JISKA (Jurnal Informatika Sunan Kalijaga)
Vol. 11, No. 2, MEI, 2026, Pp. 143 – 154
ISSN: 2527 – 5836 (print) | 2528 – 0074 (online)

Evaluasi Pemanfaatan *Online Compiler* dan *Autograder* Sebagai Pendukung Aktivitasi Praktikum Pemograman Mahasiswa

Muhammad Galih Wonoseto [(1)*], Muhammad Jibril [(2)]
Departemen Informatika, UIN Sunan Kalijaga, Yogyakarta, Indonesia
e-mail : muhammad.wonoseto@uin-suka.ac.id, 20106050019@student.uin-suka.ac.id.
* Penulis korespondensi.
Artikel ini diajukan 7 Desember 2025, direvisi 10 April 2026, diterima 12 April 2026, dan dipublikasikan 25 Mei 2026.

Abstract

The automation of programming practicum assessment using an online compiler has become increasingly important to address the limitations of manual evaluation, such as time inefficiency, subjectivity, and the lack of automated feedback support in conventional e-learning systems. This study aims to evaluate the implementation of an online compiler and autograder in a programming practicum through interviews and questionnaires with 91 respondents, comprising lecturers and students. The online compiler and autograder were implemented using the Moodle platform, integrated with the CodeRunner plugin and the JOBE server to automate code evaluation and provide real-time feedback to students. User acceptance was assessed using the Technology Acceptance Model (TAM), which includes three dimensions: perceived ease of use, perceived usefulness, and system acceptance. The evaluation results indicate positive scores across all dimensions, with 78.75% for ease of use, 78.09% for usefulness, and 76.33% for system acceptance. These findings demonstrate that the proposed system effectively improves the assessment process and shows strong potential for adoption in programming education environments.

Keywords: Evaluation, E-Learning, Online Compiler, Autograder, Technology Acceptance Model

Abstrak

Otomatisasi penilaian praktikum pemrograman menggunakan online compiler menjadi sangat penting untuk mengatasi keterbatasan penilaian manual, seperti inefisiensi waktu, subjektivitas penilaian, dan kurangnya dukungan umpan balik otomatis pada sistem e-learning konvensional. Penelitian ini bertujuan untuk mengevaluasi aplikasi online compiler dan autograder dalam praktikum pemrograman menggunakan metode wawancara dan kuesioner pada 91 responden, yaitu dosen dan mahasiswa. Online compiler dan autograder diimplementasikan menggunakan platform Moodle yang diintegrasikan dengan plugin CodeRunner dan server JOBE untuk mengotomatisasi pemeriksaan kode program serta memberikan umpan balik secara real-time kepada mahasiswa. Evaluasi penerimaan pengguna dilakukan menggunakan model *Technology Acceptance Model* yang mencakup tiga aspek, yaitu kemudahan penggunaan, manfaat, dan penerimaan sistem. Hasil evaluasi menunjukkan nilai positif pada ketiga aspek tersebut, masing-masing sebesar 78,75% kemudahan penggunaan, 78,09% kebermanfaatan, dan 76,33% penerimaan sistem. Temuan ini menunjukkan bahwa sistem yang diusulkan secara efektif meningkatkan proses penilaian dan berpotensi tinggi untuk diadopsi dalam lingkungan pendidikan pemrograman.

Kata Kunci: Evaluasi, E-Learning, Compiler Online, Penilaian Otomatis, Model Penerimaan Teknologi

1. PENDAHULUAN

Teknologi digital telah menjadi bagian integral dalam pendidikan modern, termasuk dalam bidang pendidikan pemrograman yang terus berkembang (Arikarani & Amirudin, 2021). UIN Sunan Kalijaga Yogyakarta telah memiliki platform pembelajaran daring melalui aplikasi daring.uin-suka.ac.id. Platform tersebut telah mendukung berbagai aktivitas perkuliahan seperti diskusi, audio, video, meeting, pengumpulan tugas, berbagi materi, forum, pengumuman, kuis, update



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

status, kirim presensi, kirim nilai tugas, dan lain-lain. Akan tetapi, platform tersebut belum menyediakan fitur online compiler dan auto grader untuk tugas praktikum mata kuliah pemrograman. Laboratorium Informatika UIN Sunan Kalijaga telah memiliki data center yang siap digunakan untuk mengembangkan berbagai aplikasi (Wonoseto et al., 2025). Oleh karena itu, peneliti berinisiatif membuat dan memanfaatkan platform praktikum berbasis Moodle, JOBE server, dan CodeRunner pada data center milik Laboratorium Informatika UIN Sunan Kalijaga Yogyakarta. Platform tersebut menyediakan fitur compiler online dan auto grader untuk tugas praktikum mata kuliah pemrograman. Platform *e-learning* ini diberi nama Kalisa.

Sebelum ada platform ini, dosen algoritma pemrograman mengoreksi jawaban kode program mahasiswa secara manual satu per satu. Tantangan perkuliahan dan praktikum mata kuliah algoritma pemrograman menjadi semakin besar seiring bertambahnya jumlah mahasiswa dalam satu kelas. Idealnya, praktikum mata kuliah algoritma pemrograman maksimal hanya diisi oleh 20 mahasiswa per kelas. Namun, kini jumlah mahasiswa dalam satu kelas mencapai 63 orang. Peningkatan jumlah mahasiswa dalam kelas pemrograman dapat menimbulkan tantangan dalam proses penilaian tugas secara individual, sehingga pendekatan penilaian otomatis mulai banyak dimanfaatkan untuk membantu menangani proses tersebut (Goodfellow et al., 2024). Tantangan ini tidak diiringi dengan peningkatan sarana dan prasarana penunjang, seperti jumlah ruang kelas, jumlah komputer di laboratorium, serta tidak adanya asisten dosen. Selain itu, dosen dibebani dengan tugas administrasi lainnya seperti BKD, SKP, e-kinerja, penelitian, pengabdian, serta tugas lain seperti pembimbing dan penguji (Hamukti et al., 2018). Menurut penelitian Hamukti, tugas administrasi dosen sering kali menyita lebih dari 40% waktu kerja dosen, sehingga mengurangi fokus dosen pada evaluasi, penilaian pembelajaran dan pengajaran (Hamukti et al., 2018). Beban kerja dosen yang berlebihan berpotensi mengakibatkan proses evaluasi dan penilaian menjadi tidak objektif, tidak cermat, tidak konsisten dan tidak teliti.

Melihat permasalahan yang ada, salah satu solusi yang dapat diusulkan adalah pemanfaatan Moodle sebagai dasar *Learning Management System* (LMS) yang dilengkapi dengan plugin CodeRunner (Pringuet et al., 2021). Moodle adalah platform LMS berbasis *open source* yang telah banyak digunakan di institusi pendidikan (Gamage et al., 2022). Dengan integrasi plugin CodeRunner, Moodle dapat menyediakan fitur *online compiler* dan *automation testing* (Athaya et al., 2021), sehingga memungkinkan mahasiswa untuk langsung menulis, menjalankan, dan menguji kode mereka secara daring. Sistem penilaian otomatis dalam pendidikan pemrograman umumnya memanfaatkan pengujian dinamis, terutama *unit testing*, untuk mengevaluasi kebenaran kode sekaligus memberikan umpan balik kepada mahasiswa secara otomatis (Messer et al., 2024). Hal ini tidak hanya meningkatkan efisiensi proses praktikum, tetapi juga memberikan pengalaman belajar yang lebih interaktif dan relevan dengan kebutuhan dunia kerja. Salama et al. (2020) mengembangkan sistem pembelajaran jarak jauh untuk pemrograman yang dilengkapi dengan compiler online. Sistem tersebut memungkinkan mahasiswa mengakses materi pembelajaran, menulis dan mengompilasi kode, serta berinteraksi dengan dosen secara daring, sehingga mendukung fleksibilitas pembelajaran tanpa bergantung pada lokasi fisik. Penelitian oleh Teo et al. (2019) menunjukkan bahwa penggunaan Moodle pada institusi pendidikan tinggi di Asia meningkatkan keterampilan kolaborasi mahasiswa hingga 39%.

Setelah ada platform ini, proses penilaian kode program mahasiswa dapat dilakukan secara otomatis oleh autograder dari online compiler. Penggunaan *automated assessment tools* dalam pembelajaran pemrograman dapat memberikan umpan balik secara cepat sekaligus meningkatkan efisiensi dan konsistensi proses penilaian (Cipriano et al., 2022). Hal ini merupakan salah satu upaya dalam proses transformasi digital yang sedang dilakukan. Transformasi digital yang diupayakan mencakup berbagai aspek, di antaranya penggunaan perangkat lunak e-learning hingga modul praktikum yang mudah diakses (Bangsawan, 2023). Penilaian dengan online compiler dan autograder diharapkan dapat meningkatkan objektivitas penilaian praktikum pemrograman. Sistem ini juga mampu mengurangi kebutuhan ruang untuk menyimpan dokumen fisik terkait proses penilaian, sehingga lebih efisien dalam pengelolaannya. (Gamage et al., 2022).

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

Menurut Maylawati et al. (2024), teknologi pengajaran modern, seperti CodeRunner, secara signifikan meningkatkan kualitas mata kuliah pemrograman dan berkontribusi pada revolusi pengajaran yang sedang berlangsung. CodeRunner adalah plugin Moodle sumber terbuka gratis untuk menandai kode siswa secara otomatis (Croft & England, 2020). Croft & England (2020) telah memanfaatkan Moodle, CodeRunner, dan server JOBE sebagai alat dan sarana pembelajaran pemrograman komputer di Coventry University. Hal tersebut mendorong peneliti untuk melakukan penelitian serupa di Program Studi Informatika UIN Sunan Kalijaga Yogyakarta. Penelitian ini bertujuan untuk mengembangkan dan mengevaluasi aplikasi online compiler dan otomasi penilaian (autograder) untuk praktikum pemrograman yang diberi nama Kalisa.

2. METODE PENELITIAN

2.1 Metode Pengumpulan Data

Pengumpulan data dalam penelitian ini dilakukan dengan berbagai cara untuk memperoleh informasi yang relevan mengenai kebutuhan dosen dan mahasiswa terhadap sistem penilaian praktikum pemrograman. Metode pengumpulan data yang digunakan antara lain metode wawancara dan kuesioner. Wawancara dilakukan dengan dosen pengampu mata kuliah pemrograman untuk menggali permasalahan yang dihadapi dalam proses pembelajaran serta kebutuhan fitur pada sistem. Selain itu, wawancara juga dilakukan dengan mahasiswa untuk mengetahui kendala yang dihadapi dalam menggunakan sistem yang ada serta harapan mereka terhadap sistem yang akan dikembangkan. Kuesioner diberikan kepada mahasiswa untuk mengumpulkan data kuantitatif mengenai persepsi mereka terhadap sistem penilaian yang ada, serta seberapa besar mereka menginginkan perubahan atau peningkatan pada sistem penilaian praktikum pemrograman. Penelitian ini melibatkan 91 responden yang terdiri dari mahasiswa dan dosen.

2.2 Metode Lean UX

Metode yang digunakan dalam penelitian ini mengadopsi metode Lean UX. Lean UX menawarkan berbagai kelebihan dan sangat relevan dalam lingkungan yang dinamis, seperti startup atau organisasi dengan siklus pengembangan produk yang cepat (Aarlien & Colomo-Palacios, 2020). Metode Lean UX mempercepat proses pengambilan keputusan karena langsung menggunakan data dari pengguna. Kolaborasi lintas fungsi juga mendorong kerja sama yang lebih efektif antar tim. Selain itu, fokus Lean UX yang berorientasi pada hasil memastikan bahwa solusi yang dihasilkan benar-benar sesuai dengan kebutuhan pengguna (Aarlien & Colomo-Palacios, 2020; Liikkanen et al., 2014).

Metode Lean UX digunakan untuk memastikan bahwa pengembangan sistem sesuai dengan kebutuhan pengguna melalui pendekatan iteratif. Metode ini terdiri dari empat tahapan utama: *Declare Assumptions*, *Create an MVP*, *Run an Experiment*, dan *Feedback and Research*. Setiap tahapan disusun untuk menghasilkan produk yang relevan berdasarkan validasi dan umpan balik pengguna.

a) *Declare Assumptions*
   Pada tahap ini, peneliti mengidentifikasi kebutuhan atau perilaku pengguna yang diasumsikan menjadi prioritas utama untuk diselesaikan. Dalam penelitian ini, identifikasi kebutuhan pengguna dilakukan melalui wawancara dengan dosen dan mahasiswa.

b) *Create a Minimum Variable Product* (MVP)
   Pada tahap ini, peneliti mengembangkan versi produk minimal yang berfungsi sebagai alat untuk menguji asumsi dan memperoleh umpan balik dari pengguna.

c) *Run an Experiment*
   MVP diuji langsung kepada pengguna, dan data hasil pengujian digunakan untuk menentukan langkah berikutnya.

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

d) *Feedback & Research*
Hasil eksperimen dianalisis untuk memvalidasi asumsi awal atau melakukan penyesuaian yang diperlukan. Dalam penelitian ini, salah satu *umpan balik* yang digunakan adalah instrumen *Technology Acceptance Model* (TAM). Selain menggunakan instrumen kuesioner TAM, feedback dan evaluasi juga dilakukan melalui wawancara dengan mahasiswa dan dosen.

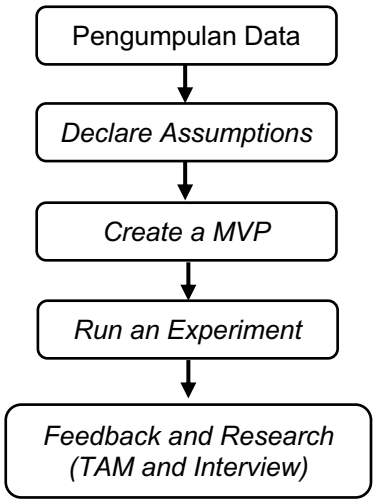
2.3 Technology Acceptance Model

Untuk mengukur sejauh mana transformasi digital dalam penilaian praktikum pemrograman diterima dan digunakan oleh dosen dan mahasiswa, penelitian ini akan menggunakan *Technology Acceptance Model* sebagai alat ukur. TAM adalah model yang dikembangkan untuk memahami dan mengevaluasi faktor-faktor yang memengaruhi penerimaan teknologi, khususnya dalam konteks penggunaan sistem informasi (Marangunić & Granić, 2015). Al-Adwan et al. (2023) juga menggunakan metode TAM untuk meneliti penggunaan teknologi metaverse dalam pendidikan di perguruan tinggi. Velayani et al. (2023) menggunakan TAM untuk mengevaluasi aplikasi telemedicine selama pandemi COVID-19. Tabel 1 menunjukkan indikator dan daftar pernyataan dalam *Technology Acceptance Model.* Alur penelitian penggabungan ketiga metode tersebut dalam penelitian ini dapat dilihat pada Gambar 1.

Tabel 1 Daftar Indikator Technology Acceptance Model

Variabel	Kode	Indikator
Kemudahan (*Perceived Ease of Use*)	PEOU 1	Aplikasi Kalisa mudah untuk diakses.
	PEOU 2	Saya merasa antarmuka aplikasi Kalisa sederhana dan mudah digunakan.
	PEOU 3	Menavigasi dalam aplikasi Kalisa sangat mudah.
	PEOU 4	Instruksi dalam menyelesaikan kursus di aplikasi Kalisa mudah untuk dipahami.
	PEOU 5	Menjawab pertanyaan terkait pemrograman pada course di aplikasi Kalisa terasa intuitif.
	PEOU 6	Mengirimkan dan meninjau jawaban saya melalui aplikasi Kalisa sangat mudah.
Kebermanfaatan (*Perceived of Usefulness*)	POU 1	Penggunaan aplikasi Kalisa membantu saya memahami konsep pemrograman dengan lebih baik.
	POU 2	Umpan balik dari aplikasi Kalisa membantu saya meningkatkan keterampilan pemrograman.
	POU 3	Aplikasi Kalisa memudahkan saya dalam mempersiapkan praktikum pemrograman.
	POU 4	Penggunaan aplikasi Kalisa meningkatkan pengalaman belajar saya dalam mata kuliah pemrograman.
	POU 5	Aplikasi Kalisa membantu saya mengidentifikasi area yang perlu saya tingkatkan dalam pemrograman.
	POU 6	Saya percaya penggunaan Aplikasi Kalisa meningkatkan kualitas penilaian saya.
Penerimaan (*Acceptance of It*)	AOT 1	Saya merasa percaya diri menggunakan aplikasi Kalisa untuk penilaian pemrograman.
	AOT 2	Saya puas dengan cara aplikasi Kalisa membantu saya dalam mata kuliah pemrograman.
	AOT 3	Aplikasi Kalisa adalah alat yang membantu dalam proses belajar saya.
	AOT 4	Saya bersedia mengandalkan aplikasi Kalisa untuk studi pemrograman saya.
	AOT 5	Modul Aplikasi Kalisa akan menjadi bagian penting dari pendidikan pemrograman saya.
	AOT 6	Saya percaya aplikasi Kalisa membuat proses penilaian menjadi adil dan transparan.

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



Gambar 1 Metode Penelitian

3. HASIL DAN PEMBAHASAN

Proses pembelajaran pemrograman di Program Studi Informatika UIN Sunan Kalijaga, sebelum menggunakan online compiler dan autograder, menghadapi tantangan signifikan seiring meningkatnya jumlah mahasiswa. Proses penilaian praktikum pemrograman sebelumnya dilakukan secara manual dengan metode yang membutuhkan waktu sangat lama. Sebelumnya, mahasiswa perlu menulis kode program mereka dengan tangan atau menjalankan kode mereka di perangkat lokal, menyimpan hasilnya dalam bentuk gambar tangkapan layar (*screenshot*), lalu mengunggahnya ke platform daring.uin-suka.ac.id. Namun, seperti yang diungkapkan oleh beberapa mahasiswa, proses ini juga memiliki tantangan. Tidak semua mahasiswa memiliki laptop yang memadai untuk menjalankan kode program. Kadang, hasil yang diunggah tidak konsisten karena masalah perangkat atau kelalaian saat membuat dokumen. Lebih lanjut, platform daring.uin-suka.ac.id belum mendukung fitur seperti compiler, automation testing, dan autograder, sehingga banyak tugas harus diperiksa secara manual oleh dosen.

Tabel 2 Hasil Pengumpulan Data

No	Pernyataan	Ya	Tidak
1	E-learning yang berjalan saat ini (https://daring.uin-suka.ac.id/) sudah menunjang kebutuhan tugas pemrograman	54%	46%
2	Perlu ada pembaruan https://daring.uin-suka.ac.id untuk menunjang kebutuhan tugas pemrograman	80%	20%

Tabel 3 Hasil Asumsi

Prioritas	Poin Utama Asumsi	
Asumsi #1	Fitur	Fitur inti seperti compiler terintegrasi dan automation testing sangat memengaruhi hasil pembelajaran mahasiswa.
Asumsi #5	Monitoring dan Pelaporan	Fitur ini membantu dosen memantau progres mahasiswa, tetapi dapat dikembangkan setelah fitur inti selesai.
Asumsi #3	Desain	Desain yang sederhana dan responsif memastikan kemudahan akses dan mendorong adopsi sistem oleh pengguna.
Asumsi #4	Adaptabilitas	Pelatihan dan dukungan teknis membantu pengguna memahami dan memanfaatkan sistem secara efektif.
Asumsi #2	Efisiensi Biaya dan Sumber Daya	Pengelolaan anggaran yang efisien penting untuk menjaga keberlanjutan sistem.

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

Berdasarkan hasil survei yang dilakukan kepada 50 mahasiswa Informatika, keberadaan platform daring.uin-suka.ac.id belum sepenuhnya memenuhi kebutuhan tugas atau praktikum pemrograman dan diperlukan pembaruan dari aplikasi sebelumnya untuk menunjang kebutuhan tugas pemrograman. Hasil tersebut dapat dilihat pada Tabel 2. Berdasarkan proses *declare assumptions* yang telah dilakukan, didapatkan hasil asumsi dengan urutan prioritas seperti pada Tabel 3.

Setelah melewati serangkaian proses *Declare Assumptions*, tahap selanjutnya adalah membangun *Minimum Viable Product* (MVP) untuk menguji sejauh mana solusi yang diusulkan mampu menjawab permasalahan dalam penilaian praktikum pemrograman. Dalam konteks penelitian ini, Moodle dipilih sebagai platform dasar karena sifatnya yang fleksibel, *open source*, serta mampu diintegrasikan dengan beragam plugin khusus untuk pendidikan. Untuk mengatasi masalah objektivitas dan konsistensi penilaian, *CodeRunner* dipilih sebagai plugin yang dapat mengotomatisasi proses pengujian kode program mahasiswa. Gambar 2 menunjukkan contoh halaman kuis praktikum pemrograman yang menampilkan soal, lembar jawaban untuk penulisan kode program, hasil pengujian kode program, serta hasil penilaian otomatis. Prototipe aplikasi diuji oleh mahasiswa yang mengikuti mata kuliah Perancangan Algoritma dan Pemrograman. Mahasiswa berpartisipasi dalam sesi demonstrasi, diskusi, dan evaluasi prototipe yang dikembangkan dalam kegiatan praktikum pemrograman. Gambar 3 adalah contoh halaman rekap nilai seluruh mahasiswa peserta eksperimen praktikum pemrograman.

Kegiatan ini bertujuan untuk memastikan bahwa prototipe yang disusun telah memenuhi kebutuhan pengguna yang telah diidentifikasi sebelumnya serta memvalidasi asumsi desain yang diterapkan pada tahap awal. Oleh sebab itu, evaluasi dilakukan berdasarkan *user requirements* yang telah dirumuskan sebelumnya. Rincian *user requirements* dan hasil pengujian atau evaluasi aplikasi dijabarkan pada Tabel 4.

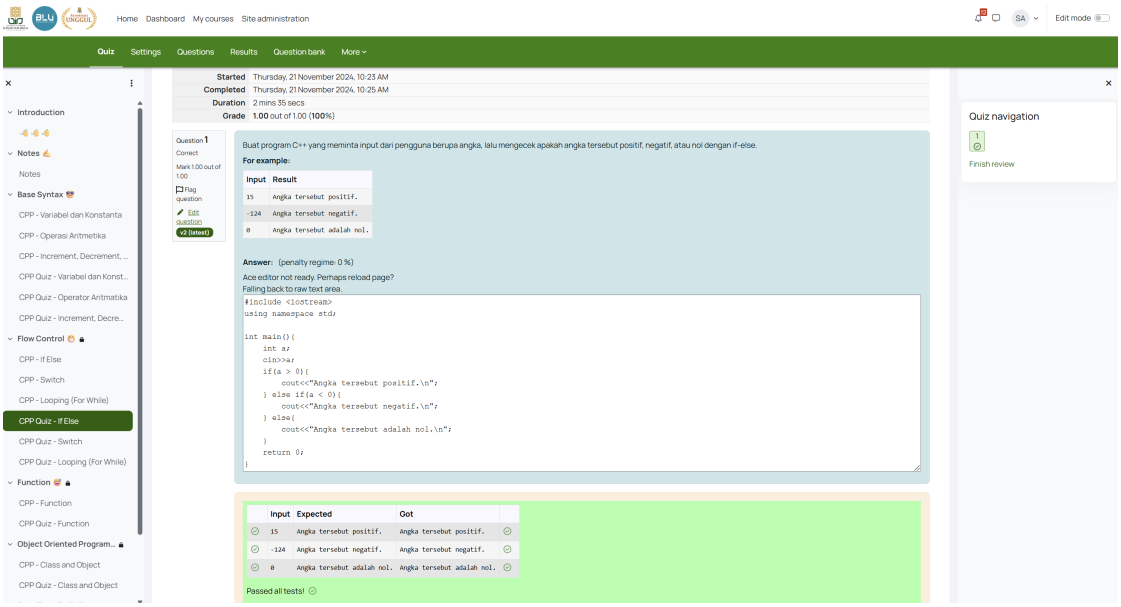
Tabel 4 Hasil Pengujian Aplikasi

No.	User Requirements	Ya/Tidak	Komentar
1	Halaman Login	Ya	*Requirement* telah terpenuhi
2	Halaman Dashboard Mahasiswa	Ya	*Requirement* telah terpenuhi
3	Halaman Dashboard Dosen	Ya	*Requirement* telah terpenuhi
4	Fitur Reset Kata Sandi	Ya	*Requirement* telah terpenuhi
5	Fitur Ubah Profil	Ya	*Requirement* telah terpenuhi
6	Navigasi Materi Praktikum	Ya	*Requirement* telah terpenuhi
7	Upload Jawaban Tugas Praktikum	Ya	*Requirement* telah terpenuhi
8	Fitur Penilaian Otomatis	Ya	*Requirement* telah terpenuhi
9	Feedback Langsung Hasil Praktikum	Ya	*Requirement* telah terpenuhi
10	Halaman Riwayat Tugas dan Nilai	Ya	*Requirement* telah terpenuhi
11	Fitur Diskusi dengan Dosen	Ya	*Requirement* telah terpenuhi
12	Fitur Statistik Nilai Mahasiswa	Ya	*Requirement* telah terpenuhi
13	Notifikasi Deadline Tugas	Ya	*Requirement* telah terpenuhi
14	Pengelolaan Konten	Ya	*Requirement* telah terpenuhi
15	Riwayat Aktifitas	Ya	*Requirement* telah terpenuhi
16	Keluar	Ya	*Requirement* telah terpenuhi

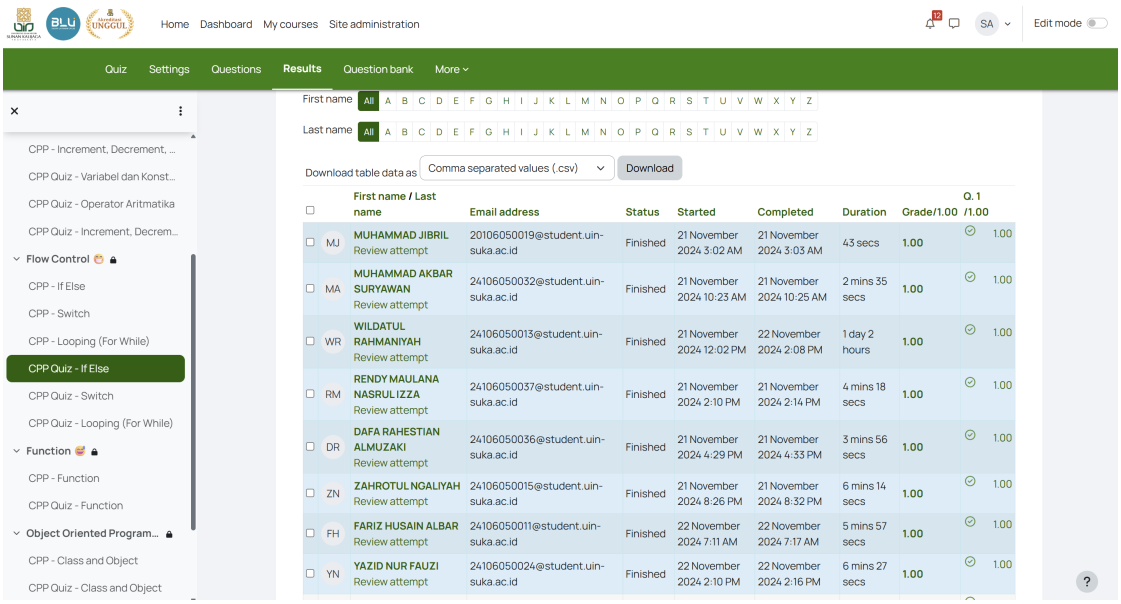
Dalam penghitungan jumlah sampel, penelitian ini menggunakan rumus Slovin seperti pada Pers. (1). Metode Slovin dipilih untuk menentukan jumlah responden yang representatif dengan mempertimbangkan populasi yang tersedia serta tingkat kesalahan yang diizinkan. Adapun rumus Slovin dinyatakan pada Pers. (1). Di mana n merupakan jumlah sampel yang diperlukan, *N* adalah jumlah populasi, dan *e* menunjukkan tingkat kesalahan (error margin) yang ditoleransi.

$$n = \frac{N}{1 + N(e^2)} \quad (1)$$

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



Gambar 2 Hasil Minimum Variable Product



Gambar 3 Hasil Penilaian Otomatis

Dalam penelitian ini, jumlah populasi (*N*) adalah 181 mahasiswa Program Studi Informatika angkatan 2023 dan 2024, dan tingkat kesalahan (*e*) ditetapkan sebesar 10% (0,1). Substitusi nilai-nilai tersebut ke dalam rumus menghasilkan perhitungan seperti pada Pers. (2). Hasil perhitungan menunjukkan bahwa jumlah sampel yang diperlukan adalah 64 responden. Namun, untuk menyesuaikan dengan kondisi kelas yang tersedia, jumlah sampel yang digunakan dalam penelitian ini ditetapkan menjadi 91 responden. Penyesuaian ini dilakukan untuk memastikan representasi yang lebih luas dari populasi yang ada sekaligus memaksimalkan keterlibatan mahasiswa yang berpartisipasi.

$$n = \frac{181}{1 + 181\,(0.1^2)} = \frac{181}{1 + 181\,(0.01)} = \frac{181}{1 + 1.81} = \frac{181}{2.81} \approx 64.41 \qquad (2)$$

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

Tabel 5 Hasil *Feedback*

Variabel	Kode	Indikator	*Corrected Item-Total Correlation*	Keterangan
Kemudahan (*Perceived Ease of Use*)	PEOU 1	Aplikasi Kalisa mudah untuk diakses.	.766	Valid
	PEOU 2	Saya merasa antarmuka aplikasi Kalisa sederhana dan mudah digunakan.	.744	Valid
	PEOU 3	Menavigasi dalam aplikasi Kalisa sangat mudah.	.748	Valid
	PEOU 4	Instruksi dalam menyelesaikan kursus di aplikasi Kalisa mudah untuk dipahami.	.746	Valid
	PEOU 5	Menjawab pertanyaan terkait pemrograman pada course di aplikasi Kalisa terasa intuitif.	.768	Valid
	PEOU 6	Mengirimkan dan meninjau jawaban saya melalui aplikasi Kalisa sangat mudah.	.691	Valid
Kebermanfaatan (*Perceived of Usefulness*)	POU 1	Penggunaan aplikasi Kalisa membantu saya memahami konsep pemrograman dengan lebih baik.	.804	Valid
	POU 2	Umpan balik dari aplikasi Kalisa membantu saya meningkatkan keterampilan pemrograman.	.817	Valid
	POU 3	Aplikasi Kalisa memudahkan saya dalam mempersiapkan praktikum pemrograman.	.772	Valid
	POU 4	Penggunaan aplikasi Kalisa meningkatkan pengalaman belajar saya dalam mata kuliah pemrograman.	.838	Valid
	POU 5	Aplikasi Kalisa membantu saya mengidentifikasi area yang perlu saya tingkatkan dalam pemrograman.	.768	Valid
	POU 6	Saya percaya penggunaan Aplikasi Kalisa meningkatkan kualitas penilaian saya.	.808	Valid
Penerimaan (*Acceptance of It*)	AOT 1	Saya merasa percaya diri menggunakan aplikasi Kalisa untuk penilaian pemrograman.	.764	Valid
	AOT 2	Saya puas dengan cara aplikasi Kalisa membantu saya dalam mata kuliah pemrograman.	.799	Valid
	AOT 3	Aplikasi Kalisa adalah alat yang membantu dalam proses belajar saya.	.815	Valid
	AOT 4	Saya bersedia mengandalkan aplikasi Kalisa untuk studi pemrograman saya.	.728	Valid
	AOT 5	Modul Aplikasi Kalisa akan menjadi bagian penting dari pendidikan pemrograman saya.	.789	Valid
	AOT 6	Saya percaya aplikasi Kalisa membuat proses penilaian menjadi adil dan transparan.	.696	Valid

Selanjutnya, hasil *feedback* perlu diuji validitas dan reliabilitasnya. Kriteria validitas yang digunakan mengikuti pedoman, yaitu suatu indikator dinyatakan valid jika memiliki koefisien

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

korelasi minimal 0,5 atau lebih (Mulyanto et al., 2020). Tabel 5 menunjukkan hasil uji validitas *feedback* dari mahasiswa terhadap aplikasi yang dikembangkan. Dapat dilihat pada Tabel 5 bahwa nilai koefisien korelasi setiap indikator lebih dari 0,5. Dengan demikian, dapat dikatakan bahwa setiap indikator valid. Selanjutnya, dilakukan penentuan tingkat reliabilitas instrumen melalui nilai *alfa Cronbach*. Instrumen yang memiliki nilai *Cronbach's Alpha* lebih dari atau sama dengan 0,7 dapat dinyatakan memiliki tingkat keandalan yang baik (Mulyanto et al., 2020). Hasil uji reliabilitas dapat dilihat pada Tabel 6.

Hasil uji reliabilitas yang ditunjukkan pada Tabel 6 mengungkapkan bahwa seluruh variabel dalam *Technology Acceptance Model* (TAM), termasuk Kemudahan (PEOU), Kebermanfaatan (POU) dan Penerimaan Teknologi (AOT), memiliki nilai *Cronbach's Alpha* di atas 0,7. Dengan demikian, dapat disimpulkan bahwa setiap variabel dalam penelitian ini telah memenuhi standar reliabilitas yang diperlukan, sehingga dapat diandalkan sebagai dasar pengambilan keputusan dan analisis lanjutan.

Transformasi digital melalui penerapan aplikasi Kalisa telah memberikan berbagai dampak pada proses pembelajaran dan penilaian praktikum pemrograman. Untuk menganalisis dampak ini, peneliti berfokus pada tiga variabel utama, yaitu kemudahan (PEOU), kebermanfaatan (POU), dan penerimaan (AOT). Berdasarkan hasil pengolahan data dalam tabel tersebut, persepsi pengguna terhadap aplikasi Kalisa diukur melalui tiga variabel utama, yaitu Kemudahan (PEOU), Kebermanfaatan (POU), serta Penerimaan (AOT).

Tabel 6 Hasil Uji Reliabilitas

Variabel	*Cronbatch Alpha*	Keterangan
PEOU (Kemudahan)	.916	Reliabel
POU (Kebermanfaatan)	.931	Reliabel
AOT (Penerimaan Teknologi)	.921	Reliabel

Variabel kemudahan (PEOU) dievaluasi melalui enam indikator pertanyaan. Merujuk pada Tabel 7, skor terendah variabel PEOU adalah 1, yang mengindikasikan bahwa terdapat responden yang merasa sangat tidak sepakat terhadap pernyataan tersebut. Di sisi lain, skor tertinggi mencapai 5, yang berarti sejumlah responden sangat mendukung pernyataan terkait kemudahan penggunaan aplikasi. Dengan nilai rata-rata sebesar 3,938, dapat disimpulkan bahwa mayoritas responden menyatakan setuju terhadap indikator-indikator PEOU dengan nilai interpretasi yang tinggi (antara 3,41 dan 4,20).

Tabel 7 Hasil Data TAM per Kategori

Variabel	Min.	Max.	Mean	Std.Deviation	Interpretasi
PEOU (Kemudahan)	1	5	3.938	.944	Tinggi
POU (Kebermanfaatan)	1	5	3.905	.936	Tinggi
AOT (Penerimaan Teknologi)	1	5	3.817	.907	Tinggi

Sementara itu, variabel kebermanfaatan (POU) juga diukur melalui enam indikator pertanyaan. Berdasarkan Tabel 7, nilai minimum POU adalah 1, yang menandakan adanya responden yang sangat tidak setuju terhadap suatu aspek kebermanfaatan. Namun, nilai maksimum yang mencapai 5 menunjukkan adanya kelompok responden yang sangat mendukung pernyataan terkait manfaat aplikasi. Rata-rata POU sebesar 3,905 mengisyaratkan bahwa sebagian besar responden memandang Kalisa sebagai sarana yang bermanfaat, baik dalam mendukung pemahaman konsep pemrograman dengan nilai interpretasi yang tinggi (antara 3,41 dan 4,20).

Terakhir, variabel penerimaan (AOT) juga diukur menggunakan enam indikator. Menurut Tabel 7, nilai minimum AOT adalah 1, yang mengindikasikan adanya responden yang sangat tidak setuju terhadap salah satu pernyataan yang diajukan. Nilai maksimum AOT kembali mencapai 5, yang menunjukkan bahwa ada pula responden yang menunjukkan penerimaan penuh terhadap aplikasi ini. Dengan rata-rata 3,817, dapat disimpulkan bahwa mayoritas responden secara

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

umum menerima Kalisa sebagai alat yang sesuai untuk mendukung proses pembelajaran dan penilaian dalam mata kuliah pemrograman dengan nilai interpretasi yang tinggi (antara 3,41 dan 4,20). Hasil ini menunjukkan bahwa Kalisa berpotensi diadopsi secara luas dan dimanfaatkan secara efektif dalam lingkungan akademik.

Perhitungan dalam penelitian ini didasarkan pada pengukuran tingkat kemudahan (PEOU), kebermanfaatan (POU), dan penerimaan teknologi (AOT) terhadap aplikasi Kalisa. Penghitungan dilakukan menggunakan dua komponen utama, yaitu ΣSK dan ΣSH, yang kemudian diolah menjadi persentase untuk memberikan gambaran tentang tingkat penerimaan aplikasi. ΣSH adalah total skor aktual yang diperoleh dari hasil kuesioner responden seperti pada Pers. (4). Sedangkan ΣSK adalah nilai skor tertinggi yang dapat dicapai pada setiap variabel. ΣSK dihitung dengan rumus Pers. (3). Nilai *Technology Acceptance Model* dihitung dengan menggunakan Pers. (5). Hasil perhitungan Pers. (3), (4), dan (5) disajikan pada Tabel 8. Nilai ini mencerminkan jawaban aktual yang diberikan responden terhadap pertanyaan pada variabel yang diukur.

$$\Sigma SK = Skor\ tertinggi\ kriterium \times Total\ jumlah\ pertanyaan \times Jumlah\ responden \quad (3)$$

$$\Sigma SH = Jumlah\ skor\ aktual\ seluruh\ responden \quad (4)$$

$$Persentase = \frac{\Sigma SH}{\Sigma SK} \times 100\% \quad (5)$$

Tabel 8 Hasil Perhitungan TAM

Variabel	Jumlah Pertanyaan	ΣSK	ΣSH	Persentase	Interpretasi
PEOU (Kemudahan)	6	2730	2150	78.75%	Tinggi
POU (Kebermanfaatan)	6	2730	2132	78.09%	Tinggi
AOT (Penerimaan)	6	2730	2084	76.33%	Tinggi

4. KESIMPULAN

Penggunaan sistem penilaian praktikum pemrograman secara konvensional sering menimbulkan kendala dalam efisiensi dan objektivitas penilaian. Hasil penelitian menunjukkan bahwa aplikasi Kalisa memberikan dampak yang signifikan (tinggi) terhadap proses pembelajaran, terutama dalam meningkatkan efisiensi waktu penilaian, objektivitas dan transparansi. Melalui fitur otomatisasi seperti pengecekan kode program menggunakan CodeRunner dan integrasi server JOBE, mahasiswa dapat langsung mengetahui letak kesalahan pada kode, sementara dosen tidak perlu memeriksa secara manual. Dengan demikian, waktu dosen dapat digunakan untuk berfokus pada aspek materi yang lebih mendalam, seperti diskusi pemecahan masalah dan pendampingan secara individual.

Hasil evaluasi menggunakan metode TAM menunjukkan bahwa aplikasi Kalisa memiliki tingkat penerimaan yang baik (tinggi) oleh mahasiswa maupun dosen. Variabel kemudahan (PEOU/*Perceived Ease of Use*) dengan nilai persentase 78,75%: aplikasi dinilai memiliki antarmuka yang sederhana dan mudah digunakan, meskipun terdapat masukan tentang kekakuan sistem yang hanya menerima kode tanpa pengecekan kesalahan. Sementara itu, variabel kebermanfaatan (POU/*Perceived of Usefulness*) dengan persentase 78,09% menunjukkan bahwa aplikasi terbukti membantu mahasiswa memahami konsep pemrograman melalui umpan balik otomatis dan fitur evaluasi mandiri, dengan tingkat penerimaan yang baik (tinggi). Variabel penerimaan (AOT/*Acceptance of It*) dengan persentase 76,33% menunjukkan tingkat penerimaan yang baik (tinggi), meskipun masih terdapat masukan dari mahasiswa untuk meningkatkan fleksibilitas dan kinerja aplikasi.

Meski demikian, ada kekhawatiran bahwa penilaian otomatis yang terlalu kaku dapat mengurangi fleksibilitas mahasiswa dalam bereksperimen dengan sintaks. Kekakuan sistem mengurangi fleksibilitas dalam pengerjaan praktikum pemrograman. Namun, sebagian pandangan melihat

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

kekakuan sistem ini justru berpotensi menjadi nilai tambah untuk membiasakan mahasiswa menulis sintaks yang sesuai dengan kebutuhan compiler di dunia kerja.

Beberapa masukan juga menyoroti pentingnya panduan dan pelatihan awal bagi pengguna baru, serta perlunya integrasi dengan database kampus agar penerapan sistem dapat berjalan lancar tanpa harus melakukan migrasi data. Transformasi digital ini diharapkan tidak hanya menghemat waktu dan meningkatkan transparansi penilaian, tetapi juga memperkaya proses pembelajaran.

Dengan efisiensi waktu yang lebih baik, dosen dapat merancang tugas yang lebih variatif, memberikan bimbingan yang lebih mendalam, serta memantau kemampuan mahasiswa secara lebih komprehensif. Pada akhirnya, pemanfaatan teknologi dalam penilaian praktikum pemrograman dianggap penting dan relevan untuk menghadapi tuntutan pendidikan, selama tetap disertai penyesuaian budaya kerja, infrastruktur yang solid, serta pendekatan yang seimbang antara penilaian otomatis dan penilaian manual.

DAFTAR PUSTAKA

Aarlien, D., & Colomo-Palacios, R. (2020). *Lean UX: A Systematic Literature Review* (pp. 500–510). Springer. https://doi.org/10.1007/978-3-030-58817-5_37

Al-Adwan, A. S., Li, N., Al-Adwan, A., Abbasi, G. A., Albelbisi, N. A., & Habibi, A. (2023). "Extending the Technology Acceptance Model (TAM) to Predict University Students' Intentions to Use Metaverse-Based Learning Platforms". *Education and Information Technologies*, *28*(11), 15381–15413. https://doi.org/10.1007/s10639-023-11816-3

Arikarani, Y., & Amirudin, M. F. (2021). Pemanfaatan Media dan Teknologi Digital dalam Mengatasi Masalah Pembelajaran di Masa Pandemi. *Edification Journal: Pendidikan Agama Islam*, *4*(1), 93–116. https://doi.org/10.37092/ej.v4i1.296

Athaya, H., Nadir, R. D. A., Indra Sensuse, D., Kautsarina, K., & Suryono, R. R. (2021). Moodle Implementation for E-Learning: A Systematic Review. *6th International Conference on Sustainable Information Engineering and Technology 2021*, 106–112. https://doi.org/10.1145/3479645.3479646

Bangsawan, G. (2023). Kebijakan Akselerasi Transformasi Digital di Indonesia: Peluang dan Tantangan untuk Pengembangan Ekonomi Kreatif. *Jurnal Studi Kebijakan Publik*, *2*(1), 27–40. https://doi.org/10.21787/jskp.2.2023.27-40

Cipriano, B. P., Fachada, N., & Alves, P. (2022). Drop Project: An Automatic Assessment Tool for Programming Assignments. *SoftwareX*, *18*, 101079. https://doi.org/10.1016/j.softx.2022.101079

Croft, D., & England, M. (2020). Computing with CodeRunner at Coventry University. *Proceedings of the 4th Conference on Computing Education Practice 2020*, 1–4. https://doi.org/10.1145/3372356.3372357

Gamage, S. H. P. W., Ayres, J. R., & Behrend, M. B. (2022). A Systematic Review on Trends in Using Moodle for Teaching and Learning. *International Journal of STEM Education*, *9*(1), Article ID: 9. https://doi.org/10.1186/s40594-021-00323-x

Goodfellow, M., Abel, A., Liaskos, K., & Levine, J. (2024). Automated Marking in Undergraduate Programming Classes. *Proceedings of the 8th Conference on Computing Education Practice*, 13–16. https://doi.org/10.1145/3633053.3633060

Hamukti, W., Andrawina, L., & Suwarsono, L. W. (2018). Analisis Beban Kerja Dosen Bidang Pendidikan dan Penunjang Menggunakan Metode Knowledge Conversion 5C-4C. *JISI: Jurnal Integrasi Sistem Industri*, *4*(2), 73–84. https://doi.org/10.24853/jisi.4.2.73-84

Liikkanen, L. A., Kilpiö, H., Svan, L., & Hiltunen, M. (2014). Lean UX. *Proceedings of the 8th Nordic Conference on Human-Computer Interaction: Fun, Fast, Foundational*, 1095–1100. https://doi.org/10.1145/2639189.2670285

Marangunić, N., & Granić, A. (2015). Technology Acceptance Model: A Literature Review from 1986 to 2013. *Universal Access in the Information Society*, *14*(1), 81–95. https://doi.org/10.1007/s10209-014-0348-1

Maylawati, D. S., Kumar, Y. J., Kasmin, F., & Ramdhani, M. A. (2024). Deep Sequential Pattern Mining for Readability Enhancement of Indonesian Summarization. *International Journal of*

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

Electrical and Computer Engineering (IJECE), *14*(1), 782. https://doi.org/10.11591/ijece.v14i1.pp782-795

Messer, M., Brown, N. C. C., Kölling, M., & Shi, M. (2024). Automated Grading and Feedback Tools for Programming Education: A Systematic Review. *ACM Transactions on Computing Education*, *24*(1), 1–43. https://doi.org/10.1145/3636515

Mulyanto, A., Sumarsono, S., Niyartama, T. F., & Syaka, A. K. (2020). Penerapan Technology Acceptance Model (TAM) dalam Pengujian Model Penerimaan Aplikasi MasjidLink. *Semesta Teknika*, *23*(1), 27–38. https://doi.org/10.18196/st.231253

Pringuet, P., Friel, A., & Vande Wiele, P. (2021). CodeRunner: A Case Study of the Transition to Online Learning of a Java Programming Course. *SSRN Electronic Journal*, 1–10. https://doi.org/10.2139/ssrn.3874475

Salama, R., Uzunboylu, H., & Alkaddah, B. (2020). Distance Learning System, Learning Programming Languages by Using Mobile Applications. *New Trends and Issues Proceedings on Humanities and Social Sciences*, *7*(2), 23–47. https://doi.org/10.18844/prosoc.v7i2.5015

Teo, T., Zhou, M., Fan, A. C. W., & Huang, F. (2019). Factors that Influence University Students' Intention to Use Moodle: A Study in Macau. *Educational Technology Research and Development*, *67*(3), 749–766. https://doi.org/10.1007/s11423-019-09650-x

Velayani, M. R., Nuruzzaman, M. T., Fatwanto, A., & Sugiantoro, B. (2023). Evaluasi Penerimaan Masyarakat terhadap Aplikasi Telemedicine pada Masa Pandemi COVID-19. *JISKA (Jurnal Informatika Sunan Kalijaga)*, *8*(2), 140–153. https://doi.org/10.14421/jiska.2023.8.2.140-153

Wonoseto, M. G., Kusuma, M., & Millatina, D. (2025). Data Center Online Compiler untuk Membangun Aplikasi Evaluasi Praktikum Pemrograman Berdasarkan Prinsip Maudhûiyyah, 'Adula, dan Istimrâr. *Prosiding Konferensi Integrasi Interkoneksi Islam dan Sains (KIIIS) Edisi 6*, *6*(1), 207–214. https://ejournal.uin-suka.ac.id/saintek/kiiis/article/view/4800

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