Jurnal Review Pendidikan Dasar:
Jurnal Kajian Pendidikan dan Hasil Penelitian
Vol 12(1), Februari 2026

e-ISSN: 2460-8475
http://journal.unesa.ac.id/index.php/PD
https://doi.org/10.26740/jrpd.v12n1.p18-31

The Effectiveness of Pair Programming in Enhancing Elementary School Students' Creative Thinking Skills in Coding Lessons

Febriarsita Eka Sasmita ✉, Madrasah Ibtidaiyah Teacher Education Universitas Sunan Giri Surabaya
Ummu Khairiyah, Madrasah Ibtidaiyah Teacher Education Universitas Islam Lamongan
Pance Mariati, Primary School Teacher Education Universitas Nahdlatul Ulama Surabaya

✉ *febriarsita@unsuri.ac.id*

Abstract: This study aims to analyze the effectiveness of pair programming in coding lessons on enhancing elementary school students' creative thinking skills. The research employed a quasi-experimental method with a nonequivalent control group design involving 44 elementary school students participating in a coding learning program. The subjects were divided into two groups: an experimental group learning through pair programming and a control group learning using the conventional direct instruction method. The research instrument used the Torrance Tests of Creative Thinking (TTCT) Figural A to measure three indicators of creative thinking: fluency, originality, and elaboration. Data analysis was conducted using paired sample t-tests and the Wilcoxon signed-rank test. The results showed a significant improvement in creative thinking skills in the experimental group compared to the control group, particularly in the fluency and originality indicators. However, the improvement in the elaboration indicator was not significant. These findings indicate that pair programming is an effective learning strategy to stimulate creative thinking in coding lessons at the elementary school level, especially in terms of idea fluency and originality.

Keywords: Pair Programming, Coding Learning, Creative Thinking, Collaborative, Elementary School.

How to Cite: Sasmita, F.E., Khairiyah, U., & Mariati, P (2026). The Effectiveness of Pair Programming in Enhancing Elementary School Students' Creative Thinking Skills in Coding Lessons . *Jurnal Review Pendidikan Dasar: Jurnal Kajian Pendidikan dan Hasil Penelitian, 12*(1), 18-31. https://doi.org/10.26740/jrpd.v12n1.p18-31

Article Info:
Received: 29 December 2025; Accepted: 18 February 2026; Published Online: 28 February 2026

Published by:
Pendidikan Dasar Universitas Negeri Surabaya

INTRODUCTION

The development of digital technology in the 21st century demands a transformation in the education system, including the mastery of 21st-century skills such as creative thinking, collaboration, critical thinking, problem solving, and technological literacy (Hilton & Pellegrino, 2012; Trilling & Fadel, 2009). One of the skills that has received increasing attention is *computational literacy*, which includes programming (coding) as a means of fostering logical, systematic, and creative thinking in solving problems (Wing, 2011). In line with this global trend, various countries such as the United States, the United Kingdom, Australia, and South Korea have integrated coding instruction into their primary education curricula as part of their readiness for the digital era (Wu et al., 2020). This demonstrates that coding is no longer a technical skill limited to future programmers but has become a fundamental competency essential from the primary school level.

In Indonesia, the urgency of digital literacy and technological competence has been addressed through the implementation of the *Merdeka Curriculum*, which explicitly includes Informatics as a subject starting from the elementary level (Awaluddin & Hadi, 2025; Mardiana & Emmiyati, 2024). This marks a shift in the educational paradigm, which previously focused on basic literacy, reading, writing, and arithmetic and has now expanded to encompass technological and information literacy (Aprilia, 2025; Nuraini et al., 2022). However, several studies show that the implementation of informatics learning in primary schools is still not optimal due to limited infrastructure, insufficient teacher preparedness, and students' low experience in programming (Naufal, 2021). In addition, the 2022 PISA results also indicate that Indonesia continues to face challenges in science, mathematics, and technology literacy, which form the foundation for learning coding (Herman et al., 2024). These conditions highlight the need for pedagogical approaches that not only introduce coding as a technical skill but also cultivate creative and collaborative ways of thinking from the primary education level.

Although coding has been introduced in primary schools, the approaches used in classrooms generally remain technical, individual, and product-oriented. Teachers often provide code examples for students to imitate through structured exercises (drill-based learning), resulting in students merely following procedural steps without understanding the creative thinking processes behind programming (Lye & Koh, 2014, 2018). Such an approach leads to the perception of coding as a linear problem-solving activity rather than a medium for exploring ideas. Consequently, the potential of coding to foster students' creative thinking skills is underutilized, even though creative thinking is one of the key elements of the Profil Pelajar Pancasila, particularly within the dimensions of critical and creative reasoning (Suhendar et al., 2021). Therefore, a learning approach to coding is needed, one that not only focuses on technical skills but also facilitates students in generating ideas, exploring strategies, and collaborating in problem-solving.

Creative thinking is a higher-order cognitive ability that enables individuals to generate new, flexible, and meaningful ideas in solving problems (Guilford, 1967; Runco & Jaeger, 2012). Torrance (1977) states that creative thinking is not merely the ability to find a single correct answer but involves divergent thinking, which is reflected in four key indicators: fluency (the ability to generate many ideas), originality (the uniqueness of ideas), elaboration (the development of detailed ideas), and flexibility (the ability to think from multiple perspectives). In the context of primary education, creative thinking is closely related to the development of problem-solving skills and the formation of learner autonomy (Cropley, 2006; Jeffrey & Craft, 2004). Therefore, creative thinking is an essential competency that must be developed through learning experiences that provide space for idea exploration, open discussion, and engagement in challenging, open-ended tasks from the elementary level (Beghetto & Kaufman, 2010; Sawyer & Henriksen, 2024). One relevant domain for fostering creative thinking is coding education, as programming activities require students to design algorithms, modify solutions, and devise new strategies for addressing digital problems (Resnick, 2017).

To optimize coding instruction so that it focuses not only on technical skills but also on the development of creative thinking, a collaborative learning approach that allows space for idea exchange is needed. One relevant approach is *pair programming*, a collaborative programming strategy in which two students work together at one computer, assuming the roles of driver and navigator (Williams & Kessler, 2000).In this model, the driver is responsible for writing the program code, while the navigator provides guidance, strategies, and alternative solutions throughout the programming process. Previous studies have shown that pair programming can enhance conceptual understanding, programming accuracy, learning motivation, and students' confidence (Ayub et al., 2019; Hanks, 2008; Hannay et al., 2009). Furthermore, the two-way interaction in pair programming encourages students to develop creative thinking skills through discussion, idea sharing, and joint evaluation of solutions (Salleh et al., 2014). Thus, pair programming holds significant potential for improving creative thinking skills in coding instruction at the primary school level.

Although numerous studies have demonstrated the effectiveness of pair programming in improving learning outcomes and programming concept comprehension (Ayub et al., 2019; Hanks, 2008; Hannay et al., 2009), most of these studies were conducted at the secondary or higher education levels, particularly within computer science education. Research specifically examining the impact of pair programming on the creative thinking skills of primary school students remains limited, especially in the context of coding instruction at the elementary level. In addition, most previous studies have focused more on cognitive aspects, such as the improvement of computational thinking skills (Grover & Pea, 2018), while affective aspects and creative thinking as part of 21st-century competencies have not been explored in depth (Angraini et al., 2022). In Indonesia, research on coding instruction for primary school students is still largely centered on the development of learning models and media (Putro & Astuti, 2020), leaving a gap in studies that examine collaborative approaches such as pair programming and their impact on students' creative thinking skills. Therefore, this study holds strong academic and empirical urgency to be developed.

Empirical studies have shown that social interaction in learning plays an important role in the development of higher-order thinking skills, including creative thinking.

Vygotsky & Cole (2018) emphasized that children learn most effectively through social interaction within the *zone of proximal development* (ZPD), where they can develop new ideas through collaboration and dialogue. In the context of coding instruction, working in pairs through pair programming creates a collaborative environment that enables the exchange of ideas, argumentation, and joint reflection (Salleh et al., 2014). This process fosters a learning atmosphere that supports the emergence of original ideas and flexible thinking, two central characteristics of creativity (Runco, 2008). In addition, several empirical studies indicate that pair programming not only enhances cognitive performance but also nurtures affective aspects such as motivation, confidence, and learning engagement (Hanks, 2008; Hannay et al., 2009). These aspects play an essential role in encouraging students to experiment and take intellectual risks, two key characteristics of creative thinking (Cropley, 2006). Therefore, the pair programming approach has strong theoretical and empirical foundations for implementation in coding instruction at the primary school level, particularly for developing creative thinking skills through collaborative and exploratory learning activities.

Based on the discussion above, it can be concluded that there is a need for a coding learning approach that not only emphasizes procedural and technical aspects but also focuses on developing creative thinking skills from the primary school level. Pair programming is considered a relevant instructional strategy because it involves social interaction, collaboration, and the exchange of ideas that can foster the emergence of creative thinking in problem solving. Therefore, this study was conducted to analyze the effectiveness of implementing pair programming in coding instruction on the creative thinking skills of primary school students. This research is expected to provide empirical contributions to the development of creative and collaborative coding learning practices in elementary schools, as well as enrich the scientific literature in the fields of 21st-century education and digital literacy in Indonesia.

METHODS

a. Reseach Design

This study employed a quasi-experimental design with the *Nonequivalent Control Group Design* type, as the sample was not selected through full randomization. However, this design still allows for testing the causal relationship between the treatment variable and the dependent variable by comparing two comparable groups (Creswell, 2013). This design was chosen because it is suitable for educational research involving intact classroom groups and maintains internal validity through the administration of a pretest and posttest for both groups. The experimental design in this study can be represented in Table 1 as follows:

Table 1. Research Experimental Design

Group	Pretest	Treatment	Posttest
Experimental Group	O_1	X (Coding with Pair Programming)	O_2
Control Group	O_3	– (Conventional Learning)	O_4

Description:

O_1 : Pretest of creative thinking skills in the experimental group
O_2 : Posttest of creative thinking skills in the experimental group
O_3 : Pretest of creative thinking skills in the control group
O_4 : Posttest of creative thinking skills in the control group
X : Treatment in the form of coding instruction using pair programming
– : Conventional learning (lecture + individual exercises)

A comparison between groups was used to analyze the effectiveness of the intervention. This design is relevant for examining moderate causal relationships in primary education settings and can minimize threats to validity, such as history and instrumentation effects, through strict procedural controls.

b. Research Subjects

This study involved fifth-grade students at Sekolah Kreatif SD Muhammadiyah 16 Surabaya during the second semester of the 2024/2025 academic year. The total number of research participants was 44 students, divided into two classes: an experimental group and a control group. The subjects were selected using purposive sampling based on several criteria, namely:

1. Students have previously participated in basic coding instruction at school.
2. Students are in the cognitive developmental stage of concrete–formal operations according to Piaget's theory.
3. Students attend learning activities regularly without attendance issues.
4. Students and their parents agree to participate in the study through informed consent.

This school was selected as the research site because it implements a curriculum policy that integrates coding instruction with mathematics through a project-based learning approach. In addition, the coding teacher also serves as the mathematics teacher, resulting in coding materials that are strongly grounded in mathematical concepts such as coordinates, geometry, patterns, and basic algorithms. This condition is highly relevant to the aims of the study, which focus on developing creative thinking skills through coding instruction. The distribution of research subjects is presented in Table 2 below:

Table 2. Distribution of Research Subjects

Group	Number of Students	Learning Intervention
Experimental Group (Class V-B)	22 students	Coding using Pair Programming
Control Group (Class V-A)	22 students	Conventional coding instruction

The study was conducted over seven meetings (one academic calendar month) during the school's regular instructional program. Each meeting lasted 2 × 35 minutes. The learning environment was supported by the school's computer laboratory and instructional tools using visual programming (Scratch).

c. Research Instruments

The primary instrument used in this study was a Creative Thinking Ability Test adapted from the Torrance Tests of Creative Thinking (TTCT) Figural Form A developed by Torrance (1966). This instrument was selected because it effectively measures creative thinking skills through nonverbal visual tasks that align with the characteristics of elementary school students. The TTCT is widely used in studies on creative thinking because it possesses strong construct validity and can be applied across various learning contexts (Lui & Chan, 2006). In this study, the TTCT consisted of three indicators of creative thinking skills assessed using the standard TTCT scoring rubric, namely:

Table 3. Three Indicators of Creative Thinking Ability Based on the TTCT Rubric

Indicator	Definition	Measurement Form
Fluency	Ability to generate many ideas	Scores based on the number of relevant responses
Originality	Ability to generate unique and uncommon ideas	Scores based on the level of originality of the responses
Elaboration	Ability to elaborate and enrich ideas	Scores based on the completeness of idea or drawing details

The instrument underwent a contextual adaptation process to align with the mathematics-based coding learning materials while retaining the original indicators in accordance with TTCT standards. Content validity was established through expert judgment conducted by three lecturers in mathematics education and elementary education. The reliability of the instrument was tested using Cronbach's Alpha through a limited pilot test, resulting in a reliability coefficient of $\alpha = 0.87$, which is categorized as high (Fraenkel & Wallen, 1990). The form of the instrument used in this study is presented in Table 4 below.

Table 4. Types of Activities for Measuring Creative Thinking

Activity	Test Type	Task Description	Example Prompt	Creative Thinking Ability Indicators
Activity 1	*Picture Construction*	Students are given a simple shape (e.g., a small semicircle) and are asked to develop it into a meaningful drawing and provide a title.	"Use this shape to create a meaningful drawing and give it a creative title."	*Originality, Elaboration*
Activity 2	*Picture Completion*	There are 10 incomplete drawings that students must complete creatively.	"Complete the following drawing according to your imagination to create something different."	*Fluency, Originality*
Activity 3	*Parallel Lines*	Students are given 30 pairs of parallel lines and must create various drawings from each pair.	"Develop each pair of lines into as many objects/drawings as possible."	*Fluency*

d. Data Analysis

The research data were analyzed to evaluate the effectiveness of pair programming in enhancing the creative thinking skills of primary school students. The analysis focused on three indicators of creative thinking: fluency, originality, and elaboration. First, the normality of the data was tested using the Shapiro-Wilk test to determine the appropriate statistical tests. Normally distributed data were analyzed using paired-sample t-tests to assess changes in creative thinking skills before and after the intervention within each group (Clark et al., 2008; Cresswell, 2014). Non-normally distributed data were analyzed using the Wilcoxon signed-rank test as a non-parametric alternative.

Subsequently, the improvement in creative thinking skills between groups was compared. For normally distributed data, independent t-tests were used, while for non-normal data, the Mann-Whitney U test was applied. A significance level of $p < 0.05$ was used to determine whether the improvement in the experimental group was greater than that in the control group. Through this procedure, the researchers could comprehensively assess the effect of pair programming on students' creative thinking skills while comparing its effectiveness with conventional methods. The analysis focused not only on total creative thinking scores but also on each individual indicator, fluency, originality, and elaboration, thus providing a deeper insight into the aspects of creative thinking stimulated by this instructional strategy.

RESULTS

This study analyzed the differences in the improvement of creative thinking skills between the experimental group, which used pair programming, and the control group, which employed direct instruction. The data were analyzed through a series of steps, including normality testing, within-group comparisons, and between-group comparisons. This study analyzed the differences in the improvement of creative thinking skills between the experimental group, which used pair programming, and the control group, which employed direct instruction. The data were analyzed through a series of steps, including normality testing, within-group comparisons, and between-group comparisons.

a. Normality Test

Before conducting comparative analyses, the gain scores for each indicator were tested for normality using the Shapiro-Wilk test. The results of the normality test are presented in Table 5.

Table 5. Results of the Normality Test

Group	Indicator	Statistic	p-value	Description
Experimental	Fluency	0.921	0.065	Normal Distribution
	Originality	0.895	0.021*	Not Normal
	Elaboration	0.881	0.012*	Not Normal
Control	Fluency	0.945	0.178	Normal Distribution
	Originality	0.903	0.031*	Not Normal
	Elaboration	0.934	0.121	Normal Distribution

Based on Table 5, the gain scores for the fluency indicator in both groups and the elaboration indicator in the control group were normally distributed ($p > 0.05$). In contrast, the originality indicator in both groups and the elaboration indicator in the experimental group were not normally distributed ($p < 0.05$).

b. Comparison of Pre-test and Post-test Within Groups

The comparison of pre-test and post-test scores was conducted using paired-sample t-tests for normally distributed data and the Wilcoxon signed-rank test for non-normal data. The results are presented in Table 6.

Table 6. Results of the Paired-Sample t-Test

Group	Indicator	Pre-test Mean	Post-test Mean	p-value	Description
Experimental	Fluency	25.36	38.91	0.000*	Significant Improvement
	Originality	18.64	29.55	0.003*	Significant Improvement
	Elaboration	22.09	24.18	0.157	Not Significant
Control	Fluency	26.14	28.23	0.085	Not Significant
	Originality	19.32	21.45	0.061	Not Significant
	Elaboration	21.86	22.95	0.214	Not Significant

Based on Table 6, the experimental group showed significant improvement in the fluency indicator (p = 0.000) and originality indicator (p = 0.003), whereas the elaboration indicator was not significant (p = 0.157). In the control group, none of the three indicators showed significant improvement.

c. Comparison of Improvement Between Groups

To assess differences in improvement between groups, independent-sample t-tests were used for normally distributed data and the Mann-Whitney U test for non-normal data. The analysis results showed that the gain in creative thinking skills in the experimental group was significantly higher than in the control group for the fluency indicator (p = 0.008) and originality indicator (p = 0.011), whereas no significant difference was found for the elaboration indicator (p = 0.325).

Table 7. Results of the Independent-Sample t-Test

Indicator	Experimental Group (Mean Gain)	Control Group (Mean Gain)	p-value	Description
Fluency	13.55	2.09	0.008*	Significant Improvement
Originality	10.91	2.13	0.011*	Significant Improvement
Elaboration	2.09	1.09	0.325	Not Significant

These results confirm that pair programming is effective in enhancing students' creative thinking skills, particularly in terms of fluency and originality, while the elaboration aspect requires additional strategies to achieve significant improvement.

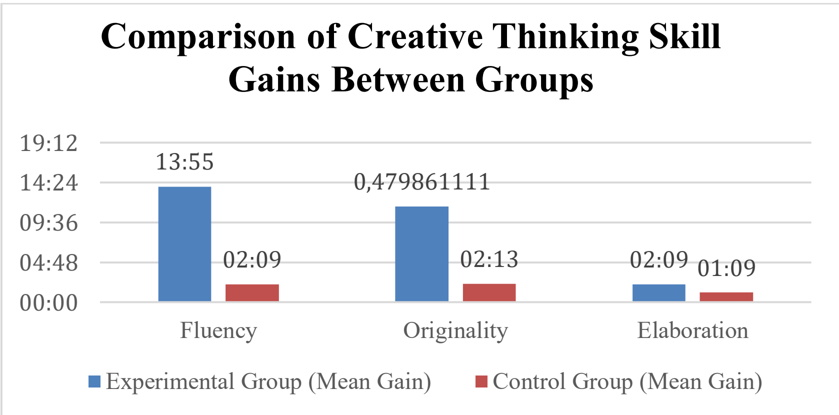

Figure 1. Comparison of Creative Thinking Skill Gains Between Groups

DISCUSSION

The results of the study indicate that the implementation of pair programming is effective in enhancing the creative thinking skills of elementary school students, particularly in terms of fluency (idea generation) and originality (idea uniqueness).

Within-group analysis showed that the experimental group using pair programming experienced significant improvement in these two indicators, whereas the control group using direct instruction demonstrated smaller, non-significant gains. Theoretically, this improvement can be explained through the principles of computational thinking and the learning-by-doing approach, in which students actively solve problems through hands-on practice (Popat & Starkey, 2019). In pair programming, the driver writes the code while the navigator provides feedback, creating a collaborative interaction that fosters real-time idea exchange and problem-solving strategies. Vygotsky and Cole's (1978) theory of the zone of proximal development (ZPD) is also relevant, as social interaction within the pair allows students to reach higher levels of creative thinking than working individually.

Fluency increased in the experimental group because the structure of pair programming encourages students to actively propose multiple alternative solutions when facing coding problems. The role of the driver, who writes the code directly, combined with the role of the navigator, who monitors, provides suggestions, and questions the steps of their partner, creates a rapid feedback cycle that repeatedly stimulates the generation of new ideas. Originality improved because this collaboration forces students to view problems from different perspectives; the navigator may challenge assumptions or offer unexpected approaches, thereby triggering unique and innovative solutions. This process aligns with the theory of divergent thinking (Torrance, 1977), which emphasizes the creation of multiple possible solutions and the exploration of ideas that deviate from habitual thinking. Moreover, the trial-and-error nature of coding strengthens the creative process in practice. Students can immediately implement ideas, evaluate outcomes, correct errors, and iteratively adjust solutions. This activity not only trains analytical thinking skills but also develops cognitive flexibility, a key element of creative thinking. In contrast, the control group, which learned through direct instruction, received less stimulation from social interaction and continuous idea exchange. As a result, the number of alternative solutions generated was lower, and the uniqueness of ideas tended to be limited, as students focused more on following teacher instructions rather than actively exploring creative possibilities.

However, the improvement in the elaboration aspect (detail enrichment) was not significant in either group. This is likely due to the focus of the tasks, which emphasized logical problem-solving and program functionality rather than the development of aesthetics, visual complexity, or detailed refinement of solutions. Elaboration requires more time, persistence, and higher metacognitive abilities, as students must review, expand, and refine initial ideas into richer and more detailed solutions. In the context of pair programming, the collaborative dynamics tend to encourage the generation of new ideas and rapid validation through driver-navigator interactions, causing students to focus more on idea creation than on in-depth idea development. Furthermore, the trial-and-error nature of coding makes students accustomed to finding quickly functional solutions, so the aspect of detailed enrichment receives less priority.

These findings are consistent with (Kim, 2017), who demonstrated that the creative stimulation from collaborative methods is stronger in generating new ideas than in refining existing ones, especially in groups with limited experience in planning and developing complex solutions. In other words, pair programming is more effective in enhancing generative creative ideas (fluency and originality) but requires additional strategies to stimulate elaboration, such as tasks that emphasize detail, aesthetics, or visual complexity.

Pedagogically, the results of this study confirm that pair programming can serve as an effective cooperative learning strategy in elementary school coding education, particularly for stimulating fluency and originality in thinking. Teachers can implement this model by emphasizing role rotation between driver and navigator, facilitating guided discussions, and encouraging joint reflection after coding sessions to enhance understanding and creative thinking skills. To support the elaboration aspect, teachers can design tasks that require detailed development, such as creating complex animations, interactive digital projects, or code-based storytelling, while providing concrete examples and clear assessment criteria so that students understand how to develop ideas more deeply.

Furthermore, integrating pair programming with scaffolding strategies can assist students who have limited experience in creative thinking by providing step-by-step guidance to organize ideas, enrich solutions, and manage coding complexity. Thus, the implementation of pair programming not only enhances short-term creative thinking skills but also fosters creative and collaborative skills that are beneficial for subsequent coding learning.

The limitations of this study, including a relatively small sample and a short intervention duration, indicate that the results cannot be fully generalized. Future research should use larger and more diverse samples, extend the intervention period, and conduct a more in-depth analysis of social interactions, collaborative strategies, and students' cognitive processes during pair programming. Additionally, further studies could explore the integration of the elaboration aspect in task design to investigate how students can be encouraged not only to generate new ideas but also to develop more detailed and complex solutions.

CONCLUSION

This study demonstrates that pair programming is effective in enhancing the creative thinking skills of elementary school students, particularly in terms of fluency (idea generation) and originality (idea uniqueness). Within-group analysis showed that the experimental group using pair programming experienced more significant improvements compared to the control group using direct instruction. This improvement was driven by collaborative interactions between the driver and navigator, which facilitated idea exchange, solution exploration, and trial-and-error processes that stimulated creative thinking. However, the elaboration aspect (detail enrichment) did not show significant improvement, likely because the tasks focused more on logical problem-solving and program functionality rather than on developing detail or visual complexity.

These findings indicate that pair programming is more effective in promoting generative creative ideas but requires additional strategies to stimulate elaboration. Pedagogically, the results confirm that pair programming can be adopted as an effective cooperative learning strategy in elementary school coding education, with an emphasis on role rotation, reflection, and task design that encourages elaboration. For future research, it is recommended to use larger and more diverse samples, extend the intervention duration, and conduct a more in-depth analysis of social interactions and students' cognitive processes to better understand the mechanisms underlying improvements in creative thinking skills.

REFERENCES

Angraini, L. M., Sohibun, S., & Arcat, A. (2022). *Computational Thinking Berbasis Multimedia Interaktif*. Global Research And Consulting Institute (Global-RCI).

Aprilia, E. F. (2025). Implementing Literacy Culture and Digital Literacy in the Early Grades of Primary School. *Jurnal Review Pendidikan Dasar: Jurnal Kajian Pendidikan Dan Hasil Penelitian*, *11*(1), 23–39.

Awaluddin, A., & Hadi, M. S. (2025). Integrasi Pembelajaran Coding Dan Kecerdasan Buatan Di Sekolah Dasar: Tantangan Dan Peluang. *Pendas: Jurnal Ilmiah Pendidikan Dasar*, *10*(01), 1081–1086.

Ayub, M., Karnalim, O., Risal, R., Senjaya, W. F., & Wijanto, M. C. (2019). Utilising Pair Programming to Enhance the Performance of Slow-Paced Students on Introductory Programming. *Journal of Technology and Science Education*, *9*(3), 357–367.

Beghetto, R. A., & Kaufman, J. C. (2010). *Nurturing creativity in the classroom*. Cambridge University Press.

Clark, V. L. P., Creswell, J. W., Green, D. O., & Shope, R. J. (2008). Mixing quantitative and qualitative approaches. *Handbook of Emergent Methods*, *363*, 363–387.

Cresswell, T. (2014). *Place: an introduction*. John Wiley & Sons.

Creswell, J. W. (2013). *Steps in conducting a scholarly mixed methods study*.

Cropley, A. (2006). In praise of convergent thinking. *Creativity Research Journal*, *18*(3), 391–404.

Fraenkel, J. R., & Wallen, N. E. (1990). *How to design and evaluate research in education.* ERIC.

Grover, S., & Pea, R. (2018). Computational thinking: A competency whose time has come. *Computer Science Education: Perspectives on Teaching and Learning in School*, *19*(1), 19–38.

Guilford, J. P. (1967). Creativity: Yesterday, today and tomorrow. *The Journal of Creative Behavior*, *1*(1), 3–14.

Hanks, B. (2008). Empirical evaluation of distributed pair programming. *International Journal of Human-Computer Studies*, *66*(7), 530–544.

Hannay, J. E., Dybå, T., Arisholm, E., & Sjøberg, D. I. K. (2009). The effectiveness of pair programming: A meta-analysis. *Information and Software Technology*, *51*(7), 1110–1122.

Herman, T., Akbar, A., Farokhah, L., Febriandi, R., Zahrah, R. F., Febriani, W. D., Kurino, Y. D., & Abidin, Z. (2024). *Kecakapan abad 21: Literasi matematis, berpikir matematis, dan berpikir komputasi*. Indonesia Emas Group.

Hilton, M. L., & Pellegrino, J. W. (2012). *Education for life and work: Developing transferable knowledge and skills in the 21st century*. National Academies Press.

Jeffrey*, B., & Craft, A. (2004). Teaching creatively and teaching for creativity: Distinctions and relationships. *Educational Studies*, *30*(1), 77–87.

Kim, K. H. (2017). The torrance tests of creative thinking-figural or verbal: Which one should we use? *Creativity. Theories–Research-Applications*, *4*(2), 302–321.

Lui, K. M., & Chan, K. C. C. (2006). Pair programming productivity: Novice–novice vs. expert–expert. *International Journal of Human-Computer Studies*, *64*(9), 915–925.

Lye, S. Y., & Koh, J. H. L. (2014). Review on teaching and learning of computational thinking through programming: What is next for K-12? *Computers in Human Behavior*, *41*, 51–61.

Lye, S. Y., & Koh, J. H. L. (2018). Case studies of elementary children's engagement in computational thinking through scratch programming. In *Computational thinking in the STEM disciplines: Foundations and research highlights* (pp. 227–251). Springer.

Mardiana, M., & Emmiyati, E. (2024). Implementasi kurikulum merdeka dalam pembelajaran: Evaluasi dan pembaruan. *Jurnal Review Pendidikan Dasar: Jurnal Kajian Pendidikan Dan Hasil Penelitian*, *10*(2), 121–127.

Naufal, H. A. (2021). Literasi digital. *Perspektif*, *1*(2), 195–202.

Nuraini, R., Pattiasina, P. J., & Ulfah, A. (2022). Peran literasi teknologi dalam dunia pendidikan. *Al-Madrasah: Jurnal Ilmiah Pendidikan Madrasah Ibtidaiyah*, *6*(3), 659–666.

Popat, S., & Starkey, L. (2019). Learning to code or coding to learn? A systematic review. *Computers & Education*, *128*, 365–376.

Putro, Y. T. M., & Astuti, R. (2020). Penerapan Scratch dalam Pembelajaran Coding Siswa Sekolah Dasar. *Emergent: Journal of Educational Discoveries and Lifelong Learning (EJEDL)*, *1*(4), 21.

Resnick, M. (2017). *Lifelong kindergarten: Cultivating creativity through projects, passion, peers, and play*. MIT press.

Runco, M. A. (2008). Creativity and education. *New Horizons in Education*, *56*(1), n1.

Runco, M. A., & Jaeger, G. J. (2012). The standard definition of creativity. *Creativity Research Journal*, *24*(1), 92–96.

Salleh, N., Mendes, E., & Grundy, J. (2014). Investigating the effects of personality traits on pair programming in a higher education setting through a family of experiments. *Empirical Software Engineering*, *19*(3), 714–752.

Sawyer, R. K., & Henriksen, D. (2024). *Explaining creativity: The science of human innovation*. Oxford university press.

Suhendar, A. M., Ali, S., & Suratman, A. (2021). Membangun Berpikir Kreatif, Sistematis Dan Logis Matematis Melalui Pembelajaran Koding. *Jurnal Perspektif*, *5*(2), 176–190.

Torrance, E. P. (1966). Torrance tests of creative thinking. *Educational and Psychological Measurement*.

Torrance, E. P. (1977). *Creativity in the Classroom; What Research Says to the Teacher.*

Trilling, B., & Fadel, C. (2009). *21st century skills: Learning for life in our times*. John Wiley & Sons.

Vygotsky, L., & Cole, M. (2018). Lev Vygotsky: Learning and social constructivism. *Learning Theories for Early Years Practice. UK: SAGE Publications Inc*, 68–73.

Williams, L. A., & Kessler, R. R. (2000). All I really need to know about pair programming I learned in kindergarten. *Communications of the ACM*, *43*(5), 108–114.

Wing, J. (2011). Research notebook: Computational thinking—What and why. *The Link Magazine*, *6*, 20–23.

Wu, L., Looi, C.-K., Multisilta, J., How, M.-L., Choi, H., Hsu, T.-C., & Tuomi, P. (2020). Teacher's perceptions and readiness to teach coding skills: A comparative study between Finland, Mainland China, Singapore, Taiwan, and South Korea. *The Asia-Pacific Education Researcher*, *29*(1), 21–34.