e-JURNAL PENYELIDIKAN DAN INOVASI
ISLAMIC INNOVATION CENTER FOR TECHNOLOGY, UIS
https://ejpi.uis.edu.my/index.php/ejpi
e-ISSN 2289-7909

VOL. 13 NO. 2 (AUGUST, 2026) Special Edition AIDI
Page 290-297
DOI: https://doi.org/10.53840/e-jpi.v13i2.434

Received: 24 July 2026 Accepted: 18 August 2026 Published: 31 August 2026

ARTIFICIAL INTELLIGENCE IN SOFTWARE DEVELOPMENT: A NARRATIVE REVIEW OF ITS EFFECTS ON STUDENT CODING EFFICIENCY IN HIGHER EDUCATION

Nur Razia Mohd Suradi[1], Azlinda Abdul Aziz[2], Nik Nordiana N Ab Rahman[3], Rahayu Handan[4], Mohd Amir Abdullah[5]

([1,3,4] Faculty Communication, Visual Art and Computing, Universiti Selangor, Bestari Jaya, Malaysia
[2] Faculty of Computer Science and Mathematics, Universiti Teknologi MARA (UiTM) Melaka, Campus Jasin,Malaysia
[5] Fakulti Pembangunan Mahasiswa Budiman, Universiti Selangor, Bestari Jaya, Malaysia
Email: [1]razia@unisel.edu.my, [2]azlindaaziz@uitm.edu.my, [2]niknordiana@unisel.edu.my, [4]raha@unisel.edu.my,[5]amir@unisel.edu.my,
[5]amir@unisel.edu.my

ABSTRACT

Artificial intelligence (AI) has become an integral component of modern software development using AI-powered coding assistants, automated debugging tools, and intelligent programming environments. In higher learning institutions, students increasingly adopt the tools to support programming tasks and enhance coding performance. This study presents a narrative review of existing literature on the effects of AI-powered development tools on students' coding efficiency in higher learning institutions. The review focuses on studies published between 2022 and 2026 and retrieved from major academic databases, including Scopus, IEEEXplore, and ScienceDirect, using predefined keywords and inclusion criteria. The selected studies were analyzed using thematic analysis to systematically identify and categorize recurring patterns across the literature, focusing on coding efficiency, learning outcomes, and challenges associated with AI-assisted programming tools. The findings indicate that AI-powered development tools significantly improve coding efficiency by increasing development speed, reducing syntax and logical errors, and enhancing debugging processes. In addition, these tools support learning by providing real-time code suggestions and improving students' problem-solving capabilities. However, certain limitations have been identified, including overdependence on AI-generated code, reduced independent thinking, issues regarding academic integrity, and limited conceptual understanding of programming fundamentals. The study concludes that AI-powered development tools provide significant advantages in enhancing students' coding efficiency; however, their application should be supported by well-designed teaching strategies to achieve balanced learning outcomes. Future research should examine the long-term impact of AI-assisted programming on students' technical competence, critical thinking, and professional readiness in software development.

Keywords: Artificial Intelligence, Software Development, AI Coding Assistants, Coding Efficiency, Higher Education Programming

INTRODUCTION

Students in higher education frequently use a wide range of AI-powered development tools to assist coding activities, with particular emphasis on AI pair programmers, code completion assistants, and generative AI applications. Tools such as GitHub Copilot, ChatGPT and similar reply are progressively transforming programming education by automating repetitive tasks, generating code, providing real-time suggestions, and assisting with debugging. ChatGPT is increasingly recognized as a useful educational tool because it can generate human-like responses, provide explanations, summarize content, and assist with academic tasks (D. de Oliveira, 2025). Recent studies have reported positive student perceptions of AI tools, particularly in STEM disciplines, where they help improve learning efficiency and provide quick access to information. Another study conducted by (Yetiştiren et al., (2023) compare the performance of these prominent code generation tools in terms of code quality metrics, such as Code Validity, Code Correctness, Code Security, Code Reliability, and Code Maintainability, to identify their strengths and shortcomings.

AI pair programmers and code completion tools utilize large language models to generate code from natural language prompts or programming contexts, helping students write code more efficiently and improve productivity. These tools are especially valuable for students with less understanding of coding because they offer immediate guidance and support during software development tasks. AI chatbots are frequently used for debugging, code explanation, and problem-solving, while AI-enhanced integrated development environments provide intelligent, context-aware code suggestions that improve the coding experience.

The fast adoption of AI-powered development tools has influenced students' learning experiences, programming confidence, and perceptions of software development. Many students examine AI positively and believe that it enhances learning, reduces repetitive coding tasks, improves efficiency, and lowers the effort required to complete programming assignments. These tools have demonstrated potential in supporting various software engineering activities, including code generation, code completion, testing, refactoring, reverse engineering, and code comprehension. However, their growing use also presents several challenges. There is still limited research focusing on students' coding efficiency in higher learning institutions.

PROBLEM STATEMENT

Even though AI-powered development tools have become progressively popular among higher education students and have proved benefits in coding efficiency and software development productivity, current literature primarily focuses on tool capabilities and short-term performance improvements Even though AI-powered development tools have become progressively popular among higher education students and have proved benefits in coding efficiency and software development productivity, current literature primarily focuses on tool capabilities and short-term performance improvements. A study conducted by Gambo et al., (2026) investigates how AI tools can influence the performance in computer programming in Nigeria which can cause academic integrity. Limited research has examined their impact on students' learning outcomes, programming competency, and independent problem-solving abilities.

Although numerous studies have investigated AI-powered development tools such as ChatGPT and GitHub Copilot, the existing literature remains fragmented and primarily focuses on tool capabilities and productivity outcomes. Limited attention has been given to synthesizing evidence on how these technologies influence student coding efficiency in higher education. Furthermore, the educational implications of AI-assisted coding, including its effects on learning, critical thinking skills, and programming competency, remain unclear due to inconsistent findings across studies. Therefore, a narrative review is required to provide a comprehensive understanding of the current state of knowledge and identify areas for future research.

This study seeks to accomplish the subsequent research objectives:

RO1: To identify the AI-powered development tools commonly used by higher education students.

RO2: To examine the effects of AI-powered development tools on students' coding efficiency.
RO3: To analyze the challenges and limitations associated with AI-assisted programming among students.

To fulfill these objectives, the study addresses the subsequent research questions:

RQ1: What AI-powered development tools are commonly used by higher education students?
RQ2: How do AI-powered development tools affect students' coding efficiency?
RQ3: What challenges and limitations do students encounter when using AI-powered development tools?

Therefore, this study aims to review existing literature on the influences of AI-powered tools on student's coding effectiveness in higher education.

METHODOLOGY

This study employed a narrative review methodology to discuss the gap by focusing on the implication of AI tools used by students in assisting them to learn programming activity.

a. Search Strategy

Preliminary Search Support

At the preliminary stage, ChatGPT (OpenAI) were used only to establish initial adaptation with the topic, refinement of candidate search terms and Boolean search strings, and organization of preliminary search criteria. Their role was exploratory and organizational rather than analytical. They were not used for self-directed study selection and eligibility purposes. The authors consequently conducted formal database searches for these studies. All retrieved records considered for inclusion were manually screened using predefined eligibility criteria.

Search Process

A structured literature search was conducted using three major electronic databases which are Scopus, IEEEXplore and Elsevier, to identify relevant peer-reviewed articles published in English between January 2022 and December 2026. The search strategy incorporated combinations of keywords and Boolean operators, including terms such as "Artificial Intelligence in Software Development", "AI Coding Assistants", "Coding Efficiency" and "Higher Education Programming". The Boolean Search operator, after combining the three research objectives (ROs) used in this study are as below:

(ChatGPT OR "GitHub Copilot" OR "Generative AI")
AND
(programming education OR programming OR coding)
AND
(student OR students)

However, in certain electronic database such as IEEE Xplore, the Boolean search did not work accordingly or failed to give the result which return 0 records. In this situation, the researcher must modify the Boolean operator which then gives a result. To ensure thorough coverage, additional relevant studies were identified through manual screening of the reference lists of selected articles. Studies were included if they were peer-reviewed original research articles, experimental studies, or review articles reporting, experimental, or technical applications of artificial intelligence in programming courses. Only articles published in English between 2022 and 2026 were considered.

Study Selection

Studies were excluded if they were editorials, letters to the editor, conference abstracts without full text, or non-peer-reviewed publications. Articles not directly related to programming or not involving artificial intelligence tools were also excluded. Furthermore, studies lacking adequate or relevance to the objectives of the review were not contemplated. A total of 52 studies that met the inclusion criteria were selected for analysis. The selection process involved initial screening of titles and abstracts followed by full-text evaluation based on relevance.

b. Study Selection Process

This section discussed the selection process as depicted in Figure 1 below. Three electronic databases are used which are Scopus, IEEE Xplore and Elsevier. For the Scopus database, the initially retrieved papers are 930 papers; however, after manual screening, the total selected paper is 20 papers. The process is then repeated with the IEEEXplore and Elsevier database which initially yielded 259 papers and 300 papers respectively. After filtering manually, 16 papers were chosen for each database. Overall, the total of selected papers from the three databases is 52 papers.

Each paper was first checked for duplication, to ensure they were not the same paper. However, there are no duplicate papers. However, upon filtering, no duplicate papers were found. The title and abstract were also screened to ensure the paper content is appropriate with the research to be studied. During this stage, 16 papers did not discuss the AI powered tools used in programming courses in higher learning and were therefore excluded, which makes the total of 36 papers. Through a thorough analysis, the papers that met the objectives of the study are 25 papers.

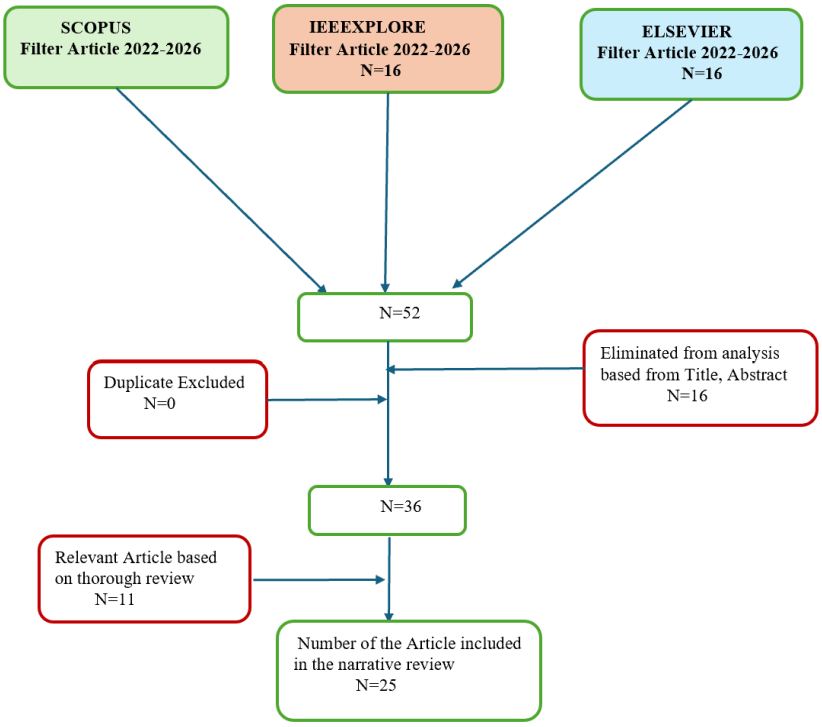

Figure 1: Study Selection Process

FINDINGS AND DISCUSSION

The research investigates the integration of generative AI tools into programming education, recognizing common tools, their impact on efficiency, and significant educational and security encounters.

AI-powered development tools

Generative AI(GenAI) tools such as ChatGPT and GitHub Copilot have been used widely in various aspects in software development such as in code automation, code debugging and in daily activities (Chiu et al., 2026). However, Cotton et al.(2024) recommend toward focusing on using GenAI in ethically in education environment despite arguing the usage of AI tools.

Higher education students predominantly employ a main group of general-purpose and specific AI tools for programming tasks:

ChatGPT (OpenAI)
Many studies utilize ChatGPT as a main tool in university which positively attracts students, critical thinking capabilities due to its accessibility and also its user-friendly interface in completing the academic tasks (Youssef et al., 2024). In programming courses, ChatGPT acts as a primary tool to assist the students in programming tasks (Gambo et al., 2026; Groothuijsen et al., 2024). According to Groothuijsen et al., (2024), ChatGPT is also used for error checking and debugging of code, increasing conceptual understanding, generating, and optimizing solution code, explaining code, and solving mathematical problems. Furthermore, students claimed that ChatGPT is used to obtain background knowledge, learning syntax and programming concepts and suggesting suitable algorithms (Haindl & Weinberger, 2024).

GitHub Copilot
Also known as "AI pair programmer" and is integrated directly as Integrated Development Environment (IDE) and act to provide real time code completion and suggestion (R. M. De Oliveira et al., 2026; Menon, 2023).

Other Tools
Other tools such as Coodey and Claude (Oyelere & Aruleba, 2025), DeepSeek (Waqas et al., 2025), Replit AI (Yuan & Li, 2025) and Amazon Q (Lee & Joe, 2025) are also used in programming courses, however, ChatGPT and Copilot are the most preferred by students.

Effects on Coding Efficiency
AI tools act as "Productivity tool" by saving times by giving solution to an error message occur on coding implementation (Menon, 2023). Besides that, students can learn programing with the assistant of AI (Bassner et al., 2026). Additionally, Haindl & Weinberger (2024) state that students learned a quality code by follow the naming of a good standard for that programming language that used. Consequently, students with AI-supported format maintained high student engagement in activity submissions throughout the semester, even after students reached the minimum passing grade (R. M. De Oliveira et al., 2026). The code quality produced by using AI tools is less complex and follow the rule (Haindl & Weinberger, 2024) compared with the group which did not used AI tool. A study conducted (Haindl & Weinberger, 2024) stated that AI tools can produce efficient code for programming tasks, however, still need human interaction.

Challenges and Limitations
While artificial intelligence has shown extensive potential in increasing students' coding competence, its acceptance in higher education is complemented by respective challenges and limitations. The reviewed literature indicates that issues such as depending on AI-generated code, reduced critical thinking and problem-solving skills, concerns regarding academic integrity, inaccuracies in AI-generated responses, data privacy risks, and unequal access to AI technologies may limit its educational effectiveness. Understanding these challenges is critical for ensuring that AI serves as a reassuring learning tool rather than an alternative for essential programming knowledge.

a. Low Critical thinking
The ability of AI tools to complete programming tasks reduces students' analytical thinking skills. Solutions are given without the need for students to understand. Students only accept the answers given by AI tools and understand only at the surface level (Nathaniel et al., 2025).

b. Information Accuracy

The answer or response given by AI tools is sometimes inaccurate and misleading as the code information based on the prompt engineering given by the user. This may be confusing to the beginners or students with insufficient beginner knowledge. (Waqas et al., 2025).

c. Ethics

The access to the information with additional fees, delayed response and sometimes gives an inaccurate result contribute to less interest for the students to know about the concept in programming (Baek et al., 2024). Besides that, the culture of dishonesty among students in completing the programming tasks is very high instead of seeking the help from the instructor or lecturer as a teacher of the course (Waqas et al., 2025).

CONCLUSION

AI powered tools are important component in learning programming in higher learning institutions. They are used for development aids, debugging, code creation, explanation and algorithmic formulation. To leverage the capabilities of AI tools, the teaching delivery can combine the traditional method in teaching the concept of the programming language whereas AI tools act as powerful personalized tutor. Eventually, AI tools cannot replace human instruction in practicing and teaching the foundation of programming skills. The assessment should be redesigned to use the AI tools and focus on the verification and understanding of the concept. However, studies constantly stated that the AI tools are not just added in teaching but should be integrated via on purpose pedagogy with clear policies that reserve critical thinking and core programming proficiencies while using AI's strengths. Refer Figure 2.

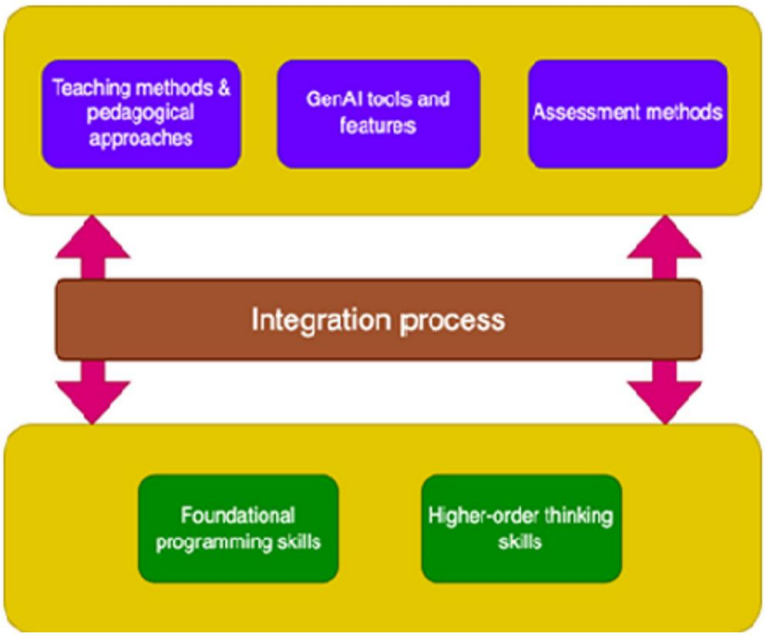

Figure 2: Alignment of teaching methods, GenAI tools, and assessment strategies in programming Education (Nathaniel et al., 2025)

ACKNOWLEDGEMENT

Thank you to the publisher for their support in the publication of this research article. We appreciate the efforts of the editorial team in reviewing and editing our work and we are thankful for the opportunity to contribute to the field of research through this publication.

FUNDING
This research did not receive any specific grant from funding agencies in the public, commercial, or non-profit sectors during preparing the manuscript.

REFERENCES

Baek, C., Tate, T., & Warschauer, M. (2024). "ChatGPT seems too good to be true": College students' use and perceptions of generative AI. *Computers and Education: Artificial Intelligence*, *7*. https://doi.org/10.1016/j.caeai.2024.100294

Bassner, P., Lenk-Ostendorf, B., Beinstingel, R., Wasner, T., & Krusche, S. (2026). Less stress, better scores, same learning: The dissociation of performance and learning in AI-supported programming education. *Computers and Education: Artificial Intelligence*, *10*. https://doi.org/10.1016/j.caeai.2025.100537

Chiu, K. Y., Saliman, K. M., Tian, R., Lam, A. H. C., Chiu, D. K. W., Choi, S. P. M., & Ho, K. K. W. (2026). Perceived effectiveness and usefulness of ChatGPT as an aid for learning programming: a quantitative study based on a novel 8E learning technology model. *Interactive Learning Environments*. https://doi.org/10.1080/10494820.2026.2646272

Cotton, D. R. E., Cotton, P. A., & Shipway, J. R. (2024). Chatting and cheating: Ensuring academic integrity in the era of ChatGPT. *Innovations in Education and Teaching International*, *61*(2), 228–239. https://doi.org/10.1080/14703297.2023.2190148

de Oliveira, D. (2025). Assessing ChatGPT in digital education: A case study on student perception. *Digital Engineering*, *7*. https://doi.org/10.1016/j.dte.2025.100067

De Oliveira, R. M., Da Silva, E. P., Wainer, J., & Azevedo, R. (2026). Generative AI in Introductory Programming for Non-CS STEM Students: A Two-Semester Study. *IEEE Access*, *14*, 88219–88236. https://doi.org/10.1109/ACCESS.2026.3701610

Gambo, O., Onifade, O. J., Olanrewaju-Smart, W., Abuya, O. J., Salawu, R. E., Massenon, R., & Gambo, I. (2026). Impact of ChatGPT on learning outcomes and performance of students in computer programming courses: a mixed-method approach. *Discover Education*, *5*(1). https://doi.org/10.1007/s44217-026-01142-4

Groothuijsen, S., van den Beemt, A., Remmers, J. C., & van Meeuwen, L. W. (2024a). AI chatbots in programming education: Students' use in a scientific computing course and consequences for learning. *Computers and Education: Artificial Intelligence*, *7*. https://doi.org/10.1016/j.caeai.2024.100290

Haindl, P., & Weinberger, G. (2024). Does ChatGPT Help Novice Programmers Write Better Code? Results From Static Code Analysis. *IEEE Access*, *12*, 114146–114156. https://doi.org/10.1109/ACCESS.2024.3445432

Lee, D. K., & Joe, I. (2025). A GPT-Based Code Review System With Accurate Feedback for Programming Education. *IEEE Access*, *13*, 105724–105737. https://doi.org/10.1109/ACCESS.2025.3581139

Menon, P. (2023). Exploring GitHub Copilot assistance for working with classes in a programming course. *Issues in Information Systems*, *24*(4), 66–81. https://doi.org/10.48009/4_iis_2023_106

Nathaniel, J., Oyelere, S. S., Suhonen, J., & Tedre, M. (2025). Literature Review on the Integration of Generative AI in Programming Education. In *International Journal of Artificial Intelligence in Education* (Vol. 35, Number 5, pp. 2724–2755). Springer. https://doi.org/10.1007/s40593-025-00524-3

Oyelere, S. S., & Aruleba, K. (2025). A comparative study of student perceptions on generative AI in programming education across Sub-Saharan Africa. *Computers and Education Open*, *8*. https://doi.org/10.1016/j.caeo.2025.100245

Waqas, M., Hasan, S., & Ali, A. (2025a). Effectiveness of Generative AI Tools in Computer Science and Engineering Education. *International Journal of Artificial Intelligence in Education*, *35*(6), 4002–4033. https://doi.org/10.1007/s40593-025-00525-2

Yetiştiren, B., Özsoy, I., Ayerdem, M., & Tüzün, E. (2023). *Evaluating the Code Quality of AI-Assisted Code Generation Tools: An Empirical Study on GitHub Copilot, Amazon CodeWhisperer, and ChatGPT*. http://arxiv.org/abs/2304.10778

Youssef, E., Medhat, M., Abdellatif, S., & Al Malek, M. (2024). Examining the effect of ChatGPT usage on students' academic learning and achievement: A survey-based study in Ajman, UAE. *Computers and Education: Artificial Intelligence*, *7*. https://doi.org/10.1016/j.caeai.2024.100316

Yuan, J., & Li, Y. (2025). The Impact of Security Mindset on the Use of AI Assistants in Computing Education. *IEEE Transactions on Education*, *68*(6), 484–491. https://doi.org/10.1109/TE.2025.3610340