Jurnal Pendidikan Progresif
e-ISSN: 2550-1313 | p-ISSN: 2087-9849
URL: https://jpp.fkip.unila.ac.id/index.php/jpp
Jounal's DOI: http://doi.org/10.23960/jpp

Vol. 16, No. 1, 2026
Article ID: 1355
Pages: 345-366

Enhancing High School Students' Basic Programming Skills and Self-Efficacy: A Drill-and-Practice Game-Based Approach

Isnania Lestari[1], Ferry Marlianto[2], Dewi Sulistiyarini[2,*], Febrianto Sabirin[2], & Noorjima Abd Wahab[3]

[1]Department of Information Systems, University PGRI Pontianak, Indonesia
[2]Department of Information Technology Education, University PGRI Pontianak, Indonesia
[3]Department of Social Sciences, University Sultan Zainal Abidin, Malaysia

*Corresponding email: dhewysulis@gmail.com

Received: 24 November 2025 *Accepted: 26 January 2026* *Published: 15 February 2026*

Abstract: Drill-and-practice educational games combine repetitive practice with gamification through rewards, achievements, challenges, feedback, and progress tracking to support gradual skill development. In this study, the drill-and-practice element focuses on repeated, structured practice of basic programming exercises with immediate corrective feedback and multiple attempts until mastery. This study examined the effect of a drill-and-practice educational game on basic programming skills and programming self-efficacy among senior high school students. This quasi-experimental study used a Non-Equivalent Control Group Design. The sample comprised 179 tenth-grade students from six intact classes, selected via cluster random sampling. Three classes constituted the control group, with 89 students, and three classes constituted the experimental group, with 90 students. Students completed a basic programming pretest and posttest, as well as a self-efficacy questionnaire. Data were analyzed using descriptive statistics and MANCOVA with pretest scores as covariates. Descriptively, students' overall basic programming scores increased from a mean of 38.70 to 70.49 in the posttest, while self-efficacy increased from a mean of 59.82 to 74.05. The multivariate test showed a significant group effect (Pillai's Trace = 0.528; $F = 97.442$; $p < 0.001$; Partial $ç^2 = 0.528$). Between-subjects tests indicated significant improvements in basic programming skills ($F = 115.102$; $p < 0.001$; Partial $ç^2 = 0.397$) and self-efficacy ($F = 86.697$; $p < 0.001$; Partial $ç^2 = 0.331$), with the experimental group achieving higher posttest outcomes than the control group. A drill-and-practice educational game can enhance both cognitive outcomes in basic programming and affective outcomes in self-efficacy, and can be considered an interactive learning alternative for programming topics at the high school level. Future studies should add performance-based coding tasks and broader samples to validate and extend these results.

Keywords: educational games, drill and practice, basic programming skills, self-efficacy.

Article's DOI: https://doi.org/10.23960/jpp.v16i1.pp345-366

INTRODUCTION

Programming has become an important skill because it is a core competence in the Industrial Revolution 4.0 and Society 5.0, where rapid technological change is expected to reshape jobs and skill demands. The Future of Jobs Report 2025 identifies programming as a prominent skill that differentiates growing from declining jobs, suggesting that programming competence underpins emerging technology domains across industries (World Economic Forum, 2025). Therefore, programming competence can be regarded as a foundational skill for developing and applying digital technologies, including AI-driven systems, data-intensive applications, and cybersecurity solutions. Research also indicates that programming skills are increasingly valued across sectors because they support efficiency and innovation in areas such as business management and process automation (Ehlinger

& Stephany, 2024), which aligns with broader evidence on digital transformation (Montandon et al., 2021). For this reason, programming skills are not only technical abilities for writing code but also foundational competencies that support innovation and future digital study and career pathways.

In Indonesia, the supply of skilled digital workers, particularly in programming-related roles, remains limited relative to projected national needs. Government analyses estimate that Indonesia will need approximately 9 million digital talent by 2030, equivalent to roughly 600,000 additional advanced digital workers per year, highlighting a sizable gap between demand and current supply (SMERU Research Institute, 2022). One example of persistent hiring difficulty is in software and application development roles, which require specific competencies, including programming skills (World Economic Forum, 2025). In addition to the quantity gap, the shortage is also associated with a competence-alignment problem between graduates' skills and industry needs, which helps explain why technical vacancies remain difficult to fill (Gayatri et al., 2023). Therefore, strengthening foundational programming at the senior high school level is important for expanding the talent pipeline earlier and cultivating students' interest and confidence, thereby preparing them to pursue advanced programming and related career pathways.

At the secondary level, students still face various obstacles in learning programming, such as difficulty understanding abstract concepts, algorithmic logic, and complex programming language syntax. Limited resources and facilities can also constrain optimal learning processes (Li, 2023). Unengaging learning methods may lead to low student motivation, further exacerbating these challenges (Napalit et al., 2023). Therefore, selecting appropriate learning methods and media can positively affect students' learning outcomes (Eteng et al., 2022).

Beyond improving cognitive outcomes, the choice of learning methods and media can also shape students' affective readiness, particularly their confidence to engage with programming tasks, making self-efficacy a key factor to consider in secondary-level programming instruction. Self-efficacy is an individual's belief in their ability to achieve goals (Schunk & DiBenedetto, 2021). In the context of education, particularly programming learning, self-efficacy plays a crucial role because it influences how students understand concepts, complete coding assignments, and address errors or bugs that arise during the learning process. Students with high self-efficacy in programming tend to be more confident in trying new algorithms, more persistent in the face of errors, and able to explore various solutions independently (Abdunabi et al., 2024; Q. Li et al., 2025). Conversely, students with low self-efficacy often feel doubtful, give up quickly when faced with coding errors, and tend to rely on others for help (Tseng et al., 2024).

Low self-efficacy in programming learning can be influenced by unsuccessful prior experiences, monotonous learning methods, limited support from instructors or peers, and the complexity of programming logic and syntax (Kovari & Katona, 2023; Tsai, 2019). These conditions may reduce learning motivation and affect achievement in informatics. Therefore, self-efficacy is important in program design for both understanding the factors that shape students' confidence and designing teaching strategies that support persistence and continued learning (Liu et al., 2023; MIHCI Türker & Pala, 2020; Smit et al., 2025).

In response to projected demand for the digital workforce, the government has expanded programming training from basic to advanced levels through initiatives such as the Digital Talent Scholarship, which has scaled to tens of thousands of participants annually and is being expanded to increase reach in the coming years

(Chen et al., 2023). However, short-course training alone is unlikely to be sufficient if students do not develop strong foundations and sustained interest in programming before entering higher education or the labor market. Therefore, strengthening programming instruction at the secondary school level is a relevant upstream effort to build readiness and continuity in students' learning pathways. This direction is supported by K–12 programming education research showing that structured programming learning in schools contributes to the development of students' programming ability and related competencies, underscoring the value of strengthening foundations earlier (Sun et al., 2022).

Integrating educational games with a drill-and-practice approach can strengthen programming learning by combining structured repetition with an engaging practice environment. Drill and practice are effective because it emphasizes repeated, systematic, and structured exercises that support gradual mastery (Sudirman & Zain, 2023). Prior studies indicate that repeated practice can improve students' understanding (Lie et al., 2023), develop logical reasoning (Cheng et al., 2023), and increase learning enthusiasm (Elfeky & Elbyaly, 2023). In this study, the proposed integration is not presented as a new concept of gamification, but rather as a structured drill-and-practice cycle implemented within a programming educational game that emphasizes repeated attempts, immediate feedback, and progressively difficult tasks for basic programming learning.

Educational games can support programming learning by providing interactive and contextual activities that help students apply concepts through simulations and challenges (Pan et al., 2024). Game elements such as challenges, points, and level progression can enhance motivation and reinforce technical understanding (Maryono et al., 2022; Videnovik et al., 2023; Zhan et al., 2022). As students engage in practice-oriented tasks, their programming performance can improve alongside the development of computational thinking and problem-solving (Giannakoulas & Xinogalos, 2024). When educational games are designed around repeated practice toward clear goals, they can encourage sustained engagement and gradual improvement in programming-related procedures (Córdova-Esparza et al., 2024).

Beyond cognitive outcomes, a drill-and-practice approach is also relevant for strengthening programming self-efficacy because repeated successful attempts, corrective feedback, and gradual task progression can create mastery-oriented learning experiences that support students' confidence in handling programming tasks. Programming self-efficacy is important because it is associated with students' confidence, persistence, and independence when facing programming challenges (Abdunabi et al., 2024; Q. Li et al., 2025; Tseng et al., 2024). In other words, improving programming performance without supporting students' confidence may not be sufficient for sustained learning, especially when learners frequently encounter errors and debugging demands during early programming practice.

Despite the importance of programming instruction in digital-era curricula, evidence evaluating the co-occurrence of basic programming skills and programming self-efficacy at the secondary school level remains limited. Recent syntheses show that studies on game-based and gamified programming predominantly assess cognitive outcomes, such as programming performance and engagement, whereas self-efficacy is less consistently examined as a primary affective outcome (Videnovik et al., 2023; Zhan et al., 2022; Gundersen & Lampropoulos, 2025). Therefore, this study strengthens its contribution by using a drill-and-practice educational game as the learning intervention and by examining two outcomes simultaneously, namely basic programming skills as a cognitive outcome and programming self-efficacy as an affective outcome, within a quasi-experimental classroom setting.

Moreover, drill-and-practice or tutor-like game approaches for programming have been reported more often in higher education settings than in secondary education. This pattern is reflected in evidence syntheses, which show that university-level implementations predominate among game-based learning studies in computer science (Videnovik et al., 2023). This trend is also illustrated by empirical studies in higher education that embed repeated practice, immediate feedback, and progress indicators in programming learning games or tutors (Grey & Gordon, 2023; Hainey & Baxter, 2024). This makes it important to examine a comparable approach at the high school level while ensuring accessibility that fits local conditions. In Indonesia, mobile access is more prevalent than computer ownership: a large share of the population owns mobile phones, whereas household computer ownership remains relatively low (Badan Pusat Statistik, 2024). Therefore, the web-based drill-and-practice educational game used in this study was selected because it can be accessed across devices and because its interaction design supports iterative practice with timely feedback, which is a benefit frequently highlighted in K–12 serious game research when game design decisions align with learners' needs (Gundersen & Lampropoulos, 2025). In addition, storing learners' activity and progress records in a database enables continuity of practice. It supports teacher monitoring through learning analytics dashboards, which are commonly used to visualize students' progress and learning behaviors for pedagogical decision-making (Masiello et al., 2024; Rahimi & Shute, 2021). The educational game interface used in this study is shown in Figure 1.

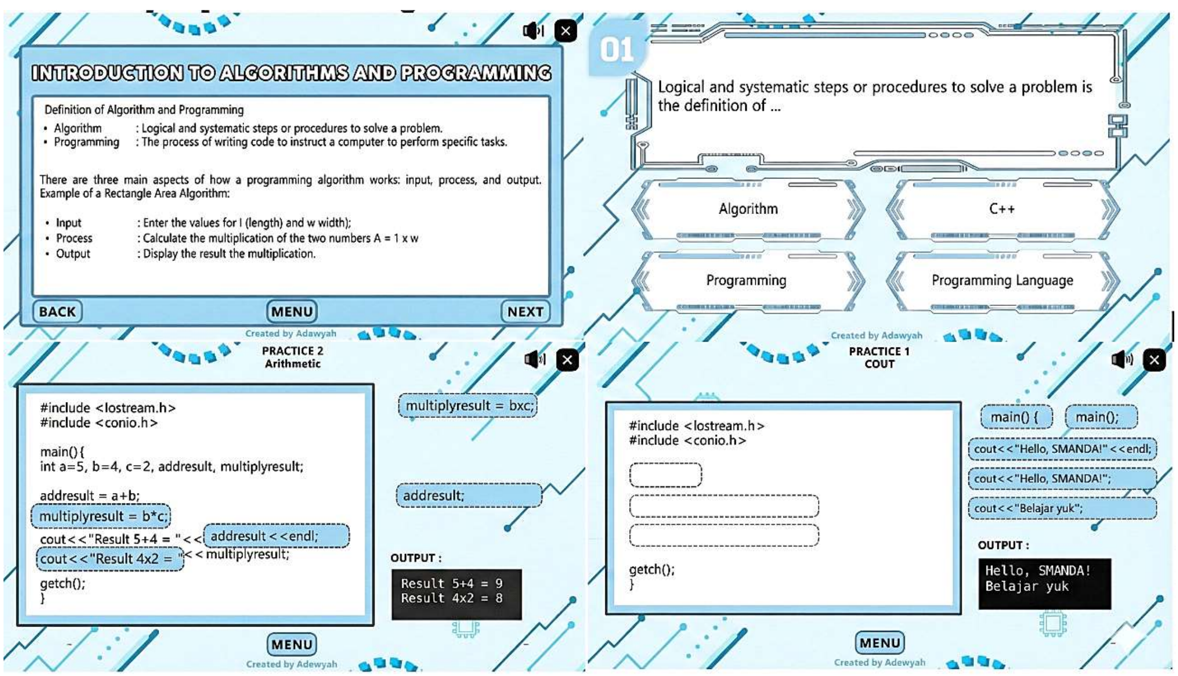

Figure 1. Interface of the drill-and-practice educational game for basic programming

Given the growing need to strengthen programming foundations in secondary education and the potential of educational games to support structured practice, this study investigates the effectiveness of a drill-and-practice educational game in senior high school programming instruction. Unlike studies that focus only on cognitive achievement or examine affective factors separately, this study evaluates basic programming skills and programming self-efficacy as complementary outcomes. To clarify the contribution and guide the analysis, the study formulates the following research questions:

RQ1. Is there a simultaneous difference in basic programming skills and programming self-efficacy between students who learn using the drill-and-practice educational game and those who learn without it?
RQ2. Is there a difference in basic programming skills between students who learn using the drill-and-practice educational game and those who learn without it?
RQ3. Is there a difference in programming self-efficacy between students who learn using the drill-and-practice educational game and those who learn without it?

■ METHOD

Participant

The study was conducted in Pontianak City, Indonesia, in collaboration with three private senior high schools, namely SMA Islam Bawari Pontianak, SMA Mujahidin Pontianak, and SMA Bina Utama Pontianak. These schools were selected because they reflect the characteristics of private education in the city and have already implemented Informatics courses for tenth-grade students, particularly in basic programming. The research population comprised all tenth-grade students at the three participating schools, representing 13 classes and a total of 376 students. The sample was determined using cluster random sampling, in which intact classes were randomly selected as the sampling unit. Six tenth-grade classes comprising 179 students were selected as the research sample. Three classes, each comprising 89 students, were assigned to the control group, and three classes, each comprising 90 students, were assigned to the experimental group. Each participating school contributed one control class and one experimental class to ensure proportional distribution across schools.

Research Design and Procedure

This study employed a quasi-experimental design, namely one that involves an experimental group and a control group without full random assignment at the individual level, yet still allows examination of cause-and-effect relationships (Capili & Anastasi, 2024; Miller et al., 2020). The specific design was a non-equivalent control group design. The experimental group received instruction supported by a drill-and-practice educational game. In contrast, the control group received regular instruction using instructional media commonly used by teachers in the participating schools, namely e-book and presentation-based materials. The purpose of selecting a regular-instruction control condition was to examine whether the drill-and-practice educational game could serve as an alternative to conventional classroom media.

The learning content was delivered in the Informatics course for tenth-grade students and covered four topics: the definition of programming algorithms; input/output and flowcharts; branching algorithms; and iteration (looping) algorithms. The study was conducted over five weeks and consisted of three stages: pre-test, treatment, and post-test. The following research procedures were carried out:

To reduce the possibility that group differences were driven primarily by novelty or interactivity rather than the intended instructional approach, instructional conditions were controlled as follows. In each school, the experimental and control classes were taught by the same teacher. Both groups received the same learning objectives, content coverage, time allocation, and equivalent assignments and practice tasks.

Table 1. Research procedures

Phase	Time	Experimental Group (90 students)	Control Group (89 students)
Pre-test	Week 1	Measurement of Self-Efficacy, followed by Basic Programming Skills Test	Measurement of Self-Efficacy, followed by Basic Programming Skills Test
Treatment	Week 2–4	Using Drill & Practice Educational Game	Without Drill & Practice Educational Game
Post-test	Week 5	Measurement of Self-Efficacy, followed by Basic Programming Skills Test	Measurement of Self-Efficacy, followed by Basic Programming Skills Test

Therefore, the planned difference between groups was primarily the learning medium used during instruction.

Instruments

This study employed two data-collection techniques: a test and a questionnaire. The Basic Programming Skills Test consisted of 25 pre-test and 25 post-test questions validated by the teachers involved in the study. The questions covered the understanding of programming algorithms, input-output-flowcharts, branching algorithms, and looping algorithms, all of which align with the core competencies of the 10th-grade Informatics subject. The questionnaire instrument was a self-efficacy questionnaire structured on a Likert scale. Indicators measured through the questionnaire included understanding basic programming concepts, algorithm design, writing basic syntax, using control structures, error correction, evaluating program logic, and perseverance in completing programming tasks (Karalar, 2023; Tsai et al., 2018). Initially, the questionnaire comprised 28 items; after a trial, 25 were deemed valid. The reliability test yielded a coefficient of 0.889, classified as very high; the instrument was deemed suitable for research use.

This study used two instruments: a Basic Programming Skills Test and a Programming Self-Efficacy Questionnaire.

Basic Programming Skills Test

Basic programming skills in this study were assessed using a multiple-choice test and a performance-based test. The multiple-choice test was administered as a pre-test and a post-test, with twenty-five items in each form. The pre-test and post-test were designed as parallel forms with comparable content coverage. The items measured students' mastery of tenth-grade informatics content, including the definition of programming algorithms, input/output and flowcharts, branching algorithms, and iteration or looping algorithms. Each correct answer was scored one, and each incorrect answer was scored zero, resulting in a total score ranging from zero to twenty-five. In addition, at the post-test, students completed a performance-based coding task comprising simple programming problems focused on input–output, branching, and looping. Students' code was scored using a concise rubric emphasizing functional correctness and basic logic, and the resulting practicum score was used as an additional measure of basic programming skills.

Content validity was evaluated through expert judgment by three Informatics teachers involved in the study. They reviewed the items for relevance to the curriculum, clarity, and alignment with the targeted competencies.

Table 2. Blueprint of the basic programming skills test

Topic Indicator	Operational Meaning	Number of Multiple-Choice Tests	Performance-Based Test
Definition of programming algorithms	Understanding basic definitions and foundational concepts in programming and algorithms, such as algorithm, compiler, variables, data types, and basic operators	6	-
Input, output, and flowcharts	Understanding standard input and output in C plus plus and interpreting flowchart symbols and outputs	6	Read and write subject score
Branching algorithms	Applying conditional reasoning using relational and logical expressions and conditional structures to determine program outputs	6	Determine the grade from a score
Iteration or looping algorithms	Applying repetition logic using for, while, do while, and nested loops, and interpreting loop outputs	7	Input "N" students' scores and compute the class average

Programming Self-Efficacy Questionnaire

The questionnaire instrument was a self-efficacy questionnaire structured on a Likert scale. The questionnaire indicators were derived from studies on computer programming self-efficacy for computer literacy education consist of understanding basic programming concepts, algorithm design, writing basic syntax, using control structures, error correction, evaluating program logic, and perseverance in completing programming tasks (Karalar, 2023; Tsai et al., 2018), and were operationalized into statements relevant to basic C++ programming for tenth-grade students. The initial questionnaire consisted of twenty-eight items rated from one, strongly disagree, to five, strongly agree.

Content validity was examined through expert judgment by three lecturers from the Information Technology Education program at Universitas PGRI Pontianak, who reviewed item relevance, clarity, and alignment with the intended indicators. Empirical item validity was tested using the Pearson product-moment correlation in a pilot study involving thirty-five tenth-grade students. Using a two-tailed significance level of 0.05, the critical correlation value was approximately 0.334. Items with item-total correlation values exceeding this value were retained. Based on the pilot results, items 6, 26, and 27 were removed, yielding 25 valid items. The reliability of the final questionnaire was evaluated using Cronbach's alpha, yielding a coefficient of 0.889, indicating very high internal consistency.

Table 3. Programming self-efficacy questionnaire indicators and items

Indicator	Meaning of Indicator	Item Number Before Tryout	Item Number After Tryout
Understanding basic concepts	Confidence in using basic programming elements such as data types, variables, arithmetic operators, and simple input output commands	1, 2, 3, 4	1, 2, 3, 4

Algorithm design	Confidence in planning solution steps before coding, including transforming problems into executable steps, writing pseudocode, and drawing flowcharts	5, 6, 7, 8	5, 7, 8
Basic C++ coding	Confidence in writing basic C++ code, implementing pseudocode into code, using correct syntax, and compiling and running a simple program	9, 10, 11, 12	9, 10, 11, 12
Use of control structures	Confidence in selecting and applying control structures such as sequence, branching, and looping to solve problems	13, 14, 15, 16	13, 14, 15, 16
Debugging	Confidence in identifying and fixing syntax errors and logical errors using compiler messages and tracing execution flow	17, 18, 19, 20	17, 18, 19, 20
Evaluating program logic	Confidence in understanding and evaluating logical expressions, combined conditions, and operator precedence in program conditions	21, 22, 23, 24	21, 22, 23, 24
Persistence	Confidence in persisting when facing difficulties, trying alternative strategies, and allocating additional time to complete programming tasks	25, 26, 27, 28	25, 28

Data Analysis

After data collection, descriptive statistics and Multivariate Analysis of Covariance (MANCOVA) were conducted. All statistical analyses were performed using IBM SPSS Statistics. Descriptive statistics were used to characterize the distribution of data, including the mean, standard deviation, and general trends in pre-test and post-test results across both groups. MANCOVA was chosen because this study had two dependent variables (basic programming skills and self-efficacy) and sought to control for initial differences among students by including pretest scores as a covariate. The multivariate approach allows simultaneous testing of the treatment effect across both outcomes and reduces the risk of errors that arise when testing them separately (Ateº et al., 2019). In addition, MANCOVA provides information on the combined effects across dependent variables, making it more suitable for evaluating the effects of using the Drill and Practice Educational Game on cognitive and affective dimensions simultaneously (Landler et al., 2022). Before running MANCOVA, a series of assumption tests were conducted: (1) multivariate normality was checked using Mahalanobis Distance; (2) linearity to ensure a linear relationship in each group; (3) slope homogeneity is used to ensure that the relationship between the covariate and the dependent variable is linear and consistent across all treatment groups; (4) Box's M is used to test the similarity of the covariance matrix between groups; and (5) Levene's test is used to test the homogeneity of error variance in each dependent variable.

RESULT AND DISCUSSION

Descriptive Results of Basic Programming Skills and Self-Efficacy

Descriptive statistics were used to provide an overview of students' basic programming skills and programming self-efficacy before and after the instructional period. The overall descriptive results are presented in Table 4.

Table 4 shows that the mean basic programming score increased from 38.70 on the pre-test to 70.49 on the post-test, indicating an improvement of 31.79 points. The mean self-efficacy score increased from 59.82 on the pre-test to 74.05 on the post-test, indicating an improvement of 14.23 points. The minimum and maximum scores also increased for both variables.

Table 4. Overall descriptive statistics

Variable	N	Mean	Median	SD	Variance	Min	Max
Basic Programming (Pre)	179	38.70	36.00	15.82	250.53	12.00	80.00
Self-Efficacy (Pre)	179	59.82	60.00	8.29	68.75	28.00	81.60
Basic Programming (Post)	179	70.49	68.00	14.65	214.90	32.00	100.00
Self-Efficacy (Post)	179	74.05	74.40	9.62	92.45	44.00	100.00

The minimum basic programming score increased from 12 to 32, and the minimum self-efficacy score increased from 28 to 44. In contrast, the maximum basic programming score increased from 80 to 100, and the maximum self-efficacy score increased from 81.60 to 100.

The improvement pattern in Table 4 suggests that Informatics instruction implemented during the study period was generally associated with increased basic programming skills. Low initial performance is commonly observed among novice learners who still adapting to programming logic, algorithms, and syntax. Such difficulties may be influenced by limited prior experience, challenges in understanding logic and algorithms, the continued use of conventional instructional approaches, and limited learning resources that support gradual practice (Cheah, 2020).

Table 4 also shows an overall increase in self-efficacy. This pattern is important because programming learning often requires students to persist through trial-and-error, interpret compiler messages, and continue working despite frequent mistakes, especially at the novice level. Higher self-efficacy is closely related to students' persistence in the face of errors, their willingness to try alternative strategies, and their readiness to learn more independently rather than relying solely on teachers or peers. Students' confidence tends to develop when they experience tangible progress and perceive that they can complete increasingly challenging learning tasks. Therefore, the observed increase in self-efficacy suggests that, alongside cognitive improvement, students' confidence to engage with programming tasks may also have strengthened during the instructional period (Tseng et al., 2024).

To provide a more detailed picture by treatment condition, descriptive statistics were also examined for the control and experimental groups. The results by group are presented in Table 5.

Table 5. Descriptive statistics by group

Variabel	Groups	N	Mean	Median	SD	Variance	Min	Max
Basic Programming (Pre)	Control	89	38.16	36.00	17.47	305.11	12.00	80.00
	Experimental	90	39.24	36.00	14.10	198.79	12.00	80.00
Basic Programming (Post)	Control	89	64.45	64.00	13.52	182.71	32.00	96.00
	Experimental	90	76.47	78.00	13.29	176.54	52.00	100.00

Self-Efficacy (Pre)	Control	89	59.09	60.00	8.19	67.145	37.60	76.80
	Experimental	90	60.54	60.40	8.36	70.05	28.00	81.60
Self-Efficacy (Post)	Control	89	69.90	68.80	7.96	63.42	48.00	92.80
	Experimental	90	78.15	78.40	9.37	87.94	44.00	100.00

Table 5 indicates that baseline scores were relatively similar across groups. The mean pre-test score for basic programming was 38.16 in the control group and 39.24 in the experimental group. The mean pre-test self-efficacy score was 59.09 in the control group and 60.54 in the experimental group. Following the instruction, both groups improved; however, the experimental group showed higher post-test means for both variables. The mean post-test basic programming score was 76.47 in the experimental group and 64.45 in the control group. The mean post-test self-efficacy score was 78.15 in the experimental group and 69.90 in the control group.

For basic programming skills, the higher post-test mean in the experimental group suggests that learning with the drill-and-practice educational game may provide additional benefits compared with conventional media such as e-books and presentation materials. In this study, the educational game was designed to support step-by-step practice through structured tasks, to provide immediate feedback on students' responses, and to allow repeated attempts until learners achieved mastery of each topic. This pattern is consistent with the core features of drill-and-practice learning, which emphasizes systematic repetition to strengthen procedural skills and improve accuracy in applying rules or steps (Elfeky & Elbyaly, 2023).

From a game-based learning perspective, higher outcomes in the experimental group may also reflect greater practice intensity and student engagement when activities are delivered as progressively challenging tasks. Prior studies have reported that gamification and educational games can improve programming-related skills by encouraging students to practice more actively and persist in completing tasks (Bozan & Ta'lidere, 2024). However, evidence also indicates that the effectiveness of gamification varies across contexts, depending on the discipline, the design principles employed, the duration of the gameful experience, and the learning environment (Videnovik et al., 2023; Zhan et al., 2022). Therefore, in this study, students' learning was supported not only by the game's interaction but also by teacher monitoring of students' practice progress. Practice and activity records were used to help teachers identify students who experienced difficulties with specific topics and to provide targeted responses such as concept reinforcement, problem-solving guidance, or additional support when needed.

For self-efficacy, the experimental group also showed a higher post-test mean than the control group. Increased self-efficacy may occur when repeated practice, feedback, and opportunities to try again create more frequent mastery experiences. When students perceive clear progress, their confidence in handling programming challenges may increase. This pattern has been reported in studies indicating that gamification and practice-oriented learning experiences can support programming self-efficacy and persistence in learning programming (Chinchua et al., 2022).

Nevertheless, evidence also suggests that increased engagement does not always translate into higher self-efficacy. A study using a specific gamification element, such as badges, reported improved engagement but no significant differences in learning performance or self-efficacy compared with a control condition (Ortiz-Rojas et al., 2025). Such mixed findings indicate that self-efficacy outcomes depend strongly on implementation quality and the alignment between game design and learning objectives (Aljamaan et al., 2025).

Although the descriptive results indicate differences between students who learned with the drill-and-practice educational game and those who learned with conventional media, further analysis is required to determine whether these differences are statistically significant after controlling for students' initial conditions. Therefore, the next section presents inferential analyses to provide a stronger answer to the research question.

Differences in Basic Programming Skills and Self-Efficacy between the Experimental and Control Groups

This section examines whether there were simultaneous differences in basic programming skills and self-efficacy between students who learned using the drill-and-practice educational game and those who learned using conventional media, such as e-books and presentation materials. The analysis employed Multivariate Analysis of Covariance, namely MANCOVA, with two dependent variables, basic programming skills and self-efficacy. Pre-test scores in basic programming skills and self-efficacy were included as covariates to control for students' initial conditions, thereby allowing more meaningful interpretation of post-test comparisons between groups. Before conducting MANCOVA, a series of prerequisite tests was carried out, and the results indicated that the assumptions of multivariate normality, linearity, homogeneity of regression slopes, equality of covariance matrices, and homogeneity of variances were met.

Table 6. Multivariate MANCOVA results

Effect	Pillai's Trace	F	Sig.	Partial Eta Squared	Conclusion
Programming Pre-test	0.601	131.064	p < .001	0.601	Significant
Self-efficacy Pre-test	0.508	89.706	p < .001	0.508	Significant
Group	0.528	97.442	p < .001	0.528	Significant

As shown in Table 6, the programming pre-test had a significant multivariate effect on the combined post-test outcomes, with Pillai's Trace of 0.601, F of 131.064, and $p < .001$. The partial eta squared of 0.601 indicates a very strong effect. The self-efficacy pre-test also had a significant multivariate effect, with Pillai's Trace of 0.508, F of 89.706, and $p < .001$, and a partial eta squared of 0.508 indicating a strong effect. After controlling for both covariates, the group factor remained significant, with Pillai's Trace of 0.528, F of 97.442, and $p < .001$, and a partial eta squared of 0.528, indicating that group differences had a strong effect on the combined outcomes.

The strong effect of the programming pre-test indicates that students' initial programming skills were a significant determinant of their outcomes. Higher initial performance suggests that students entered the learning process with more established conceptual and procedural foundations, enabling them to process new material more efficiently and correct errors more effectively during practice. This pattern is consistent with evidence that prior knowledge in entry-level programming learning contributes to meaningful variation in students' performance, supporting the need to control baseline differences when evaluating treatment effects (Smith, Hao, Jagodzinski, Liu, & Gupta, 2019). In addition, improvements in basic programming concepts and learners' confidence often co-develop during structured programming instruction, suggesting that students who start with stronger foundations may also experience smoother progression through learning activities (Tsai et al., 2023).

The strong effect of the self-efficacy pre-test suggests that students' initial confidence was related to their engagement in the programming learning process, particularly in addressing errors, logical impasses, and the need to try alternative strategies. Students with higher self-efficacy tend to persist longer, explore solutions more confidently, and sustain effort across repeated practice, thereby increasing their likelihood of achieving better learning outcomes. This pattern aligns with prior findings that programming self-efficacy is associated with persistence and learning strategies and plays an important role in sustaining effort during programming learning (Q. Li et al., 2025). Moreover, motivational mechanisms common in gamified learning, such as perceived autonomy, relatedness, and enjoyment, can support sustained effort during practice, which helps explain why initial confidence can matter when students face repeated trial-and-error in programming tasks (Kaya & Ercag, 2023; Li et al., 2024).

The group effect remained significant after controlling for initial programming skills and self-efficacy, indicating that the treatment condition contributed meaningfully to the combined outcomes. This finding supports the interpretation that learning with the drill-and-practice educational game provided added benefits compared with conventional learning using e-books and presentation media. From a cognitive perspective, prior research indicates that games and gamification in programming learning can improve programming-related achievement by encouraging active practice and persistence in completing tasks (Alsuhaymi & Alotaibi, 2023; Alharbi et al., 2024). From an affective perspective, repeated practice accompanied by immediate feedback and stepwise progression can foster mastery experiences that strengthen learners' confidence, consistent with research reporting that gamification-based learning ecosystems can increase programming self-efficacy (Hsiao et al., 2023).

Differences in Basic Programming Skills between the Experimental and Control Groups

This section examines whether there were differences in post-test basic programming skills between students who learned using the drill-and-practice educational game and students who learned using conventional media in the form of e-books and presentation materials after controlling for students' initial conditions. Follow-up univariate analysis was conducted using the Tests of Between-Subjects Effects from the MANCOVA model, with the basic programming pre-test and self-efficacy pre-test included as covariates.

Table 7. Tests of between-subjects effects for basic programming skills

Source	Dependent Variable	F	Sig.	Partial Eta Squared
Programming Pre-test	Basic Programming Post-test	258.124	$p < .001$	0.596
Self-efficacy Pre-test	Basic Programming Post-test	10.315	$p = .002$	0.056
Group	Basic Programming Post-test	115.102	$p < .001$	0.397

As shown in Table 7, the group factor had a statistically significant effect on post-test basic programming skills, with an F of 115.102 and $p < .001$. The partial eta squared of 0.397 indicates that group differences had a strong effect on post-test basic programming skills after controlling for baseline conditions. In addition, the programming pre-test had a significant and very strong effect on post-test scores. In contrast, the self-efficacy pre-test also contributed significantly, but with a smaller effect size. To clarify the direction of the difference, the between-group comparison is presented in Table 8.

The pairwise comparison shows that the experimental group scored higher than the control group on the basic programming post-test, with a mean difference of 10.96 points (p < .001).

Table 8. Pairwise comparisons for basic programming skills

Dependent Variable	Group Comparison	Mean Difference	Std. Error	Sig.
Basic Programming Post-test	Control vs Experimental	-10.96	1.02	p < .001

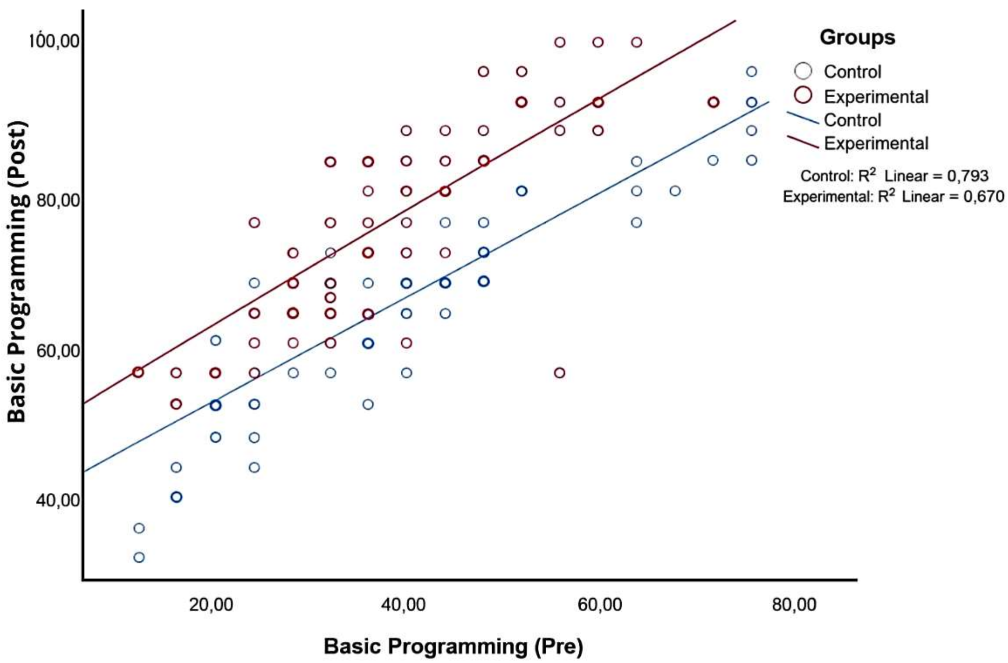

Figure 2. Scatter plot of basic programming pre-test and post-test scores by group

Figure 2 shows a positive relationship between pre-test and post-test scores in both groups, indicating that students with higher baseline scores tended to achieve higher post-test scores. However, within similar ranges of pre-test scores, the experimental group's points tend to appear at higher post-test levels than those of the control group. This pattern supports the inferential findings that learning with the drill-and-practice educational game was associated with higher basic programming outcomes than learning with conventional media.

The higher basic programming outcomes in the experimental group are plausible because the drill-and-practice educational game encouraged more active and repeated practice, allowing students to engage more frequently with basic programming concepts and procedures. In programming instruction, performance gains are often associated with sufficient and consistent practice opportunities, particularly when practice requires students to complete tasks, receive feedback, and correct mistakes across repeated cycles. Such iterative practice can strengthen procedural accuracy and consolidate foundational understanding. Empirical evidence indicates that gamification and educational games in programming education can improve programming achievement by increasing students' engagement and persistence in practice activities (Cuervo-Cely et al., 2022).

From a pedagogical perspective, the effectiveness of structured and repetitive practice environments can be explained by the theory of deliberate practice, which posits that goal-directed, structured engagement in specific tasks, accompanied by informative feedback, leads to gradual refinement of performance and stronger procedural skills (Ericsson, 2008; Ericsson & Harwell, 2019). In the context of basic programming, deliberate practice helps learners strengthen procedural fluency by reinforcing correct command usage and logical sequencing through continuous rehearsal. The game structure in this study provided repeated opportunities to complete tasks and clear success criteria, which likely facilitated this process. The combination of repeated practice, explicit feedback, and structured, progressively challenging learning activities likely supported gradual procedural refinement by directing students' cognitive focus toward accuracy and logical sequencing rather than superficial task completion.

Beyond increasing practice intensity, educational games may also support understanding through more contextual and challenging activities. Stepwise challenges can provide opportunities for learners to apply concepts across a wider range of problem types, helping them learn not only rules and definitions but also when and how to apply procedures. This gradual increase in task difficulty is consistent with cognitive theories of skill acquisition, which propose that learning is strengthened when learners progress from simple to more complex tasks while maintaining manageable cognitive demands and opportunities for consolidation (Krell et al., 2022; Ouwehand et al., 2025; Sweller, 2011). Consistent with this view, prior research reports that educational games can help students grasp complex programming concepts through interactive and meaningful learning experiences (Villareale, States, & Villareale, 2020). Findings from game-based assessment studies also suggest that practice embedded in game environments can reinforce technical competence by encouraging learners to rehearse skills repeatedly toward clear goals (Gomez et al., 2023).

Nevertheless, these findings should be interpreted in light of the measurement approach. Basic programming skills were assessed using a multiple-choice test, which primarily reflects introductory conceptual and procedural understanding rather than students' ability to write, run, and debug executable programs in authentic performance tasks. Therefore, conclusions about practical programming competence should be made cautiously, and future research could strengthen the evidence by incorporating performance-based coding and debugging tasks.

Differences in Self-Efficacy between the Experimental and Control Groups

This section examines whether there were differences in post-test self-efficacy between students who learned using the drill-and-practice educational game and students who learned using conventional media in the form of e-books and presentation materials after controlling for students' initial conditions. Follow-up univariate analysis was conducted using the Tests of Between-Subjects Effects from the MANCOVA model, with the basic programming pre-test and self-efficacy pre-test included as covariates.

Table 9. Tests of between-subjects effects for self-efficacy

Source	Dependent Variable	F	Sig.	Partial Eta Squared
Programming Pre-test	Self-Efficacy Post-test	7.989	p = .005	0.044
Self-efficacy Pre-test	Self-Efficacy Post-test	172.487	p < .001	0.496
Group	Self-Efficacy Post-test	86.697	p < .001	0.331

As shown in Table 9, the group factor had a statistically significant effect on post-test self-efficacy, with F of 86.697 and p < .001. The partial eta squared of 0.331 indicates that group differences had a strong effect on self-efficacy after controlling for baseline conditions. In addition, the pre-test self-efficacy had a substantial effect on post-test self-efficacy. In contrast, the programming pre-test also contributed significantly but with a smaller effect size. To clarify the direction of the difference, the between-group comparison is presented in Table.

The pairwise comparison shows that the experimental group scored higher than the control group on the post-test self-efficacy, with a mean difference of 8.84 points (p < .001).

Table 10. Pairwise comparisons for self-efficacy

Dependent Variable	Group Comparison	Mean Difference	Std. Error	Sig.
Self-Efficacy Post-test	Control vs Experimental	-8.84	0.95	p < .001

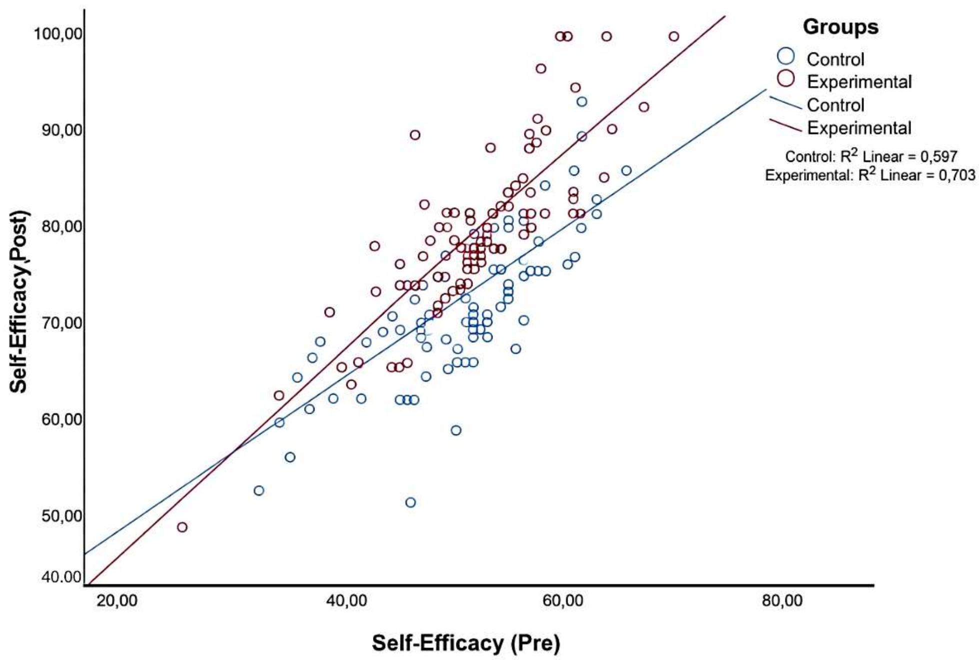

Figure 3. Scatter plot of self-efficacy pre-test and post-test scores by group

Figure 3 shows a positive relationship between pre-test and post-test self-efficacy in both groups, indicating that students with higher baseline confidence tended to report higher post-test confidence. However, within similar ranges of pre-test self-efficacy, the experimental group's points tend to appear at higher post-test levels than those of the control group. This pattern supports the inferential results, which show that learning with the drill-and-practice educational game was associated with greater gains in self-efficacy than learning with conventional media.

Table 11. Summary of game log on level completion and attempts

Level	Completion Rate	Mean	SD
Level 1	100.00 %	1.90	0.82
Level 2	100.00 %	2.56	1.19
Level 3	96.67 %	3.55	1.10
Level 4	92.22 %	3.57	1.21

Table 11 presents behavioral data from the game logs across four levels of the educational game. Completion rates remained high across all levels, with only a modest decline at higher levels, indicating that most students were able to progress despite increasing task demands. At the same time, the mean number of attempts rose with each level, suggesting that students engaged in repeated practice before achieving successful completion. Taken together, these patterns indicate sustained effort and persistence, as students continued to work through challenges rather than disengaging when tasks became more difficult.

The higher self-efficacy outcomes in the experimental group are plausible because the drill-and-practice educational game provided repeated opportunities to practice, receive feedback, and experience incremental success. Such mastery-oriented experiences can strengthen students' confidence in handling programming tasks, especially when early learning often involves errors and trial-and-error. This pattern is reflected in the present study, where the game logs indicate that students consistently completed the levels but required more attempts on later levels, suggesting persistence and repeated practice as task demands increased. Studies on gamified and game-based programming learning have reported improvements in programming self-efficacy when learners engage in structured challenges, receive feedback, and perceive progress toward goals, which supports the interpretation of the group difference observed in this study (Chinchua et al., 2022).

In addition, the significant contribution of the self-efficacy pre-test indicates that students' initial confidence shaped how they benefited from learning activities, which is consistent with research showing that self-efficacy is closely related to persistence, willingness to try alternative strategies, and sustained effort when learning programming (Schunk & DiBenedetto, 2021; Abdunabi et al., 2024). From this perspective, the educational game may have amplified positive learning experiences by making practice more engaging and by providing clearer indicators of progress, thereby supporting students' sense of mastery during learning (Chen & Tu, 2021). Nevertheless, self-efficacy in this study was measured using a self-report questionnaire, so the results reflect students' perceived confidence rather than directly observed behavior. To address this limitation, the study triangulated the questionnaire findings with behavioral evidence from the game logs. Future research could extend this approach by incorporating additional fine-grained indicators, such as time-on-task and help-seeking frequency, and by examining how these behaviors relate to changes in self-efficacy over time (Masiello et al., 2024).

■ CONCLUSION

Research on the effects of drill-and-practice educational games on basic programming skills and self-efficacy indicates that their implementation is effective in improving these outcomes among students in private high schools in Pontianak. The results of univariate tests and MANCOVA showed a significant difference between the experimental and control groups, with students who received the treatment achieving higher scores. This improvement occurred because drill-and-practice educational games allow students to gradually practice skills, receive direct feedback, correct errors independently, and learn in an interactive, contextual environment. Gamification elements such as challenges, rewards, and goal achievement also increase students' intrinsic motivation and self-confidence. The implications of this study indicate that drill-and-practice educational games can serve as a learning medium for programming materials, particularly algorithms, at the high school level. They may be applied to other materials that require gradual practice and concept reinforcement. Limitations of the study include the use of multiple-choice skills tests without performance assessments and self-efficacy measures that remain based on self-report. Further research is recommended to develop more diverse skill measurement tools, including direct practice tests, and to expand the range of materials and educational levels to improve representativeness.

ACKNOWLEDGEMENT

The authors would like to thank the Ministry of Higher Education, Science, and Technology of the Republic of Indonesia for providing financial support through a research grant for this study.

DECLARATION OF GENERATIVE AI USAGE IN THE WRITING PROCESS

During the drafting of this manuscript, the author(s) utilized ChatGPT and Quillbot for the purpose of refining sentence structure and translating text. Following the use of this tool, the authors reviewed and revised the content as necessary and accept full responsibility for the final content of the article.

REFERENCES

Abdunabi, R., Hbaci, I., & Nyambe, T. (2024). Examining factors predicting programming self-efficacy for computer information systems students. *Information Systems Education Journal*, 22(5), 46–58. https://doi.org/10.62273/kdpz6290

Alharbi, A., Nurfianti, A., Mullen, R. F., McClure, J. D., & Miller, W. H. (2024). The effectiveness of simulation-based learning (SBL) on students' knowledge and skills in nursing programs: a systematic review. *BMC Medical Education*, 24(1). https://doi.org/10.1186/s12909-024-06080-z

Aljamaan, A., Alsaber, A., Emmanuel, C., & Alkandari, A. (2025). Leveling up learning effectiveness in STEM education through gamification: an empirical study on behavioral intention and digital literacy among undergraduate students in Kuwait. *Frontiers in Education*, 10, 1586466. https://doi.org/10.3389/feduc.2025.1586466

Alsuhaymi, D. S., & Alotaibi, O. M. (2023). Gamification's efficacy in enhancing students' HTML programming skills and academic achievement motivation. *Journal of Education and E-Learning Research*, 10(3), 397–407. https://doi.org/10.20448/jeelr.v10i3.4738

Ateº, C., Kaymaz, Ö., Kale, H. E., & Tekindal, M. A. (2019). Comparison of test statistics of nonnormal and unbalanced samples for multivariate analysis of variance in terms of type-i error rates. *Computational and Mathematical Methods in Medicine*, 15(1), 2173638. https://doi.org/10.1155/2019/2173638

Badan Pusat Statistik. (2024). *Statistik telekomunikasi indonesia 2023 (telecommunication statistics in indonesia 2023)*. In Direktorat Statistik Keuangan Teknologi Informasi dan Pariwisata (Ed.), Badan Pusat Statistik (2024th ed., Vol. 12). Badan Pusat Statistik. https://www.bps.go.id/id/publication/2024/08/30/f4b846f397ea452bdc2178b3/statistik-telekomunikasi-indonesia-2023.html

Bozan, I., & Ta'lidere, E. (2024). The effect of digital game design-supported coding education on gifted. *International Journal of Contemporary Educational Research*, 11(1), 20–28. https://doi.org/10.52380/ijcer.2024.11.1.531

Capili, B., & Anastasi, J. K. (2024). An introduction to types of quasi-experimental designs. *AJN The American Journal of Nursing*, 124(11), 50–52. https://doi.org/10.1097/01.NAJ.0001081740.74815.20

Cheah, C. S. (2020). Factors contributing to the difficulties in teaching and learning of computer programming: A literature review. *Contemporary Educational Technology*, 12(2), 1–14. https://doi.org/10.30935/cedtech/8247

Chen, C. C., & Tu, H. Y. (2021). The effect of digital game-based learning on learning motivation and performance under social cognitive theory and entrepreneurial thinking. *Frontiers in Psychology*, 12. https://doi.org/10.3389/fpsyg.2021.750711

Chen, L., Ramli, K., Hastiadi, F. F., & Suryanegara, M. (2023). *Accelerating digital transformation in indonesia/ : technology, market, and policy*. Economic Research Institute for ASEAN and East Asia (ERIA). https://www.eria.org/uploads/Accelerating-Digital-Transformation-Indonesia-rev3.pdf

Cheng, Y.-P., Lai, C.-F., Chen, Y.-T., Wang, W.-S., Huang, Y.-M., & Wu, T.-T. (2023). Enhancing student's computational thinking skills with student-generated questions strategy in a game-based learning platform. *Computers & Education*, 200, 104794. https://doi.org/10.1016/j.compedu.2023.104794

Chinchua, S., Kantathanawat, T., & Tuntiwongwanich, S. (2022). Increasing programming self-efficacy (pse) through a problem-based gamification digital learning ecosystem (DLE) model. *Journal of Higher Education Theory and Practice*, 22(9), 131–143. https://doi.org/10.33423/jhetp.v22i9.5370

Córdova-Esparza, D. M., Romero-González, J. A., Córdova-Esparza, K. E., Terven, J., & López-Martínez, R. E. (2024). Active learning strategies in computer science education: a systematic review. *Multimodal Technologies and Interaction*, 8(6). https://doi.org/10.3390/mti8060050

Cuervo-Cely, K. D., Restrepo-Calle, F., & Ramírez-Echeverry, J. J. (2022). Effect of gamification on the motivation of computer programming students. *Journal of Information Technology Education: Research*, 21, 1–16. https://doi.org/10.28945/4917

Ehlinger, E. G., & Stephany, F. (2024). Skills or degree? the rise of skill-based hiring for ai and green jobs. *Technological Forecasting and Social Change*, 214, 124042. https://doi.org/10.1016/j.techfore.2025.124042

Elfeky, A., & Elbyaly, M. (2023). Managing drill and practice programs with a motivational design and their effects on improving students' attitudes toward information and communication technology courses. *European Chemical Bulletin*, 12(6), 6567–6574. https://doi.org/10.31838/ecb/2023.12.si6.579

Ericsson, K. A. (2008). Deliberate practice and acquisition of expert. *Academic Emergency Medicine/ : Official Journal of the Society for Academic Emergency Medicine*, *15*(11), 988–994. https://doi.org/10.1111/j.1553-2712.2008.00227.x

Ericsson, K. A., & Harwell, K. W. (2019). Deliberate practice and proposed limits on the effects of practice on the acquisition of expert performance/ : why the original definition matters and recommendations for future. *Frontiers in Psychology*, *10*, 2396. https://doi.org/10.3389/fpsyg.2019.02396

Eteng, I., Akpotuzor, S., Akinola, S. O., & Agbonlahor, I. (2022). A review on effective approach to teaching computer programming to undergraduates in developing countries. *Scientific African*, 16, e01240. https://doi.org/10.1016/j.sciaf.2022.e01240

Gayatri, G., Jaya, I. G. N. M., & Rumata, V. M. (2023). The indonesian digital workforce gaps in 2021–2025. *Sustainability (Switzerland)*, 15(1). https://doi.org/10.3390/su15010754

Giannakoulas, A., & Xinogalos, S. (2024). Studying the effects of educational games on cultivating computational thinking skills to primary school students: a systematic literature review. *Journal of Computers in Education*, 11(4), 1283–1325. https://doi.org/10.1007/s40692-023-00300-z

Gómez, M. J., Ruipérez-Valiente, J. A., & Clemente, F. J. G. C. (2023). A systematic literature review of game-based assessment studies: trends and challenges. *IEEE Transactions on Learning Technologies*, 16(4), 500–515. https://doi.org/10.1109/TLT.2022.3226661

Grey, S., & Gordon, N. A. (2023). Education sciences: motivating students to learn how to write code using a gamified programming tutor. *Education Sciences*, *13*(3), 230. https://doi.org/10.3390/educsci13030230

Gundersen, S. W., & Lampropoulos, G. (2025). Using serious games and digital games to improve students ' computational thinking and programming skills in k-12 education/ : a systematic literature review. *Technologies*, *13*(3), 113. https://doi.org/10.3390/technologies13030113

Hainey, T., & Baxter, G. (2024). A Serious game for programming in higher education. *Computers & Education: X Reality*, 4, 100061. https://doi.org/10.1016/j.cexr.2024.100061

Hsiao, H.-S., Chen, J.-C., Chen, J.-H., Chien, Y.-H., Chang, C.-P., & Chung, G.-H. (2023). A study on the effects of using gamification with the 6E model on high school students' computer programming self-efficacy, IoT knowledge, hands-on skills, and behavioral patterns. *Educational Technology Research and Development*, *71*(4), 1821–1849. https://doi.org/10.1007/s11423-023-10216-1

Karalar, H. (2023). Adaptation of computer programming self-efficacy scale for computer literacy education into Turkish for middle scho ol students. *International Technology and Education Journal*, 7(2), 51–59. Retrieved from https://files.eric.ed.gov/fulltext/EJ1413438.pdf

Kaya, O. S., & Ercag, E. (2023). The impact of applying challenge-based gamification program on students' learning outcomes: Academic achievement, motivation and flow. *Education and Information Technologies*, 28(8), 10053–10078. https://doi.org/10.1007/s10639-023-11585-z

Kovari, A., & Katona, J. (2023). Effect of software development course on programming self-efficacy. *Education and Information Technologies*, 28(9), 10937–10963. https://doi.org/10.1007/s10639-023-11617-8

Krell, M., Xu, K. M., Rey, G. D., & Paas, F. (2022). Editorial/ : Recent approaches for assessing cognitive load from a validity perspective. *Frontiers in Education*, *6*, 838422. https://doi.org/10.3389/feduc.2021.838422

Landler, L., Ruxton, G. D., & Malkemper, E. P. (2022). The multivariate analysis of variance as a powerful approach for circular data. *Movement Ecology*, 10(1), 1–10. https://doi.org/10.1186/s40462-022-00323-8

Li, L., Hew, K. F., & Du, J. (2024). Gamification enhances student intrinsic motivation, perceptions of autonomy and relatedness, but minimal impact on competency: a meta-analysis and systematic review. *Educational Technology Research and Development*, 72(2), 765–796. https://doi.org/10.1007/s11423-023-10337-7

Li, Q., Jiang, Q., Liang, J. C., Xiong, W., & Zhao, W. (2025). Roles of programming self-efficacy, cognitive styles, and self-regulated learning strategies on computational thinking in computer programming. *Humanities and Social Sciences Communications*, 12(1), 1–12. https://doi.org/10.1057/s41599-025-05686-y

Li, Z. (2023). Research on the application of programming in high school teaching. *Lecture Notes in Education Psychology and Public Media*, 19(1), 186–189. https://doi.org/10.54254/2753-7048/19/20231428

Lie, H., Studer, K., Zhao, Z., Thomson, B., Turakhia, D. G., & Liu, J. (2023). Training for open-ended drilling through a virtual reality simulation. *Proceedings - 2023 IEEE International Symposium on Mixed and Augmented Reality*, ISMAR 2023, 366–375. https://doi.org/10.1109/ISMAR59233.2023.00051

Liu, J., Li, Q., Sun, X., Zhu, Z., & Xu, Y. (2023). Factors influencing programming self-efficacy: an empirical study in the context of Mainland China. *Asia Pacific Journal of Education*, 43(3), 835–849. https://doi.org/10.1080/02188791.2021.1985430

Maesschalck, S. (2024). Critical thinking: the code to crack computer science education. *Journal of Information Technology Education: Innovations in Practice*, 23. https://doi.org/10.28945/5387

Maryono, D., Budiyono, Sajidan, & Akhyar, M. (2022). Implementation of gamification in programming learning: literature review. *International Journal of Information and Education Technology*, 12(12), 1448–1457. https://doi.org/10.18178/ijiet.2022.12.12.1771

Masiello, I., Mohseni, Z. A., Palma, F., Nordmark, S., Augustsson, H., & Rundquist, R. (2024). A current overview of the use of learning analytics dashboards. *Education Sciences*, *14*(1), 82. https://doi.org/10.3390/educsci14010082

MIHCI Türker, P., & Pala, F. K. (2020). The effect of algorithm education on students' computer programming self-efficacy perceptions and computational thinking skills. *International Journal of Computer Science Education in Schools*, 3(3), 19–32. https://doi.org/10.21585/ijcses.v3i3.69

Miller, C. J., Smith, S. N., & Pugatch, M. (2020). Experimental and quasi-experimental designs in implementation research. *Psychiatry Research*, 283(March), 112452. https://doi.org/10.1016/j.psychres.2019.06.027

Montandon, J. E., Politowski, C., Silva, L. L., Valente, M. T., Petrillo, F., & Guéhéneuc, Y. G. (2021). What skills do IT companies look for in new developers? A study with Stack Overflow jobs. *Information and Software Technology*, 129, 1–6. https://doi.org/10.1016/j.infsof.2020.106429

Napalit, F., Tanyag, B., So, C. L., Sy, C., & San Pedro, J. R. (2023). Examining student experiences: Challenges and perception in computer programming. *International Journal of Research Studies in Education*, 12(8), 101–112. https://doi.org/10.5861/ijrse.2023.71

Ortiz-Rojas, M., Chiluiza, K., Valcke, M., & Bolanos-Mendoza, C. (2025). How gamification boosts learning in STEM higher education: a mixed methods study. *International Journal of STEM Education*, 12(1), 1–14. https://doi.org/10.1186/s40594-024-00521-3

Ouwehand, K., Lespiau, F., Tricot, A., & Paas, F. (2025). Cognitive load theory/ : emerging trends and innovations. *Education Sciences*, *15*(4), 458. https://doi.org/10.3390/educsci15040458

Pan, Y., Adams, E. L., Ketterlin-Geller, L. R., Larson, E. C., & Clark, C. (2024). Enhancing middle school students' computational thinking competency through game-based learning. *Educational Technology Research and Development*, 72(6), 3391–3419. https://doi.org/10.1007/s11423-024-10400-x

Rahimi, S., & Shute, V. (2021). Learning analytics dashboards in educational games. In M. Sahin & D. Ifenthaler (Eds.), *Visualizations and dashboards for learning analytics* (pp. 527–546). Springer International Publishing. https://doi.org/10.1007/978-3-030-81222-5_24

Schunk, D. H., & DiBenedetto, M. K. (2021). Chapter four - self-efficacy and human motivation. In A. J. B. T.-A. in M. S. Elliot (Ed.), *Advances in Motivation Science* (Vol. 8, pp. 153–179). Elsevier. https://doi.org/10.1016/bs.adms.2020.10.001

Smit, R., Schmid, R., & Robin, N. (2025). Experiencing enjoyment in visual programming tasks promotes self-efficacy and reduces the gender gap. *British Journal of Educational Technology*, 56(3), 1231–1247. https://doi.org/10.1111/bjet.13523

Smith, D. H., Hao, Q., Jagodzinski, F., Liu, Y., & Gupta, V. (2019). Quantifying the effects of prior knowledge in entry-level programming courses. *Proceedings of the ACM Conference on Global Computing Education*, 30–36. New York, NY, USA: Association for Computing Machinery. https://doi.org/10.1145/3300115.3309503

Sudirman, S., & Zain, Moh. I. (2023). Application of the drill method to improving science learning outcomes. *Jurnal Penelitian Pendidikan IPA*, 9(4), 1886–1891. https://doi.org/10.29303/jppipa.v9i4.3649

Sun, L., Guo, Z., & Zhou, D. (2022). Developing K-12 students' programming ability: A systematic literature review. *Education and Information Technologies*, *27*(5), 7059–7097. https://doi.org/10.1007/s10639-022-10891-2

Sweller, J. (2011). Cognitive Load Theory. In J. P. Mestre & B. Ross (Eds.), *Psychology of Learning and Motivation* (Vol. 55, pp. 37–76). Academic Press. https://doi.org/10.1016/B978-0-12-387691-1.00002-8

The SMERU Research Institute. (2022). *Diagnostic report - digital skills landscape in indonesia*. https://pathwayscommission.bsg.ox.ac.uk/sites/default/files/2022-03/FINAL_Diagnostic Report_Accessible.pdf

Tsai, C.-Y., Chen, Y.-A., Hsieh, F.-P., Chuang, M.-H., & Lin, C.-L. (2023). Effects of a programming course using the game model on undergraduates' self-efficacy and basic programming concepts. *Journal of Educational Computing Research*, 62(3), 482–504. https://doi.org/10.1177/07356331231206071

Tsai, C.-Y. (2019). Improving students' understanding of basic programming concepts through visual programming language: The role of self-efficacy. *Computers in Human Behavior*, 95, 224–232. https://doi.org/10.1016/j.chb.2018.11.038

Tsai, M.-J., Wang, C.-Y., & Hsu, P.-F. (2018). Developing the computer programming self-efficacy scale for computer literacy education. *Journal of Educational Computing Research*, 56(8), 1345–1360. https://doi.org/10.1177/0735633117746747

Tseng, C. Y., Cheng, T. H., & Chang, C. H. (2024). A novel approach to boosting programming self-efficacy: issue-based teaching for non-cs undergraduates in interdisciplinary education. *Information (Switzerland)*, 15(12), 1–17. https://doi.org/10.3390/info15120820

Videnovik, M., Vold, T., Kiønig, L., Bogdanova, A. M., & Trajkovik, V. (2023). Game based learning in computer science education/ : a scoping literature review. *International Journal of STEM Education*. https://doi.org/10.1186/s40594-023-00447-2

Villareale, J., Biemer, C. F., El-Nasr, M. S., & Zhu, J. (2020). Reflection in game-based learning/ : a survey of programming games reflection in game-based learning/ : a survey of programming games. *FDG '20: Proceedings of the 15th International Conference on the Foundations of Digital Games*, *September*, 1–9. https://doi.org/10.1145/3402942.3403011

World Economic Forum. (2025). *Future of Jobs Report 2025: Insight Report*. Geneva: World Economic Forum. Retrieved from https://www.weforum.org/reports/the-future-of- jobs-report-2025/

Zhan, Z., He, L., Tong, Y., Liang, X., Guo, S., & Lan, X. (2022). Computers and Education/ : Artificial intelligence the effectiveness of gamification in programming education/ : Evidence from a meta-analysis. *Computers and Education: Artificial Intelligence*, 3, 100096. https://doi.org/10.1016/j.caeai.2022.100096