IJIE (Indonesian Journal of Informatics Education)

Research Paper ISSN: 2549-0389

The Role of Artificial Intelligence in Programming Education and Its Impact on the Learning Process

Bagas Dwiantoro, *Yudianto Sujana, Puspanda Hatta
Informatics Education Department, Universitas Sebelas Maret, Indonesia
Corresponding Email: yudianto.sujana@staff.uns.ac.id

Abstract:

The integration of Artificial Intelligence (AI) into programming education has rapidly expanded, offering both promising opportunities and complex challenges. This study conducts a systematic literature review (SLR) using the PRISMA framework to examine the influence of AI tools on creativity, collaboration, and technology acceptance in programming education. A total of 33 peer-reviewed studies published between 2022 and 2024 were analysed to explore the pedagogical impact of AI. The findings indicate that AI tools support creative problem-solving, enhance collaborative learning, and increase student engagement through personalized feedback and adaptive learning environments. Despite these benefits, concerns remain about the potential for over-reliance on AI, reduced critical thinking, and ethical issues such as bias and authorship. While AI encourages iterative and imaginative approaches to programming, its successful implementation depends on instructional strategies that promote reflection, responsible tool use, and alignment with real-world programming practices. This study emphasizes the importance of balancing the advantages of AI with thoughtful pedagogy to support meaningful learning. Future research is recommended to investigate the long-term effects of AI on student development and to refine frameworks for integrating AI tools into programming education.

Keywords: *Artificial intelligence, learning process, programming education.*

IJIE (Indonesian Journal of Informatics Education)
Vol 9 Issue 1 2025
DOI: http://dx.doi.org/10.20961/ijie.v9i1.103884
Received: 11/06/25 Revised: 30/07/25 Accepted: 31/07/25 Online: 31/07/25
© 2025 The Authors. Published by Universitas Sebelas Maret. This is an open-access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

Introduction

The integration of Artificial Intelligence (AI) in education, particularly in programming learning, has rapidly progressed and significantly impacted various aspects of the learning process. AI-driven tools have shown immense potential in improving learning outcomes, fostering creativity, and enhancing collaborative dynamics in education (Fan et al., 2025; Xu, 2024). As AI technologies evolve, they continue to reshape the way students approach complex subjects such as programming, providing opportunities for personalized learning, real-time feedback, and problem-solving (Kohen-Vacs et al., 2025). Despite the transformative potential of AI in education, the effects of AI tools on students' programming skills, creativity, and collaboration remain areas of active investigation.

Previous studies have shown that AI tools can augment learning in various domains (Wang et al., 2024), but their specific effects on programming education remain underexplored. The influence of AI on developing critical thinking, coding skills, and group interactions in programming education is particularly unclear. AI tools, such as code generators and machine learning assistants, can aid students in generating coding solutions and debugging (Alanazi et al., 2025; Qureshi, 2023). However, it is uncertain whether these tools enhance or hinder students' problem-solving autonomy and independent thinking. Research on this topic has yielded mixed results, with some studies indicating that AI tools can develop programming skills and foster creativity (Alanazi et al., 2025; Zhou & Peng, 2025), while others suggest that over-reliance on AI may reduce critical thinking and autonomy (Kosmyna et al., 2025; Monib et al., 2024).

AI's potential to enhance collaboration in programming education is another critical area of focus. Collaborative learning in programming encourages teamwork, idea-sharing, and peer-based problem-solving, essential for skill development in real-world software development. Specific AI-based platforms, such as GitHub Copilot and Replit Teams, have been used to facilitate collaborative coding experiences, offering features like real-time feedback and collaborative code editing (Lyu et al., 2025; Welter et al., 2025). These platforms can enhance collaboration by streamlining communication, facilitating pair programming, and providing immediate feedback. However, research is limited on how AI tools influence group dynamics, communication, and project outcomes in programming education settings. Further investigation is needed to optimize AI's effectiveness in facilitating collaborative programming (Berrezueta-Guzman et al., 2025; Song et al., 2024).

Despite the growing body of literature on AI in education, there remains a gap in fully understanding the comprehensive impact of AI on programming education, particularly regarding creativity, collaboration, and independent problem-solving. This systematic literature review aims to examine how AI tools impact programming education in terms of creativity, collaboration, and independent thinking. It seeks to inform educators on the effective integration of AI technologies into curriculum and instructional practices, contributing to the development of more effective programming education.

Related Work

The role of AI in programming education has attracted considerable attention, with studies examining its impact on learning outcomes, creativity, and collaboration. Previous research highlights how AI tools like code assistants and automated feedback systems support students. For instance, AI-driven tools can help students overcome challenges like debugging and code optimization, facilitating a more efficient learning process (Alanazi et al., 2025; Lyu et al., 2025; Qureshi, 2023). AI-assisted learning environments, such as intelligent visual feedback systems, have also proven effective in improving user performance and cognitive engagement (Fan et al., 2025; Qureshi, 2023; Xu, 2024). While these studies demonstrate clear benefits, their scope is often limited to specific, measurable tasks. This narrow focus can overlook broader pedagogical challenges, including the risk of students becoming procedurally fluent without developing deep conceptual understanding.

A significant concern raised in the literature is that excessive reliance on AI tools may hinder the development of independent problem-solving and critical thinking skills. Chan (2023) highlighted the ethical concerns and potential for over-reliance on AI to diminish critical thought and creativity. This suggests that while AI tools can support learning, they must be carefully integrated to avoid creating dependency. To mitigate this risk, researchers suggest pedagogical strategies that shift the focus from the final product to the learning process itself. For example, educators can design assignments that require students to document their interaction with AI, submit their prompts, and explain the revisions they made, thereby fostering metacognitive skills (Rahe & Maalej, 2025). Another proposed solution is to design tasks that are less susceptible to direct AI solutions, such as complex debugging, code refactoring, and critical code reviews, which compel students to engage with the material on a deeper level (Takerngsaksiri et al., 2024; Zhai et al., 2024).

Research on AI's role in collaborative programming projects has yielded mixed results. On one hand, AI tools can enhance collaboration by providing real-time feedback and enabling students to share ideas more effectively (Ma et al., 2023). On the other hand, some studies identify potential drawbacks, such as AI creating dependency among students, which can reduce their ability to engage in critical discussions and undermine essential soft skills like teamwork and communication (Stamer et al., 2023). A key limitation of many of these studies is their focus on the technology itself, often in isolation from the pedagogical framework. The effectiveness of collaborative AI tools is not inherent in the technology but is highly dependent on structured educational strategies that guide students toward responsible AI use while actively fostering human-to-human interaction.

The acceptance of AI in programming education is another critical area. Studies applying the Technology Acceptance Model (TAM) have shown that students are more likely to adopt AI tools if they perceive them as easy to use and beneficial for enhancing their programming skills (Pan et al., 2024). However, significant barriers to adoption, such as ethical concerns and the fear of diminished personal skill development, have also been identified (Shrivastava, 2025). Furthermore, these studies face methodological limitations. The TAM, while useful, was not designed for the specific context of education and can oversimplify the complex factors at play. Critics argue that it often neglects crucial elements such as pedagogical goals, institutional policies, and the nuanced instructor-student dynamic, all of which heavily influence whether a tool is not just adopted, but meaningfully integrated (Liu, 2025; Ursavaş et al., 2025). These findings underscore the need for a more holistic evaluation framework that considers technological, pedagogical, and ethical dimensions when integrating AI into programming education.

Research Method

This study employs a systematic literature review (SLR) approach, guided by the Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA) framework. PRISMA ensures a transparent, replicable, and rigorous process in reviewing and analysing literature related to the influence of artificial intelligence (AI) on creativity, collaboration, and technology acceptance in programming education (Chan, 2023).

Design of Research

To effectively address the research question asking how AI tools impact creativity, collaboration, and technology acceptance in programming education, the inclusion criteria for this systematic literature review were deliberately formulated. The primary criterion mandated that selected studies must investigate the use of AI tools specifically within a programming education context. This ensures that all included research is directly relevant to the core subject matter.

Furthermore, to ensure a comprehensive analysis that directly corresponds to the distinct components of the research question, the criteria required that studies address at least one of the central themes: creativity, collaboration, or the acceptance of technology. By focusing on studies that explore creative problem-solving, collaborative dynamics, and how students and educators adopt and interact with these recent technologies, the review can synthesize targeted evidence for each aspect of the research inquiry. Conversely, studies were excluded if they did not pertain to AI in a programming setting or failed to cover these specific themes, thereby maintaining a focused analysis directly aligned with the research objectives.

Data Collection

The data collection process involved a comprehensive search of peer-reviewed articles indexed in major academic databases, focusing on recent studies within the last decade. Keywords such as "AI in programming education," "creativity and AI tools," "collaboration with AI platforms," and "technology acceptance in education" were used.

An initial pool of articles was screened by evaluating titles and abstracts to align with the study's inclusion criteria. Subsequently, full-text reviews ensured the selected studies provided significant insights into the roles of AI in fostering creativity, enhancing collaboration, and influencing technology acceptance in programming education. This systematic approach resulted in the selection of thirty-three relevant studies for analysis, offering a comprehensive understanding of the integration and impact of AI tools in educational settings. The inclusion and exclusion criteria are outlined in Table 1.

Table 1. Article Selection Criteria

Inclusion Criteria	Exclusion Criteria
Studies examining the integration of Artificial Intelligence in programming education, focusing on AI usage and its technical applications.	Studies that do not involve Artificial Intelligence as a central focus in programming education or lack technical details about AI usage.
Research investigating the impact of AI on pedagogical aspects, such as creativity, collaboration, and learning outcomes, in programming education.	Research that fails to address pedagogical outcomes, such as creativity, collaboration, or learning enhancement, in programming education.
Articles exploring innovative pedagogical approaches using AI in programming education.	Studies that do not provide measurable evidence or outcomes related to the impact of AI on programming education.

Data Analysis

Data analysis for this review followed a structured synthesis process. The articles were organised into three main themes based on the research questions: creativity, collaboration, and technology acceptance. For each theme, the findings from the selected studies were summarised, compared, and analysed to identify common patterns and trends.

Each article was evaluated based on several factors, including the study's methodology, sample size, and key findings. The impact of generative AI on each theme was carefully examined, and a thematic synthesis approach was used to identify key insights from the selected studies. Additionally, any inconsistencies or gaps in the research were noted, helping to highlight areas that require further investigation.

PRISMA Process and Architecture

In this study, a SLR methodology was applied according to the Preferred Reporting Items for PRISMA 2020 guidelines (Page et al., 2021). PRISMA offers a structured, transparent framework for systematically identifying, selecting, and synthesizing research, ensuring methodological rigor. The review process followed three main phases: planning, conducting, and reporting, as recommended by PRISMA. During the planning phase, the research question was defined to examine the role of artificial intelligence (AI) in programming education and its impact on the learning process. This research question focused on key themes such as challenges, objectives, methodologies, and findings from studies published between 2023 and 2024.

The conducting phase began by identifying relevant literature. A comprehensive search strategy was developed using keywords like "artificial intelligence," "programming education," "AI in education," and "AI impact on learning." The search was conducted in January 2024 across Scopus, ensuring a broad and thorough coverage. Inclusion and exclusion criteria were established to limit the review to empirical studies that used data collection to answer research questions rigorously. Opinion-based or theoretical articles were excluded. The PRISMA flow diagram visually represents the results of the search, showing the number of articles identified, screened, and included.

The article selection process was conducted systematically in the Scopus database, focusing on articles published between January 1, 2022, and January 1, 2024, in English. A search equation was used to identify articles related to artificial intelligence (AI) and programming education. Initially, 250 articles were retrieved, with 230 remaining after duplicates and inaccessible articles were removed. The titles, abstracts, and keywords of these articles were reviewed to ensure their relevance to the research questions, leaving 60 articles for further assessment.

In the eligibility phase, full-text articles were reviewed based on criteria that focused on AI's use in programming education and its impact on learning. Opinion-based and non-empirical articles were excluded, narrowing the selection to 20 high-quality studies that met the inclusion criteria. These studies were analysed to explore AI's role in enhancing learning outcomes, cognitive engagement, and skill development in programming education.

The process was documented in accordance with PRISMA guidelines to ensure transparency and repeatability. A PRISMA flow diagram shown in Figure 1summarizes each step, from the initial identification of 250 articles to the inclusion of the final 20 studies. The findings from these studies highlight how AI in programming education can improve learning outcomes by providing adaptive feedback, increasing motivation, and fostering autonomous learning.

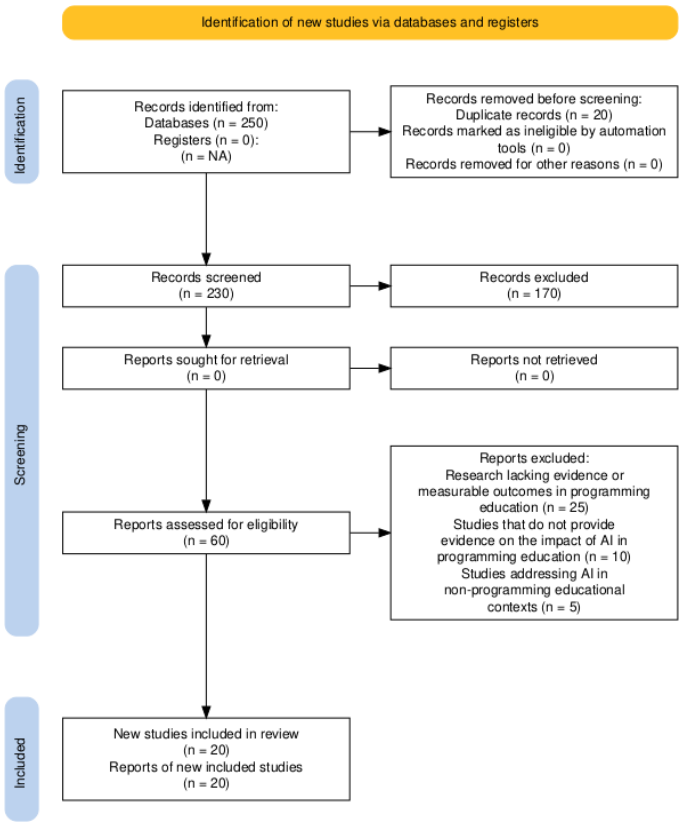

Figure 1. PRISMA flow diagram.

Implementation

After selecting the relevant studies, their findings were systematically organized and synthesized around the main themes of programming education and AI's impact on the learning process. Each article was carefully reviewed to extract key insights related to how AI influences learning outcomes, cognitive engagement, and the development of programming skills. The results were categorized and analysed to assess the overall impact of AI on programming education. The analysis also explored whether AI tools enhanced student performance, fostered independent learning, and influenced students' perceptions and acceptance of AI in the educational process.

Evaluation

The quality and relevance of each study were assessed using several criteria: methodological rigor, alignment with the research question, and their contribution to understanding AI's role in programming education. Methodological rigor was evaluated by examining the study's research design, sample size, and the robustness of its data collection and analysis methods. The relevance of each study was evaluated based on its direct focus on AI's effects on programming education, particularly concerning learning outcomes, motivation, and student engagement. Lastly, the contribution of each study to advancing the field was assessed, focusing on identifying any gaps or limitations in current research, with particular attention given to the integration of AI and its potential benefits and challenges in programming education.

Result and Discussion

Result

The findings of this systematic literature review emphasize the impact of artificial intelligence (AI) on programming education, specifically in relation to learning outcomes, cognitive engagement, and technology acceptance. The results are synthesized to answer the two research questions (RQs).

RQ1: What Are the Impacts and Benefits of AI in the Educational Process of Programming?

The integration of AI in programming education brings several transformative benefits, notably in enhancing creativity, fostering collaboration, and promoting technology acceptance. AI tools help students engage in more creative problem-solving processes by generating diverse solutions and suggesting new approaches to coding challenges (Bucea-Manea-Ţoniş et al., 2022; Lin & Chen, 2024; Qian, 2025; Zhong et al., 2024). These AI-driven systems encourage students to think outside the box, much like the effects seen in creative disciplines, where AI tools facilitate innovative thinking and idea generation (Alanazi et al., 2025; Fan et al., 2025; Kohen-Vacs et al., 2025; Lyu et al., 2025; Rahe & Maalej, 2025; Zhong et al., 2024).

Moreover, AI promotes collaboration among students by facilitating teamwork and enhancing communication through AI-powered discussion platforms and collaborative learning environments. Students working with AI often engage in deeper, more dynamic discussions about coding practices, which improves collective problem-solving skills and fosters a collaborative learning culture (Fan et al., 2025; Lyu et al., 2025; Song et al., 2024; Stamer et al., 2023; Yan et al., 2024).

AI tools also contribute to greater technology acceptance in education. As AI-powered platforms become more integrated into the learning process, students and educators alike increasingly recognize their value in enhancing educational outcomes (Cervantes & Navarro, 2025; Pan et al., 2024; Rahe & Maalej, 2025; Shrivastava, 2025; Ursavaş et al., 2025). The rapid adoption of AI tools in programming education shows how they help bridge the gap between traditional teaching methods and the evolving technological landscape of education (Berrezueta-Guzman et al., 2025; Cervantes & Navarro, 2025; Fan et al., 2025; Kohen-Vacs et al., 2025; Lemke et al., 2023; Lyu et al., 2025; Ma et al., 2023; Ursavaş et al., 2025; Xu, 2024; Zastudil et al., 2023).

RQ2: How Does the Implementation of AI in Programming Education Assist in the Learning Process?

The implementation of AI in programming education assists in the learning process by enhancing problem-solving abilities, facilitating deeper engagement with coding tasks, and promoting critical thinking. Studies have shown that AI-driven platforms enable students to experiment with different approaches to problem-solving without fear of failure, stimulating critical thinking and increasing motivation (Bucea-Manea-Ţoniş et al., 2022; Cervantes & Navarro, 2025; Fan et al., 2025; Kohen-Vacs et al., 2025; Lin & Chen, 2024; Qian, 2025; Rahe & Maalej, 2025; Stamer et al., 2023; Takerngsaksiri et al., 2024; Xu, 2024; Zhai et al., 2024; Zhong et al., 2024). Additionally, AI helps in breaking down

complex programming concepts into manageable parts, which improves students' comprehension and application skills (Rahe & Maalej, 2025; Xu, 2024). Notably, AI tools facilitate hands-on learning experiences, allowing students to engage with complex programming concepts in a more practical and interactive way (Cervantes & Navarro, 2025; Lin & Chen, 2024; Rahe & Maalej, 2025; Xu, 2024; Zastudil et al., 2023). By leveraging these capabilities, AI-powered platforms can create more effective learning environments that support students' diverse needs and abilities.

Discussion

Addressing the Learning Process with AI in Programming Education

AI tools have significantly transformed the way programming education is approached by fostering more efficient and adaptive learning methodologies. The ability of AI to provide personalized feedback and real-time assistance allows students to engage with coding tasks in an iterative manner, reducing cognitive load and enabling faster progress (Kosmyna et al., 2025; Qian, 2025; Rahe & Maalej, 2025; Wang et al., 2024; Xu, 2024). This iterative process encourages students, particularly those with different learning styles, to experiment with solutions and refine their approaches, which enhances both their coding skills and problem-solving abilities (Berrezueta-Guzman et al., 2025; Lyu et al., 2025; Pan et al., 2024; Rahe & Maalej, 2025; Song et al., 2024; Takerngsaksiri et al., 2024; Welter et al., 2025; Zhong et al., 2024). According to the Technology Acceptance Model (TAM), AI tools must be easy to use and beneficial for students in order to increase their acceptance and integration into educational environments (Cervantes & Navarro, 2025; Chan, 2023; Kohen-Vacs et al., 2025; Lemke et al., 2023; Monib et al., 2024; Pan et al., 2024; Qian, 2025; Rahe & Maalej, 2025; Shrivastava, 2025; Ursavaş et al., 2025; Wang et al., 2024; Zhai et al., 2024).

AI in programming education also encourages creative exploration by offering diverse coding solutions and suggesting alternative approaches, which cultivates an environment where students feel empowered to think beyond conventional programming paradigms (Bucea-Manea-Ţoniş et al., 2022; Kohen-Vacs et al., 2025; Lin & Chen, 2024). This aligns with studies indicating that AI tools can inspire students to visualize new possibilities and generate innovative solutions to complex programming problems (Lemke et al., 2023; Xu, 2024; Yan et al., 2024). By facilitating speculative thinking, AI aids students in tackling open-ended programming challenges and contributes to fostering a more imaginative approach to coding tasks.

However, challenges remain in fully integrating AI into programming education. While AI tools excel in generating novel solutions and automating repetitive coding tasks, students may face difficulties in applying these solutions to real-world scenarios or in meeting industry standards without proper guidance (Bucea-Manea-Ţoniş et al., 2022; Rahe & Maalej, 2025; Yan et al., 2024). This highlights the importance of balancing the use of AI with traditional learning methods that emphasize practical application and critical evaluation (Cervantes & Navarro, 2025; Fan et al., 2025; Lyu et al., 2025; Ma et al., 2023; Xu, 2024). Educators must also ensure that students understand the limitations of AI, such as the potential for bias in AI-driven tools and the ethical implications of relying too heavily on these systems in educational settings (Cervantes & Navarro, 2025; Lyu et al., 2025; Monib et al., 2024; Qureshi, 2023; Rahe & Maalej, 2025; Shrivastava, 2025; Ursavaş et al., 2025; Zhai et al., 2024).

In addition, AI-powered tools play a key role in fostering collaboration among students. Collaborative learning, supported by AI technologies, enhances both technical and soft skills, such as communication and teamwork, which are crucial in programming education (Stamer et al., 2023; Yan et al., 2024). AI encourages peer interactions by offering collaborative platforms where students can collectively solve programming challenges and share knowledge. This collaborative approach not only improves coding skills but also cultivates a sense of shared responsibility and creative problem-solving, which are essential elements for success in the field of programming (Fan et al., 2025; Lyu et al., 2025; Song et al., 2024).

Impact on Programming Outcomes and Educational Benefits

The effects of AI in programming education extend beyond the learning process to tangible student outcomes. AI-driven platforms have been shown to improve students' productivity and creativity by automating repetitive coding tasks and generating multiple programming solutions (Kohen-Vacs et al., 2025; Lin & Chen, 2024; Qian, 2025; Rahe & Maalej, 2025; Song et al., 2024; Takerngsaksiri et al., 2024; Zhou & Peng, 2025). This allows students to focus on more complex and higher-order tasks, which enhances their overall learning experience and encourages deeper engagement with the material. Studies suggest that students using AI tools in programming education often produce more original and sophisticated outputs, as AI helps to refine their ideas and generate diverse coding possibilities (Lin & Chen, 2024; Zastudil et al., 2023).

Despite these advancements, the practical application of AI-generated solutions in real-world programming scenarios remains a subject of ongoing discussion. While AI tools are effective in creating innovative and unconventional

programming solutions, students may require additional support in aligning these outputs with industry requirements and ensuring they meet the necessary standards (Bull & Kharrufa, 2024). The importance of integrating AI tools within a pedagogical framework that emphasizes both creativity and practicality cannot be overstated.

Furthermore, the ethical implications of AI use in programming education must be carefully considered. Issues such as authorship, bias, and the potential for over-reliance on AI are critical concerns that must be addressed by educators (Monib et al., 2024; Qureshi, 2023; Shrivastava, 2025; Zhai et al., 2024). By encouraging students to reflect on these ethical dilemmas, educators can transform potential challenges into opportunities for critical thinking and professional growth. This ensures that students not only learn how to effectively use AI tools but also become responsible creators who understand the ethical dimensions of technology in the programming field (Cervantes & Navarro, 2025; Kohen-Vacs et al., 2025; Shrivastava, 2025).

Conclusion

This study highlights the transformative impact of generative AI (GenAI) tools in programming education, emphasizing their role in enhancing both the creative and cognitive aspects of learning. AI facilitates iterative learning, enabling students to explore a variety of coding solutions and engage with programming tasks in a more personalized and adaptive manner. By reducing cognitive load, AI empowers students to focus on higher-order skills like problem-solving, critical thinking, and creative ideation.

Furthermore, AI tools foster enhanced collaboration among students, facilitating peer-to-peer learning and shared problem-solving. However, the integration of AI tools into educational contexts requires careful consideration of ethical implications, such as bias in AI algorithms and over-reliance on technology. Instructors must ensure students understand AI's limitations and can critically evaluate these tools.

The educational implications of this study are significant. The main benefits of AI in programming education include enhanced creativity, productivity, and problem-solving skills. However, limitations such as ethical concerns and potential over-reliance on technology must be addressed. A key research agenda for the future includes investigating the long-term effects of AI on students' professional development and optimizing AI tools to support students in meeting industry standards while encouraging innovation.

Acknowledgement

This research was supported by a Research Group Grant from Universitas Sebelas Maret under Contract Number 194.2/UN27.22/PT.01.03/2024. The authors sincerely acknowledge the assistance provided by Universitas Sebelas Maret, which facilitated the successful completion of this study.

References

Alanazi, M., Soh, B., Samra, H., & Li, A. (2025). The Influence of Artificial Intelligence Tools on Learning Outcomes in Computer Programming: A Systematic Review and Meta-Analysis. *Computers*, *14*(5), 185. https://doi.org/10.3390/computers14050185

Berrezueta-Guzman, S., Krusche, S., & Wagner, S. (2025). *From Coders to Critics: Empowering Students through Peer Assessment in the Age of AI Copilots*. https://arxiv.org/pdf/2505.22093

Bucea-Manea-Țoniș, R., Kuleto, V., Gudei, S. C. D., Lianu, C., Lianu, C., Ilić, M. P., & Păun, D. (2022). Artificial Intelligence Potential in Higher Education Institutions Enhanced Learning Environment in Romania and Serbia. *Sustainability*, *14*(10), 5842. https://doi.org/10.3390/su14105842

Bull, C., & Kharrufa, A. (2024). Generative Artificial Intelligence Assistants in Software Development Education: A Vision for Integrating Generative Artificial Intelligence into Educational Practice, Not Instinctively Defending Against It. *IEEE Software*, *41*(2), 52–59. https://doi.org/10.1109/MS.2023.3300574

Cervantes, J., & Navarro, E. (2025). Business Students' Perceptions of AI in Higher Education: An Analysis Using the Technology Acceptance Model. *Journal of Interdisciplinary Perspectives*, *3*(6). https://doi.org/10.69569/jip.2025.194

Chan, C. K. Y. (2023). A comprehensive AI policy education framework for university teaching and learning. *International Journal of Educational Technology in Higher Education*, *20*(1), 38. https://doi.org/10.1186/s41239-023-00408-3

Fan, G., Liu, D., Zhang, R., & Pan, L. (2025). The impact of AI-assisted pair programming on student motivation, programming anxiety, collaborative learning, and programming performance: a comparative study with traditional pair programming and individual approaches. *International Journal of STEM Education*, *12*(1), 1–17. https://doi.org/10.1186/S40594-025-00537-3

Kohen-Vacs, D., Usher, M., & Jansen, M. (2025). Integrating Generative AI into Programming Education: Student Perceptions and the Challenge of Correcting AI Errors. *International Journal of Artificial Intelligence in Education*, 1–19. https://doi.org/10.1007/s40593-025-00496-4

Kosmyna, N., Hauptmann, E., Yuan, Y. T., Situ, J., Liao, X.-H., Beresnitzky, A. V., Braunstein, I., & Maes, P. (2025). *Your Brain on ChatGPT: Accumulation of Cognitive Debt when Using an AI Assistant for Essay Writing Task*. https://arxiv.org/pdf/2506.08872

Lemke, C., Kirchner, K., Anandarajah, L., & Herfurth, F. (2023). Exploring the Student Perspective: Assessing Technology Readiness and Acceptance for Adopting Large Language Models in Higher Education. *European Conference on E-Learning*, *22*(1), 156–164. https://doi.org/10.34190/ecel.22.1.1828

Lin, H., & Chen, Q. (2024). Artificial intelligence (AI) -integrated educational applications and college students' creativity and academic emotions: students and teachers' perceptions and attitudes. *BMC Psychology*, *12*(1), 487. https://doi.org/10.1186/s40359-024-01979-0

Liu, N. (2025). Exploring the factors influencing the adoption of artificial intelligence technology by university teachers: the mediating role of confidence and AI readiness. *BMC Psychology*, *13*(1), 311. https://doi.org/10.1186/s40359-025-02620-4

Lyu, W., Wang, Y., Sun, Y., & Zhang, Y. (2025). Will Your Next Pair Programming Partner Be Human? An Empirical Evaluation of Generative AI as a Collaborative Teammate in a Semester-Long Classroom Setting. *Proceedings of the Twelfth ACM Conference on Learning @ Scale*, 83–94. https://doi.org/10.1145/3698205.3729544

Ma, Q. C., Wu, S. T., & Koedinger, K. (2023). Is AI the better programming partner? Human-Human Pair Programming vs. Human-AI pAIr Programming. *CEUR Workshop Proceedings*, *3487*, 64–77. https://arxiv.org/pdf/2306.05153

Monib, W. K., Qazi, A., Apong, R. A., Azizan, M. T., De Silva, L., & Yassin, H. (2024). Generative AI and future education: a review, theoretical validation, and authors' perspective on challenges and solutions. *PeerJ Computer Science*, *10*, e2105. https://doi.org/10.7717/peerj-cs.2105

Page, M. J., McKenzie, J. E., Bossuyt, P. M., Boutron, I., Hoffmann, T. C., Mulrow, C. D., Shamseer, L., Tetzlaff, J. M., Akl, E. A., Brennan, S. E., Chou, R., Glanville, J., Grimshaw, J. M., Hróbjartsson, A., Lalu, M. M., Li, T., Loder, E. W., Mayo-Wilson, E., McDonald, S., … Moher, D. (2021). The PRISMA 2020 statement: an updated guideline for reporting systematic reviews. *BMJ*, *372*, n71. https://doi.org/10.1136/bmj.n71

Pan, Z., Xie, Z., Liu, T., & Xia, T. (2024). Exploring the Key Factors Influencing College Students' Willingness to Use AI Coding Assistant Tools: An Expanded Technology Acceptance Model. *Systems*, *12*(5), 176. https://doi.org/10.3390/systems12050176

Qian, Y. (2025). Pedagogical Applications of Generative AI in Higher Education: A Systematic Review of the Field. *TechTrends*, 1–16. https://doi.org/10.1007/s11528-025-01100-1

Qureshi, B. (2023). ChatGPT in Computer Science Curriculum Assessment: An analysis of Its Successes and Shortcomings. *2023 9th International Conference on E-Society, e-Learning and e-Technologies*, 7–13. https://doi.org/10.1145/3613944.3613946

Rahe, C., & Maalej, W. (2025). How Do Programming Students Use Generative AI? *Proceedings of the ACM on Software Engineering*, *2*(FSE), 978–1000. https://doi.org/10.1145/3715762

Shrivastava, P. (2025). Understanding acceptance and resistance toward generative AI technologies: a multi-theoretical framework integrating functional, risk, and sociolegal factors. *Frontiers in Artificial Intelligence*, *8*, 1565927. https://doi.org/10.3389/frai.2025.1565927

Song, F., Agarwal, A., & Wen, W. (2024). The Impact of Generative AI on Collaborative Open-Source Software Development: Evidence from GitHub Copilot. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.4856935

Stamer, T., Steinhäuser, J., & Flägel, K. (2023). Artificial Intelligence Supporting the Training of Communication Skills in the Education of Health Care Professions: Scoping Review. *Journal of Medical Internet Research*, *25*(1), e43311. https://doi.org/10.2196/43311

Takerngsaksiri, W., Warusavitarne, C., Yaacoub, C., Keng Hou, M. H., & Tantithamthavorn, C. (2024). Students' Perspectives on AI Code Completion: Benefits and Challenges. *2024 IEEE 48th Annual Computers, Software, and Applications Conference (COMPSAC)*, 1606–1611. https://doi.org/10.1109/COMPSAC61105.2024.00252

Ursavaş, Ö. F., Yalçın, Y., İslamoğlu, H., Bakır-Yalçın, E., & Cukurova, M. (2025). Rethinking the importance of social norms in generative AI adoption: investigating the acceptance and use of generative AI among higher education students. *International Journal of Educational Technology in Higher Education*, *22*(1), 38. https://doi.org/10.1186/s41239-025-00535-z

Wang, Y., Hanafi Zaid, Y., Li, J., & Pan, Y. (2024). Opportunities and Challenges of Using ChatGPT as a Teaching Assistant in English Language Teaching: A Systematic Literature Review. *Proceedings of the 2024 International Symposium on Artificial Intelligence for Education*, 375–382. https://doi.org/10.1145/3700297.3700362

Welter, A., Schneider, N., Dick, T., Weis, K., Tinnes, C., Wyrich, M., & Apel, S. (2025). *From Developer Pairs to AI Copilots: A Comparative Study on Knowledge Transfer*. https://arxiv.org/pdf/2506.04785

Xu, Z. (2024). AI in education: Enhancing learning experiences and student outcomes. *Applied and Computational Engineering*, *51*(1), 104–111. https://doi.org/10.54254/2755-2721/51/20241187

Yan, W., Nakajima, T., & Sawada, R. (2024). Benefits and Challenges of Collaboration between Students and Conversational Generative Artificial Intelligence in Programming Learning: An Empirical Case Study. *Education Sciences*, *14*(4), 433. https://doi.org/10.3390/educsci14040433

Zastudil, C., Rogalska, M., Kapp, C., Vaughn, J., & MacNeil, S. (2023). Generative AI in Computing Education: Perspectives of Students and Instructors. *2023 IEEE Frontiers in Education Conference (FIE)*, 1–9. https://doi.org/10.1109/FIE58773.2023.10343467

Zhai, C., Wibowo, S., & Li, L. D. (2024). The effects of over-reliance on AI dialogue systems on students' cognitive abilities: a systematic review. *Smart Learning Environments*, *11*(1), 28. https://doi.org/10.1186/s40561-024-00316-7

Zhong, T., Zhu, G., You Lim, K., Soon Ong, Y., Zhong Candidate, T., & Zhu Assistant Professor, G. (2024). *Generative AI as a Tool or Leader? Exploring AI-Augmented Thinking in Student Programming Tasks*. https://arxiv.org/pdf/2411.19490

Zhou, M., & Peng, S. (2025). The Usage of AI in Teaching and Students' Creativity: The Mediating Role of Learning Engagement and the Moderating Role of AI Literacy. *Behavioral Sciences*, *15*(5), 587. https://doi.org/10.3390/bs15050587