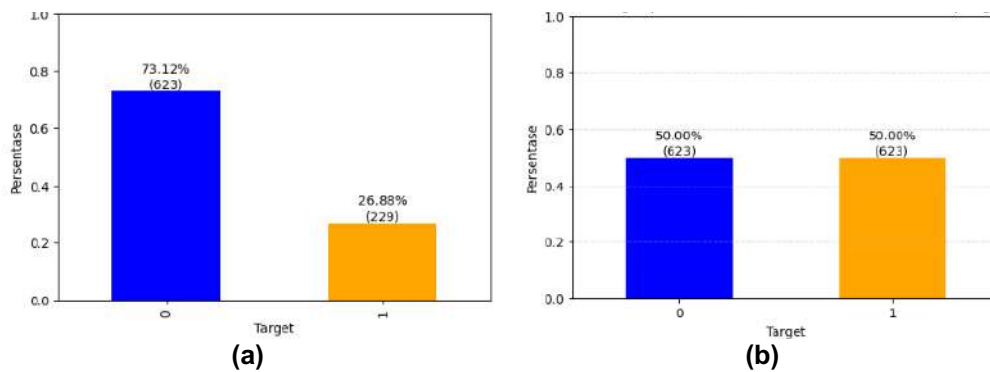
Pada skenario 1 data dibagi menjadi 90% data *train* dan 10% data *test*. Data *train* yang dihasilkan pada kelas bukan diabetes adalah 623 data sedangkan data diabetes sebanyak 229 data pada Gambar 3(a). Maka, penyeimbangan data dilakukan yang kemudian menghasilkan jumlah yang sama antara bukan diabetes dan diabetes. Setelah dilakukan SMOTE, data pada kelas diabetes yang awalnya jauh berbeda dengan data bukan diabetes menjadi sama yaitu dengan jumlah 623 pada data diabetes yang bisa dilihat di Gambar 3(b).

Pada skenario 2 data dibagi menjadi 70% data *train* dan 30% data *test*. Data *train* yang dihasilkan pada kelas bukan diabetes yaitu 483 data dan kelas diabetes sebanyak 179 seperti pada Gambar 4(a). Kemudian menghasilkan jumlah yang sama antara bukan diabetes dan diabetes setelah

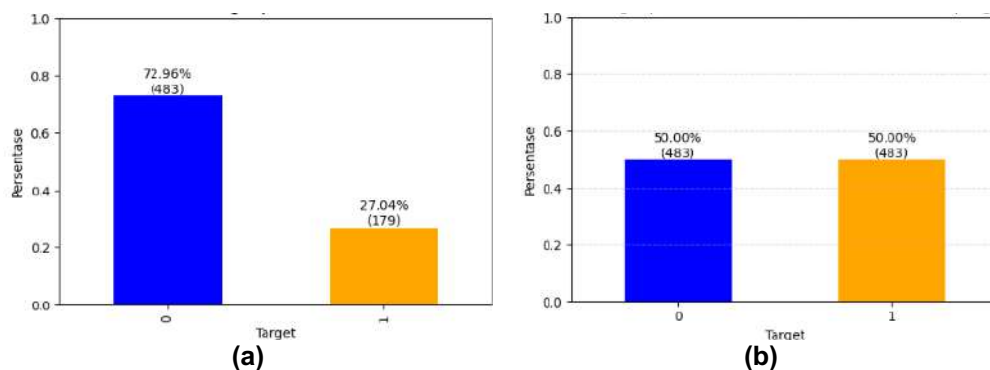


dilakukan SMOTE. Data pada kelas diabetes menjadi sama dengan data bukan diabetes yaitu menjadi berjumlah 483 pada data diabetes yang bisa dilihat di Gambar 4(b).

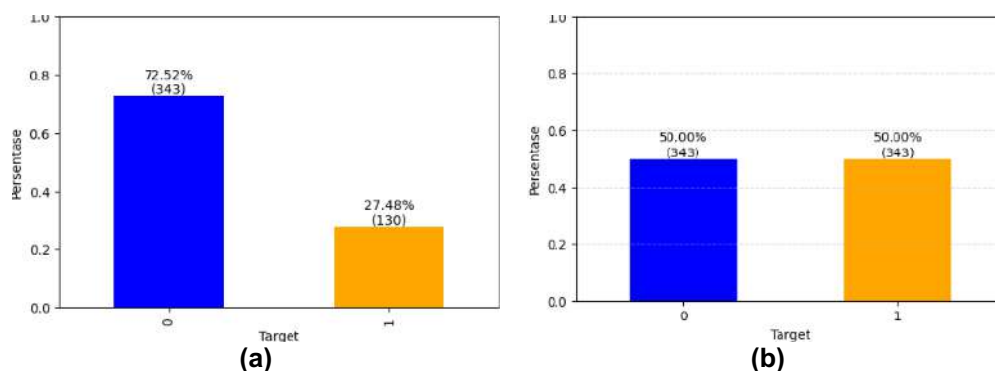
Pada skenario 3 data dibagi menjadi 50% data *train* dan 50% data *test*. Data *train* yang dihasilkan pada kelas bukan diabetes sebanyak 343 data dan pada kelas diabetes sebanyak 130 data seperti pada Gambar 5(a). Setelah itu, melakukan penyeimbangan data yang menghasilkan jumlah yang sama antara kelas bukan diabetes dan kelas diabetes. Setelah SMOTE, kelas diabetes menjadi 343 yang bisa dilihat di Gambar 5(b).



Gambar 3 Sebelum (a) dan Setelah (b) SMOTE 90:10



Gambar 4 Sebelum (a) dan Setelah (b) SMOTE 70:30



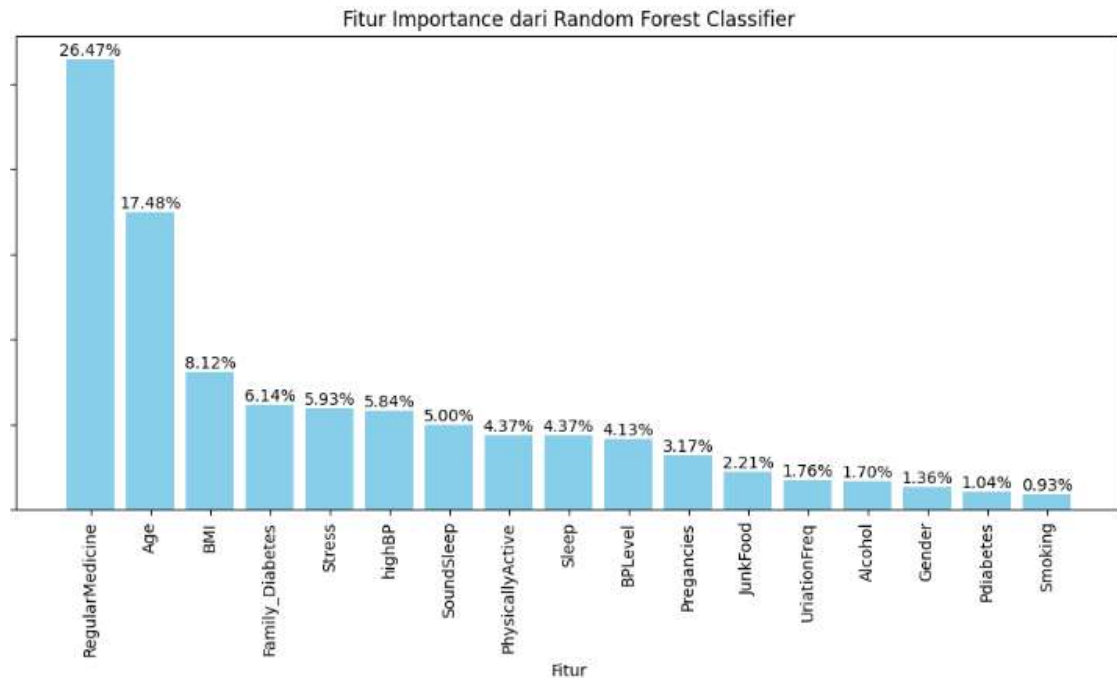
Gambar 5 Sebelum (a) dan Setelah (b) SMOTE 50:50

3.3 Seleksi Fitur

Seleksi fitur dilakukan untuk memilih fitur yang relevan pada *dataset* untuk dilakukan pemodelan. *Dataset* yang digunakan oleh peneliti memiliki fitur yang berjumlah 17. Dari seluruh fitur tersebut dilakukan seleksi untuk mengambil fitur yang paling relevan yaitu 10 dari 17 fitur yang telah



diseleksi. Hasil dari 17 fitur pada *dataset*, dilakukan seleksi dengan teknik *Feature Importance* dari Random Forest. Hasil dari seleksi menunjukkan ada 10 fitur yang paling relevan antara lain, *regular medicine*, *age*, BMI, *family diabetes*, *stress*, *highbp*, *soundsleep*, *physically active*, *sleep*, dan *bplevel* seperti yang ditunjukkan pada Gambar 6. Dari Tabel 2, fitur *regular medicine*, *age*, dan BMI memiliki pengaruh terbesar terhadap hasil model, sedangkan fitur lain tetap relevan dengan bobot lebih kecil. Fitur-fitur yang memiliki hasil *importance score* kurang dari 4,00% diabaikan untuk meningkatkan efisiensi model.



Gambar 6 Grafik Hasil Seleksi Fitur

Tabel 2 Hasil Seleksi Fitur

No.	Fitur	Importance Score
1	Regular Medicine	26,47%
2	Age	17,48%
3	BMI	8,12%
4	Family_Diabetes	6,14%
5	Stress	5,93%
6	HighBP	5,84%
7	SoundSleep	5,00%
8	Physically Active	4,37%
9	Sleep	4,37%
10	BPLLevel	4,13%

3.4 Evaluasi Random Forest dengan SMOTE

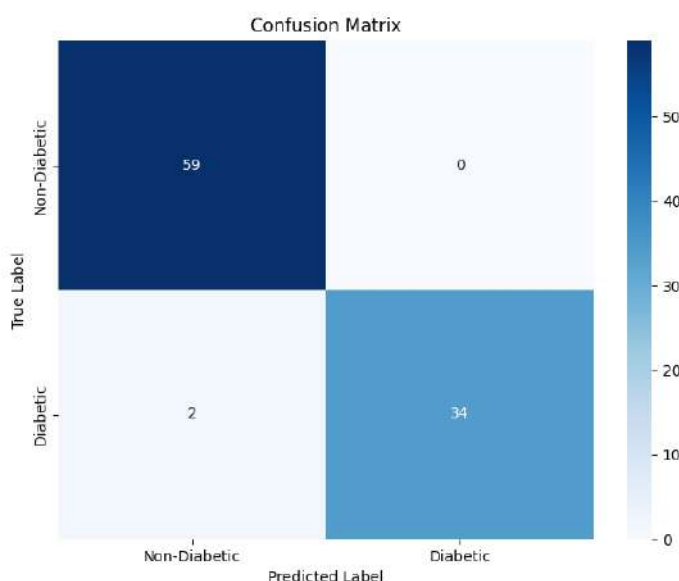
Terdapat tiga skenario yang dilakukan pada penelitian ini, yaitu membagi data latih dan data uji menjadi rasio antara lain 90:10, 70:30, 50:50. Setiap skenario digunakan untuk menguji kestabilan dan performa model dalam melakukan klasifikasi. Hasil evaluasi model dengan algoritma Random Forest sebagai berikut. Setelah dilakukan pembagian data pada skenario 1 dengan rasio 90:10, selanjutnya melakukan penyeimbangan data pada data *training*. Kemudian dilakukan seleksi fitur dengan menggunakan 10 fitur terbaik. Setelah itu dilakukan *tuning* parameter untuk mendapatkan *hyperparameter* terbaik, yaitu *max_depth* sebesar 50 dan *n_estimators* sebanyak 10. Skenario 1 dengan tahap SMOTE ini, menghasilkan *confusion matrix*



pada Tabel 3 dengan visualisasi grafik pada Gambar 7. Pada tabel dan gambar tersebut memperlihatkan bahwa ada 34 yang benar diprediksi diabetes dan 59 yang diprediksi dengan benar bukan diabetes. Ada 2 yang merupakan diabetes tetapi diprediksi bukan diabetes. Dari *confusion matrix* tersebut menghasilkan evaluasi model berupa akurasi sebesar 97%, presisi 100%, *recall* 94%, dan *F1-score* sebesar 97%.

Tabel 3 Confusion Matrix Skenario 1 dengan SMOTE

Prediksi	Aktual	
	Diabetes	Bukan Diabetes
Diabetes	34	0
Bukan Diabetes	2	59



Gambar 7 Grafik Confusion Matrix Skenario 1 dengan SMOTE

Setelah melakukan pembagian data (*split*) dan penyeimbangan data pada data *training* menggunakan skenario 2 dengan rasio 70:30, kemudian dilakukan seleksi fitur dengan menggunakan 10 fitur terbaik. Selanjutnya dilakukan *tuning* parameter untuk mendapatkan *hyperparameter* terbaik, yaitu *max_depth* sebesar 100 dan *n_estimators* sebanyak 20. Skenario 2 dengan tahap SMOTE menghasilkan *confusion matrix* pada Tabel 4 dengan visualisasi *confusion matrix* pada Gambar 8. Hasil tersebut memperlihatkan bahwa ada 80 yang diprediksi secara benar diabetes dan 194 bukan diabetes. Ada 6 yang diabetes tetapi diprediksi bukan diabetes dan 5 bukan diabetes diprediksi sebagai diabetes. Dari hasil *confusion matrix*, evaluasi model menghasilkan akurasi sebesar 96%, presisi 94%, *recall* 93%, dan *F1-score* sebesar 94%.

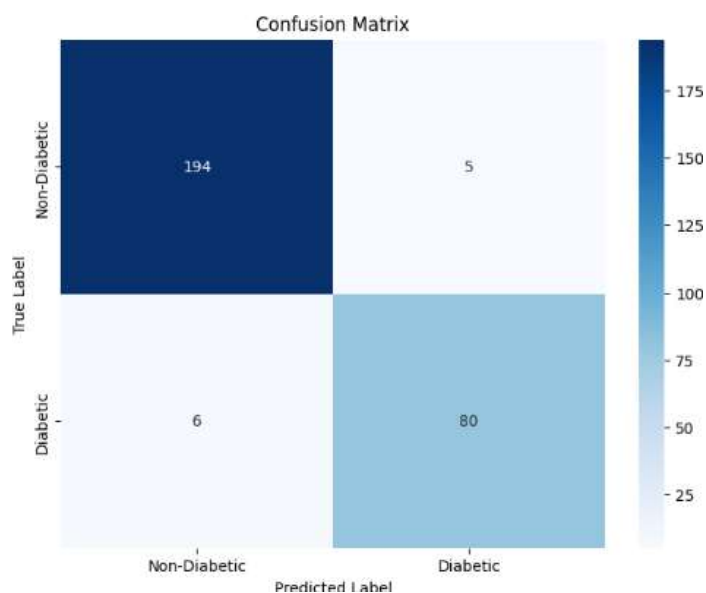
Setelah dilakukan pembagian data (*split*) pada skenario 3 dengan rasio 50:50, selanjutnya menerapkan SMOTE pada data *training*. Kemudian dilakukan seleksi fitur dengan menggunakan 10 fitur terbaik untuk pemodelan. Selanjutnya melakukan *tuning* parameter untuk mendapatkan *hyperparameter* terbaik, yaitu *max_depth* sebesar 50 dan *n_estimators* sebanyak 15. Skenario 3 yang menggunakan SMOTE, hasil evaluasi ditampilkan dalam *confusion matrix* pada Tabel 5.

Gambar 9 merupakan hasil *confusion matrix* skenario 3 dengan SMOTE yang menunjukkan ada 121 data yang diprediksi secara benar diabetes dan 327 bukan diabetes. Ada 14 data diabetes tetapi diprediksi bukan diabetes dan 12 data bukan diabetes diprediksi diabetes. Dari data tersebut, evaluasi model menghasilkan akurasi sebesar 95%, presisi 91%, *recall* 90%, dan *F1-score* sebesar 90%.

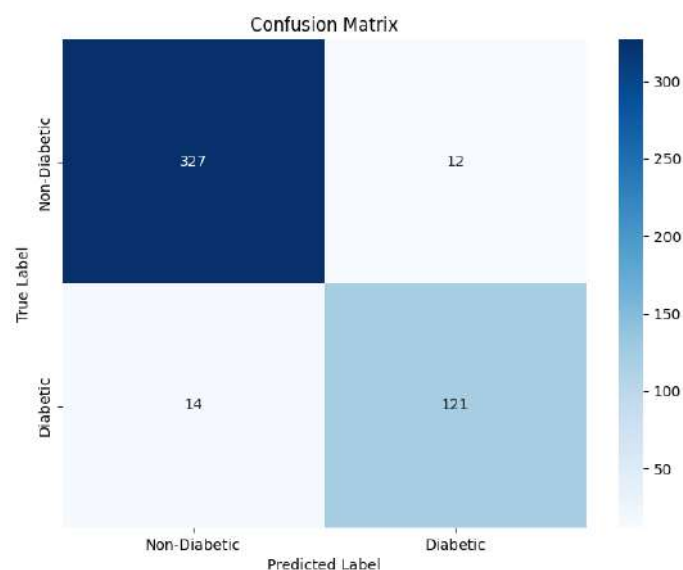


Tabel 4 *Confusion Matrix* Skenario 2 dengan SMOTE

Prediksi	Aktual	
	Diabetes	Bukan Diabetes
Diabetes	80	5
Bukan Diabetes	6	194

Gambar 8 Grafik *Confusion Matrix* Skenario 2 dengan SMOTETabel 5 *Confusion Matrix* Skenario 3 dengan SMOTE

Prediksi	Aktual	
	Diabetes	Bukan Diabetes
Diabetes	121	12
Bukan Diabetes	14	327

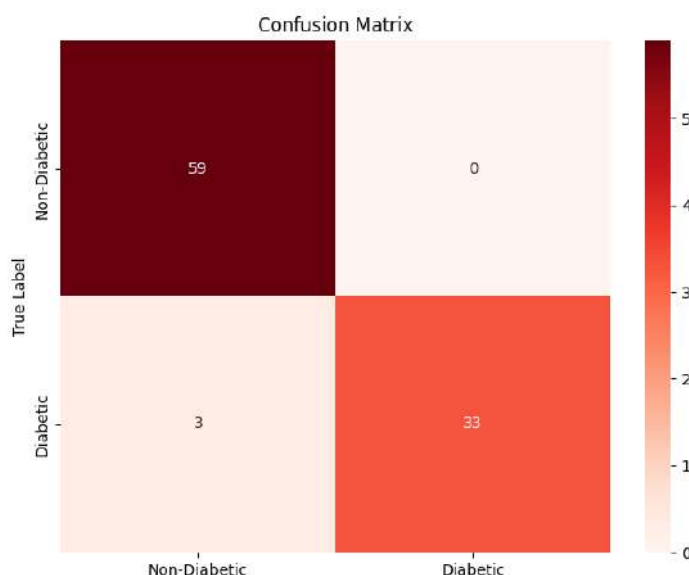
Gambar 9 Grafik *Confusion Matrix* Skenario 3 dengan SMOTE

3.5 Evaluasi Random Forest Tanpa SMOTE

Setelah dilakukan pembagian data pada skenario 1 dengan rasio 90:10, selanjutnya melakukan penyeimbangan data pada data training. Kemudian dilakukan seleksi fitur dengan menggunakan 10 fitur terbaik untuk pemodelan. Selanjutnya dilakukan *tuning* parameter untuk menemukan *hyperparameter* terbaik, yaitu *max_depth* sebesar 50 dan *n_estimators* sebanyak 10. Skenario 1 tanpa SMOTE ini menghasilkan *confusion matrix* pada Tabel 6 dengan visualisasi pada Gambar 10. Hasil *confusion matrix* tersebut menunjukkan bahwa ada 33 yang benar diprediksi diabetes dan 59 yang diprediksi dengan benar bukan diabetes. Ada 3 yang merupakan diabetes tetapi diprediksi bukan diabetes. Maka skenario ini menghasilkan evaluasi model berupa akurasi sebesar 96%, presisi 100%, *recall* 92%, dan *F1-score* sebesar 96%.

Tabel 6 Confusion Matrix Skenario 1 Tanpa SMOTE

Prediksi	Aktual	
	Diabetes	Bukan Diabetes
Diabetes	33	0
Bukan Diabetes	3	59



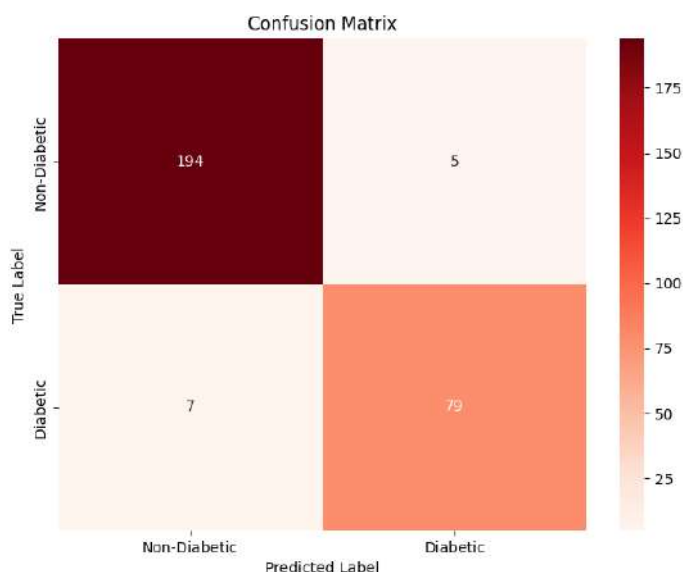
Gambar 10 Grafik Confusion Matrix Skenario 1 Tanpa SMOTE

Setelah dilakukan pembagian data pada skenario 2 dengan rasio 70:30, selanjutnya melakukan penyeimbangan data pada data *training*. Kemudian dilakukan seleksi fitur dengan menggunakan 10 fitur terbaik untuk pemodelan. Selanjutnya dilakukan *tuning* parameter untuk mendapatkan *hyperparameter* terbaik, yaitu *max_depth* sebesar 50 dan *n_estimators* sebanyak 10. Skenario 2 tanpa tahap SMOTE menghasilkan *confusion matrix* pada Tabel 7 dengan grafik pada Gambar 11. Hasil *confusion matrix* tersebut memperlihatkan bahwa ada 79 yang diprediksi secara benar diabetes dan 194 bukan diabetes. Ada 7 yang diabetes tetapi diprediksi bukan diabetes dan 5 bukan diabetes diprediksi sebagai diabetes. Dari hasil tersebut maka evaluasi model menghasilkan akurasi sebesar 96%, presisi 94%, *recall* 92%, dan *F1-score* sebesar 93%.

Tabel 7 Confusion Matrix Skenario 2 Tanpa SMOTE

Prediksi	Aktual	
	Diabetes	Bukan Diabetes
Diabetes	79	5
Bukan Diabetes	7	194

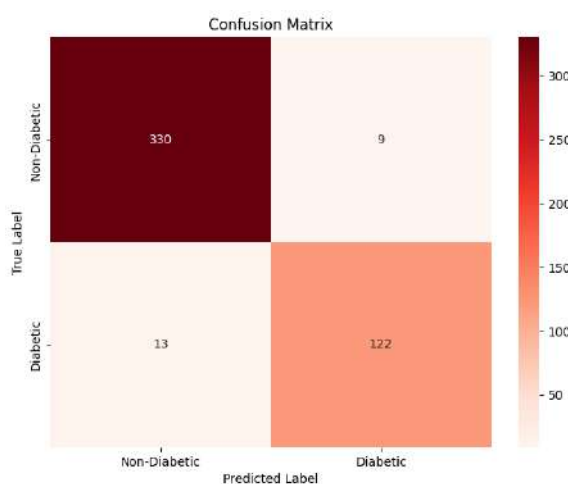


Gambar 11 Grafik *Confusion Matrix* Skenario 2 Tanpa SMOTE

Setelah dilakukan pembagian data pada skenario 1 dengan rasio 50:50, lalu dilakukan penerapan SMOTE pada data *training*. Kemudian dilakukan seleksi fitur dengan menggunakan 10 fitur terbaik. Selanjutnya melakukan tuning parameter untuk mendapatkan *hyperparameter* terbaik, yaitu *max_depth* sebesar 50 dan *n_estimators* sebanyak 10. Skenario 3 tanpa SMOTE menghasilkan *confusion matrix* pada Tabel 8 dengan visualisasi pada Gambar 12.

Tabel 8 *Confusion Matrix* Skenario 3 Tanpa SMOTE

Prediksi	Aktual	
	Diabetes	Bukan Diabetes
Diabetes	122	9
Bukan Diabetes	13	330

Gambar 12 Grafik *Confusion Matrix* Skenario 3 Tanpa SMOTE

Hasil *confusion matrix* skenario 3 tanpa SMOTE yang menunjukkan ada 122 data yang diprediksi secara benar diabetes dan 330 bukan diabetes. Ada 13 data diabetes tetapi diprediksi bukan diabetes dan 9 data bukan diabetes diprediksi diabetes. Dari data tersebut, evaluasi model menghasilkan akurasi sebesar 95%, presisi 93%, *recall* 90%, dan *F1-score* sebesar 92%.



Dari hasil evaluasi model dengan Random Forest yang telah terbagi dalam tiga skenario, hasil paling baik yaitu pada skenario 1 yang merupakan membagi rasio data 90% data latih dan 10% data uji. Skenario 1 memiliki data latih yang lebih besar daripada skenario 2 dan 3. Maka pembagian rasio data berpengaruh pada bagaimana model dilatih. Pembagian data latih yang lebih banyak dari data uji cenderung menghasilkan performa model yang lebih baik. Data latih digunakan untuk melatih model. Model yang memiliki informasi pada data latih yang lebih banyak menyebabkan model mampu mempelajari pola data dengan lebih kompleks (Barus, 2022). Dengan demikian, model dapat generalisasi pada data baru dan menghasilkan model yang lebih akurat.

Evaluasi dengan SMOTE juga menghasilkan hasil yang lebih baik daripada tanpa SMOTE. Penggunaan SMOTE dapat meningkatkan performa model pada data yang tidak seimbang dengan menyeimbangkan jumlah sampel antara kelas mayoritas dan minoritas. Tanpa SMOTE, model cenderung hanya memperhatikan pada kelas mayoritas saja dan mengabaikan kelas minoritas (Sutoyo et al., 2020). Dengan SMOTE, lebih banyak contoh dari kelas minoritas yang dihasilkan secara sintesis, serta membantu model untuk lebih baik dalam mengenali dan memprediksi kedua kelas target sehingga dapat meningkatkan performa model.

4. KESIMPULAN

Penelitian ini berhasil mengembangkan metode deteksi dini diabetes menggunakan algoritma Random Forest yang dikombinasikan dengan teknik SMOTE untuk menyeimbangkan data. Hasil menunjukkan bahwa kualitas model prediksi sangat dipengaruhi oleh proporsi data latih dan keseimbangan distribusi kelas target. Skenario terbaik adalah skenario 1 dengan pembagian data 90% untuk latih dan 10% untuk uji, yang menghasilkan kinerja tertinggi dengan akurasi 97%, presisi 100%, *recall* 94%, dan F1-score 97%. Ini membuktikan bahwa data latih yang lebih banyak dan seimbang memungkinkan model belajar lebih efektif dalam mengenali tanda-tanda awal diabetes.

Penelitian ini juga terdapat pemilihan fitur menggunakan *Feature Importance* dari *Random Forest* dan penerapan SMOTE dalam berbagai skenario pembagian data. Hal ini memberikan dasar penting bagi pengembangan sistem deteksi dini penyakit diabetes, khususnya untuk data penyakit dengan ketidakseimbangan data yang tinggi seperti diabetes. Penelitian selanjutnya bisa menggunakan model lain, seperti *cross-validation* dan menambahkan data klinis untuk meningkatkan kemampuan model pada kasus nyata.

DAFTAR PUSTAKA

- Aris, F., & Benyamin, B. (2019). Penerapan Data Mining untuk Identifikasi Penyakit Diabetes Melitus dengan Menggunakan Metode Klasifikasi. *Router Research*, 1(1), 1–6. <https://doi.org/10.29239/j.router.2019.313>
- Daghistani, T., & Alshammari, R. (2020). Comparison of Statistical Logistic Regression and RandomForest Machine Learning Techniques in Predicting Diabetes. *Journal of Advances in Information Technology*, 11(2), 78–83. <https://doi.org/10.12720/jait.11.2.78-83>
- Elfaladonna, F., & Rahmadani, A. (2019). Analisa Metode Classification-Decision Tree dan Algoritma C.45 untuk Memprediksi Penyakit Diabetes dengan Menggunakan Aplikasi Rapid Miner. *SINTECH (Science and Information Technology) Journal*, 2(1), 10–17. <https://doi.org/10.31598/sintechjournal.v2i1.293>
- Faida, A. N., & Santik, Y. D. P. (2020). Kejadian Diabetes Melitus Tipe I pada Usia 10-30 Tahun. *Higeia Journal of Public Health Research and Development*, 4(1), 33–42. <https://doi.org/10.15294/higeia/v4i1/31763>
- Hana, F. M. (2020). Klasifikasi Penderita Penyakit Diabetes Menggunakan Algoritma Decision Tree C4.5. *Jurnal SISKOM-KB (Sistem Komputer dan Kecerdasan Buatan)*, 4(1), 32–39. <https://doi.org/10.47970/siskom-kb.v4i1.173>



- Junus, C. Z. V., Tarno, T., & Kartikasari, P. (2023). Klasifikasi Menggunakan Metode Support Vector Machine dan Random Forest untuk Deteksi Awal Risiko Diabetes Melitus. *Jurnal Gaussian*, 11(3), 386–396. <https://doi.org/10.14710/j.gauss.11.3.386-396>
- Karyadiputra, E., & Setiawan, A. (2022). Penerapan Data Mining untuk Prediksi Awal Kemungkinan Terindikasi Diabetes. *Teknosains: Media Informasi Sains dan Teknologi*, 16(2), 221–232. <https://doi.org/10.24252/teknosains.v16i2.28257>
- Kementerian Kesehatan Republik Indonesia. (2023). *Rencana Aksi Kerja Kegiatan Direktorat P2PTM 2021-2024* (1st ed.). Kementerian Kesehatan Republik Indonesia. <https://www.scribd.com/document/757455987/RAK-Dit-P2PTM-1-465827-02-4tahunan-070>
- Li, Y., & Mu, Y. (2024). Research and Performance Analysis of Random Forest-Based Feature Selection Algorithm in Sports Effectiveness Evaluation. *Scientific Reports*, 14(1), Article ID: 26275. <https://doi.org/10.1038/s41598-024-76706-1>
- Magliano, D., & Boyko, E. J. (2013). Five Questions on the IDF Diabetes Atlas. *Diabetes Research and Clinical Practice*, 102(2), 147–148. <https://doi.org/10.1016/j.diabres.2013.10.013>
- Mostafa, G., Mahmoud, H., Abd El-Hafeez, T., & ElAraby, M. E. (2024). The Power of Deep Learning in Simplifying Feature Selection for Hepatocellular Carcinoma: A Review. *BMC Medical Informatics and Decision Making*, 24(1), Article ID: 287. <https://doi.org/10.1186/s12911-024-02682-1>
- Mulia, C., & Kurniasih, A. (2023). Teknik SMOTE untuk Mengatasi Imbalance Class dalam Klasifikasi Bank Customer Churn Menggunakan Algoritma Naïve bayes dan Logistic Regression. *Prosiding Seminar Ilmiah Nasional Online Mahasiswa Ilmu Komputer dan Aplikasinya*, 4(2), 552–559. <https://conference.upnvj.ac.id/index.php/senamika/article/view/2590>
- Rahman, M. S., Hossain, K. S., Das, S., Kundu, S., Adegoke, E. O., Rahman, Md. A., Hannan, Md. A., Uddin, M. J., & Pang, M.-G. (2021). Role of Insulin in Health and Disease: An Update. *International Journal of Molecular Sciences*, 22(12), Article ID: 6403. <https://doi.org/10.3390/ijms22126403>
- Rajaraman, A., & Ullman, J. D. (2011). Data Mining. In *Mining of Massive Datasets* (Vol. 2, Issue January 2013, pp. 1–17). Cambridge University Press. <https://doi.org/10.1017/CBO9781139058452.002>
- Sun, J., Hu, W., Ye, S., Deng, D., & Chen, M. (2023). The Description and Prediction of Incidence, Prevalence, Mortality, Disability-Adjusted Life Years Cases, and Corresponding Age-Standardized Rates for Global Diabetes. *Journal of Epidemiology and Global Health*, 13(3), 566–576. <https://doi.org/10.1007/s44197-023-00138-9>
- Tigga, N. P., & Garg, S. (2020). Prediction of Type 2 Diabetes Using Machine Learning Classification Methods. *Procedia Computer Science*, 167, 706–716. <https://doi.org/10.1016/j.procs.2020.03.336>
- Tulu, T. W., Wan, T. K., Chan, C. L., Wu, C. H., Woo, P. Y. M., Tseng, C. Z. S., Vodencarevic, A., Menni, C., & Chan, K. H. K. (2023). Machine Learning-Based Prediction of COVID-19 Mortality Using Immunological and Metabolic Biomarkers. *BMC Digital Health*, 1(1), Article ID: 6. <https://doi.org/10.1186/s44247-022-00001-0>
- Witjaksana, E. C. P., Saedudin, Rd. R., & Widartha, V. P. (2021). Perbandingan Akurasi Algoritma Random Forest dan Algoritma Artificial Neural Network untuk Klasifikasi Penyakit Diabetes. *EProceedings of Engineering*, 8(5), 9773–9781. <https://openlibrarypublications.telkomuniversity.ac.id/index.php/engineering/article/view/15758>
- Yang, Y., Khorshidi, H. A., & Aickelin, U. (2024). A Review on Over-Sampling Techniques in Classification of Multi-Class Imbalanced Datasets: Insights for Medical Problems. *Frontiers in Digital Health*, 6, Article ID: 1430245. <https://doi.org/10.3389/fdgth.2024.1430245>
- Zailani, A. U., & Hanun, N. L. (2020). Penerapan Algoritma Klasifikasi Random Forest untuk Penentuan Kelayakan Pemberian Kredit di Koperasi Mitra Sejahtera. *Infotech: Journal of Technology Information*, 6(1), 7–14. <https://doi.org/10.37365/jti.v6i1.61>



LAMPIRAN A

Tabel 9 Dataset

Age	Gender	Family Diabetes	highBP	Physically Active	BMI	Smoking	Alcohol	Sleep	Sound Sleep	Regular Medicine	Junkfood	Strees	BPLevel	Pregancie	Pdiabetes	Urination Freq
<40	F	No	No	>0.5 hr	21	No	No	7	7	No	Often	Not at all	Normal	0	0	Not much
<40	F	No	No	≥1hr	20	No	No	6	6	No	Very often	Not at all	Normal	0	0	Not much
50-	M	No	No	≥1hr	27	No	No	7	7	No	Very often	Not at all	Normal	0	0	Not much
59																
40-	M	Yes	No	None	29	No	No	6	6	No	Occasionally	Not at all	Normal	0	0	Not much
49																
≥60	F	No	No	None	18	No	No	6	6	Yes	Occasionally	Sometimes	Normal	3	0	Quite often

Tabel 10 Hasil Setelah Pengubahan Data

Age	Gender	Family Diabetes	highBP	Physically Active	BMI	Smoking	Alcohol	Sleep	Sound Sleep	Regular Medicine	Junkfood	Strees	BPLevel	Pregancie	Pdiabetes	Urination Freq
3.0	0.0	0.0	0.0	1.0	21.0	0.0	0.0	7	7	0.0	2.0	1.0	4.0	0.0	0.0	0.0
3.0	0.0	0.0	0.0	3.0	20.0	0.0	0.0	6	6	1.0	3.0	1.0	4.0	0.0	0.0	0.0
1.0	1.0	0.0	0.0	3.0	27.0	0.0	0.0	7	7	0.0	3.0	1.0	4.0	0.0	0.0	0.0
0.0	1.0	1.0	0.0	2.0	29.0	0.0	0.0	6	6	0.0	1.0	1.0	4.0	0.0	0.0	0.0
2.0	0.0	0.0	0.0	2.0	18.0	0.0	0.0	6	6	1.0	1.0	2.0	4.0	3.0	0.0	2.0



Klasifikasi Penyakit pada Tanaman Berdasarkan Citra Daun Menggunakan Metode Convolutional Neural Network

Denis Aji Pangestu ^{(1)*}, Okta Qomaruddin Aziz ⁽²⁾, Cahyo Crysdian ⁽³⁾

Departemen Teknik Informatika, UIN Maulana Malik Ibrahim, Malang, Indonesia

e-mail : denisajipangestu07@gmail.com, okta.qomaruddin@uin-malang.ac.id, cahyo@ti.uin-malang.ac.id.

* Penulis korespondensi.

Artikel ini diajukan 26 Juni 2024, direvisi 1 Maret 2025, diterima 2 Maret 2025, dan dipublikasikan 31 Mei 2025.

Abstract

The agricultural sector is a vital part of the economy, providing food, raw materials, and employment opportunities. In Indonesia, this sector faces significant challenges, such as low interest from younger generations and plant disease issues. Plant disease identification typically requires the expertise of experienced professionals, but this process is time-consuming and costly. This research aims to develop a plant disease classification model using a Convolutional Neural Network (CNN) to assist farmers in identifying diseases in rice, corn, tomato, and potato plants based on leaf images. Testing was conducted with data splitting ratios of 70:30, 80:20, and 90:10, using both single-stage and multi-stage classification methods. The best results were achieved with an 80:20 data ratio using single-stage classification, with an average accuracy of 80%, precision of 80%, recall of 81%, and F1-score of 79%. This study demonstrates that the CNN method is effective in plant disease classification, achieving optimal performance at a 80:20 data ratio and in single-stage classification. It is hoped that this research can help farmers quickly and accurately identify and manage plant diseases, as well as encourage innovation in the agricultural sector. The implementation of CNN in plant disease classification shows great potential in enhancing the efficiency and accuracy of disease detection, ultimately supporting the sustainability and development of the agricultural sector.

Keywords: Classification, Image Classification, Convolutional Neural Network, Plant Disease, Agricultural

Abstrak

Sektor pertanian merupakan bagian penting dalam perekonomian, menyediakan pangan, bahan baku, dan kesempatan kerja. Di Indonesia, sektor ini menghadapi tantangan serius, seperti kurangnya minat generasi muda dan masalah penyakit tanaman. Identifikasi penyakit tanaman biasanya memerlukan ahli yang berpengalaman, namun proses ini memakan waktu dan biaya yang signifikan. Penelitian ini bertujuan untuk mengembangkan model klasifikasi penyakit tanaman menggunakan metode Convolutional Neural Network (CNN) untuk membantu petani dalam mengidentifikasi penyakit pada tanaman padi, jagung, tomat, dan kentang berdasarkan citra daun. Pengujian dilakukan dengan rasio pembagian data 70:30, 80:20, dan 90:10, serta menggunakan metode klasifikasi *single stage* dan *multi stage*. Hasil terbaik dicapai dengan rasio data 80:20 menggunakan *single-stage classification*, dengan rata-rata akurasi 80%, presisi 80%, *recall* 81%, dan *F1-score* 79%. Penelitian ini menunjukkan bahwa metode CNN efektif dalam klasifikasi penyakit tanaman, dengan performa optimal pada rasio data 80:20 dan *single-stage classification*. Diharapkan penelitian ini dapat membantu petani dalam mengenali dan mengatasi penyakit tanaman secara cepat dan tepat, serta mendorong inovasi di sektor pertanian. Implementasi CNN dalam klasifikasi penyakit tanaman menunjukkan potensi besar dalam meningkatkan efisiensi dan akurasi deteksi penyakit, yang pada akhirnya mendukung keberlanjutan dan perkembangan sektor pertanian.

Kata Kunci: Klasifikasi, Klasifikasi Gambar, Convolutional Neural Network, Penyakit Tanaman, Pertanian



1. PENDAHULUAN

Sektor pertanian memegang peranan yang krusial dalam perekonomian, tidak hanya sebagai penyedia kebutuhan pangan dan bahan baku, tetapi juga sebagai sumber lapangan pekerjaan yang signifikan (Chellapandi et al., 2021). Di Indonesia, sektor pertanian menghadapi berbagai tantangan yang kompleks, antara lain menurunnya minat generasi muda untuk ikut berkontribusi dalam bidang pertanian serta adanya ancaman serius yang datang dari penyakit tanaman yang berdampak pada produktivitas (Ngadi et al., 2023). Berbagai macam penyakit tanaman yang disebabkan oleh patogen seperti virus, bakteri, jamur atau nematoda telah menyebabkan penurunan hasil panen yang cukup besar (Suganda & Wulandari, 2018). Sebagai contoh, pada tahun 2020, produksi beras di Indonesia mengalami penurunan sebesar 9 persen akibat serangan penyakit tanaman dan faktor lingkungan lainnya. Dampak ekonomi yang disebabkan penyakit tanaman tidak hanya dirasakan secara mikro oleh petani, namun juga memengaruhi ketahanan pangan nasional dan stabilitas harga komoditas pertanian (Tineges et al., 2020).

Identifikasi penyakit tanaman secara konvensional bergantung pada keahlian ahli atau petani berpengalaman melalui pengamatan visual. Namun, pendekatan ini memiliki beberapa keterbatasan, seperti memerlukan waktu yang lama, biaya konsultasi yang tinggi dan ketergantungan pada ketersediaan tenaga ahli (Oktaviana et al., 2021). Dalam konteks ini, identifikasi otomatis penyakit tanaman menjadi solusi yang menjanjikan. Dengan menggunakan teknik klasifikasi citra, gejala penyakit dapat dideteksi secara dini, bahkan pada tahap awal infeksi (Singh & Misra, 2017). Penerapan metode ekstraksi fitur dan *machine learning* memungkinkan identifikasi penyakit tanaman secara akurat tanpa memerlukan kehadiran ahli secara langsung (Ambarwari et al., 2023).

Dalam beberapa tahun terakhir, penelitian tentang identifikasi penyakit tanaman menggunakan teknologi *machine learning* dan *deep learning* telah berkembang pesat. Salah satu metode yang paling banyak digunakan adalah Convolutional Neural Network (CNN), yang dikenal dengan kemampuannya dalam memproses citra digital secara kompleks (Anggiratih et al., 2021; Rodríguez-Lira et al., 2024; Tugrul et al., 2022; Upadhyay et al., 2025). CNN meniru cara kerja korteks visual manusia dan mampu mempelajari fitur-fitur penting dari citra secara otomatis. Metode ini terdiri dari tiga lapisan utama, yaitu *convolution layer*, *pooling layer*, dan *fully-connected layer*, yang bekerja bersama untuk mengklasifikasikan citra dengan akurasi tinggi (Guo & Wang, 2022).

Pengembangan model CNN untuk klasifikasi penyakit tanaman berdasarkan citra daun pernah dilakukan oleh Anggraeni et al. (2022) terkait klasifikasi penyakit tanaman cabai menggunakan Convolutional Neural Network (CNN) dengan arsitektur terdiri dari 3 lapisan konvolusi, 3 *fully connected layer*, dan satu lapisan *output*. *Dataset* diambil dari Kaggle, terdiri dari tiga kelas: *health*, *leaf curl*, dan *yellowish*. Data dibagi menjadi 80% untuk *training*, 10% untuk validasi, dan 10% untuk pengujian. Dari total 300 data citra, model mencapai akurasi 80% pada *training* dan 60% pada validasi dengan 100 data.

Irfansyah et al. (2021) melakukan penelitian yang mengkaji klasifikasi penyakit akibat hama pada tanaman kopi menggunakan Convolutional Neural Network (CNN) dengan arsitektur AlexNet. Identifikasi dilakukan melalui citra daun menggunakan *dataset* ReCoLe dari Mendeley, yang terdiri dari 300 data terbagi dalam tiga kelas: *health*, *rust*, dan *red spider mite*. Data dibagi dengan rasio 80:20 untuk *training* dan *testing*, dengan akurasi dianggap baik jika >70%. *Training* dengan 260 data latih menghasilkan akurasi 69,44% – 80,56%, sementara pengujian dengan 40 data uji mencapai 81,6%. Hasil ini menunjukkan bahwa AlexNet akurat untuk klasifikasi penyakit tanaman kopi berbasis citra daun.

Penelitian yang dilakukan oleh Sabrina & Maki (2022) membahas klasifikasi penyakit tanaman apel dari citra daun menggunakan Convolutional Neural Network (CNN). *Dataset* diambil dari Kaggle dan disusun dalam direktori *train* (7.700 data) dan *valid* (1.943 data), dengan total 9.643 citra yang terbagi dalam 4 kelas: *apple scab*, *black rot*, *rust*, dan *healthy*. Citra berukuran 256×256



piksel dalam format RGB. Model dapat dijalankan pada *smartphone* dengan Android Lollipop ke atas. Hasil *final test accuracy* dari seluruh proses pelatihan mencapai 97,1%.

Penelitian berikutnya oleh Zalvadila (2023) tentang klasifikasi penyakit bawang merah menggunakan Convolutional Neural Network (CNN) dan Support Vector Machine (SVM) dengan kernel RBF, *linear*, sigmoid, dan *polynomial*. Penelitian ini menerapkan Gray Level Co-occurrence Matrix (GLCM) untuk ekstraksi fitur. *Dataset* terdiri dari 320 citra, terbagi dalam dua kelas: bercak ungu dan moler, masing-masing 160 gambar. Hasil pengujian menunjukkan bahwa CNN dan SVM dengan kernel *linear*, RBF, dan *polynomial* mencapai akurasi, presisi, *recall*, dan *F1-score* 100%. Sementara itu, SVM dengan kernel sigmoid menggunakan GLCM memperoleh akurasi 75%, presisi 75%, *recall* 73%, dan *F1-score* 74%.

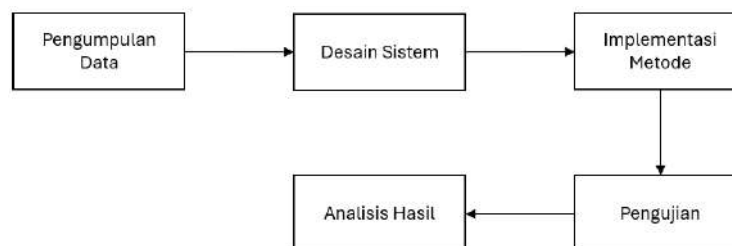
Berdasarkan penelitian sebelumnya, Convolutional Neural Network (CNN) terbukti mampu melakukan klasifikasi citra digital dengan tingkat akurasi tinggi dan kemudahan dalam penggunaannya. Oleh karena itu, penelitian ini memilih CNN sebagai metode utama untuk klasifikasi buah segar dan busuk. Dengan menerapkan metode ini, diharapkan dapat diperoleh hasil klasifikasi yang akurat dan andal dalam mengidentifikasi kondisi buah berdasarkan citra digital.

2. METODE PENELITIAN

Penelitian ini akan menggunakan metode *Convolutional Neural Network* (CNN). *Convolutional Neural Network* (CNN) adalah algoritma Deep Learning yang digunakan untuk pengenalan citra, dengan fitur seperti struktur sederhana, sedikit parameter *training*, dan adaptabilitas. CNN juga dapat digunakan untuk analisis suara dan pengenalan pola (Irfansyah et al., 2021).

CNN biasanya digunakan untuk mengidentifikasi perpindahan, *zoom*, dan distorsi lain dari grafik dua dimensi. Karena lapisan ekstraksi fitur CNN belajar dari data *training*, ekstraksi fitur secara eksplisit dari data *training* dapat dihindari (Hasan et al., 2021). Neuron pada bidang *feature map* yang sama memiliki bobot yang sama, memungkinkan jaringan belajar secara bersamaan, yang merupakan keuntungan utama dibandingkan jaringan saraf yang terhubung satu sama lain, dan mengurangi kompleksitas jaringan (Felix et al., 2020).

Dalam penelitian ini, dijelaskan secara sistematis langkah-langkah yang dilakukan dari awal hingga akhir. Proses ini mencakup pengumpulan data, desain sistem, implementasi metode menggunakan algoritma Convolutional Neural Network, pengujian, dan analisis hasil. Penelitian ini menggunakan bahasa pemrograman Python untuk membangun modelnya. Gambar 1 akan menampilkan urutan tahapan penelitian sesuai dengan yang dijelaskan sebagai berikut.



Gambar 1 Desain Tahap Penelitian













2.1 Pengumpulan Data

Dalam penelitian ini, untuk melakukan klasifikasi penyakit tanaman berdasarkan citra daun, dibutuhkan *dataset* citra daun dari empat jenis tanaman, yaitu padi, jagung, tomat, dan kentang. *Dataset* yang digunakan adalah “*New Plant Disease Dataset*” dan “*Rice Leaf Disease Image Sample*” yang diperoleh dari situs web data.mendeley.com dan kaggle.com. *Dataset* ini mencakup empat spesies tanaman dengan total 12 kelas yang berbeda, terdiri dari 1200 citra



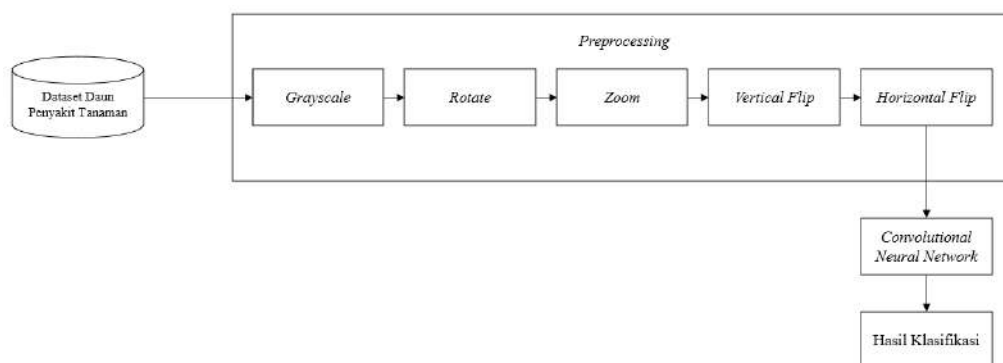
daun dalam format .JPG, dengan masing-masing kelas memiliki 100 citra daun untuk klasifikasi penyakit tanaman. Tabel 1 menunjukkan beberapa contoh data yang digunakan dalam penelitian.

Tabel 1 Contoh Data

No.	Nama Tanaman	Nama Penyakit Tanaman	Citra Daun
1	Jagung	Common Rust	
2	Jagung	Healthy	
3	Jagung	Northern Leaf Blight	
4	Kentang	Early Blight	
5	Kentang	Healthy	
6	Kentang	Late Blight	
7	Padi	Brown Spot	
8	Padi	Healthy	
9	Padi	Scald	
10	Tomat	Bacterial Spot	
11	Tomat	Healthy	
12	Tomat	Mosaic Virus	

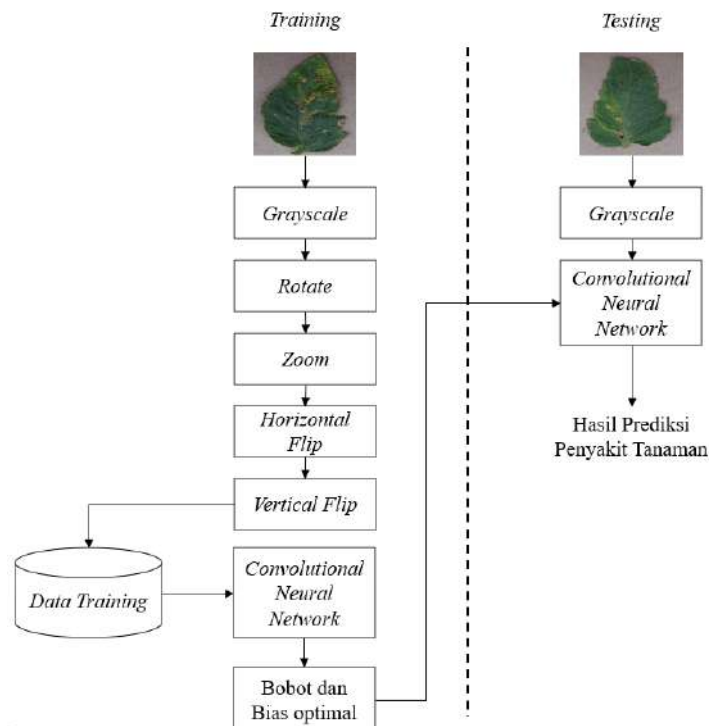
2.2 Desain Sistem

Gambar 2 dan 3 menunjukkan alur proses klasifikasi penyakit tanaman pangan. Proses ini dimulai dengan input data yang dibagi menjadi data pelatihan (*training data*) dan data pengujian (*testing data*). Setiap kelas memiliki 100 citra daun sebagai input. Selanjutnya, data tersebut melalui tahap *preprocessing* yang mencakup pengolahan seperti *rotate*, *zoom*, vertikal *flip*, dan horizontal *flip*. Setelah *preprocessing*, data diproses menggunakan metode Convolutional Neural Network (CNN). Penelitian ini menghasilkan 12 kelas dari proses klasifikasi, yaitu *corn common rust*, *corn healthy*, *corn northern leaf blight*, *potato early blight*, *potato healthy*, *potato late blight*, *rice brown spot*, *rice healthy*, *rice scald*, *tomato bacterial spot*, *tomato healthy*, dan *tomato mosaic virus*. Hasil klasifikasi tersebut kemudian ditampilkan sebagai *output* untuk memberikan informasi tentang penyakit pada tanaman pangan.



Gambar 2 Desain Sistem Klasifikasi Penyakit Tanaman

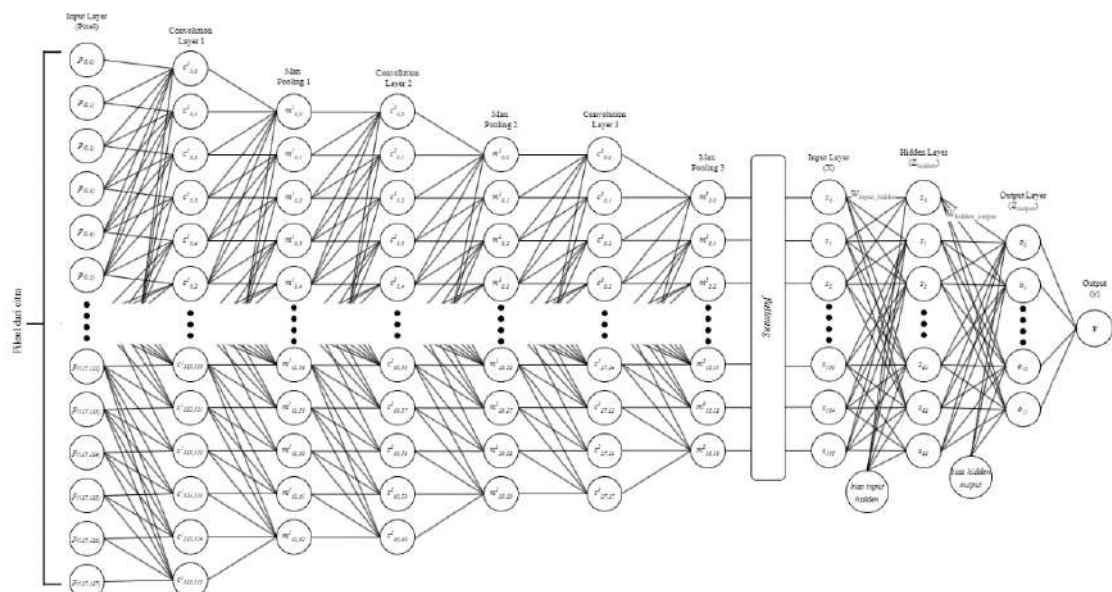




Gambar 3 Alur Proses Pengolahan Data

2.3 Metode Convolutional Neural Network

Dalam penelitian ini, Convolutional Neural Network (CNN) diaplikasikan untuk mengklasifikasikan penyakit tanaman berdasarkan citra daun. Arsitektur Convolutional Neural Network (CNN) yang digunakan adalah LeNet-5 dengan beberapa modifikasi, sebagaimana ditunjukkan pada Gambar 4. Arsitektur ini terdiri dari beberapa lapisan utama, yaitu *input layer*, 3 *convolution layer*, 3 *max pooling layer*, *flatten layer*, *dense layer*, dan *output layer*. Berikut penjelasan aplikatif mengenai bagaimana data gambar hasil *preprocessing* diproses oleh CNN.



Gambar 4 Arsitektur CNN LenNet-5



Citra daun yang telah melalui tahap *preprocessing* (seperti *rotate*, *zoom*, *vertikal flip*, dan *horizontal flip*) diubah menjadi matriks piksel dengan ukuran 128x128x1 (*grayscale*). Matriks ini menjadi input untuk CNN. Citra *grayscale* dipilih untuk mengurangi kompleksitas komputasi karena hanya memiliki satu saluran (*channel*) dibandingkan dengan citra RGB yang memiliki tiga saluran.

Pada tahap *convolution layer*, fitur-fitur penting dari citra daun diekstraksi menggunakan kernel/filter. Dalam penelitian ini digunakan kernel dengan ukuran 3x3 akan melakukan konvolusi pada matriks input untuk menghasilkan *feature map*. Proses ini dilakukan secara berulang pada tiga *convolution layer* yang ada, di mana setiap *layer* akan menghasilkan *feature map* yang lebih abstrak dan representatif. Pada citra *grayscale*, konvolusi dilakukan pada satu saluran (*channel*), sehingga komputasi menjadi lebih efisien. Pers. (1) yang ditunjukkan merupakan perhitungan digunakan dalam perhitungan proses konvolusi. Di mana $c_{(a,b)}$ adalah nilai *feature maps*, $X_{(a,b)}$ adalah nilai *input*, dan $W_{(i,j)}$ adalah nilai *kernel*.

$$c_{(a,b)} = \sum_{i=0}^{k_h-1} \sum_{j=0}^{k_w-1} X_{(a+i,b+j)} W_{(i,j)} \quad (1)$$

Setelah proses konvolusi, *feature map* yang dihasilkan akan direduksi dimensinya menggunakan *max pooling*. Dalam penelitian ini digunakan *max pooling* dengan ukuran 2x2 akan mengambil nilai maksimum dari setiap wilayah 2x2 pada *feature map*, sehingga mengurangi dimensi *feature map* menjadi setengahnya. Hal ini membantu mengurangi kompleksitas komputasi dan mencegah *overfitting*. Berikut adalah Pers. (2) yang digunakan dalam perhitungan ini, di mana $m_{(p,q)}$ merupakan *output feature map* hasil perhitungan, $\max(c_{a \times s_x + p, b \times s_y + q})$ adalah perhitungan untuk mencari nilai tertinggi, serta s_x dan s_y adalah *stride*.

$$m_{(p,q)} = \max(c_{a \times s_x + p, b \times s_y + q}) \quad (2)$$

Feature map yang dihasilkan dari *convolution* dan *pooling layer* berbentuk 2D. Sebelum dilanjutkan ke proses klasifikasi, *feature map* tersebut harus diubah menjadi vektor 1D. Vektor satu dimensi ini kemudian digunakan sebagai input untuk *fully connected layer*. Pada tahap *fully connected layer*, vektor hasil *flattening* akan diproses melalui beberapa *dense layer* untuk melakukan klasifikasi. *Hidden layer* menggunakan fungsi aktivasi ReLU untuk memperkenalkan non-linearitas, sedangkan *output layer* menggunakan fungsi aktivasi *softmax* untuk menghasilkan probabilitas dari setiap kelas penyakit tanaman. Tujuan utama *fully connected layer* adalah mengklasifikasikan data dengan tepat.

Pers. (3) sampai (5) menunjukkan proses perhitungan yang terjadi dalam *feed forward*. Di mana z_s adalah node pada *hidden layer*, x_d adalah nilai *input*, w_r adalah bobot, b_n adalah nilai bias, a_h adalah nilai aktivasi fungsi ReLU, serta \hat{y} adalah perhitungan *softmax* dan perhitungan *loss function* menggunakan *cross-entropy loss*. Pers. (6) sampai (10) menunjukkan proses perhitungan yang terjadi dalam *backward*. Di mana δ_n adalah nilai *error*, $\frac{dL}{dW}$ adalah turunan dari bobot, $\frac{dL}{db_n}$ adalah turunan dari bias, serta $W_n(new)$ dan $b_n(new)$ adalah *update* bobot dan bias.

$$z_s = \sum_{r=0} x_d w_r + b_n \quad (3)$$

$$a_h = \max(0, a_g) \quad (4)$$

$$f(p_i) = \frac{\exp(p_i)}{\sum_{j=0}^{12} \exp(p_j)} \quad (5)$$

$$\delta_n = (y - o_t) \quad (6)$$



$$\frac{dL}{dV} = \delta_n(z)^T \quad (7)$$

$$\frac{dL}{db_n} = \delta_n \quad (8)$$

$$W_n(new) = W_n - \eta \frac{dL}{dW} \quad (9)$$

$$b_n(new) = b_n - \eta \frac{dL}{db_n} \quad (10)$$

2.4 Spesifikasi Komputasi

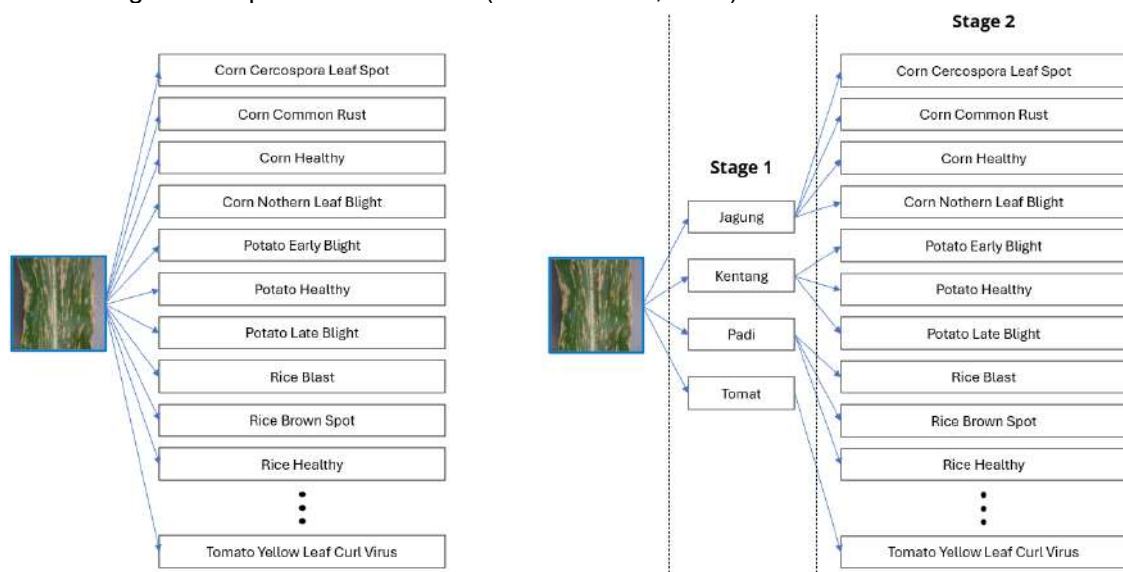
Penelitian ini menggunakan platform komputasi *cloud* yang disediakan oleh Kaggle.com. Spesifikasi komputasi yang digunakan adalah sebagai berikut:

- CPU: 2 vCPU Intel Xeon.
- GPU: NVIDIA Tesla P100 (16GB VRAM) atau Tesla T4 (16GB VRAM) tergantung pada ketersediaan sumber daya Kaggle.
- RAM: 13GB
- Penyimpanan: 20GB (Kaggle Dataset Storage), 5 GB untuk penyimpanan sementara selama proses pelatihan dan pengujian.

Penggunaan GPU pada platform Kaggle memungkinkan proses pelatihan model CNN berjalan lebih cepat karena kemampuan GPU dalam menangani komputasi paralel yang intensif, terutama dalam operasi matriks yang dominan dalam proses konvolusi dan backpropagation.

3. HASIL DAN PEMBAHASAN

Penelitian ini menggunakan pembagian dataset sebagai skenario pengujian dengan rasio 70:30, 80:20, dan 90:10 untuk *training* dan *testing*. Selain itu, diterapkan dua metode klasifikasi, yaitu *single-stage classification* dan *multi-stage classification* (Żarski et al., 2021). Gambar 5 dan Tabel 2 menampilkan skema *stage classification* dan pembagian skenario model. Enam skenario tersebut dilatih selama 300 *epoch* dengan *learning rate* sebesar 0,001, yang bertujuan untuk memperoleh hasil yang optimal. Setelah model dilatih, dilakukan perhitungan *confusion matrix* untuk mengevaluasi performa klasifikasi (Nawwar et al., 2021).



Gambar 5 Contoh Stage Classification



Skenario model 1 menggunakan *single-stage classification* dengan rasio 70:30, melibatkan 840 data latih dan 360 data uji. Tabel 3 menampilkan confusion matrix multi-kelas beserta nilai *precision*, *recall*, dan *F1-score* untuk 12 kelas penyakit tanaman berdasarkan citra daun. Skenario model 2 menggunakan pendekatan yang sama dengan rasio 80:20, memanfaatkan 960 data latih dan 240 data uji; hasilnya ditampilkan pada Tabel 4. Skenario model 3 juga menerapkan *single-stage classification* dengan rasio 90:10, terdiri atas 1.080 data latih dan 120 data uji, dengan hasil yang ditunjukkan pada Tabel 5.

Tabel 2 Pembagian Skenario Model

Uji Coba	Model Klasifikasi	Data Latih	Data Uji
1	<i>Single-stage classification</i>	70%	30%
2		80%	20%
3		90%	10%
4	<i>Multi-stage classification</i>	70%	30%
5		80%	20%
6		90%	10%

Tabel 3 Nilai dari Confusion Matrix Skenario Model 1

Kelas Jenis Penyakit	Precision	Recall	F1-Score
Corn Common Rust	45%	60%	51%
Corn Healthy	70%	78%	74%
Corn Northern Leaf Blight	90%	60%	72%
Potato Early Blight	95%	100%	97%
Potato Healthy	80%	76%	78%
Potato Late Blight	55%	69%	61%
Rice Brown Spot	60%	86%	71%
Rice Healthy	85%	100%	92%
Rice Scald	100%	77%	87%
Tomato Bacterial Spot	100%	83%	91%
Tomato Healthy	75%	100%	86%
Tomato Mosaic Virus	100%	80%	89%
Accuracy			74%

Tabel 4 Nilai dari Confusion Matrix Skenario Model 2

Kelas Jenis Penyakit	Precision	Recall	F1-Score
Corn Common Rust	20%	67%	31%
Corn Healthy	90%	68%	77%
Corn Northern Leaf Blight	83%	58%	68%
Potato Early Blight	87%	84%	85%
Potato Healthy	53%	80%	64%
Potato Late Blight	63%	58%	60%
Rice Brown Spot	27%	100%	42%
Rice Healthy	93%	78%	85%
Rice Scald	97%	67%	79%
Tomato Bacterial Spot	90%	96%	93%
Tomato Healthy	97%	85%	91%
Tomato Mosaic Virus	93%	80%	86%
Accuracy			80%



Tabel 5 Confusion Matrix Skenario Model 3

Kelas Jenis Penyakit	Precision	Recall	F1-Score
Corn Common Rust	10%	25%	14%
Corn Healthy	80%	80%	80%
Corn Northern Leaf Blight	90%	50%	64%
Potato Early Blight	90%	100%	95%
Potato Healthy	90%	75%	82%
Potato Late Blight	50%	83%	62%
Rice Brown Spot	50%	100%	67%
Rice Healthy	100%	83%	91%
Rice Scald	100%	83%	91%
Tomato Bacterial Spot	90%	90%	90%
Tomato Healthy	90%	100%	95%
Tomato Mosaic Virus	100%	77%	87%
Accuracy			78%

Tabel 6 Confusion Matrix Hasil Stage 1 dan 2 Skenario Model 4

Kelas Jenis Penyakit	Precision	Recall	F1-Score
Corn Common Rust	30%	68%	42%
Corn Healthy	70%	74%	71%
Corn Northern Leaf Blight	83%	50%	62%
Potato Early Blight	78%	93%	86%
Potato Healthy	45%	83%	59%
Potato Late Blight	81%	61%	69%
Rice Brown Spot	13%	100%	24%
Rice Healthy	97%	54%	71%
Rice Scald	95%	94%	94%
Tomato Bacterial Spot	97%	86%	90%
Tomato Healthy	87%	87%	86%
Tomato Mosaic Virus	100%	86%	92%
Accuracy			73%

Tabel 7 Confusion Matrix Hasil Stage 1 dan 2 Skenario Model 5

Kelas Jenis Penyakit	Precision	Recall	F1-Score
Corn Common Rust	30%	64%	40%
Corn Healthy	75%	79%	77%
Corn Northern Leaf Blight	90%	52%	67%
Potato Early Blight	87%	87%	86%
Potato Healthy	57%	72%	63%
Potato Late Blight	61%	74%	67%
Rice Brown Spot	49%	100%	66%
Rice Healthy	97%	74%	83%
Rice Scald	97%	87%	91%
Tomato Bacterial Spot	93%	79%	86%
Tomato Healthy	88%	84%	86%
Tomato Mosaic Virus	98%	88%	93%
Accuracy			76%

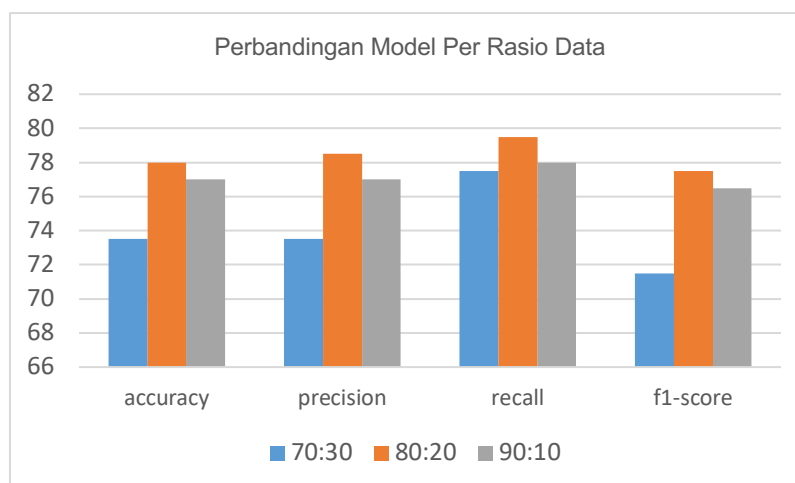


Tabel 8 *Confusion matrix* Hasil Stage 1 dan 2 Skenario Model 6

Kelas Jenis Penyakit	Precision	Recall	F1-Score
Corn Common Rust	54%	67%	60%
Corn Healthy	45%	83%	59%
Corn Northern Leaf Blight	72%	53%	61%
Potato Early Blight	93%	85%	88%
Potato Healthy	74%	68%	71%
Potato Late Blight	65%	82%	73%
Rice Brown Spot	100%	83%	90%
Rice Healthy	90%	82%	86%
Rice Scald	90%	91%	90%
Tomato Bacterial Spot	65%	73%	69%
Tomato Healthy	74%	83%	78%
Tomato Mosaic Virus	93%	77%	85%
Accuracy			76%

Skenario model 4 menggunakan *multi-stage classification* dengan rasio 70:30, terdiri dari 840 data latih dan 360 data uji, sebagaimana tercantum pada Tabel 6. Skenario model 5 menerapkan metode yang sama dengan rasio 80:20, melibatkan 960 data latih dan 240 data uji; hasilnya tercantum pada Tabel 7. Terakhir, skenario model 6 menggunakan *multi-stage classification* dengan rasio 90:10, menggunakan 1.080 data latih dan 102 data uji, dengan hasil klasifikasi yang disajikan pada Tabel 8.

Berdasarkan analisis hasil penelitian ini, dari enam model yang dievaluasi, model 2 menunjukkan performa tertinggi dengan akurasi 80%, presisi 80%, *recall* 81%, dan *F1-score* 79%. Keunggulan ini menunjukkan bahwa metode yang digunakan lebih efektif dalam mengklasifikasikan penyakit tanaman pangan dibandingkan skenario lainnya seperti yang ditunjukkan pada Gambar 6. Berdasarkan *confusion matrix* yang ditunjukkan pada Gambar 7 dengan nilai rinciannya pada Tabel 9, kesalahan klasifikasi lebih sering terjadi pada kelas dengan kemiripan warna dan tekstur, seperti Corn Common Rust vs. Corn Northern Leaf Blight, serta Potato Late Blight vs. Potato Healthy. Kesalahan ini kemungkinan dipengaruhi oleh pencahayaan saat pengambilan gambar dan kemiripan fitur tekstur antar kelas penyakit.



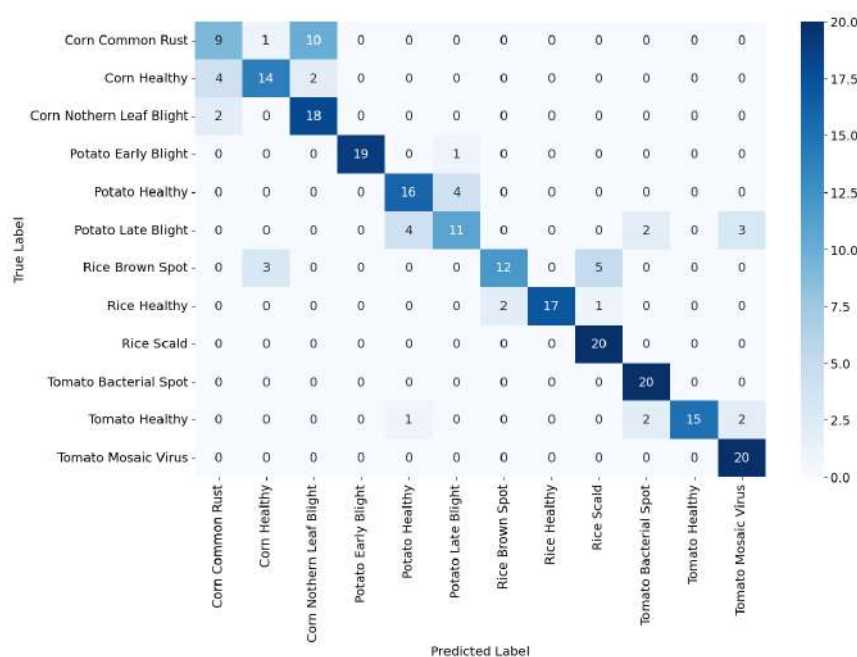
Gambar 6 Perbandingan Model Per Rasio Data

Pada *multi-stage classification*, meskipun akurasi keseluruhan lebih rendah dibandingkan single-stage, beberapa kelas menunjukkan peningkatan *recall* dan *F1-score*, yang berarti metode ini



mampu mengenali beberapa jenis penyakit dengan lebih baik. Gambar 8 membandingkan performa *single-stage* dan *multi-stage classification*. Hasil menunjukkan bahwa *single-stage classification* memiliki performa terbaik secara keseluruhan. Namun, perbedaan performa antara kedua metode ini tidak terlalu signifikan.

Perbedaan utama terletak pada kecepatan konvergensi, di mana *single-stage classification* mencapai konvergensi pada *epoch* ke-300, sementara *multi-stage classification* masih mengalami sedikit fluktuasi. Hal ini menunjukkan bahwa *single-stage* lebih efisien dalam waktu pelatihan, sedangkan *multi-stage* dapat meningkatkan performa klasifikasi pada kelas tertentu. Meskipun *single-stage classification* lebih cepat dalam mencapai kestabilan, klasifikasi bertahap memberikan hasil yang lebih akurat dan stabil untuk beberapa kelas penyakit tanaman. Ini menegaskan bahwa meskipun *single-stage* lebih efisien dalam hal waktu pelatihan, *multi-stage classification* memiliki potensi untuk meningkatkan akurasi pada kelas-kelas tertentu yang lebih sulit diklasifikasikan.

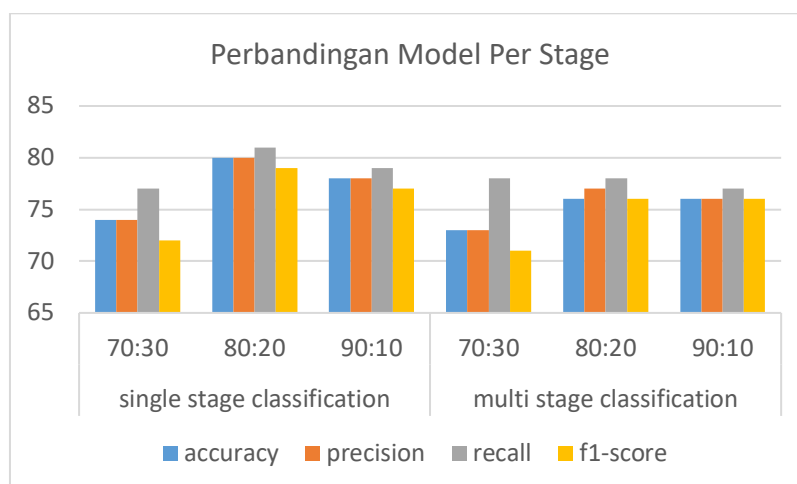


Gambar 7 Confusion Matrix Data Uji Skenario 2

Tabel 9 Nilai dari Confusion Matrix Skenario Model 2

Kelas Jenis Penyakit	Precision	Recall	F1-Score
Corn Common Rust	20%	67%	31%
Corn Healthy	90%	68%	77%
Corn Northern Leaf Blight	83%	58%	68%
Potato Early Blight	87%	84%	85%
Potato Healthy	53%	80%	64%
Potato Late Blight	63%	58%	60%
Rice Brown Spot	27%	100%	42%
Rice Healthy	93%	78%	85%
Rice Scald	97%	67%	79%
Tomato Bacterial Spot	90%	96%	93%
Tomato Healthy	97%	85%	91%
Tomato Mosaic Virus	93%	80%	86%
Accuracy	80%		





Gambar 8 Perbandingan Model Per Stage

4. KESIMPULAN

Penelitian ini mengevaluasi performa metode Convolutional Neural Network (CNN) dengan arsitektur LeNet-5 dalam klasifikasi penyakit tanaman pangan. Pengujian mencakup *preprocessing* data dengan metode *resize*, *rotate*, *zoom*, *flip*, dan *grayscale*, serta skenario klasifikasi *single-stage* dan *multi-stage* dengan rasio 70:30, 80:20, dan 90:10 untuk data latih dan uji. Hasil pengujian menunjukkan bahwa model 2 memiliki performa terbaik dengan akurasi 80%, presisi 80%, *recall* 81%, dan *F1-Score* 79%, serta berhasil mendeteksi 191 dari 240 data uji. Meskipun terdapat 49 kesalahan prediksi, terutama pada klasifikasi penyakit tanaman jagung, model ini menunjukkan potensi besar dalam penerapan nyata untuk membantu petani mengidentifikasi penyakit tanaman.

Namun, model lain dalam penelitian ini tidak mencapai performa yang setinggi model 2. Model dengan rasio 70:30 dan 90:10 menunjukkan akurasi yang lebih rendah karena beberapa faktor. Pada rasio 70:30, jumlah data latih yang lebih sedikit menyebabkan model kurang mampu mengenali pola dengan baik, sehingga performa prediksi menurun. Sementara itu, pada rasio 90:10, meskipun data latih lebih banyak, jumlah data uji yang lebih sedikit membuat model berisiko mengalami *overfitting*, di mana model lebih cenderung menghafal data latih daripada belajar pola umum yang dapat digeneralisasi. Selain itu, metode *multi-stage classification* menunjukkan konvergensi yang lebih lambat dibandingkan *single-stage classification*, yang mengakibatkan model belum sepenuhnya stabil dalam mendeteksi beberapa jenis penyakit tertentu.

Untuk meningkatkan akurasi, terutama pada penyakit yang sulit diklasifikasikan, penelitian selanjutnya dapat memanfaatkan *dataset* yang lebih besar dan beragam agar model lebih mampu melakukan generalisasi. Teknik augmentasi data yang lebih kompleks, seperti variasi warna dan pencahayaan, juga dapat digunakan untuk mengatasi perbedaan citra daun. Selain itu, penggabungan CNN dengan metode lain, misalnya Hybrid CNN-SVM, dapat dipertimbangkan untuk meningkatkan ketepatan klasifikasi. Penyesuaian arsitektur CNN dengan kedalaman dan jumlah filter yang optimal juga berpotensi memperkuat representasi fitur tanpa menambah risiko *overfitting*. *Fine-tuning hyperparameter*, seperti *learning rate*, jumlah *epoch*, dan fungsi aktivasi, perlu dilakukan secara menyeluruh agar proses pembelajaran lebih efektif. Dengan berbagai penyempurnaan tersebut, model diharapkan mampu menjadi solusi yang lebih akurat dan andal dalam deteksi penyakit tanaman, sehingga memberikan manfaat besar bagi sektor pertanian.

DAFTAR PUSTAKA



- Ambarwari, A., Husni, E. M., & Mahayana, D. (2023). Perkembangan Paradigma Metode Klasifikasi Citra Penginderaan Jauh dalam Perspektif Revolusi Sains Thomas Kuhn. *Jurnal Filsafat Indonesia*, 6(3), 465–473. <https://doi.org/10.23887/jfi.v6i3.53865>
- Anggiratih, E., Siswanti, S., Octaviani, S. K., & Sari, A. (2021). Klasifikasi Penyakit Tanaman Padi Menggunakan Model Deep Learning EfficientNet B3 dengan Transfer Learning. *Jurnal Ilmiah SINUS*, 19(1), 75–83. <https://doi.org/10.30646/sinus.v19i1.526>
- Anggraeni, D. S., Widayana, A., Rahayu, P. D., & Rozikin, C. (2022). Metode Algoritma Convolutional Neural Network pada Klasifikasi Penyakit Tanaman Cabai. *STRING (Satuan Tulisan Riset dan Inovasi Teknologi)*, 7(1), 73–78. <https://doi.org/10.30998/string.v7i1.13304>
- Chellapandi, B., Vijayalakshmi, M., & Chopra, S. (2021). Comparison of Pre-Trained Models Using Transfer Learning for Detecting Plant Disease. *2021 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS)*, 383–387. <https://doi.org/10.1109/ICCCIS51004.2021.9397098>
- Felix, F., Wijaya, J., Sutra, S. P., Kosasih, P. W., & Sirait, P. (2020). Implementasi Convolutional Neural Network untuk Identifikasi Jenis Tanaman Melalui Daun. *Jurnal SIFO Mikroskil*, 21(1), 1–10. <https://doi.org/10.55601/jsm.v21i1.672>
- Guo, Y., & Wang, X. (2022). Modulation Signal Classification Algorithm Based on Denoising Residual Convolutional Neural Network. *IEEE Access*, 10, 121733–121740. <https://doi.org/10.1109/ACCESS.2022.3221475>
- Hasan, Moh. A., Riyanto, Y., & Riana, D. (2021). Grape Leaf Image Disease Classification Using CNN-VGG16 Model. *Jurnal Teknologi dan Sistem Komputer*, 9(4), 218–223. <https://doi.org/10.14710/jtsiskom.2021.14013>
- Irfansyah, D., Mustikasari, M., & Suroso, A. (2021). Arsitektur Convolutional Neural Network (CNN) Alexnet untuk Klasifikasi Hama pada Citra Daun Tanaman Kopi. *Jurnal Informatika: Jurnal Pengembangan IT*, 6(2), 87–92. <https://doi.org/10.30591/jpit.v6i2.2802>
- Nawwar, N., Kasban, H., & Salama, M. (2021). Improvement of Confusion Matrix for Hand Vein Recognition Based on Deep-Learning Multi-Classfier Decisions. *Arab Journal of Nuclear Sciences and Applications*, 54(4), 133–146. <https://doi.org/10.21608/ajnsa.2021.70450.1460>
- Ngadi, N., Zaelany, A. A., Latifa, A., Harfina, D., Asiati, D., Setiawan, B., Ibnu, F., Triyono, T., & Rajagukguk, Z. (2023). Challenge of Agriculture Development in Indonesia: Rural Youth Mobility and Aging Workers in Agriculture Sector. *Sustainability*, 15(2), Article ID: 922. <https://doi.org/10.3390/su15020922>
- Oktaviana, U. N., Hendrawan, R., Annas, A. D. K., & Wicaksono, G. W. (2021). Klasifikasi Penyakit Padi berdasarkan Citra Daun Menggunakan Model Terlatih ResNet101. *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, 5(6), 1216–1222. <https://doi.org/10.29207/resti.v5i6.3607>
- Rodríguez-Lira, D.-C., Córdova-Esparza, D.-M., Álvarez-Alvarado, J. M., Terven, J., Romero-González, J.-A., & Rodríguez-Reséndiz, J. (2024). Trends in Machine and Deep Learning Techniques for Plant Disease Identification: A Systematic Review. *Agriculture*, 14(12), Article ID: 2188. <https://doi.org/10.3390/agriculture14122188>
- Sabrina, S. A., & Maki, W. F. A. (2022). Klasifikasi Penyakit pada Tanaman Kopi Robusta Berdasarkan Citra Daun Menggunakan Convolutional Neural Network. *EProceedings of Engineering*, 9(3), 1919–1927. <https://openlibrarypublications.telkomuniversity.ac.id/index.php/engineering/article/view/17997>
- Singh, V., & Misra, A. K. (2017). Detection of Plant Leaf Diseases Using Image Segmentation and Soft Computing Techniques. *Information Processing in Agriculture*, 4(1), 41–49. <https://doi.org/10.1016/j.inpa.2016.10.005>
- Suganda, T., & Wulandari, D. Y. (2018). Curvularia sp. Jamur Patogen Baru Penyebab Penyakit Bercak Daun pada Tanaman Sawi. *Jurnal Agrikultura*, 29(3), 119–123. <https://doi.org/10.24198/agrikultura.v29i3.22716>
- Tineges, R., Triayudi, A., & Sholihati, I. D. (2020). Analisis Sentimen Terhadap Layanan Indihome Berdasarkan Twitter dengan Metode Klasifikasi Support Vector Machine (SVM). *Jurnal Media Informatika Budidarma*, 4(3), 650–658. <https://doi.org/10.30865/mib.v4i3.2181>



- Tugrul, B., Elfatimi, E., & Eryigit, R. (2022). Convolutional Neural Networks in Detection of Plant Leaf Diseases: A Review. *Agriculture*, 12(8), Article ID: 1192. <https://doi.org/10.3390/agriculture12081192>
- Upadhyay, A., Chandel, N. S., Singh, K. P., Chakraborty, S. K., Nandede, B. M., Kumar, M., Subeesh, A., Upendar, K., Salem, A., & Elbeltagi, A. (2025). Deep Learning and Computer Vision in Plant Disease Detection: A Comprehensive Review of Techniques, Models, and Trends in Precision Agriculture. *Artificial Intelligence Review*, 58(3), Article ID: 92. <https://doi.org/10.1007/s10462-024-11100-x>
- Zalvadila, A. (2023). Klasifikasi Penyakit Tanaman Bawang Merah Menggunakan Metode SVM dan CNN. *Jurnal Informatika: Jurnal Pengembangan IT*, 8(3), 255–260. <https://doi.org/10.30591/jpit.v8i3.5341>
- Żarski, M., Wójcik, B., & Miszczak, J. A. (2021). Krakn: Transfer Learning Framework and Dataset for Infrastructure Thin Crack Detection. *SoftwareX*, 16, Article ID: 100893. <https://doi.org/10.1016/J.SOFTX.2021.100893>



Prediksi Kualitas Udara Menggunakan Metode CatBoost

Mohamad Arif Abdul Syukur ^{(1)*}, Suhartono ⁽²⁾, Totok Chamidy ⁽³⁾

Departemen Teknik Informatika, UIN Maulana Malik Ibrahim, Malang, Indonesia

e-mail : 200605110044@student.uin-malang.ac.id, {suhartono,to2k2013}@ti.uin-malang.ac.id.

* Penulis korespondensi.

Artikel ini diajukan 26 Juni 2024, direvisi 2 Januari 2025, diterima 8 Januari 2025, dan dipublikasikan 31 Mei 2025.

Abstract

Air is essential for life, but industrial activities, forest fires, cigarette smoke, and transportation contribute to air pollution. AirVisual AQI 2024 data ranks Jakarta in 11th place globally, with the highest level of pollution, reaching 127, which is unhealthy for sensitive groups and poses a risk of causing serious illnesses, including skin and respiratory diseases. This research uses the CatBoost method to predict the air quality index using Jakarta SPKU data taken from Kaggle. The data is processed through pre-processing and divided into four models with different comparisons of training and testing data. Each model was tested with the parameters iteration, depth, learning rate, and l2_leaf_reg, using GridSearchCV to find the optimal combination. The results show that the model with 90% training data and 10% testing data provides the best accuracy of 97%, due to the larger proportion of training data. This research demonstrates that the CatBoost method can yield accurate air quality predictions, which is crucial in supporting efforts to mitigate the impact of pollution and enhance public health.

Keywords: Prediction, Air Quality, Gradient Boosting, CatBoost, GridSearchCV

Abstrak

Udara penting bagi kehidupan, namun aktivitas industri, pembakaran hutan, asap rokok dan transportasi meningkatkan polusi udara. Data AirVisual AQI 2024 menempatkan Jakarta pada peringkat 11 dunia dengan tingkat polusi tertinggi, mencapai 127 yang tidak sehat bagi kelompok sensitif, dan berisiko menimbulkan penyakit serius seperti penyakit kulit dan pernapasan. Penelitian ini menggunakan metode CatBoost untuk memprediksi indeks kualitas udara dengan data SPKU Jakarta yang diambil dari Kaggle. Data tersebut diolah melalui pra-pemrosesan dan dibagi menjadi empat model dengan perbandingan data pelatihan dan pengujian yang berbeda. Setiap model diuji dengan parameter *iteration*, *depth*, *learning_rate*, dan *l2_leaf_reg*, menggunakan GridSearchCV untuk menemukan kombinasi terbaik. Hasilnya menunjukkan bahwa model dengan 90% data pelatihan dan 10% data pengujian memberikan akurasi terbaik sebesar 97%, karena proporsi data pelatihan yang lebih besar. Penelitian ini menunjukkan bahwa metode CatBoost dapat memberikan prediksi kualitas udara yang akurat, yang penting untuk mendukung upaya mengurangi dampak polusi dan meningkatkan kesehatan masyarakat.

Kata Kunci: Prediksi, Kualitas Udara, Gradient Boosting, CatBoost, GridSearchCV

1. PENDAHULUAN

Prediksi kualitas udara di kota Jakarta sangat penting karena dampaknya yang signifikan terhadap kesehatan masyarakat dan keadilan lingkungan (Apte et al., 2017). Variasi tingkat kualitas udara dapat mempunyai implikasi yang besar, sehingga penting untuk mengukur dan mengelola polusi udara secara efektif. Tingginya variasi tingkat polusi udara di Jakarta, khususnya polutan seperti PM10, SO₂, NO₂, O₃, dan CO, mempunyai implikasi langsung terhadap kesehatan masyarakat dan kesejahteraan lingkungan (Chandra et al., 2022; Iqbal et al., 2025). Terbatasnya jumlah stasiun pemantauan kualitas udara di Jakarta mengharuskan pengembangan model statistik untuk mengkarakterisasi secara komprehensif distribusi polusi udara secara spasial dan temporal (Lestari et al., 2022; Syuhada et al., 2023).

Pengembangan model prediksi berdasarkan data untuk memperkirakan tingkat polusi udara di Jakarta pada tahun-tahun mendatang menandakan diperlukannya pendekatan proaktif untuk



mengelola dan meningkatkan kualitas udara di kota tersebut (Ramadhani et al., 2022). Analisis terpadu terhadap polusi udara dan kondisi meteorologi di Jakarta menunjukkan adanya variasi musiman dalam konsentrasi polutan, sehingga menunjukkan perlunya model prediktif untuk mengantisipasi perubahan kualitas udara sepanjang tahun (Handhayani, 2023). Penggunaan AI dan sistem data besar untuk prediksi kualitas udara semakin menekankan kemajuan teknologi yang diperlukan untuk memperkirakan kualitas udara secara akurat dan efisien (Jufriansah et al., 2023).

Penelitian sebelumnya telah menggunakan berbagai metode pembelajaran secara efektif dalam memprediksi tingkat kualitas udara seperti Adaptive Boosting (AdaBoost), jaringan saraf tiruan (ANN), Random Forest, *stacking ensemble*, dan Support Vector Machine (SVM) dalam memperkirakan tingkat indeks kualitas udara (AQI) (Lei et al., 2023; Liang et al., 2020; Ravindiran et al., 2025). Akan tetapi sebagian besar penelitian menghadapi tantangan dalam menangani data yang tidak seimbang, fitur kategori yang kompleks, serta efisiensi komputasi yang rendah ketika mengelola *dataset* yang besar. Selain itu eksplorasi terhadap optimasi parameter yang kurang pada model yang digunakan dapat mengurangi potensi akurasi prediksi.

Berbagai model pembelajaran mesin telah digunakan untuk memprediksi kualitas udara dengan memasukkan parameter meteorologi. Support Vector Machine (SVM) terkenal karena fleksibilitas dan skalabilitasnya dalam prakiraan kualitas udara (Castelli et al., 2020). Selain itu, jaringan memori jangka pendek yang dioptimalkan dengan *hyperparameter* telah digunakan untuk prediksi tingkat kualitas udara, yang menunjukkan semakin besarnya adopsi metode berbasis pembelajaran mesin dalam mengatasi tantangan prediksi kualitas udara (Kim & Kim, 2015).

Pada penelitian ini, peneliti mencoba menggunakan metode CatBoost untuk mengatasi permasalahan tersebut dalam memprediksi indeks kualitas udara di wilayah Jakarta. Metode CatBoost unggul dalam menangani data kategori tanpa memerlukan pra-pemrosesan yang ekstensif dan memiliki performa yang baik pada data yang tidak seimbang serta dibandingkan dengan metode lain, CatBoost memiliki kemampuan mengurangi *overfitting* melalui algoritma *boosting* berbasis *gradient* dan pemrosesan yang lebih cepat pada *dataset* yang besar. Metode ini juga telah terbukti memberikan akurasi tinggi pada *dataset* polusi udara di India (Ramesh, 2023).

Penelitian ini menguji algoritma CatBoost dan menggabungkannya dengan teknik GridSearchCV untuk mencari parameter terbaik dan optimal. CatBoost digunakan dengan data polusi di wilayah Jakarta. Hasilnya terdapat empat model dengan pembagian data yang berbeda, model 1 dengan pembagian data latih 90% dan pengujian 10%, model 2 dengan pembagian data latih 80% dan pengujian 20%, model 3 dengan pembagian data latih 75% dan pengujian 25%, dan model 4 dengan pelatihan data latih 70% dan pengujian 30%. Data yang digunakan dalam penelitian ini adalah data indeks kualitas udara di wilayah Jakarta tahun 2020 yang diperoleh dari website penyedia data yaitu Kaggle.

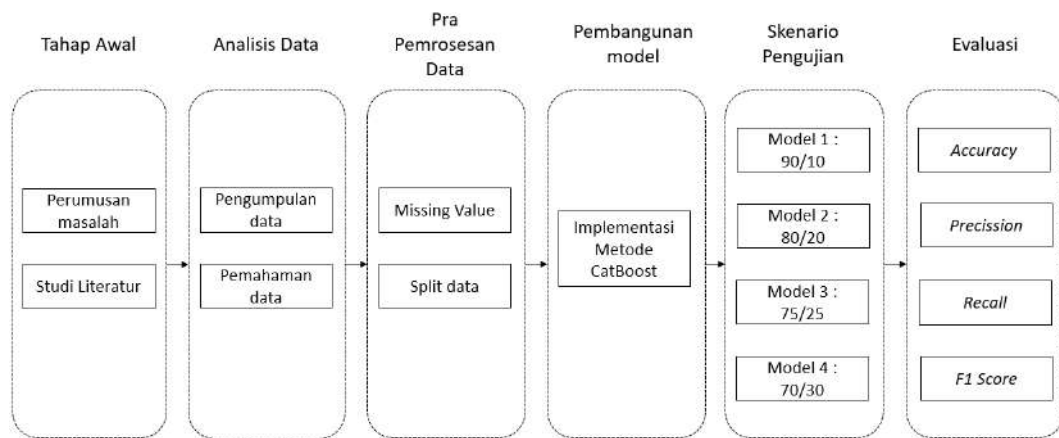
Oleh karena itu, penelitian ini bertujuan untuk mengetahui kinerja metode CatBoost dalam memprediksi indeks kualitas udara berdasarkan data indeks standar polutan udara di *dataset* wilayah Jakarta yang bersumber dari Kaggle untuk meningkatkan akurasi dalam klasifikasi prediksi. Harapannya penelitian ini dapat memberikan kontribusi signifikan terhadap penggunaan metode Catboost dan teknik GridSearchCV untuk memprediksi indeks kualitas udara di wilayah Jakarta yang lebih akurat, mendukung pengambilan keputusan untuk pengelolaan lingkungan yang lebih baik dan dapat memberikan solusi yang lebih komprehensif, efektif dan efisien.

2. METODE PENELITIAN

Metode Penelitian merupakan alur proses dari awal sampai akhir yang dilakukan untuk penelitian. Agar penelitian dapat efektif, diperlukan alur sistem yang terstruktur. Dalam penelitian ini sistem dimulai dari pendekatan penelitian perumusan masalah dan dilanjutkan dengan analisis data, kemudian setelah itu pengolahan data dengan pra-pemrosesan dan pengolahan data menggunakan Jupyter Notebook dilanjutkan dengan evaluasi model dan pembahasan skenario



pengujian hingga penarikan kesimpulan. Penelitian ini memiliki kerangka diagram alir proses untuk mencapai tujuan tersebut, seperti pada Gambar 1.



Gambar 1 Desain Penelitian

2.1 Pengumpulan Data

Pada tahap pengumpulan data yang dilakukan adalah mengumpulkan informasi mengenai topik kualitas udara. Data yang dikumpulkan merupakan data standar indeks polusi udara di wilayah Jakarta yang diambil dari Kaggle. Data ini dapat digunakan karena bersifat publik dan dapat diunduh dengan mudah melalui *website* Kaggle. Namun data tersebut merupakan data yang valid dan dapat dijadikan bahan pengolahan data yang kemudian dapat membantu dalam pengambilan keputusan. Sehingga data indeks standar pencemar udara di Jakarta dapat digunakan untuk penelitian tugas akhir. Datanya berjumlah 1830 dan terdiri dari 10 fitur yaitu tanggal, stasiun, pm10, so2, co, o3, no2, max, kritis, dan kategori. Tipe data setiap fitur berbeda-beda, ada yang bertipe objek, ada pula yang bertipe numerik. Data yang dikumpulkan akan digunakan sebagai masukan dalam proses pembangunan model dalam penelitian ini.

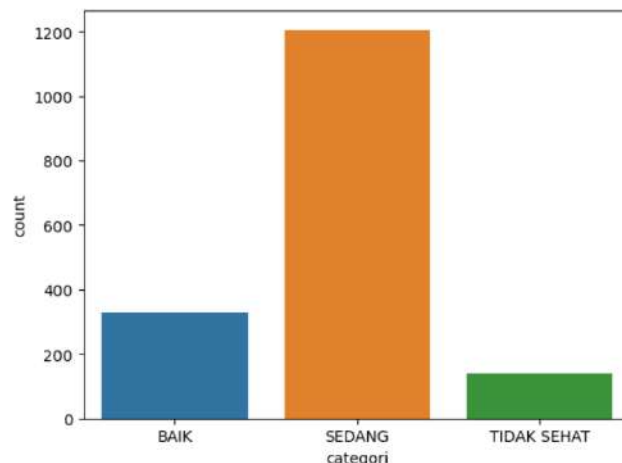
2.2 Pemahaman Data

Terdapat fitur tipe numerik yaitu fitur pm10, so2, co, o3, no2, dan max. Dan terdapat fitur tipe objek atau kategori yaitu tanggal, stasiun, kritis, dan kategori. Pada data ini terdapat tiga kategori indeks kualitas udara pada *dataset*, yaitu udara baik, sedang, dan tidak sehat. Hal ini dikarenakan pada *dataset* yang digunakan nilai tertinggi untuk indeks kualitas udara adalah 191 sehingga tidak ada kategori berbahaya di atas 300. Tabel 1 dan Gambar 2 menunjukkan deskripsi fitur dan gambar histogram analisis sebaran kategori *dataset*.

Tabel 1 Fitur *Dataset*

No.	Fitur	Keterangan
1	Tanggal	Tanggal Pengambilan Data
2	Stasiun	Lokasi Pengambilan Data
3	Pm10	Partikulat salah satu yang diukur
4	So2	Sulfida
5	Co	Karbon Monoksida
6	O3	Ozon
7	No2	Nitrogen Dioksida
8	Max	Nilai tertinggi dari semua parameter yang diukur
9	Critical	Parameter tertinggi yang diukur
10	Kategori	Kategori yang dihasilkan dari perhitungan indeks standar pencemaran udara





Gambar 2 Data Fitur Kategori

2.3 Missing Values

Pada pengolahan data *pre-processing* biasanya terdapat data yang tidak seimbang atau data yang berada di luar jangkauan. Sehingga perlu adanya perbaikan data dengan pengolahan untuk menemukan dan memperbaiki data yang tidak sesuai keakuratannya atau terdapat nilai yang hilang pada *dataset*. Dalam penelitian ini data yang tidak sesuai dan data yang tidak digunakan akan dihilangkan. Data yang akan dihapus adalah fitur “tanggal”, “stasiun”, dan “kritis”. Hal ini dilakukan karena fitur tersebut merupakan tipe objek atau tipe kategori dan pengaruhnya terhadap indeks kualitas udara juga kecil, bahkan hampir tidak ada.

2.4 Split Data

Pemisahan data dalam pengolahan data sangat penting dalam penelitian ini dengan tujuan untuk membagi data menjadi dua bagian yaitu data latih dan data uji. Data latih digunakan sebagai data untuk membangun suatu model, sedangkan data uji digunakan untuk menguji model yang telah dibuat dan juga untuk mengevaluasi kinerja model. Untuk pembagian data split pada penelitian ini dibagi menjadi 4 model yaitu model 1 dengan split 90:10 artinya 90% data latih dan 10% data uji. Model 2 dengan pembagian 80:20, artinya 80% data pelatihan dan 20% data pengujian. Model 3 dengan pembagian 75:25, artinya 75% data latih dan 25% data uji. Dan model 4 adalah pembagian 70:30, artinya 70% data pelatihan dan 30% data pengujian. Pembagian ini dapat memberikan performa yang baik untuk evaluasi keempat model dan juga dapat mencegah terjadinya *overfitting*. Selain itu juga dapat memberikan hasil kinerja yang akurat.

Dalam proses pembentukan data latih, penelitian ini menggunakan setiap komposisi model yang telah ditentukan dari keseluruhan jumlah data yang digunakan. Berdasarkan data pelatihan yang diperoleh dapat digunakan untuk melatih model CatBoost. Sedangkan data uji dengan komposisi masing-masing model yang ada merupakan data sisa dari keseluruhan data yang telah digunakan untuk data latih. Jadi data yang digunakan untuk data latih tidak digunakan untuk data uji. Dengan cara ini, data pengujian dapat digunakan untuk menguji model CatBoost dengan tingkat akurasi terbaik. Dengan keempat model yang telah ditentukan maka akan diambil akurasi terbaik untuk proses pengujian dan model tersebut akan digunakan dalam memprediksi indeks kualitas udara.

2.5 Implementasi CatBoost

Dalam pengolahan data ada langkah yang sangat penting yang harus dilakukan yaitu pengembangan model. Pengembangan model pada penelitian ini di mana algoritma CatBoost mempelajari data latih yang telah ditentukan pada proses sebelumnya. Model pelatihan kerja

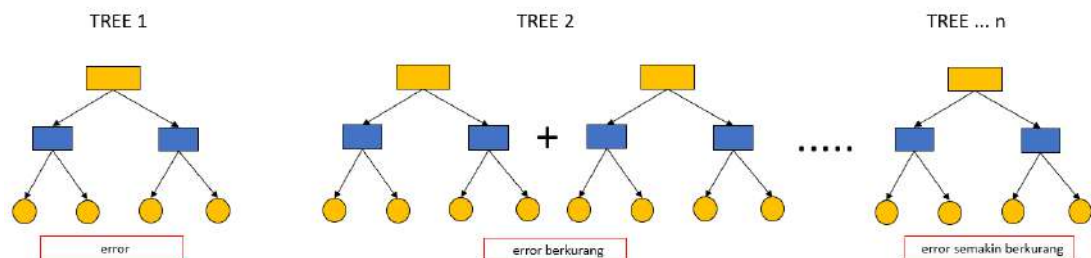


adalah menyesuaikan bobot dan bias terbaik pada algoritma dengan tujuan mengurangi fungsi kerugian dari fungsi kerugian yang ada dalam rentang nilai prediksi.

Pembuatan model menerapkan gradien yang setiap contoh dihitung berdasarkan contoh sebelumnya. Gradien yang sesuai dengan prediksi pertama dihitung menggunakan persamaan pada Pers. (1) (Dewi, 2021). Ketika pohon baru ditambahkan ke ansambel, setiap skor dihitung dari jumlah kandidat yang terpisah. Dalam hal ini gradien diperkirakan menggunakan kesamaan kosinus (*cosine similarity*). Dari fungsi skor tersebut perlu dilakukan pemilihan pohon di antara calon pohon yang ada. Misalnya, diberikan kandidat T_C, maka nilai fungsi skor kesamaan kosinus dapat dihitung menggunakan rumus pada Pers. (2) (Dewi, 2021). Visualisasi dari proses pembentukan model menggunakan algoritma CatBoost ditunjukkan pada Gambar 3.

$$grad_{r,j}(i) = \frac{\partial L(y_i, s)}{\partial s} \Big|_{s=Mr,j(i)} \quad (1)$$

$$Cosine = \frac{\sum_{i=1}^n w_i \cdot \Delta_i \cdot g_i}{\sqrt{\sum_{i=1}^n w_i \Delta_i^2} \cdot \sqrt{\sum_{i=1}^n w_i g_i^2}} \quad (2)$$



Gambar 3 Visualisasi CatBoost

2.6 Skenario Pengujian

Dalam penelitian ini, peneliti membagi data menjadi empat model seperti yang ditunjukkan pada Tabel 2. Model 1 merupakan model dengan rasio 90:10 yaitu 90% untuk data latih dan 10% untuk data uji. Model 2 merupakan model dengan rasio 80:20 yaitu 80% untuk data latih dan 20% untuk data uji. Model 3 merupakan model perbandingan 75:25 yaitu 75% untuk data latih dan 25% untuk data uji. Model 4 perbandingannya 70:30 yaitu 70% untuk data latih dan 30% untuk data uji. Dengan perbandingan yang berbeda maka akan ditemukan prediksi yang lebih akurat.

Tabel 2 Skenario Pengujian

No.	Model	Training	Testing
1	Model 1	90%	10%
2	Model 2	80%	20%
3	Model 3	75%	25%
4	Model 4	70%	30%

2.7 Evaluasi

Pada tahap evaluasi model dalam penelitian ini digunakan matriks konfusi (*confusion matrix*) sebagai dasar perhitungan metrik evaluasi seperti akurasi, presisi, *recall*, dan skor F1. Akurasi adalah sejauh mana nilai prediksi yang dihasilkan model mendekati nilai sebenarnya pada data dengan mengetahui jumlah data klasifikasi yang benar, dengan rumus perhitungannya disajikan pada Pers (3). Presisi adalah perbandingan nilai-nilai relevan berdasarkan seluruh nilai yang dipilih dengan membandingkan jumlah informasi relevan dengan jumlah total informasi yang dipilih dengan rumus perhitungannya pada Pers. (4).



Recall merupakan pemilihan rasio nilai relevan berdasarkan jumlah nilai relevan yang tersedia dengan cara membandingkan jumlah informasi relevan dengan jumlah total informasi relevan dalam informasi tersebut dengan rumus perhitungannya pada Pers. (5). Nilai presisi dan *recall* dihitung untuk mengevaluasi keakuratan dan cakupan model dalam memprediksi kategori tertentu. Selanjutnya, skor F1 sebagai rata-rata harmonis dari nilai presisi dan *recall* untuk mendapatkan gambaran seimbang antara kedua metrik tersebut, yang dijelaskan pada Pers. (6) (Amalia et al., 2022).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \times 100\% \quad (3)$$

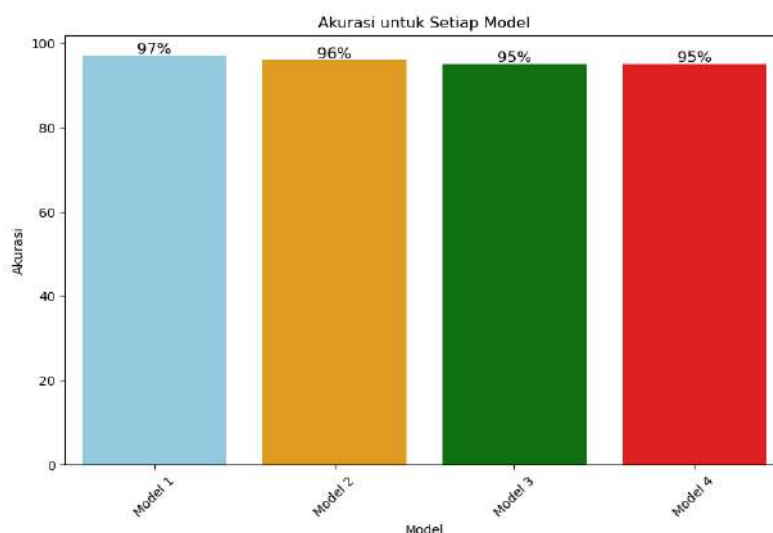
$$Precision = \frac{TP}{TP + FP} \times 100\% \quad (4)$$

$$Recall = \frac{TP}{TP + FN} \times 100\% \quad (5)$$

$$F1 \text{ score} = 2 \frac{precision \times recall}{precision + recall} \times 100\% \quad (6)$$

3. HASIL DAN PEMBAHASAN

Setiap model diuji menggunakan beberapa kombinasi parameter *depth*, *learning_rate*, *iteration*, dan *l2_leaf_reg*. Sehingga dalam pengujian semua model menggunakan nilai parameter yang sama. Nilai parameter *depth* yang digunakan adalah 6, 8, dan 10. Nilai parameter *learning_rate* yang digunakan adalah 0,1 dan 0,01. Nilai parameter *iteration* yang digunakan adalah 500, 1000, dan 1500. Sedangkan nilai parameter *l2_leaf_reg* yang digunakan adalah 1, 2, dan 3. Pada proses pengujian ini, pencarian parameter optimal dilakukan dengan menggunakan GridSearchCV. Sehingga mencari kombinasi parameter terbaik yang akan digunakan untuk model klasifikasi. Selanjutnya akan dievaluasi hasil masing-masing model.



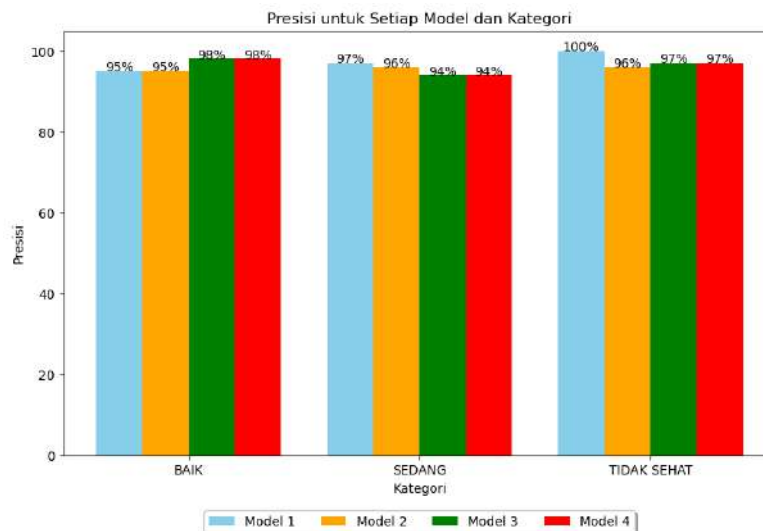
Gambar 4 Hasil Akurasi

Pada Gambar 4, Model 1 mendapatkan nilai akurasi yang lebih tinggi dibandingkan ketiga model lainnya. Hal ini dikarenakan oleh pembagian data antara data latih dan data uji. Model 1 menggunakan proporsi data latih sebesar 90% dan data uji 10%, sehingga model memiliki lebih banyak data untuk proses pelatihannya dan mampu mempelajari pola dengan lebih baik. Sementara, jumlah data uji yang lebih sedikit memungkinkan sistem melakukan pengujian secara



optimal karena model telah cukup terlatih. Model 1 terbukti menjadi model terbaik dibandingkan ketiga model lainnya berdasarkan hasil evaluasi akurasi. Pembagian data latih dan data uji sangat penting karena sangat mempengaruhi seberapa baik model dalam mengenali pola dari data baru (Okprana & Winanjaya, 2022). Dengan pembagian ini, dapat dilihat seberapa baik model dalam memprediksi data yang sebelumnya belum pernah terlihat.

Evaluasi yang dilakukan secara presisi memudahkan keakuratan model dalam mendeteksi data positif. Dengan demikian, jika nilai presisi meningkat maka jumlah kesalahan dalam memprediksi data positif akan berkurang karena presisi memberikan pemahaman mengenai keakuratan model dalam memprediksi data positif (Nainggolan & Sinaga, 2023). Hasil nilai presisi masing-masing model ditunjukkan pada Gambar 5. Dari kombinasi parameter optimal yang ditemukan yaitu pada parameter *depth*=6, *iteration*=1500, *l2_leaf_reg*=1, dan *learning_rate*=0.1 dengan hasil *mean_test_score cosine similirity* sebesar 0,958870. Seperti pada Gambar 5, model 1 memiliki nilai presisi yang paling tinggi dibandingkan dengan model lainnya. Walaupun nilai presisi pada kategori “baik” hanya mencapai 95%, namun pada kategori “sedang” mencapai 97% dan “tidak sehat” memperoleh presisi sempurna 100%.



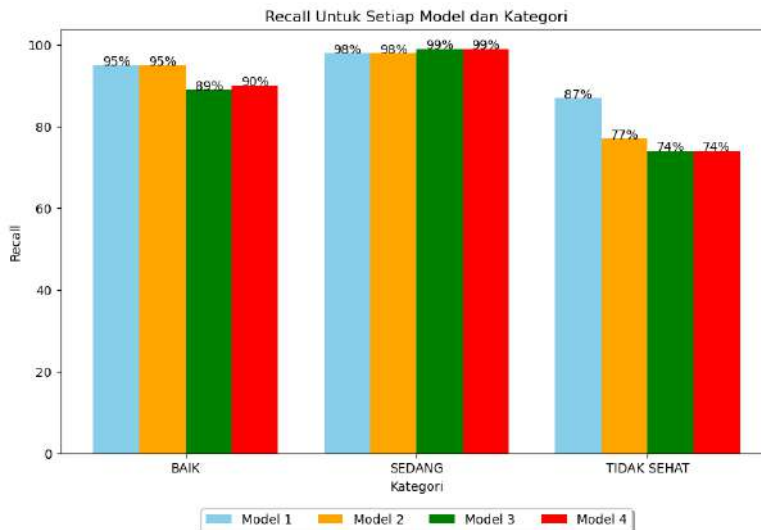
Gambar 5 Hasil Presisi

Berikutnya adalah *recall* yang merupakan perbandingan antara prediksi *True Positives* (TP) dengan seluruh data *true* positif. Sehingga evaluasi yang dilakukan dengan *recall* memberikan pemahaman seberapa sering model memprediksi positif padahal data sebenarnya positif (Saputro & Sari, 2020). Hasil nilai *recall* masing-masing model ditunjukkan pada Gambar 6. Semua model mendapatkan nilai *recall* yang tinggi sehingga dapat dikatakan banyak data positif yang diprediksi benar oleh model. Model 1 memperoleh nilai *recall* tertinggi pada kategori “tidak sehat” sebesar 87%, meskipun pada kategori “baik”, nilai *recall* model 1 dan model 2 bernilai sama yaitu 95%, dan pada kategori “sedang”, nilai *recall* model 1 sedikit lebih rendah dibandingkan dengan model 3 dan 4, yaitu sebesar 98%.

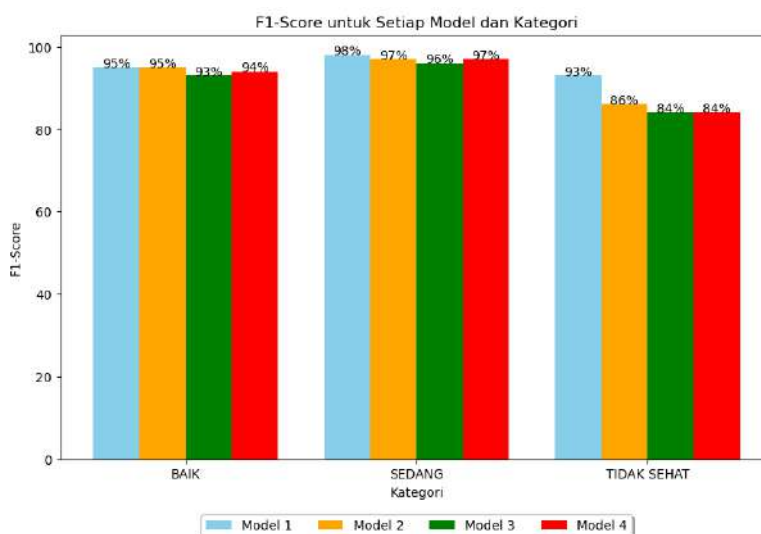
Selanjutnya evaluasi dilakukan menggunakan metrik *F1-score* yang merupakan nilai rata-rata harmonik tingkat presisi dan *recall* (Baharuddin et al., 2019). *F1-score* merupakan integrasi antara nilai presisi dan *recall* yang menjadi satu nilai sebagai kualitas model prediksi secara keseluruhan. Sehingga model dapat dinilai apakah akurat atau tidak dalam memprediksi data positif dengan benar. Hasil nilai *F1-score* masing-masing model ditunjukkan pada Gambar 7. Nilai *F1-score* pada semua model mencapai hasil yang baik. Sehingga dapat dikatakan bahwa semua model memiliki nilai keseimbangan untuk memprediksi sebagian besar kategori yang tepat (*recall*) dan menemukan prediksi yang akurat (*precision*). Nilai *F1-score* model 1 pada kategori “baik” hasilnya sama dengan model 2 yaitu 95%. Nilai *F1-score* tertinggi dihasilkan oleh model 1



pada kategori “sedang” sebesar 98%, nilai tersebut tertinggi dibandingkan ketiga model lainnya. Begitupun pada kategori “tidak sehat”, nilai pada model 1 juga menghasilkan nilai *F1-score* tertinggi dibandingkan dengan ketiga model lainnya, yaitu 93%.



Gambar 6 Hasil Recall



Gambar 7 Hasil F1-Score

4. KESIMPULAN

Berdasarkan skenario pengujian yang dilakukan pada penelitian ini, model dibagi menjadi 4 bagian, dengan perbandingan data latih dan data uji sebesar 90:10 pada model 1, 80:20 pada model 2, 75:25 pada model 3, serta 70:30 pada model 4. Model 1 menghasilkan nilai akurasi paling tinggi sehingga dapat dikatakan model 1 merupakan model terbaik dibandingkan model lainnya dalam memprediksi indeks kualitas udara di wilayah Jakarta. Data indeks standar pencemar udara di wilayah Jakarta diambil dari Kaggle dengan pemodelan menggunakan metode CatBoost.

Fungsi GridSearchCV diterapkan untuk mendapatkan kombinasi parameter optimal dari beberapa parameter yang ditentukan. Kombinasi parameter optimal yang didapat adalah pada



depth=6, iteration=1500, l2_leaf_reg=1, dan learning_rate=0.1 dengan hasil mean_test_score cosine similarity sebesar 0,958870. Model 1 menunjukkan performa yang sangat baik dengan nilai akurasi sebesar 97%. Pada evaluasi berdasarkan kategori, nilai presisi pada kategori “baik” mencapai 95%, kategori “sedang” 97%, dan kategori “tidak sehat” 100%. Sementara itu, nilai recall pada kategori “baik” mencapai 95%, kategori “sedang” 98%, dan kategori “tidak sehat” 87%. Selain itu, nilai F1-score yang merupakan kombinasi dari presisi dan recall menunjukkan hasil pada kategori “baik” sebesar 95%, kategori “sedang” 98%, dan kategori “tidak sehat” 93%.

Dapat disimpulkan bahwa pada penelitian ini, metode CatBoost dengan penerapan GridSearchCV dapat meningkatkan nilai akurasi pada proses prediksi indeks kualitas udara. Model terbaik diperoleh pada scenario pembagian data sebesar 90% untuk data latih dan 10% untuk data uji, dengan hasil prediksi masuk dalam kategori sangat baik. Pembagian data tersebut memberikan keuntungan karena jumlah data latih yang besar memungkinkan model untuk belajar secara optimal, sedangkan jumlah data uji yang lebih sedikit memudahkan sistem dalam melakukan evaluasi terhadap data yang belum pernah dilihat sebelumnya.

Penelitian ini menunjukkan kontribusi dalam pengembangan metode prediksi indeks kualitas udara dengan mengkombinasikan metode CatBoost dan GridSearchCV serta variasi parameter yang berbeda. Tentunya perlu adanya pengembangan untuk penelitian lebih lanjut mengenai penggunaan metode CatBoost atau penggunaan GridSearchCV dalam model prediksi atau klasifikasi sehingga dapat meningkatkan nilai akurasi yang ada. Selain itu, pengujian terhadap dataset yang lebih besar dan kompleks, serta perbandingan dengan algoritma pembelajaran mesin lainnya seperti XGBoost atau LightGBM, juga dapat memberikan wawasan tambahan mengenai keunggulan relatif dari pendekatan ini.

DAFTAR PUSTAKA

- Amalia, A., Zaidiah, A., & Isnainiyah, I. N. (2022). Prediksi Kualitas Udara Menggunakan Algoritma K-Nearest Neighbor. *JUPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)*, 7(2), 496–507. <https://doi.org/10.29100/jupi.v7i2.2843>
- Apte, J. S., Messier, K. P., Gani, S., Brauer, M., Kirchstetter, T. W., Lunden, M. M., Marshall, J. D., Portier, C. J., Vermeulen, R. C. H., & Hamburg, S. P. (2017). High-Resolution Air Pollution Mapping with Google Street View Cars: Exploiting Big Data. *Environmental Science & Technology*, 51(12), 6999–7008. <https://doi.org/10.1021/acs.est.7b00891>
- Baharuddin, M. M., Azis, H., & Hasanuddin, T. (2019). Analisis Performa Metode K-Nearest Neighbor untuk Identifikasi Jenis Kaca. *ILKOM Jurnal Ilmiah*, 11(3), 269–274. <https://doi.org/10.33096/ilkom.v11i3.489.269-274>
- Castelli, M., Clemente, F. M., Popovič, A., Silva, S., & Vanneschi, L. (2020). A Machine Learning Approach to Predict Air Quality in California. *Complexity*, 2020, 1–23. <https://doi.org/10.1155/2020/8049504>
- Chandra, W., Resti, Y., & Suprihatin, B. (2022). Implementation of a Breakpoint Halfway Discretization to Predict Jakarta's Air Quality. *Inovasi Matematika (Inomatika)*, 4(1), 1–10. <https://doi.org/10.35438/inomatika.v4i1.310>
- Dewi, N. K. (2021). *Deteksi Fake Follower Instagram Menggunakan Catboost Classifier* [UIN Syarif Hidayatullah]. <https://repository.uinjkt.ac.id/dspace/handle/123456789/56737>
- Handhayani, T. (2023). An Integrated Analysis of Air Pollution and Meteorological Conditions in Jakarta. *Scientific Reports*, 13(1), Article ID: 5798. <https://doi.org/10.1038/s41598-023-32817-9>
- Iqbal, M., Susilo, B., & Hizbaron, D. R. (2025). How Local Pollution and Transboundary Air Pollution Impact Air Quality in Jakarta? *Papers in Applied Geography*, 11(1), 49–62. <https://doi.org/10.1080/23754931.2024.2399626>
- Jufriansah, A., Khusnani, A., Pramudya, Y., Sya'bania, N., Leto, K. T., Hikmatiar, H., & Saputra, S. (2023). AI Big Data System to Predict Air Quality for Environmental Toxicology Monitoring. *Journal of Novel Engineering Science and Technology*, 2(01), 21–25. <https://doi.org/10.56741/jnest.v2i01.314>
- Kim, D. J., & Kim, J. Y. (2015). Generation Technique of Dynamic Monster's Behavior Pattern Based on User's Behavior Pattern Using FuSM. *Journal of Next-Generation Convergence*



- Information Services Technology*, 1(1), 9–18. <https://www.kci.go.kr/kciportal/ci/sereArticleSearch/ciSereArtiView.kci?sereArticleSearchBean.artid=ART002141142>
- Lei, T. M. T., Ng, S. C. W., & Siu, S. W. I. (2023). Application of ANN, XGBoost, and Other ML Methods to Forecast Air Quality in Macau. *Sustainability*, 15(6), Article ID: 5341. <https://doi.org/10.3390/su15065341>
- Lestari, P., Arrohman, M. K., Damayanti, S., & Klimont, Z. (2022). Emissions and Spatial Distribution of Air Pollutants from Anthropogenic Sources in Jakarta. *Atmospheric Pollution Research*, 13(9), Article ID: 101521. <https://doi.org/10.1016/j.apr.2022.101521>
- Liang, Y. C., Maimury, Y., Chen, A. H. L., & Juarez, J. R. C. (2020). Machine Learning-Based Prediction of Air Quality. *Applied Sciences*, 10(24), Article ID: 9151. <https://doi.org/10.3390/app10249151>
- Nainggolan, S. P., & Sinaga, A. (2023). Comparative Analysis of Accuracy of Random Forest and Gradient Boosting Classifier Algorithm for Diabetes Classification. *Sebatik*, 27(1), 97–102. <https://doi.org/10.46984/sebatik.v27i1.2157>
- Okprana, H., & Winanjaya, R. (2022). Analisis Pengaruh Komposisi Data Training dan Testing Terhadap Akurasi Algoritma Resilient Backpropagation (RProp). *BRAHMANA: Jurnal Penerapan Kecerdasan Buatan*, 4(1), 89–95. <https://doi.org/10.30645/brahmana.v4i1.138>
- Ramadhani, R. F., Prasetyowati, S. S., & Sibaroni, Y. (2022). Performance Analysis of Air Pollution Classification Prediction Map with Decision Tree and ANN. *Journal of Computer System and Informatics (JoSYC)*, 3(4), 536–543. <https://doi.org/10.47065/josyc.v3i4.2117>
- Ramesh, L. (2023). Prediction of Air Pollution and an Air Quality Index Using Machine Learning Techniques. *International Journal of Advanced Research in Computer Science*, 14(02), 51–55. <https://doi.org/10.26483/ijarcs.v14i2.6972>
- Ravindiran, G., Karthick, K., Rajamanickam, S., Datta, D., Das, B., Shyamala, G., Hayder, G., & Maria, A. (2025). Ensemble Stacking of Machine Learning Models for Air Quality Prediction for Hyderabad City in India. *IScience*, 28(2), Article ID: 111894. <https://doi.org/10.1016/j.isci.2025.111894>
- Saputro, I. W., & Sari, B. W. (2020). Uji Performa Algoritma Naïve Bayes untuk Prediksi Masa Studi Mahasiswa. *Creative Information Technology Journal*, 6(1), 1–11. <https://doi.org/10.24076/citec.2019v6i1.178>
- Syuhada, G., Akbar, A., Hardiawan, D., Pun, V., Darmawan, A., Heryati, S. H. A., Siregar, A. Y. M., Kusuma, R. R., Driejana, R., Ingole, V., Kass, D., & Mehta, S. (2023). Impacts of Air Pollution on Health and Cost of Illness in Jakarta, Indonesia. *International Journal of Environmental Research and Public Health*, 20(4), Article ID: 2916. <https://doi.org/10.3390/ijerph20042916>





9 772527 583007



LABORATORIUM AGAMA
MASJID SUNAN KALIJAGA