https://research.adra.ac.id /index.php/ijen/
P - ISSN: 2988-1579
E - ISSN: 2988-0092

Narratives of Performance Assessment-Based Competency Testing for Measuring Students' Programming Competencies

Erwinsyah Simanungkalit[1], Jenny Sari Taringan[2], Mardhiatul Husna[3], Djames Siahaan[4]
[1]Politeknik Negeri Medan, Indonesia
[2]Politeknik Negeri Medan, Indonesia
[3]Politeknik Negeri Medan, Indonesia
[4]Politeknik Negeri Medan, Indonesia

ABSTRACT

Background. Programming competence is a fundamental skill for students in information technology education. Despite its importance, assessment practices continue to rely predominantly on theoretical examinations that inadequately represent students' practical abilities to solve authentic programming problems.

Purpose. This study aims to explore the narratives of performance assessment-based competency testing for measuring students' programming competencies through an authentic and comprehensive assessment model that captures both learning processes and performance outcomes.

Method. This study employed a quantitative approach using a quasi-experimental design supported by a literature review. Performance assessment narratives were developed from students' competency test results using a structured rubric encompassing four dimensions: algorithm planning, code implementation, debugging, and program optimization.

Results. The findings reveal narratives of students' programming competency development that are more comprehensive than those generated through conventional assessment methods. The performance assessment model demonstrated greater accuracy in evaluating authentic programming competencies. Students achieved the highest performance in code implementation, whereas debugging and program optimization emerged as the primary challenges. Furthermore, the assessment narratives provided deeper insights into students' critical thinking, problem-solving abilities, creativity, and the progression of their programming skills.

Conclusion. he narratives generated through performance assessment-based competency testing offer an authentic and effective evaluation framework for measuring students' programming competencies. Beyond providing reliable evidence of learning outcomes, these narratives deliver meaningful feedback that supports continuous competency development and better prepares students to meet the professional demands of the information technology industry.

KEYWORDS

Authentic Assessment, Computational Skills, Higher Education

Citation: Simanungkalit, E., Taringan, S, R., Husna, M & Siahan, D. (2026). Performance Assessment-Based Competency Test Model to Measure Students' Programming Competencies. *International Journal of Educational Narrative*, *4*(3), 988–1000. https://doi.org/10.70177/ijen.v4i3.3510

Correspondence:

Erwinsyah Simanungkalit,
erwinsyahsimanungkalit@polmed.ac.id

Received: January 12, 2026

Accepted: March 10, 2026

Published: June 28, 2026

INTRODUCTION

The rapid advancement of information technology demands that every individual, especially students, master adequate digital competencies. One key skill is programming, as the professional world demands not only theoretical understanding but also the ability to apply

concepts practically to solve real-world problems. Universities have a strategic role in preparing students to compete and adapt to the complexities of technology-based work environments.

This situation emphasizes the importance of developing effective evaluation strategies, so that programming skill mastery can be measured accurately and objectively. With the right evaluation approach, lecturers can comprehensively assess students' abilities, including critical thinking skills, problem-solving skills, and the ability to implement solutions in real-world contexts (Wahyudi et al., 2025). This type of evaluation also supports a more reflective learning process, increases student motivation, and ensures that acquired competencies are relevant to the demands of the workplace. Furthermore, practice-based skill measurement allows for the identification of areas that need strengthening, allowing for appropriate improvements to the curriculum and teaching methods. Thus, the development of authentic assessment methods is a crucial element in modern higher education to prepare competent and adaptable graduates. This strategy not only improves the quality of learning but also strengthens the link between theory and practice, preparing students to face professional challenges in a more applicable and relevant way (Saputra & Juniawan, 2025).

Although programming mastery is becoming an increasingly vital competency, the learning process in many universities still tends to focus on theoretical aspects and memorization of basic concepts. Student competency assessment is generally conducted through written exams or multiple-choice tests, which only measure declarative knowledge without reflecting real-world practical abilities. As a result, students' skills in developing algorithms, writing code, debugging, and optimizing programs are not fully monitored. This condition makes evaluation less relevant to the needs of the professional world, because students are not tested on their ability to apply concepts to solve real-world problems (Ashari et al., 2025). Thus, conventional assessment models are less able to comprehensively describe the level of programming skill mastery. Evaluation approaches that emphasize practice and critical thinking processes are essential to closing this gap. Assessments that are solely theory-oriented are insufficient to prepare students to face the challenges of a dynamic technological world. Therefore, a more authentic and comprehensive competency measurement strategy is needed so that evaluation results better reflect students' practical abilities. The integration of performance-based assessment methods is expected to increase the validity of measurements and the relevance of programming education to the needs of industry and contemporary technology (Fitri & Habibie, 2025).

The gap between theoretical understanding and practical application is a crucial issue in programming education. Although students are able to master basic concepts, they often struggle when trying to apply them to solve real-world problems. This situation emphasizes the need for an evaluation model that assesses not only the final results, but also students' critical thinking processes, problem-solving skills, and creativity in programming. This type of assessment is expected to improve the quality of learning and prepare students to face professional demands in a more applicable way. Performance assessment, or performance-based assessment, has emerged as an approach that emphasizes competency evaluation through real-world, authentic tasks (Dewanto et al., 2025). In the programming context, this method provides students with the opportunity to demonstrate their skills in developing algorithms, writing code, testing programs, and solving practical problems. This approach differs from traditional evaluations because it assesses not only the final product but also the student's thought process, problem-solving strategies, and ability to adapt solutions to real-world situations. Thus, performance assessment offers a more holistic and relevant assessment method for measuring students' programming competencies comprehensively (Salsabila et al., n.d.).

The development of a performance assessment-based competency testing model is expected to produce more accurate and representative data regarding student abilities. This model is designed to allow lecturers to assess various dimensions of programming skills, from algorithm planning and code implementation to debugging and program optimization. Furthermore, the implementation of this model allows for constructive feedback, allowing students to reflect on their abilities and improve their mastery of practical skills (Putri et al., 2025). This study uses a quantitative-descriptive approach combined with a literature review to design a comprehensive performance-based evaluation instrument. Data were obtained through model trials on Informatics Engineering students, with assessments using a rubric that measures the four main dimensions of programming skills. Furthermore, data analysis was conducted using descriptive statistics to map the distribution of student abilities and correlation analysis to assess the validity and reliability of the instrument. With this approach, the model can be applied systematically and consistently, providing a comprehensive picture of student competencies, and supporting the development of more effective, real-life skills-based learning strategies (Mandowen et al., 2025).

The research findings revealed that performance-based assessments were able to assess students' abilities with a higher degree of accuracy than traditional evaluation methods. The analysis showed that students were relatively more proficient in code implementation, while debugging and program optimization skills still needed improvement. This confirms that performance-based evaluations not only assess final results but also examine students' critical thinking processes, problem-solving abilities, and creativity in completing programming assignments. Therefore, this performance assessment model can be used as an effective evaluation strategy to strengthen the quality of programming learning. Furthermore, this approach provides constructive feedback for students to continuously develop practical skills (Tanjung et al., 2025). The application of this model also helps lecturers identify areas for improvement and focus on areas of lesser mastery. Overall, this strategy facilitates the development of students' comprehensive competencies. Beyond technical aspects, this evaluation encourages students to think more analytically and creatively. In other words, performance assessment combines practical competency measurement with reflective learning oriented toward mastering real-world skills. This approach is expected to prepare students to face professional challenges in the world of information technology in a more applicable and relevant way (Wonoseto et al., 2025).

Several previous studies are relevant to this study, although their focus and scope differ. Smith (2018) highlighted project-based evaluation in programming learning, but this study was limited to students' ability to implement code. Li and Chen (2020) discussed the use of rubric-based assessment, but their focus was on only one skill aspect: debugging. Furthermore, Ahmad (2021) explored the application of performance assessment to assess students' practical skills, but this study did not specifically measure program optimization skills. Unlike the three previous studies, this study designed a more comprehensive competency testing model by assessing four key dimensions of programming: algorithm planning, code implementation, debugging, and program optimization. With this approach, the evaluation not only assesses the final results but also students' thinking processes, problem-solving, and creativity. This model is expected to provide a more comprehensive picture of students' programming skills mastery while offering opportunities for improvement and strengthening learning strategies at the tertiary level. Thus, this research makes a more comprehensive contribution to the development of authentic and applicable programming competency assessment methods.

RESEARCH METHODOLOGY

This research applies a quantitative-descriptive approach combined with experimental methods and literature review, with the aim of designing and implementing a performance-based evaluation instrument for students. This approach allows for authentic and comprehensive assessment of programming competency, as it focuses on students' ability to complete real-world tasks, not just theoretical mastery (Pare et al., 2026). Literature studies were used to build a conceptual framework and develop assessment rubric references, ensuring that the developed instrument has both scientific validity and practical relevance to the programming learning context in higher education. Research data was obtained through model trials on Informatics Engineering students, followed by assessment using a rubric that evaluates four main dimensions: algorithm planning, code implementation, debugging, and program optimization. With this approach, the assessment not only assesses the final result but also the students' thinking processes, problem-solving skills, and creativity in programming. The applied research flow facilitates visualization of the research stages and ensures that each step supports the overall goal of performance-based competency evaluation (Alfiyanti et al., 2025).

Figure 1. Research process

Data analysis was conducted using descriptive statistics to map the distribution of student abilities across each assessed programming dimension. Correlation analysis was also applied to assess the validity and reliability of the instrument, ensuring that the competency testing model delivers consistent and accurate results. This approach allows for a comprehensive evaluation of students' cognitive and practical skills, including critical thinking, problem-solving, and creativity in program development. This method allows for a more comprehensive and authentic assessment model development, while also providing constructive feedback for both students and lecturers (Bhuwaneswari et al., 2025)

The analysis results not only show the level of student competency mastery, but also serve as a reference for refining learning strategies, designing more effective practicum materials, and preparing students to face professional challenges in the field of information technology. Thus, the application of a quantitative-descriptive approach supported by experiments and literature studies has proven effective in improving the quality of evaluation and programming teaching practices, while simultaneously encouraging the improvement of student abilities in a holistic manner that is relevant to industry needs (Permana et al., 2025).

RESULT AND DISCUSSION

Measuring students' programming skills using a performance assessment-based testing model provides a more comprehensive understanding than traditional evaluation methods. The data analysis shows the distribution of students' competencies across four key programming dimensions: algorithm planning, code implementation, debugging, and program optimization. This model not only assesses the final outcome of programming assignments but also evaluates students' critical thinking processes, problem-solving abilities, and creativity in compiling and developing programs. Initial findings indicate that most students have the strongest skills in code implementation, while debugging and program optimization still require more attention through intensive practice and additional guidance. Therefore, implementing a performance assessment model allows for a more authentic and comprehensive evaluation, while providing a solid foundation for developing more effective and tailored learning strategies. This method allows instructors to gain a detailed understanding of students' strengths and weaknesses in each programming dimension and provides useful information for designing more targeted teaching interventions. Evaluation conducted using this approach not only assesses technical skills but also emphasizes analytical and creative thinking processes, which are essential competencies for mastering modern programming. In this way, the performance assessment-based competency testing model becomes a relevant tool for improving the quality of programming learning at the tertiary level.

Furthermore, the implementation of a performance-based assessment rubric enabled researchers to obtain valid and consistent quantitative data. Scores on each dimension were evaluated using descriptive statistics, while correlation analysis between dimensions was used to ensure the consistency and reliability of the model. These findings provide a deeper understanding of students' competency mastery levels and identify areas that require reinforcement in programming learning at universities. With the data obtained, instructors can design more effective learning intervention strategies, adjust lab materials, and pay special attention to programming aspects that students lack mastery, thus making the learning process more focused, relevant, and practically beneficial.

Table 1. Distribution of student scores on the algorithm planning and code implementation dimensions

Student	Algorithm Planning (0-100)	Code Implementation (0-100)
M1	78	88
M2	82	91
M3	75	85
M4	80	89
M5	77	87

The table indicates that students' scores on code implementation were consistently higher than those on algorithm planning. This pattern suggests that students tend to rely on procedural and trial-and-error approaches when completing programming tasks, rather than engaging in structured problem decomposition at the planning stage. The relatively lower performance in algorithm planning reflects a gap in students' ability to translate problems into systematic logical steps before coding, which is a fundamental aspect of computational thinking. This condition may be influenced by learning practices that emphasize immediate coding activities without sufficient reinforcement of pre-coding design processes such as flowchart development, pseudocode construction, or problem abstraction.

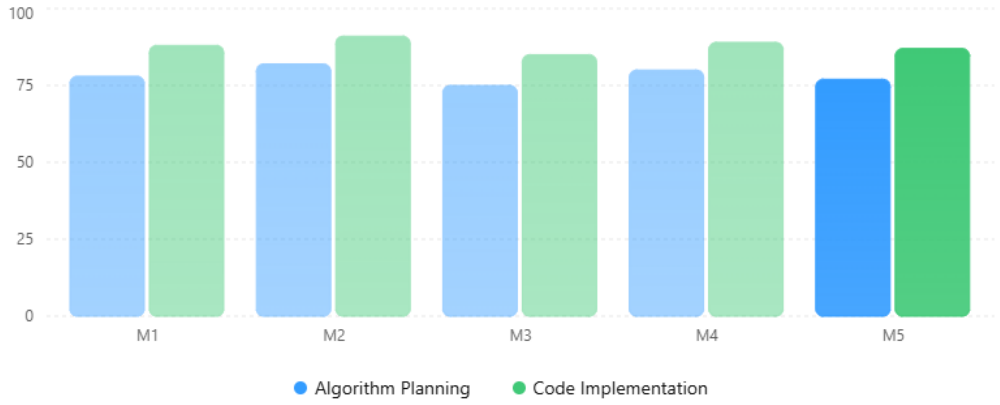

Figure 2. Distribution of student scores on algorithm planning and code implementation

The chart illustrates the performance of five students (M1–M5) in two dimensions: Algorithm Planning and Code Implementation. Overall, students achieved higher scores in Code Implementation than in Algorithm Planning. Student M2 obtained the highest scores in both dimensions, with 82 points in Algorithm Planning and 91 points in Code Implementation. Meanwhile, M3 recorded the lowest scores, achieving 75 and 85 points, respectively. The results indicate that students generally demonstrated stronger abilities in implementing code than in planning algorithms, suggesting a relatively consistent level of practical programming proficiency across the group.

From a cognitive perspective, code implementation is often associated with lower to mid-level thinking skills, where students apply known syntax and patterns. In contrast, algorithm planning requires higher-order thinking skills, including analysis, abstraction, and logical structuring. The disparity between these two dimensions indicates that students have not yet fully developed the ability to think algorithmically, even though they can execute code effectively. This finding is consistent with previous studies that highlight that novice programmers often struggle more with problem formulation than with code execution.

Furthermore, this result has important implications for instructional design. It indicates that programming instruction should not only focus on coding practice but also systematically integrate activities that strengthen algorithmic thinking. Structured interventions such as guided problem decomposition, iterative design tasks, and explicit teaching of algorithm design strategies are needed to bridge this gap. By reinforcing the planning stage, students can develop more efficient, structured, and scalable programming solutions, which ultimately improves both the quality of code implementation and overall problem-solving performance.

Table 2. Distribution of student scores on the debugging dimension

Student	Debugging (0-100)
M1	65
M2	70
M3	60
M4	68
M5	62

The table shows that students' debugging skills tend to be lower compared to other programming dimensions. These results emphasize the importance of providing additional, focused training to hone students' skills in detecting, analyzing, and correcting code errors in a structured and systematic manner. A learning approach geared toward improving debugging skills can help students understand common error patterns, improve problem-solving accuracy, and build confidence in facing complex programming challenges. Therefore, consistent and targeted practice is an important strategy for strengthening students' practical competencies in the program development process. This effort will also support the achievement of more comprehensive programming mastery, enabling students not only to become proficient in writing code but also to optimize the quality and effectiveness of the resulting code. Significantly improving debugging skills will strengthen the foundation of programming competencies, enabling students to approach real-world programming projects or assignments with greater confidence, and enhancing their readiness to face professional challenges in the world of information technology. Therefore, focusing on debugging practice is a strategic step in enhancing the programming learning process, while also supporting the development of students' critical thinking and problem-solving skills.

Additionally, the relatively low debugging scores indicate that students face difficulties in higher-order cognitive processes, particularly in identifying the root causes of errors and applying systematic correction strategies. This suggests that students may still rely on surface-level approaches, such as guessing or repetitive code modification, rather than using structured debugging techniques. From an instructional perspective, this finding highlights a gap in the learning process, where debugging is often treated as a secondary skill rather than an integral part of programming competence. Consequently, there is a need to integrate explicit debugging instruction, including error analysis frameworks, step-by-step tracing methods, and reflective evaluation of coding mistakes. Strengthening this aspect will not only improve students' accuracy in resolving errors but also enhance their analytical thinking and ability to develop more reliable and efficient programs.

Table 3. Distribution of student scores on the program optimization dimension

Student	Program Optimization (0-100)
M1	58
M2	61
M3	55
M4	60
M5	57

The analysis indicates that the program optimization dimension is the area that requires the most attention and further development. Students face challenges in designing effective code and

maximizing the efficient use of resources, making this capability relatively weak compared to other dimensions. This situation indicates the need for specific interventions, such as additional practice, best practice guides, and direct guidance from instructors, to enable students to improve their program optimization skills. Improving competency in this aspect is not only crucial for completing practical assignments but also for preparing students to face the professional demands of the programming world. By focusing on strengthening optimization, students can learn to design more concise, efficient, and maintainable code, while simultaneously developing analytical and strategic thinking skills. Furthermore, improving optimization skills also encourages creativity in finding optimal programming solutions, strengthens problem-solving abilities, and improves the overall quality of program output. This additional guidance can be provided through practical modules, real-world case studies, and structured performance-based evaluations. With this support, it is hoped that students will be able to balance code implementation skills with optimization, resulting in a more comprehensive and applicable mastery of programming competencies. These results confirm that performance assessment-based evaluations can identify specific areas of weakness and serve as the basis for more targeted learning improvement strategies.

The comprehensive analysis shows that the performance assessment-based testing model is able to assess student competency with a higher level of accuracy than traditional examination methods. This approach emphasizes students' strengths in code implementation skills while identifying areas that still need development, such as debugging and program optimization. With comprehensive capability mapping, lecturers can provide constructive feedback and design learning strategies that are more targeted and tailored to student needs. Overall, the implementation of the performance assessment model has proven effective in measuring programming skills authentically and comprehensively, which are not only relevant to the academic context, but also to the professional demands of the information technology world. This model assesses not only the final results, but also students' critical thinking processes, creativity, and problem-solving abilities, thus becoming an evaluation tool that plays a role in improving the quality of learning, strengthening practical understanding, and preparing students to face career challenges in the information technology field in a more applicable and realistic manner.

The study results indicate that the implementation of a performance assessment-based competency testing model can provide a more comprehensive evaluation of students' programming abilities compared to conventional examination methods. Performance-based assessment allows for authentic measurement, encompassing not only theoretical aspects but also students' practical skills in solving real-world programming problems. This model highlights students' strengths in code implementation while also revealing areas that still need to be strengthened, such as debugging and program optimization skills. With this approach, the assessment focuses not only on the final results but also examines students' critical thinking processes, creativity, and problem-solving abilities.(Gavriel, 2025)The results provide a holistic picture of individual competencies, which can be used to design more effective learning strategies. Furthermore, this evaluation supports lecturers in providing constructive feedback, tailoring lab materials to student needs, and encouraging the development of applicable technical skills. The implementation of performance assessment also helps students understand the relationship between concepts and practice, thereby increasing the relevance of learning to professional challenges. Thus, this model serves as a comprehensive, authentic, and relevant evaluation tool that measures not only knowledge but also critical and creative thinking skills in a programming context. This approach supports active learning and student preparation for the information technology workforce. This performance-based evaluation

also serves as a reference for improving the quality of the curriculum and teaching methods in related study programs.(Wipayana, 2025).

The comprehensive analysis indicates that the implementation of a performance assessment model can more accurately assess student competency than traditional examination methods. Students tend to demonstrate the strongest abilities in code implementation, reflecting their practical skills in executing algorithms accurately and efficiently. On the other hand, debugging and program optimization remain challenging areas for some students, requiring additional attention in planning learning strategies. By using this performance-based approach, instructors can provide specific and focused feedback and guide students to strengthen their technical skills and programming strategy abilities. This model not only assesses the final results but also allows for evaluation of students' thought processes in developing and executing programs.(Setiawan & Wandri, 2025). In addition, the performance assessment model encourages students to simultaneously develop logic, creativity, and problem-solving skills. The use of this instrument helps identify individual strengths and weaknesses, allowing learning to be more effectively directed. Overall, this approach provides a comprehensive picture of students' mastery of programming competencies and serves as a basis for improving the quality of teaching at the university level. This authentic evaluation ensures that students not only master theoretical concepts but also apply them in practical contexts relevant to the demands of the world of information technology.(Septiyanti et al., n.d.).

Furthermore, the implementation of a performance assessment model emphasizes that evaluations do not solely assess the final result, but also highlight students' thought processes when facing programming problems. Students are encouraged to hone their logical, creative, and analytical skills as they develop code. This approach emphasizes active and reflective learning, making each step in programming a meaningful learning experience. By assessing both the process and the product, students gain a deeper understanding of concepts.(Akadji & Eka, 2025). Furthermore, this model enhances their ability to systematically design, test, and modify programs. Learning becomes more contextual as students work with assignments that mimic real-world challenges in the professional world. As a result, they not only master programming techniques but also prepare themselves for complex situations in the information technology industry. This performance-based evaluation facilitates the development of authentic problem-solving skills. Furthermore, students learn to harmoniously integrate theory and practice. This model also supports the development of soft skills, such as perseverance, focus, and analytical decision-making. Thus, performance assessment provides a comprehensive and applicable learning experience, shaping students into technically competent and prepared to face professional demands.(Putra, 2025).

The results of this study indicate that the implementation of performance-based competency testing can serve as an effective evaluation strategy in the context of programming learning. This approach allows educators to identify the strengths and weaknesses of each student, allowing the learning process to be designed more personally and adaptively to suit individual needs. The integration of performance-based assessment into the curriculum opens up opportunities for lecturers to develop more applicable practicum activities, simultaneously combining theory and practice. Furthermore, this approach encourages students to develop critical thinking skills, problem-solving strategies, and analytical skills throughout the learning process.(Erlena, 2025)Unlike traditional examination methods that only assess conceptual understanding, performance-based evaluation captures students' creative processes in solving programming challenges. Thus, the implementation of this model not only assesses the final results but also considers the quality of students' thinking processes and practical skills. This strategy also allows for more targeted and constructive feedback, which can increase students' motivation and readiness

to face professional challenges. Overall, the performance-based competency assessment model makes a significant contribution to the development of a relevant and applicable curriculum. This approach emphasizes the importance of comprehensive, authentic evaluation, which measures not only material mastery but also students' thinking process competencies, creativity, and problem-solving in an integral manner.(Sariana & Billy, 2025).

Furthermore, research findings indicate that implementing performance assessment can increase students' active participation in the learning process, as they are faced with challenges that reflect real-world professional conditions. By assessing aspects of algorithm planning, code implementation, debugging, and program optimization, this model is able to provide a comprehensive picture of students' programming competencies. The long-term impact of using this model is seen in improving the quality of learning and preparing students to face professional demands with relevant technical skills and strategies in the information technology industry.(Bhuana 2025). Furthermore, this approach encourages the development of a skills-based curriculum that is authentic, applicable, and aligned with modern needs. Students not only learn theory but also hone practical skills that can be applied in real life. This type of performance assessment fosters awareness of the importance of process quality, not just the end result. Thus, performance assessment provides added value in developing holistic competencies, including problem-solving skills, critical thinking, and creativity in programming.(Nisa et al., 2026)This model also emphasizes the importance of continuous evaluation to identify areas of improvement and potential student development. Integrating this method into higher education ensures learning is relevant to industry demands and technological developments. Therefore, performance assessment is an effective strategic tool in preparing students to compete professionally and adaptably in the digital era.(Puspitasari et al., 2025).

CONCLUSION

This study demonstrates that a performance assessment-based competency testing model provides a more authentic and comprehensive evaluation of students' programming skills than traditional examination methods. The findings reveal that students perform better in code implementation, while debugging and program optimization remain areas that require further improvement.

The main contribution of this study lies in the development of an authentic assessment model that integrates multiple dimensions of programming competence, including algorithm planning, implementation, debugging, and optimization. This model not only evaluates learning outcomes but also captures students' critical thinking, problem-solving, and creativity in a structured manner. Therefore, it offers a practical and relevant evaluation approach that supports the improvement of programming education and aligns with professional demands in the field of information technology.

DECLARATION OF AI AND AI ASSISTED TECHNOLOGIES IN THE WRITING PROCESS

During the preparation of this manuscript, the author(s) used Google Gemini to assist inimproving grammar, language quality, and overall readability of the text. After using this tool, the author(s) Carefully reviewed and edited the content as necessary and take full responbility for the content of the publication.

AUTHORS' CONTRIBUTION

Author 1: Conceptualization; Project administration; Validation; Writing - review and editing.
Author 2: Conceptualization; Data curation; In-vestigation.
Author 3: Data curation; Investigation.
Author 4: Formal analysis; Methodology; Writing - original draft.

DECLARATION OF COMPETING INTEREST

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

REFERENCES

Akadji, A., & Eka, M. (2025). Internalisasi Prinsip Clean Code dan Validasi Logika pada Pengembangan Aplikasi Python bagi Mahasiswa Teknik Komputer. *Embun Harapan: Jurnal ...*, (Query date: 2026-03-07 18:07:22). https://ejournalhealth.pustakariset.my.id/index.php/ehjpm/article/view/71

Alfiyanti, N., Herlambang, A., & ... (2025). Hubungan Antara Efikasi Diri dan Kecemasan Belajar Pemrograman Pada Siswa SMK Mata Pelajaran Pemrograman Dasar. ... *Teknologi Informasi Dan ...*, (Query date: 2026-03-07 18:07:22). https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/14888

Ashari, I., Perdana, A., & Aksha, A. (2025). … Aplikasi Roadmap Pemrograman Berbasis Website dengan Implementasi Algoritma Divide and Conquer untuk Meningkatkan Kualitas Hard Skill Mahasiswa …. ... *Dan Teknologi Informasi*, (Query date: 2026-03-07 18:07:22). https://www.jiktif.ft-uim.ac.id/index.php/jiktif/article/view/78

BHUANA, C. & ... (2025). … PEMBELAJARAN BLENDED LEARNING BERBASIS LMS MOODLE PADA MATA KULIAH ALGORITMA DAN PEMROGRAMAN: ANALISIS EFEKTIVITAS MODEL …. *JURNAL PENDIDIKAN ...*, (Query date: 2026-03-07 18:07:22). https://ojs.stkiprosaliametro.ac.id/index.php/JPR/article/view/842

Bhuwaneswari, T., Herlambang, A., & ... (2025). Hubungan Antara Frekuensi Latihan Pemrograman dan Tingkat Grit Dengan Hasil Belajar Siswa SMK Mata Pelajaran Pemrograman Dasar. ... *Teknologi Informasi Dan ...*, (Query date: 2026-03-07 18:07:22). https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/14882

Dewanto, A., Herlambang, A., & ... (2025). … Belajar Pedagogik, Hasil Belajar Teknologi Informasi, Persepsi Profesi Keguruan terhadap Minat Profesi Keguruan pada Mahasiswa Pendidikan Teknologi Informasi. ... *Teknologi Informasi Dan ...*, (Query date: 2026-03-07 18:07:22). https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/15693

Erlena, M. (2025). Langkah awal menuju pemrograman. *Maliki Interdisciplinary Journal*, (Query date: 2026-03-07 18:07:22). https://urj.uin-malang.ac.id/index.php/mij/article/download/18633/6055

Fitri, S., & Habibie, A. (2025). … Bahasa Pemrograman Python untuk Analisis Perbandingan Kinerja Akademik Mahasiswa Penerima Beasiswa dan Non Beasiswa (Studi Kasus Pada Mahasiswa …. ... *Ilmiah Pendidikan Teknologi Informasi*, (Query date: 2026-03-07 18:07:22). https://journal.umtas.ac.id/index.php/produktif/article/view/6139

Gavriel, A. (2025). Implementasi Pengelolaan Data Mahasiswa Menggunakan Pointer dan Sorting Berbasis Bahasa C++. *Jurnal Aplikasi Teknologi Dan Komputasi*, (Query date: 2026-03-07 18:07:22). https://journals.arces.org/jatekom/article/view/124

Mandowen, S., Rustiyana, H., Nistrina, K., Prayudani, S., & ... (2025). *Dasar Pemrograman: Teori &Aplikasi*. books.google.com. https://books.google.com/books?hl=en&lr=&id=c8qREQAAQBAJ&oi=fnd&pg=PA7&dq=pemrograman+mahasiswa+teknologi+informasi&ots=WOsYyYw_rI&sig=MWBQk_4zaTpMs780qHV9H6EXnKA

Nisa, K., Mursalati'Aini, N., Riyadi, S., Rosyid, A., & ... (2026). Pemikiran Komputasional dan Pemrograman Dasar: Konsep, Abstraksi, Dekomposisi, dan Penerapannya dalam Kehidupan Sehari-Hari Mahasiswa Pendidikan …. ... *Teknologi Informasi …*, (Query date: 2026-03-07 18:07:22). http://calamus.id/index.php/edutik/article/view/303

Pare, S., Kati, O., Bahrin, B., & ... (2026). Efektivitas ChatGPT sebagai AI-Co-Researcher dalam Meningkatkan Kualitas Proposal Penelitian Mahasiswa. *RIGGS: Journal of …*, (Query date: 2026-03-07 18:07:22). https://journal.ilmudata.co.id/index.php/RIGGS/article/view/5766

Permana, M., Wilonotomo, W., & ... (2025). Identifikasi Teknologi Yang Dominan Dalam Pengembangan Sistem Informasi Pemesanan Online: Systematic Literature Review. *JATI (Jurnal Mahasiswa …*, (Query date: 2026-03-07 18:07:22). https://www.ejournal.itn.ac.id/index.php/jati/article/view/14071

Puspitasari, Y., Razaki, R., & Efendi, C. (2025). Pendampingan Pemograman Kartu Rencana Studi (KRS) Online Bagi Mahasiswa Di Lingkungan STKIP PGRI Situbondo: Pengabdian. *Jurnal Pengabdian Masyarakat …*, (Query date: 2026-03-07 18:07:22). http://jerkin.org/index.php/jerkin/article/view/2793

Putra, H. (2025). Kecerdasan Buatan Dalam Pembelajaran Pemrograman Mahasiswa Informatika: Peran, Peluang, dan Tantangan. *SUBMIT: Jurnal Ilmiah Teknologi Infomasi Dan Sains*, (Query date: 2026-03-07 18:07:22). https://ejurnal.unim.ac.id/index.php/submit/article/view/4528

Putri, D., Alkhusna, S., & ... (2025). CODEBLOKS: Program Penjualan Alat Tulis Sederhana Menggunakan Bahasa Pemograman C++. *Jurnal Komputasi Dan …*, (Query date: 2026-03-07 18:07:22). https://journals.arces.org/jukompak/article/view/94

Salsabila, N., Wulandari, N., & Tyaningsih, R. (n.d.). ChatGPT sebagai asisten pemrograman: Studi deskriptif pada mahasiswa pendidikan matematika dalam pembelajaran MATLAB. *Unram.Sgp1.Digitaloceanspaces.Com*, (Query date: 2026-03-07 18:07:22). Retrieved https://unram.sgp1.digitaloceanspaces.com/simlitabmas/kinerja/penelitian/jurnal/9a8e9e31-a9ea-4dcf-8c8f-9af1752800ba-722-Article%20Text-5526-1-10-20250630.pdf

Saputra, D., & Juniawan, F. (2025). Aplikasi Penjadwalan Bimbingan Skripsi Program Studi Teknologi Informasi Universitas Bangka Belitung Berbasis Website. *Jurnal Ilmiah Teknologi …*, (Query date: 2026-03-07 18:07:22). https://pustakailmiah.web.id/index.php/jitaku/article/view/111

Sariana, N., & Billy, M. (2025). … MAHASISWA BARU DENGAN MENGGUNAKAN DESIGN THINKING STUDI KASUS: UNIVERSITAS TRILOGI: Jurnal Sistem Informasi dan Teknologi Informasi. ... *Sistem Informasi Dan Teknologi …*, (Query date: 2026-03-07 18:07:22). https://jursistekni.nusaputra.ac.id/article/view/461

Septiyanti, N., Luthfi, M., Cinthya, M., & Abdillah, R. (n.d.). Integrasi ChatGPT Sebagai Asisten Virtual Untuk Meningkatkan Efektivitas Pembelajaran Pemrograman Web Berbasis Project Based Learning. *Researchgate.Net*, (Query date: 2026-03-07 18:07:22). Retrieved https://www.researchgate.net/profile/Nisa-Dwi-Septiyanti-2/publication/398896267_Integrasi_ChatGPT_Sebagai_Asisten_Virtual_Untuk_Meningkatkan_Efektivitas_Pembelajaran_Pemrograman_Web_Berbasis_Project_Based_Learning_Integrating_ChatGPT_as_a_Virtual_Assistant_in_Project-Based_Learni/links/694601bf9aa6b4649dc265b1/Integrasi-ChatGPT-Sebagai-Asisten-Virtual-Untuk-Meningkatkan-Efektivitas-Pembelajaran-Pemrograman-Web-Berbasis-Project-Based-Learning-Integrating-ChatGPT-as-a-Virtual-Assistant-in-Project-Based-Learni.pdf

Setiawan, P., & Wandri, R. (2025). Inovasi Teknologi Melalui Pembelajaran Flutter: Menyongsong Era Aplikasi Mobile. *Jurnal Sains Dan Teknologi Dalam …*, (Query date: 2026-03-07 18:07:22). https://jurnal-rjb.com/index.php/jstpm/article/view/29

Tanjung, A., Hutagalung, L., Batubara, M., & ... (2025). DASAR TEKNOLOGI INFORMASI. *YPAD …*, (Query date: 2026-03-07 18:07:22). https://journal.yayasanpad.org/index.php/ypadbook/article/view/378

Wahyudi, M., Fanani, L., & Brata, A. (2025). Aplikasi Berbasis Progressive Web App (Mentoree) sebagai Layanan Pendampingan Pembelajaran Pemrograman di Universitas Brawijaya. *Jurnal Teknologi Informasi Dan Ilmu …*, (Query date: 2026-03-07 18:07:22). https://jtiik.ub.ac.id/index.php/jtiik/article/view/8733

Wipayana, I. (2025). *… Informasi Manajemen Praktek Kerja Lapangan Dengan Pendekatan Microservice Berbasis Web (Studi Kasus: Jurursan Teknologi Informasi Politeknik Negeri Bali)*. repository.pnb.ac.id. https://repository.pnb.ac.id/id/eprint/18886/

Wonoseto, M., Kusuma, M., & ... (2025). DATA CENTER ONLINE COMPILER UNTUK MEMBANGUN APLIKASI EVALUASI PRAKTIKUM PEMROGRAMAN BERDASARKAN PRINSIP MAUDHÛIYYAH,' …. *… Interkoneksi Islam Dan …*, (Query date: 2026-03-07 18:07:22). https://ejournal.uin-suka.ac.id/saintek/kiiis/article/view/4800