Educational Journal of Learning Technology
Vol. 4, No. 2, pp. 443-452, January 2027
https://10.58230/edutech.v4i1.153
Received October 24, 2026; Revised November 13, 2026; Accepted November 16, 2026

Generative AI-Based Adaptive Learning to Enhance Computational Thinking and Self-Regulated Learning

Monica Fransisca[1], Renny Permata Saputri[2]
[1,2] Universitas Putra Indonesia YPTK
Jl. Raya Lubuk Begalung, Lubuk Begalung District, Padang City, West Sumatra, Indonesia.
[1]monicafransisca@upiyptk.ac.id

Abstract. This research aimed to develop and evaluate a Generative AI-based adaptive learning model to support computational thinking and self-regulated learning in Algorithm and Programming courses. This study used a Research and Development (R&D) approach combined with the ADDIE model and involved 35 Informatics Engineering Education students at Universitas Putra Indonesia YPTK. The developed model integrated diagnostic assessment, learner profiling, adaptive learning pathways, Generative AI scaffolding, programming practice, feedback, and reflection. The model used ChatGPT and Gemini to provide explanations, hints, examples, and learning feedback tailored to students' needs. The results showed a 14.46-percentage-point improvement in computational thinking and a 12.37-percentage-point improvement in self-regulated learning after implementing the model. These improvements suggest positive changes in students' systematic problem-solving abilities and their capacity to manage their learning processes. The findings highlight the potential of integrating adaptive learning and Generative AI scaffolding to support cognitive and self-regulatory development in programming education. The study also emphasizes the importance of instructional guidance, lecturer readiness, equitable access to GenAI tools, and differentiated support for students with varying levels of prior knowledge.

Keywords: Adaptive Learning; Computational Thinking; Generative AI; Programming Education; Self-Regulated Learning.

1. Introduction

Algorithm and Programming is an important course in Informatics Engineering Education because it develops students' ability to analyze problems, construct logical solutions, and translate solutions into algorithms and computer programs. However, learning programming goes beyond understanding syntax. Students must also develop problem-solving processes, organize information, identify relationships between problems, and construct systematic procedures. These abilities relate to computational thinking (CT), such as breaking down complex problems, identifying recurring patterns, developing abstractions, and formulating algorithmic solutions. Research on computational thinking and coding interventions has examined their relationship with problem-solving and other cognitive skills (Kong & Wang, 2024). In addition, research in programming education has determined the relationship between computational thinking, self-regulated learning, and problem-solving in programming skills (Susilowati et al., 2024).

In addition to computational thinking, students' competencies to regulate their learning are important in programming education. Programming requires continuous practice, error identification, strategy selection, monitoring, and reflection. Students who can plan their learning, monitor their understanding, select appropriate strategies, and evaluate their progress are more likely to engage effectively with programming problems. Research on computer programming

©Authors
This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).

education has identified self-regulated learning strategies such as problem solving, knowledge acquisition, study environment management, metacognitive activities, and time management (Cheng et al., 2024). An experimental study examined the relationship between CT, SRL, and programming skills, including integrating instructional approaches that foster both students' cognitive competencies and self-regulatory skills in programming education (Susilowati et al., 2024).

The increasing use of Generative Artificial Intelligence (GenAI) has created new potentials for supporting programming courses. GenAI applications such as ChatGPT provide elaborations, cases, debugging assistance, feedback, and alternative solutions during learning tasks. ChatGPT is now used in higher education and is linked to research (Baig & Yadegaridehkordi, 2024). In programming education, an experimental study with undergraduate students found that using a GenAI-based tool was associated with changes in CT, programming self-efficacy, and learning enthusiasm (Yilmaz & Karaoglan Yilmaz, 2023). Another study examined how AI-generated content can support programming learning and computational thinking (Shang & Geng, 2024).

Despite these opportunities, GenAI in programming education has some concerns in pedagogical contexts. Students may use AI tools to obtain completed code rather than to understand the reasoning behind the solution. Some research has also examined ChatGPT's integration into programming instruction by assessing its influence on students' programming behaviors, learning performance, and perceptions of learning progress. These findings provide insights into how AI-assisted programming environments may affect students' engagement with programming tasks and their learning experiences (Sun et al., 2024). Research on AI chatbots in programming courses has also examined how students use these tools and the implications for learning (Groothuijsen et al., 2024). Therefore, the presence of GenAI alone does not guarantee meaningful learning. The way AI assistance is structured within the learning process is an important consideration.

One possible approach is adaptive learning. Adaptive learning provides learning content, activities, feedback, and learning paths based on learners' characteristics, prior knowledge, and performance. This approach is particularly relevant in programming courses because students often enter the course with different levels of programming knowledge and problem-solving ability. In Indonesia, researchers have explored adaptive learning through technology-supported learning media that integrate Android and web platforms (Saputri & Fransisca, 2026). Research on adaptive learning in higher education reported potential effects on academic results and student participation (Hooshyar et al., 2024). Relevant research on personalized, technology-enhanced learning also found positive effects on learning outcomes and highlighted the potential of combining learner modeling with adaptive support and scaffolding (Hooshyar et al., 2024).

Combining GenAI with adaptive learning offers an opportunity to address some limitations of using GenAI as a general-purpose learning tool. Rather than providing the same assistance to every student, an adaptive GenAI-based model can use information about students' performance and learning needs to determine the type and level of support provided. For example, students who struggle to understand a programming concept can receive additional explanations and examples, while students who demonstrate stronger understanding can receive more challenging problems. GenAI can also provide hints, guiding questions, debugging feedback, and reflection prompts rather than immediately providing complete solutions. Although research has examined GenAI-supported programming and personalized adaptive learning, further empirical work is needed to examine how to integrate these approaches into programming education (Yilmaz & Karaoglan Yilmaz, 2023; Sun et al., 2024; Hooshyar et al., 2024).

The existing literature therefore indicates an important research gap. Previous studies have investigated the use of GenAI in programming education (Yilmaz & Karaoglan Yilmaz, 2023;

Groothuijsen et al., 2024), while other studies have examined adaptive or personalized learning in higher education (Saputri & Fransisca, 2026; Hooshyar et al., 2024). Research has also examined CT and SRL as important elements in programming learning (Kong & Wang, 2024), (Susilowati et al., 2024), (Cheng et al., 2024), (Ramírez-Echeverry et al., 2025), (Cheng et al., 2024). However, empirical evidence on the effectiveness of integrating Generative AI into an adaptive learning model that simultaneously supports CT and SRL in Algorithm and Programming courses at the higher education level remains limited. This gap matters because GenAI's effectiveness should be evaluated not only by its ability to generate programming solutions but also by its contribution to students' thinking processes and learning regulation. The novelty lies in integrating GenAI, adaptive learning, computational thinking, and self-regulated learning into a single instructional model.

To fill in this gap, this research applies a Generative AI-based adaptive learning model in an Algorithm and Programming course. The model was developed through Research and Development (R&D) and integrated with the ADDIE model, comprising Analysis, Design, Development, Implementation, and Evaluation. The learning model integrates diagnostic assessment, learner profiling, adaptive learning paths, Generative AI-based scaffolding, adaptive practice, feedback, and reflection. The model's main principle is to support students according to their learning needs rather than simply providing direct programming solutions. Through this approach, students are expected to remain actively involved in analyzing problems, developing algorithms, checking solutions, and reflecting on their learning progress.

This research is novel in integrating Generative AI, adaptive learning, Algorithm and Programming, computational thinking, and self-regulated learning into a single instructional model and evaluating its effectiveness in a higher education context. The study focuses specifically on Informatics Engineering Education students at Universitas Putra Indonesia YPTK. The study evaluates the model's effectiveness based on changes in students' computational thinking and self-regulated learning across two measurement periods. Thus, this research aims to provide research-based evidence on using Generative AI-based adaptive learning as an instructional approach to support both cognitive and self-regulatory aspects of programming courses. According to the identified problem and research gap, this research aims to examine the effectiveness of a Generative AI-based adaptive learning model in enhancing students' computational thinking and self-regulated learning in Algorithm and Programming.

2. Method

This research used a Research and Development (R&D) approach combined with the ADDIE model across five stages: Analysis, Design, Development, Implementation, and Evaluation. This research primarily examined the effectiveness of a Generative AI-based adaptive learning model in Algorithm and Programming learning. The effectiveness test used a one-group pretest-posttest design. The design can be represented as follows:

$$O_1 \rightarrow X \rightarrow O_2$$

Where:

O_1 = pretest

X = implementation of the Generative AI-based adaptive learning model

O_2 = posttest.

The pretest was administered before implementing the model, while the posttest was administered after students completed the learning activities. The differences between the pretest and posttest results were used to determine changes in students' computational thinking and self-

regulated learning. The study was conducted over one academic semester, covering the implementation of the learning model and the evaluation of students' learning outcomes.

The participants were 35 students enrolled in the Informatics Engineering Education program at Universitas Putra Indonesia YPTK. The students were taking an Algorithm and Programming course. The participants were selected based on their involvement in the course in which the developed learning model was implemented. Students had different levels of prior knowledge and programming ability. This condition was relevant to implementing an adaptive learning model because the model was designed to provide learning support based on students' individual learning needs and progress.

The learning model integrated adaptive learning principles with Generative AI-based scaffolding to provide support based on students' individual needs and progress. The implementation consisted of seven main components: (1) diagnostic assessment, (2) learner profiling, (3) adaptive learning paths, (4) Generative AI scaffolding, (5) adaptive practice, (6) feedback, and (7) reflection and progress evaluation. ChatGPT and Gemini served as Generative AI tools to support students with explanations, examples, hints, guiding questions, feedback, additional exercises, and reflection prompts. The learning materials covered fundamental Algorithm and Programming concepts, including sequence, selection, repetition, arrays, functions, and problem-solving. Students' diagnostic results and learning progress guided the selection of learning activities and the level of support provided. The learning process followed a continuous cycle: Diagnostic Assessment → Learner Profiling → Adaptive Learning Path → GenAI Scaffolding → Practice → Feedback → Reflection → Progress Evaluation. This approach enabled students to receive additional support when needed and progress to more challenging activities as their understanding developed. The computational thinking instrument contained four indicators to assess students' computational thinking abilities in Algorithm and Programming courses.

Table 1. Computational Thinking Indicators

No.	Indicator	Measurement Description
1	Decomposition	The ability to decompose complex programming challenges into simpler, manageable parts.
2	Pattern Recognition	The competencies to identify similarities, patterns, and relationships among programming problems.
3	Abstraction	The competencies to identify relevant information and eliminate unnecessary details when solving problems.
4	Algorithmic Thinking	The competencies to formulate logical, systematic, and sequential steps to solve programming problems.

The self-regulated learning instrument consisted of five indicators to assess students' competencies to manage and regulate their learning progress.

Table 2. Self-Regulated Learning Indicators

No.	Indicator	Measurement Description
1	Goal Setting	Ability to establish learning objectives and identify expected learning outcomes.
2	Planning	Ability to organize learning activities, allocate time, and determine appropriate learning steps.
3	Self-Monitoring	Ability to monitor learning progress and identify difficulties during learning activities.

4	Learning Strategies	Ability to select and apply appropriate strategies to understand learning materials and solve problems.
5	Self-Evaluation and Reflection	Ability to evaluate learning outcomes, identify weaknesses, and reflect on learning experiences for improvement.

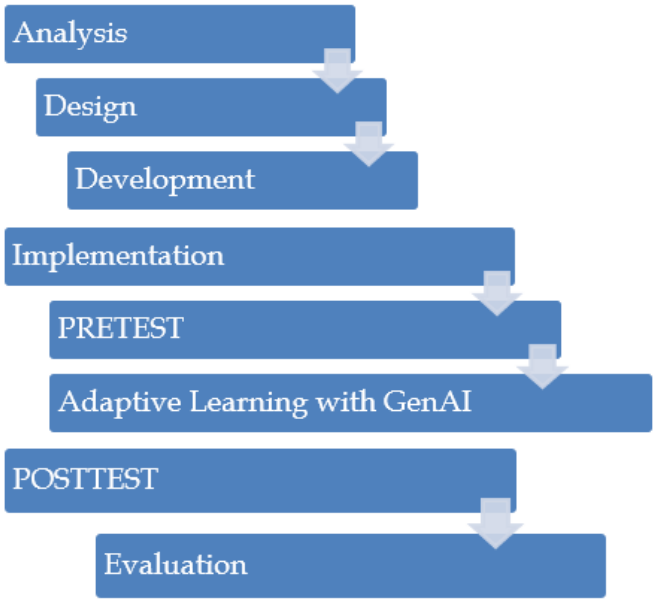


Figure 1. ADDIE Procedure

The analysis stage identified learning characteristics, students' prior knowledge, learning difficulties, and instructional requirements for Algorithm and Programming. The design stage defines the structure of the adaptive learning model, learning activities, assessment instruments, adaptive learning paths, and Generative AI scaffolding. The development stage involved developing the learning courses and the adaptive learning model. The Generative AI support was integrated into learning activities to provide explanations, hints, examples, feedback, and reflection prompts. In the Implementation stage, the model was implemented in the Algorithm and Programming course with 35 students. Before the learning activities began, students completed the pretest. They then participated in the adaptive learning activities supported by Generative AI. After completing the learning activities, students completed the posttest. The evaluation stage analyzed pretest and posttest results to determine changes in CT and SRL. The evaluation focused on the learning model's effectiveness based on improvements in students' scores.

The data were evaluated through a percentage-based analysis to assess effectiveness. This value was obtained by expressing the achieved score as a proportion of the maximum attainable score:

$$P = \frac{X}{X_{max}} \times 100\%$$

Where:
P = percentage score,
X = obtained score,
Xmax = maximum possible score.
The pretest and posttest percentages were compared to determine the improvement in students' computational thinking and self-regulated learning. The percentage-point improvement was calculated as:

$$\Delta P = P_{posttest} - P_{pretest}$$

The effectiveness analysis was conducted separately for the computational thinking and self-regulated learning measures. Higher posttest scores than pretest scores indicated improvement after implementing the Generative AI-based adaptive learning model.

3. Result

The study evaluated the effectiveness of the Generative AI-based adaptive learning model by comparing students' CT and SRL scores between the two measurement periods. The analysis used percentage-based scores obtained from the pretest and posttest. The research findings are presented in tables to provide a clear overview of the collected data.

Table 3. Pretest and Posttest Results on Computational Thinking

No.	Computational Thinking Indicators	Pretest Percentage (%)	Posttest Percentage (%)
1	Decomposition	62.50	77.50
2	Pattern Recognition	65.00	79.00
3	Abstraction	63.50	78.00
4	Algorithmic Thinking	65.68	80.02
	Overall CT	**64.17**	**78.63**

Table 4. Pretest and Posttest Results on Self-Regulated Learning

No.	Self-Regulated Thinking Indicators	Pretest Percentage (%)	Posttest Percentage (%)
1	Goal setting	68.00	80.50
2	Planning	66.50	79.00
3	Self-Monitoring	67.00	79.50
4	Learning Strategies	68.50	80.50
5	Self-Evaluation and Reflection	67.45	79.90
	Overall SRL	**67.49**	**79.86**

The results show an increase in both measured outcomes after implementing the learning model. Computational thinking increased from 64.17% in the pretest to 78.63% in the posttest, representing an improvement of 14.46 percentage points. Self-regulated learning increased from 67.49% to 79.86%, representing an improvement of 12.37 percentage points. This improvement indicates that students performed better in both CT and SRL after participating in the Generative AI-based adaptive learning activities.

Figure 2 compares students' computational thinking and self-regulated learning scores before and after implementing the Generative AI-based adaptive learning model.

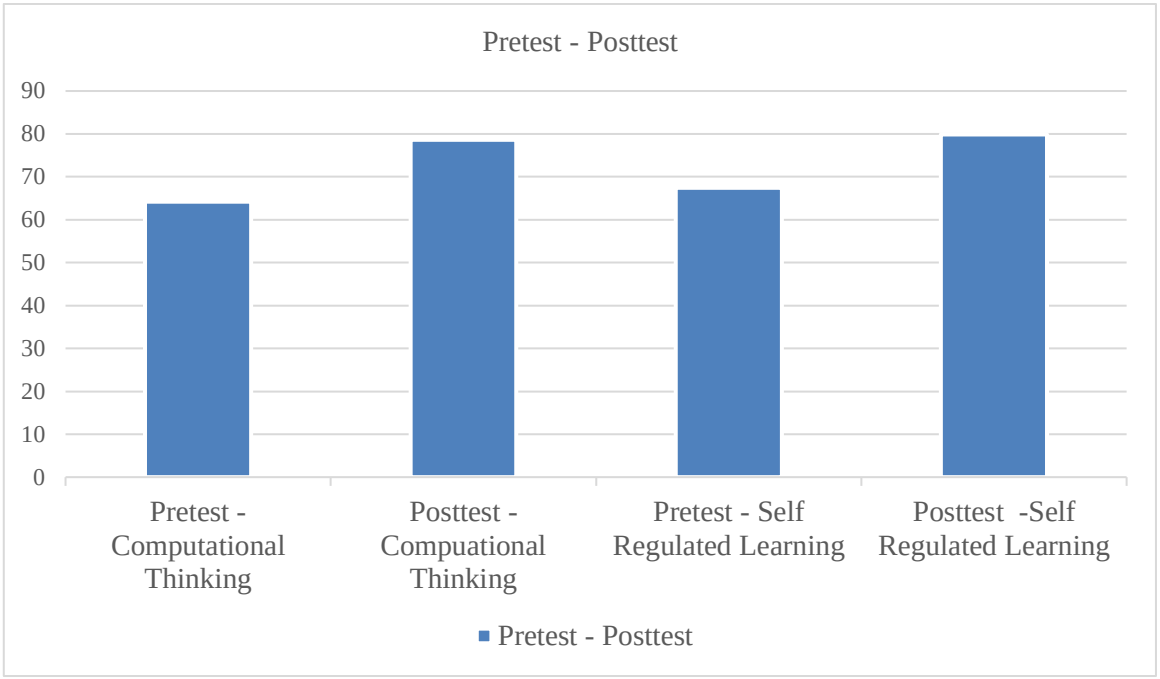


Figure 2. Comparison of Pretest and Posttest on CT and SRL

The computational thinking results indicate a positive change after implementing the learning model. The pretest score of 64.17% increased to 78.63% in the posttest. This improvement suggests that students became better able to approach learning tasks that required systematic problem solving. The improvement can be related to the combination of adaptive learning and Generative AI scaffolding. The adaptive learning component provided activities based on students' learning needs, while Generative AI provided additional explanations, examples, hints, and feedback when students experienced difficulties. This combination allowed students to receive more targeted support during the learning process.

The finding is consistent with Yilmaz and Karaoglan Yilmaz (2023), who reported positive effects of a generative AI-based tool on students' computational thinking skills. Similar findings have been reported in studies examining AI-generated content and AI-supported programming learning (Shang & Geng, 2024; Sun et al., 2024; Groothuijsen et al., 2024). These studies suggest that GenAI can provide useful learning assistance when students interact with it as part of an instructional process.

The present finding also supports the importance of structured learning support. Rather than allowing students to rely entirely on AI-generated solutions, the model positioned GenAI as a scaffold. Students were encouraged to understand the problem, consider possible solutions, receive guidance, and revise their understanding. This approach may have contributed to improved computational thinking because students were still required to engage in the problem-solving process.

The result is also consistent with previous research showing a relationship between computational thinking and self-regulated learning (Kong & Wang, 2024; Susilowati et al., 2024). Students who actively monitor their learning and evaluate their progress may be better prepared to engage with complex problem-solving tasks.

Self-regulated learning increased from 67.49% in the pretest to 79.86% in the posttest, resulting in an improvement of 12.37 percentage points. This finding indicates that students demonstrated stronger learning-regulation behaviors after participating in the adaptive learning activities.

The improvement may be related to several features of the proposed model. First, the adaptive learning pathway encouraged students to work according to their current level of understanding. Second, the model gave students opportunities to monitor their progress through practice and feedback. Third, reflection activities encouraged students to evaluate their understanding and identify areas that still needed improvement.

These characteristics align with self-regulated learning, which involves planning, monitoring, strategy use, and self-evaluation. Cheng et al. (2024) emphasized the importance of self-regulated learning strategies in programming education, while Radović et al. (2024) showed that appropriate regulation support can influence students' learning processes and outcomes. The finding is also relevant to research on adaptive feedback. Mejeh et al. (2024) reported that adaptive feedback can provide personalized support for developing self-regulated learning. In the present study, feedback from the learning environment and GenAI provided students with opportunities to identify difficulties and reconsider their learning strategies.

However, the improvement in SRL should not be attributed solely to Generative AI. The observed improvement resulted from the overall learning model, which combined adaptive learning pathways, practice, feedback, and reflection. GenAI functioned as one component of this learning environment.

4. Discussion

The findings indicate that the Generative AI-based adaptive learning model may have contributed to both computational thinking and self-regulated learning in programming education. These outcomes complement each other, as computational thinking focuses on systematic problem-solving, while SRL emphasizes students' ability to manage their learning. Their simultaneous improvement highlights the importance of integrating cognitive development with independent learning strategies.

These findings are consistent with previous studies reporting the benefits of adaptive and technology-supported learning in higher education (Hooshyar et al., 2024; Mejeh et al., 2024; Fransisca & Saputri, 2023; Saputri & Fransisca, 2026). The present study extends this work by integrating Generative AI as a scaffold within adaptive learning pathways. Rather than providing complete solutions, ChatGPT and Gemini delivered explanations, hints, examples, feedback, and reflection prompts based on students' learning needs. This approach encourages active problem-solving while reducing the risk of excessive dependence on AI-generated answers, a concern also highlighted in previous research (Denny et al., 2024; Baig & Yadegaridehkordi, 2024).

Overall, the study demonstrates the potential of integrating adaptive learning and Generative AI to support both computational thinking and self-regulated learning. Nevertheless, the findings should be interpreted in consideration of the study context and design. Further research involving larger samples, comparison groups, and longer implementation periods is needed to examine the model's effectiveness and applicability across different learning contexts.

5. Conclusion

This study examined changes in students' computational thinking and self-regulated learning after implementing a Generative AI-based adaptive learning model. The findings showed that the computational thinking score increased from 64.17% in the pretest to 78.63% in the posttest, representing an improvement of 14.46 percentage points. Similarly, the self-regulated learning score increased from 67.49% to 79.86%, representing an improvement of 12.37 percentage points. These

results indicate positive changes in both learning outcomes after implementing the model. However, they do not establish that the model alone caused the observed improvements.

Several limitations should be considered when interpreting these findings. The study involved only 35 students from a single Informatics Engineering Education program and ran for one academic semester. In addition, the one-group pretest-posttest design did not include a comparison group, limiting the ability to attribute the observed changes specifically to the intervention. The findings are also dependent on the instruments used to assess computational thinking and self-regulated learning, which may not capture all dimensions of these constructs.

6. References

Baig, M. I., & Yadegaridehkordi, E. (2024). ChatGPT in higher education: A systematic literature review and research challenges. *International Journal of Educational Research, 127*, Article 102411. https://doi.org/10.1016/j.ijer.2024.102411

Cheng, G., Zou, D., Xie, H., & Wang, F. L. (2024). Exploring differences in self-regulated learning strategy use between high- and low-performing students in introductory programming: An analysis of eye-tracking and retrospective think-aloud data from program comprehension. *Computers & Education, 208*, Article 104948. https://doi.org/10.1016/j.compedu.2023.104948

Denny, P., Prather, J., Becker, B. A., Finnie-Ansley, J., Hellas, A., Leinonen, J., Luxton-Reilly, A., Reeves, B. N., Santos, E. A., & Sarsa, S. (2024). Computing education in the era of generative AI. *Communications of the ACM, 67*(2), 56–67. https://doi.org/10.1145/3624720

Fransisca, M., & Saputri, R. P. (2023). Digital technology's implementation in hybrid learning at higher educational level. *The Indonesian Journal of Computer Science, 12*(3). https://doi.org/10.33022/ijcs.v12i3.3213

Groothuijsen, S., van den Beemt, A., Remmers, J. C., & van Meeuwen, L. W. (2024). AI chatbots in programming education: Students' use in a scientific computing course and consequences for learning. *Computers and Education: Artificial Intelligence, 7*, Article 100290. https://doi.org/10.1016/j.caeai.2024.100290

Hooshyar, D., Weng, X., Sillat, P. J., Tammets, K., Wang, M., & Hämäläinen, R. (2024). The effectiveness of personalized technology-enhanced learning in higher education: A meta-analysis with association rule mining. *Computers & Education, 223*, Article 105169. https://doi.org/10.1016/j.compedu.2024.105169

Jing, Y., Wang, H., Chen, X., [lengkapi nama seluruh penulis]. (2024). What factors will affect the effectiveness of using ChatGPT to solve programming problems? A quasi-experimental study. *Humanities and Social Sciences Communications, 11*, Article 319. https://doi.org/10.1057/s41599-024-02751-w

Kong, S.-C., & Wang, Y.-Q. (2024). Dynamic interplays between self-regulated learning and computational thinking in primary school students through animations and worksheets. *Computers & Education, 220*, Article 105126. https://doi.org/10.1016/j.compedu.2024.105126

Mejeh, M., Sarbach, L., & Hascher, T. (2024). Effects of adaptive feedback through a digital tool: A mixed-methods study on the course of self-regulated learning. *Education and Information Technologies, 29*, 1–43. https://doi.org/10.1007/s10639-024-12510-8

Radović, S., Seidel, N., Menze, D., & Kasakowskij, R. (2024). Investigating the effects of different levels of students' regulation support on the learning process and outcome: In search of the optimal level of support for self-regulated learning. *Computers & Education, 215*, Article 105041. https://doi.org/10.1016/j.compedu.2024.105041

Ramírez-Echeverry, J. J., Restrepo-Calle, F., & Jiménez, S. T. (2025). Self-regulated learning strategies in computer programming education. *European Journal of Education, 60*(1), Article e70052. https://doi.org/10.1111/ejed.70052

Saputri, R. P., & Fransisca, M. (2026). Development of hybrid learning media integrating Android and web platforms with an adaptive learning approach. *Jurnal Profesi Keguruan, 12*(1), 71–85. https://doi.org/10.15294/jpk.v12i1.52243

Shang, S., & Geng, S. (2024). Empowering learners with AI-generated content for programming learning and computational thinking: The lens of extended effective use theory. *Journal of Computer Assisted Learning, 40*(4), 1941–1958. https://doi.org/10.1111/jcal.12996

Sun, D., Boudouaia, A., Zhu, C., & Li, Y. (2024). Would ChatGPT-facilitated programming mode impact college students' programming behaviors, performances, and perceptions? An empirical study. *International Journal of Educational Technology in Higher Education, 21*, Article 14. https://doi.org/10.1186/s41239-024-00446-5

Susilowati, D., Rahim, A., Ananta, G. P., Saputri, D. S. C., Hasanah, U., & Ria, R. R. P. (2024). Do computational thinking and self-regulated learning affect computer programming problem-solving skills? An experimental study. *Jurnal Kependidikan: Jurnal Hasil Penelitian dan Kajian Kepustakaan di Bidang Pendidikan, Pengajaran dan Pembelajaran, 10*(3), 1145–1157. https://doi.org/10.33394/jk.v10i3.12415

Yilmaz, R., & Karaoglan Yilmaz, F. G. (2023). The effect of generative artificial intelligence (AI)-based tool use on students' computational thinking skills, programming self-efficacy, and motivation. *Computers & Education: Artificial Intelligence, 4*, Article 100147. https://doi.org/10.1016/j.caeai.2023.100147