Research and Development Journal Of Education
Vol. 11, No. 2, Oktober 2025, Pp: 1395 – 1406
DOI: http://dx.doi.org/10.30998/rdje.v11i2.28782
Available online at https://journal.lppmunindra.ac.id/index.php/RDJE
p–ISSN 2406-9744
e–ISSN 2657-1056

PICOLEARN: WEB-BASED LEARNING MEDIA DEVELOPMENT ON PRIMM APPROACH TO ENHANCE COGNITIVE PROCESS IN PROGRAMMING

S. Sarah Nuraisya Gustin[1], Enjun Junaeti[2(*)], Andini Setya Arianti[3]
Universitas Pendidikan Indonesia, Bandung, Indonesia[123]
sarahgustin02@upi.edu[1], enjun@upi.edu[2], andini.sa@upi.edu[3]

Received: 09 April 2025
Revised: 15 Oktober 2025
Accepted: 27 Oktober 2025

Abstract
The Merdeka Curriculum emphasizes the attainment of Higher-Order Cognitive Processes. However, achieving these cognitive goals remains a significant challenge, particularly in programming education. This study focuses on the development and evaluation of PICOLEARN, a web-based learning platform utilizing the PRIMM (Predict, Run, Investigate, Modify, Make) approach to enhance students' cognitive processes in looping structure control programming. The research was conducted in Bandung, Indonesia, involving 35 vocational high school (SMK) students as participants. The object of the study was the learning of looping structures, a fundamental topic in programming. The learning media was developed following the ADDIE model, beginning with a needs analysis and culminating in the design of instructional materials, questions, and learner activities. The media was validated by experts, achieving an 81% feasibility score, classifying it as "very good" for use. The effectiveness of the learning media was assessed using a normalized gain (n-Gain) test, which showed an average increase of 0.40 across all class groups, indicating moderate improvements in students' cognitive abilities. The results highlight the potential of the PRIMM approach to support cognitive development in programming learning. Furthermore, student responses to the media, measured by the the Technology Acceptance Model (TAM), resulted in an average score of 76%, reflecting a positive reception. The findings demonstrate that PRIMM-based learning media is both effective in enhancing cognitive processes and well-received by students. However, further refinements are recommended to maximize its impact on learning outcomes.

Keywords: ADDIE; Cognitive Process; Looping Programming; PRIMM Approach; Web-Based Learning Media

(*) Corresponding Author: Junaeti, enjun@upi.edu

How to Cite: Gustin, S. S. N., Junaeti, E., & Arianti, A. S. (2025). PICOLEARN: WEB-BASED LEARNING MEDIA DEVELOPMENT ON PRIMM APPROACH TO ENHANCE COGNITIVE PROCESS IN PROGRAMMING. *Research and Development Journal of Education*, 11(2), 1395-1406.

INTRODUCTION

Merdeka Curriculum require students to master essential 21st-century skills, such as critical and creative thinking (Kusumasari et al., 2024; Masjudin, 2024; Pulungan et al., 2024), which align with the higher-order cognitive processes outlined by Anderson & Krathwohl (2001). These cognitive processes are structured into three key dimensions: analysing, evaluating, and creating. According to Pacheco & Herrera (2021), the cognitive process encompasses a series of mental activities through which students receive, process, and interpret knowledge. This view is corroborated by Amalia & Yunianta (2019), who describe the cognitive process as a synthesis of newly acquired information with pre-existing knowledge stored in long-term memory. This process reflects students' abilities to engage in complex thinking, logical reasoning, and problem-solving, all of which are crucial for successful lifelong learning (González et al., 2021).

Despite the importance of these cognitive processes, many students struggle with cognitive tasks, particularly in programming (Ramírez et al., 2020; Elçiçek & Karal, 2020; Shah, 2023). Research indicates that students face difficulties in grasping abstract concepts, applying algorithms, and solving programming problems (Cheah, 2020). Survey study to 31 vocational school students in Bandung revealed that 51.6% of tenth-grade students encountered challenges in understanding programming language syntax, 32.3% struggled with programming structures, and 16.1% found it difficult to design algorithms. Teachers identified these difficulties as stemming from a weak foundational understanding and limited practical programming experience. These obstacles also extend to higher-order cognitive processes such as analysis, evaluation, and creation, impeding students' ability to devise logical solutions and complete programming tasks (Mangaroska et al., 2022). Consequently, there is a need for a pedagogical approach that addresses these barriers and enhances students' cognitive abilities in programming.

Several instructional methods, such as Problem-Based Learning (PBL) and Project-Based Learning (PjBL), have been proven effective in improving students' cognitive skills. PBL engages students in analyzing real-world problems and developing solutions through the application of programming code, thereby fostering analytical, problem-solving, and critical thinking skills (Anggraini et al., 2013; Ma'rifah et al., 2019). PjBL, which emphasizes the creation of tangible projects such as applications or websites, helps students gain a deeper understanding of programming concepts. Studies have shown that PjBL enhances students' creativity, technical skills, and collaborative abilities (Putri et al., 2024; Nugraha et al., 2023). However, both approaches have limitations: PBL demands a thorough understanding of the problem at hand, while PjBL requires considerable time for project development. Hence, an alternative approach tailored specifically to introductory programming education is needed.

PRIMM (Predict, Run, Investigate, Modify, Make) approach, developed by Sentance et al. (2019), provides a gradual framework for helping students comprehend programming logic and structure. The approach encourages learners to predict program behavior before running it, closely investigate the code, and modify it to create new solutions (Bradley & Ramezani, 2024; Henley et al., 2021). For instance, the Predict phase engages lower-order cognitive processes, such as recalling and understanding, while later phases like Investigate and Modify require higher-order cognitive processes such as analysis and evaluation. These steps promote logical and critical thinking, while also establishing a solid foundation in programming education (Sentance et al., 2019). PRIMM is well-suited to addressing the complex challenges of programming education, particularly when it comes to abstract concepts like variables, control structures, and algorithms. Additionally, its emphasis on interactive exploration enhances student motivation and engagement. By making programming instruction more structured and aligned with students' cognitive process needs, PRIMM provides a valuable pedagogical tool.

PRIMM-based learning necessitates specially designed media to effectively support its various stages. However, the current media used in programming instruction—such as PowerPoint presentations, YouTube videos, and interactive platforms like Quizizz—do not fully support the interactivity required for in-depth exploration of each PRIMM phase. Therefore, more integrated and interactive learning media is essential. Web-based media offers a promising solution to support programming instruction (Mustaqim & Mahamad, 2023). According to Dio & Ekohariadi (2021), web-based media can enhance the effectiveness of programming education by offering flexibility in terms of time and location, allowing students to learn at their own pace and style. In programming education, such media provide interactive features like code simulations, hands-on exercises, and discussion forums, making it easier for students to grasp abstract programming concepts. The benefits of web-based media are reflected in increased student engagement, easier access to materials, and the promotion of independent, self-directed learning.

Web-based learning media is particularly well-suited to facilitating the implementation of the PRIMM approach, as it integrates each phase into a single platform. During the Predict and Run stages, the media can offer interactive simulations that allow students to forecast the program output before running it. In the Investigate stage, students can delve into the code structure through an intuitive interface. The Modify and Make stages are supported by interactive code-editing and debugging features, enabling students to alter the code and develop new solutions. This approach emphasizes exploration-based learning and collaboration, making students more active participants in understanding and applying programming concepts (Sentance et al., 2019). Web-based learning media effectively integrates all stages of the PRIMM approach within a unified platform.

Moreover, web-based media can boost student motivation by providing an engaging interface and exploration-based activities (Mustaqim & Mahamad, 2023; Salam et al., 2023; Izzah et al., 2024). With features that support student exploration, collaboration, and active involvement, web-based media offers an innovative approach to programming education. Based on this background, the present study aims to develop PRIMM-based web learning media that can improve students' cognitive processes in looping structure control programming education.

METHODS

This study employs a research-based methodology for learning media development (i. e. PICOLEARN), utilizing the ADDIE model framework. The ADDIE model is a systematic and iterative instructional design approach that encompasses five key phases: Analyze (identifying needs and challenges), Design (developing instructional solutions or prototypes), Develop (creating and refining the media or product), Implement (deploying the media in real-world settings), and Evaluate (assessing effectiveness and identifying areas for improvement). This structured approach ensures that the resulting learning media aligns with user needs and effectively supports learning objectives (Branch & Kopcha, 2014). The procedural steps for this research are depicted in Figure 1 (see Figure 1).

The instructional design for web-based media integrates activities aligned with the PRIMM approach, which consists of Predict, Run, Investigate, Modify, and Make stages (Sentance et al., 2019). The learning content focuses on looping constructs within structured programming, a foundational concept in computer science education. Prior to implementation, the media underwent expert validation to assess its quality and usability. The evaluation employed the Learning Object Review Instrument (LORI), a widely recognized tool for assessing digital learning resources on dimensions such as content quality, usability, and engagement (Nesbit et al., 2002).

A pre-experimental design with a one-group pre-test and post-test format was employed to evaluate the implementation of PICOLEARN. The study participants were 35 tenth-grade vocational school students from the Software and Game Development program in Bandung, focusing on recurrence material with the objective of enhancing students' cognitive processes. The implementation began with the administration of a pre-test to assess students' baseline knowledge. This was followed by instructional treatment using PRIMM-based learning media. At the conclusion of the intervention, a post-test was administered alongside a questionnaire to collect student feedback on their experiences with PICOLEARN used throughout the teaching and learning process.

The normalized gain (n-Gain) test was applied to quantify the improvement in students' cognitive processes following the intervention. The n-Gain formula provides a standardized measure of learning effectiveness, enabling an objective comparison of pre-test and post-test results. Additionally, the Technology Acceptance Model (TAM) instrument was utilized to assess user experience, focusing on four key components: Perceived Usefulness, Perceived Ease of Use, Attitude Toward Using, and Behavioural Intention to Use. The TAM framework ensures a comprehensive evaluation of student perceptions and their willingness to adopt PICOLEARN as the learning media.

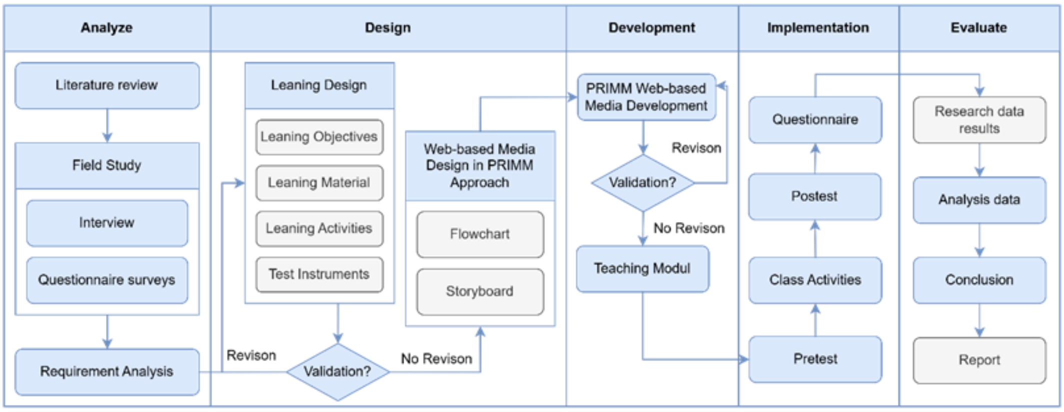

Figure 1.
Research Procedure to develop PICOLEARN with ADDIE framework

RESULTS & DISCUSSION

Results

The development of PICOLEARN is centred on providing educational resources and activity pages for students, grounded in the stages of the PRIMM framework. The instructional content includes an introduction to looping control structures, enabling students to understand fundamental concepts, followed by detailed modules on For Loops, While Loops, and Do-While Loops. The instructional design incorporates clearly defined learning objectives, materials, and activities, all aligned with the sequential stages of PRIMM to promote active engagement and comprehension.

PRIMM Approach in PICOLEARN

The PRIMM approach in PICOLEARN for For Loops learning activities is illustrated in Figures 2-6. The Predict stage interface (Figure 2) presents students with a program code to analyze and predict its behaviour (Sentance & Waite, 2021), complemented by an input field for entering their responses. This stage features three sample program codes for students to predict, with a representative example depicted in

Figure 2 to predict of the output if i=8 and n=44, then submit their answer to PICOLEARN.

```
1  int i, n;
2
3  for (i; i<=n; i++){
4
5      if (i%2 == 0){
6          cout << i << endl;
7      }
8      i+=2;
9  }
```

Figure 2.
Example code to predict For Loops control in PICOLEARN Predict page.

The Run page (Figure 3) presents the actual execution results of the code snippets initially provided on the Predict page, allowing students to compare these outcomes with their predictions. This page includes an input field where students can answer questions designed to analyze (C4) the differences between their expectations and the actual outputs (Sentance et al., 2019).

Figure 3.
Output results in PICOLEARN Run page for code of Figure 2.

The Investigate page (Figure 4) features a code snippet from Code No. 3 in Figure 2 on the Predict page. It was paired with exploratory questions as in (Sentance, 2020; Sentance & Waite, 2021) to promote application (C3), analyze (C4) and evaluate (C5) cognitive process and an input field for students to input their answers. Examples of such questions include:

a. What happens if line 8 is moved above line 4, and what will the output be when i=3 and n=35?
b. If i=5 and n=45, what will the output be if line 8 is moved to line 4?
c. Describe the loop's execution flow from lines 3 to 9 if line 8 is removed and the inputs are i=1 and n=10.
d. Explain the behaviour if line 9 is modified to i+=3 with i=1 and n=10.

Additionally, the page features a Materials button that directs students to supplementary learning resources.

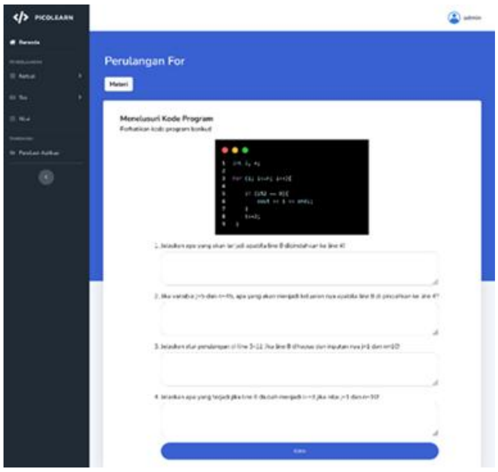

Figure 4.
The Investigate page in PICOLEARN.

The Modify page (Figure 5) presents the program code from Code No. 3 (referenced in Figure 2 from the Predict page) along with prompts encouraging students to make specific modifications (C6) to the code (Sentance et al., 2019). This page includes an input field for students to submit their modified solutions. An example of a modification task is to alter the program such that it converts even numbers to odd numbers while displaying the odd numbers in descending order. To provide clarity and guidance, this page also includes an example of the expected output, ensuring students understand the desired result of their modifications.

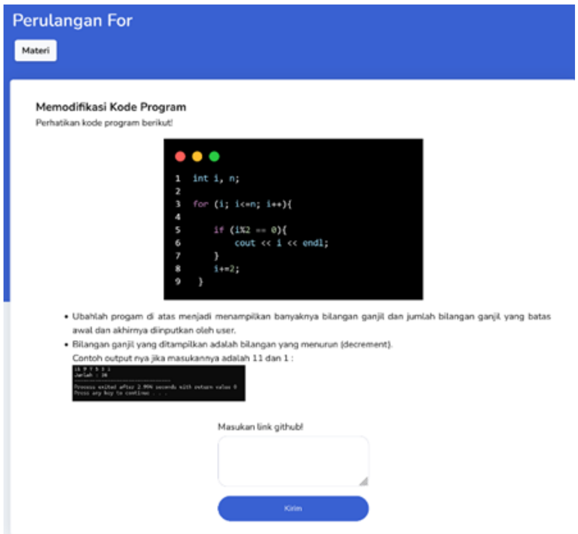

Figure 5.
The Modify page in PICOLEARN

The Make page (Figure 6) introduces a new task designed to extend students' understanding of the code structure explored in previous stages (Sentance et al., 2019).

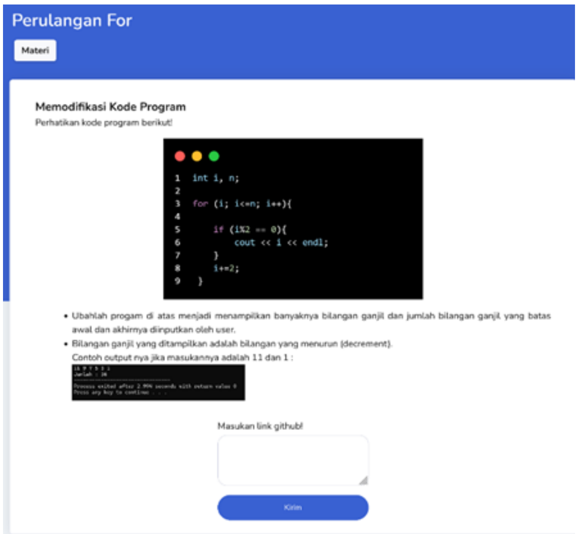

Figure 6.
The Make page in PICOLEARN.

For example, at this stage, students are tasked with creating a program that calculates the sum of even numbers, where both the number of terms and the values to be summed are provided by the user. This aligns with the objective of encouraging students to develop a completely new program inspired by the code from earlier stages but with new functionality or a different problem to solve (Peserico et al., 2023). The page also includes an input field where students can submit their solutions for review. Additionally, it includes a snippet of the expected output to guide students in verifying the correctness of their programs.

PICOLEARN Validation

Preceding implementation, PICOLEARN underwent validation by subject matter and media experts using the Learning Object Review Instrument (LORI). The evaluation was conducted using a Likert scale ranging from 1 to 5. The results of this validation process are summarized in Table 1. As presented in Table 1, content validation conducted by academic experts resulted in an overall average score of 79%, placing it in the "Very Good" category across all evaluated aspects. Similarly, media validation achieved an average score of 83%, which also falls under the "Very Good" classification. Overall, the combined validation scores yielded an average of 81%. These results demonstrate the robustness and high quality of the developed media in terms of both content and design, consistent with established criteria for effective educational resources (Nesbit et al., 2002).

Table 1.
Result of PICOLEARN Validation

Material Aspect	Percentages	Media Aspect	Percentages
Content Quality	80%	Presentation Design	80%
Learning Goal Alignment	90%	Interaction Usability	100%
Feedback and Adaptation	60%	Accessibility	70%
Motivation	80%	Reusability	80%
Average	79%	Average	83%

Cognitive Process

To assess cognitive processes, the implementation involved administering a pretest, conducting three learning sessions, and concluding with a posttest. Each session focused on a specific type of Loop Control Structure and was carried out in groups, as recommended in (Sentance et al., 2019), with 3-4 students per group. Students worked collaboratively to complete structured worksheets (LKPD) following the stages of the PRIMM framework within PICOLEARN. The effectiveness of PICOLEARN as a learning medium be evaluated through n-Gain analysis, which compares pretest and posttest scores to assess improvement in learning outcomes. This method has been successfully applied in various educational contexts (Arsyad et al., 2023; Pebriantika et al., 2024), demonstrating significant enhancements in student performance. Each test included 30 questions aimed at measuring students' cognitive process achievements (C1-C6) throughout the learning activities.

The results, presented in Table 2, show that the average improvement in learning outcomes across all groups falls within the moderate category.

Table 2.
Result of PICOLEARN Validation

Group	Average Score Pre-test	Postest	N-gain Value	Criteria
Upper	50,33	73,89	0,49	Moderate
Middle	30,00	55,22	0,36	Moderate
Lower	13,33	55,56	0,48	Moderate

Before calculating the n-Gain, a normality test was conducted to ensure that the pre-test and post-test data followed a normal distribution. The n-Gain test was then employed to evaluate the formulated hypotheses and measure the improvement in students' cognitive processing abilities after engaging with the learning media. Students were categorized into three groups—upper, middle, and lower—based on their pre-test scores.

Student's Acceptance PICOLEARN

Students' responses to PICOLEARN were evaluated using the TAM instrument. The results yielded an average score of 76%, categorized as "Very Good." These findings indicate that the PRIMM-based learning media is not only effective in enhancing cognitive learning outcomes but is also well-received by students due to its user-friendly design and practical utility. The evaluation underscores that the media successfully addresses students' needs in programming education, fostering improvements in cognitive achievement. However, to further enhance its impact, additional refinements are recommended to increase its relevance and engagement. By addressing these areas, the media can better support students' learning outcomes and enthusiasm for programming education, aligning with existing frameworks for technology-based instructional tools (Shahid et al., 2019).

Discussion

The implementation of the Predict step in this context aligns with prior research that highlights the importance of tracing and reading skills as foundational to developing programming writing skills (Fowler et al., 2022; Xie et al., 2019; Zhang et al., 2023).

Students are required to predict and identify looping control structures and basic syntax, activities that engage cognitive processes at the recall (C1) and recognition (C2) levels according to Bloom's Taxonomy (Anderson & Krathwohl, 2001). Through these predictive activities, students reinforce their fundamental comprehension of programming concepts, laying a foundation for acquiring more advanced skills in subsequent learning stages. Furthermore, the comparative activities presented on the Run page promote critical thinking, reflective learning, and error analysis, addressing issues associated with inadequate tracing skills [30] and facilitating the development of debugging abilities (Ahmadzadeh et al., 2005). This practice is consistent with educational research that underscores the significance of iterative learning and feedback mechanisms in computer science education (Messer et al., 2024).

The interactive nature of the Investigate page corresponds to educational studies advocating the use of scaffolding and inquiry-based learning to enhance programming education (Guzdial, 2023). Additionally, integrating tasks involving code modification on the Modify page supports active learning methodologies, fostering the development of problem-solving capabilities and computational thinking skills, as emphasized by research on effective programming pedagogies (Kikuchi & Hamamoto, 2016). Lastly, the focus on output expectations within the Make page reinforces problem-solving and debugging competencies crucial for mastering programming principles, aligning closely with constructivist pedagogical approaches that advocate hands-on and iterative learning experiences (Basuhail, 2019).

The evaluation of cognitive achievements, as measured by n-Gain values, generally indicated moderate improvement. Specifically, the upper group demonstrated an average n-Gain of 0.49, the middle group 0.36, and the lower group 0.48, all falling into the moderate category. The overall average n-Gain score was 0.40, signifying a moderate enhancement in students' learning outcomes. These results imply that the implementation of PICOLEARN was effective in improving student performance across various ability levels. Nevertheless, the moderate classification suggests that further refinements are necessary to maximize its effectiveness on students' cognitive development. Additionally, the overall average post-test score of 58.38 highlights the potential for further improvement in student performance. A more detailed examination indicated that the lowest scores were recorded on items related to Do-While Loops, which encompassed various cognitive dimensions. This outcome points to students' relatively limited comprehension of Do-While Loops compared to their improved understanding of For Loops and While Loops. These results align with expectations, given the distinct characteristics of Do-While Loops compared to For and While Loops (Thanyaphongphat et al., 2020).

CONCLUSION

The development of PICOLEARN as learning media utilizing the PRIMM approach has proven effective in enhancing students' cognitive processes. Following the ADDIE development model, the media was designed after conducting a needs analysis and creating materials, questions, and learner activities. It was validated by experts, achieving a feasibility score of 79% for the material aspect and 83% for the media aspect, indicating readiness for use. The use of PRIMM-based learning media resulted in a moderate improvement in students' cognitive abilities, as shown by the normalized gain (n-Gain) test, which revealed an average increase of 0.40 across all class groups. This demonstrates the effectiveness of the PRIMM approach in supporting cognitive development, particularly in mastering looping structures in programming. Additionally, the media received a positive response from students, with an average score of 76% on the TAM instrument, indicating both its effectiveness and favorable reception by users. These results imply that integrating structured programming strategies like PRIMM into digital learning platforms can significantly support students' higher-order thinking skills and engagement.

The implications of this study suggest that PICOLEARN can serve as a scalable model for programming instruction in vocational education settings, especially in regions implementing the Merdeka Curriculum. It highlights the importance of combining pedagogical frameworks with technology acceptance principles to ensure both instructional quality and user satisfaction. Future research may explore the long-term impact of PRIMM-based learning on students' problem-solving abilities, as well as its adaptability to other programming topics beyond looping structures. Moreover, iterative refinement of the media—based on student feedback and performance data—can further enhance its effectiveness and broaden its applicability across diverse educational contexts.

REFERENCES

Ahmadzadeh, M., Elliman, D., & Higgins, C. (2005, June). An analysis of patterns of debugging among novice computer science students. In *Proceedings of the 10th annual SIGCSE conference on Innovation and technology in computer science education* (pp. 84-88).

Amalia, A. R., & Yunianta, T. N. H. (2019). Deskripsi Proses Kognitif Siswa SMP dalam Menyelesaikan Masalah Matematika Berdasarkan Modes Of Representation Teori Bruner. *JRPM (Jurnal Review Pembelajaran Matematika)*, *4*(1), 58-71.

Anderson, L. W., & Krathwohl, D. R. (2001). *A taxonomy for learning, teaching, and assessing: A revision of Bloom's taxonomy of educational objectives: complete edition*. Addison Wesley Longman, Inc..

Anggraini, V. D., Mukhadis, A., & Muladi, M. (2013). Problem based learning, motivasi belajar, kemampuan awal, dan hasil belajar siswa SMK. *Jurnal Ilmu Pendidikan Universitas Negeri Malang*, *19*(2), 107342.

Arsyad, S. N., Runniarsiti, R., Swandi, A., Fauzan, A., Viridi, S., Burhan, B., & Rahmadhanningsih, S. (2023). Uji Coba Gim Edukasi dalam Meningkatkan Hasil Belajar Bahasa Inggris. *Journal of Education and Instruction (JOEAI)*, *6*(1), 47-54.

Basuhail, A. (2019). e-Learning Objects Designing Approach for Programming-Based Problem Solving. *International Journal of Technology in Education*, *2*(1), 32-41.

Bradley, S., & Ramezani, A. (2024). PRIMM and Proper: Authentic Investigation in HE Introductory Programming with PeerWise and GitHub. In *Proceedings of the 8th Conference on Computing Education Practice* (pp. 33-36).

Branch, R. M., & Kopcha, T. J. (2013). Instructional design models. In *Handbook of research on educational communications and technology* (pp. 77-87). New York, NY: Springer New York.

Cheah, C. S. (2020). Factors contributing to the difficulties in teaching and learning of computer programming: A literature review. *Contemporary Educational Technology*, *12*(2), ep272.

Dio, Y. A., & Ekohariadi, E. (2021). Pengembangan Media Pembelajaran Berbasis Web Dalam Meningkatkan Efektivitas Pembelajaran Pemrograman Web (Studi Kasus: Smks Semen Gresik). *IT-Edu: Jurnal Information Technology and Education*, *6*(3).

Elçiçek, M., & Karal, H. (2020). Cognitive problems in the process of programming teaching in higher education: learner-instructor experiences. *Turkish Online Journal of Qualitative Inquiry*, *11*(1), 140-160.

Fowler, M., Smith IV, D. H., Hassan, M., Poulsen, S., West, M., & Zilles, C. (2022). Reevaluating the relationship between explaining, tracing, and writing skills in CS1 in a replication study. *Computer Science Education*, *32*(3), 355-383.

González, A., Gallego-Sánchez, I., Gavilán-Izquierdo, J. M., & Puertas, M. L. (2021). Characterizing levels of reasoning in graph theory. *Eurasia Journal of Mathematics, Science and Technology Education*, *17*(8), em1990.

Guzdial, M. (2023). Scaffolding to support humanities students programming in a human language context. In *Proceedings of the 2023 Conference on Innovation and Technology in Computer Science Education V. 2* (pp. 579-580).

Henley, A., Ball, J., Klein, B., Rutter, A., & Lee, D. (2021, May). An inquisitive code editor for addressing novice programmers' misconceptions of program behavior. In *2021 IEEE/ACM 43rd International Conference on Software Engineering: Software Engineering Education and Training (ICSE-SEET)* (pp. 165-170). IEEE.

Izzah, D. W., Nuryantini, A. Y., & Pitriana, P. (2024). Development of Web-based Online Learning Media Using Google Sites to Increase Student Motivation and Learning Outcomes on Elasticity Material. *KnE Social Sciences*, 637-650.

Kikuchi, S., & Hamamoto, K. (2016). Investigating the relationship between tracing skill and modification skill for different programming statements. *Proceedings of the School of Information and Telecommunication Engineering, Tokai University*, *9*(1).

Kusumasari, E. D., Sumarno, S., & Dwijayanti, I. (2024). Meningkatkan kemampuan berpikir kritis siswa sekolah dasar pada pembelajaran bahasa Indonesia berbasis literasi digital pada Kurikulum Merdeka. *Tematik: Jurnal Penelitian Pendidikan Dasar*, *3*(1), 22-29.

Ma'rifah, S. N., Wijoyo, S. H., & Wicaksono, S. A. (2019). Penerapan Problem Based Learning Berbantuan Lembar Kerja Siswa Untuk Meningkatkan Hasil Belajar Pada Mata Pelajaran Pemrograman Dasar (Studi Pada: SMK Negeri 4 Malang). *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, *3*(11), 10586-10594.

Mangaroska, K., Sharma, K., Gašević, D., & Giannakos, M. (2022). Exploring students' cognitive and affective states during problem solving through multimodal data: Lessons learned from a programming activity. *Journal of Computer Assisted Learning*, *38*(1), 40-59.

Masjudin, M. (2024). Strengthening 21st century skills through an independent curriculum in mathematics education in indonesia: challenges, potential, and strategies. *International Journal of Applied Science and Sustainable Development (IJASSD)*, *6*(2), 92-113.

Messer, M., Brown, N. C., Kölling, M., & Shi, M. (2024). Automated grading and feedback tools for programming education: A systematic review. *ACM Transactions on Computing Education*, *24*(1), 1-43.

Mustaqim, M. F. & Mahamad, A. K. B. (2023). Development of Interactive Web-Based Learning Media with the Discovery Learning Model to Increase Student Learning Motivation in Object Oriented Programming XI Vocational High Schools. *Letters in Information Technology Education (LITE), 6*(1).

Nesbit, J., Belfer, K., & Vargo, J. (2002). A convergent participation model for evaluation of learning objects. *Canadian Journal of Learning and Technology/La revue canadienne de l'apprentissage et de la technologie*, *28*(3).

Nugraha, I. R. R., Supriadi, U., & Firmansyah, M. I. (2023). Efektivitas strategi pembelajaran project based learning dalam meningkatkan kreativitas siswa. *Jurnal*

Penelitian Dan Pendidikan IPS, *17*(1), 39-47.

Pacheco, C. S., & Herrera, C. I. (2021). A conceptual proposal and operational definitions of the cognitive processes of complex thinking. *Thinking skills and creativity*.

Pebriantika, L., Rahmi, J., Adesti, A., & Eriyanti, E. (2024). Efektifitas Penerapan Metode Microlearning untuk Meningkatkan Hasil Belajar Mahasiswa. *Edu Cendikia: Jurnal Ilmiah Kependidikan*.

Peserico, G., Serafini, M., Voltolini, F., Picasso, F., Agostini, D., Fiore, F., Serbati, A. S., & Montresor, A. (2023). the Primm Method for Teaching Programming: Experimentation and Validation. *ICERI2023 Proceedings*, *1*(November).

Pulungan, H., Maharani, T., Sulistyani, S., Lubis, I. H., Harahap, H., Studi, P., Indonesia, S., Medan, U. N., & Utara, P. S. (2024). *Analisis Dampak Implementasi Kurikulum Merdeka pada Pengembangan Berpikir Kritis Peserta Didik*. *3*(2), 2962–2965.

Putri, N., Musril, H. A., & Yahdi, Y. (2024). Penerapan Project Based Learning pada Mata Pelajaran Informatika di Pondok Pesantren Sematera Thawalib Parabek untuk Meningkatkan Keaktifan Belajar Peserta Didik. *Jurnal Ilmiah Teknik Informatika Dan Komunikasi*, *4*(1), 21-29.

Ramírez, C. E., Herrero, M. R., & Sosa, L. M. V. (2020). Proposed framework for employing cognitive process for resolving programming problems for novice programmers. In *2020 8th International Conference in Software Engineering Research and Innovation (CONISOFT)* (pp. 193-197). IEEE.

Salam, A. U. A., Komarayanti, S., & Sita Warsita, K. (2023). The Use of Website-Based PBL Model to Improve the Interest and Motivation in Learning Class X Students. *International Social Sciences and Humanities*, *2*(2), 479–483.

Sentance, S. (2020). *its role in dialogue and vocabulary development in programming lessons PRIMM : November*.

Sentance, S., & Waite, J. (2021). Teachers' Perspectives on Talk in the Programming Classroom : Language as a Mediator. In *ICER 2021 - Proceedings of the 17th ACM Conference on International Computing Education Research* (Vol. 1, Issue 1). Association for Computing Machinery. https://doi.org/10.1145/3446871.3469751

Sentance, S., Waite, J., & Kallia, M. (2019). Teachers' Experiences of using PRIMM to Teach Programming in School. *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*, 476–482.

Shah, A. (2023). Improving Students' Programming Processes using Cognitive Apprenticeship Methods. *Proceedings of the 2023 ACM Conference on International Computing Education Research - Volume 2*, 102–106.

Shahid, F., Aleem, M., Islam, M. A., Iqbal, M. A., & Yousaf, M. M. (2019). A review of technological tools in teaching and learning computer science. *Eurasia journal of mathematics, science and technology Education*, *15*(11), em1773.

Thanyaphongphat, J., Thongkoo, K., Daungcharone, K., & Areeprayolkij, W. (2020, March). A game-based learning approach on robotics visualization for loops in programming concepts. In *2020 Joint International Conference on Digital Arts, Media and Technology with ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunications Engineering (ECTI DAMT & NCON)* (pp. 381-385). IEEE.

Xie, B., Loksa, D., Nelson, G. L., Davidson, M. J., Dong, D., Kwik, H., Tan, A. H., Hwa, L., Li, M., & Ko, A. J. (2019). A theory of instruction for introductory programming skills. *Computer Science Education*, *29*(2–3), 205–253.

Zhang, Y., Paquette, L., Pinto, J. D., & Fan, A. X. (2023). Utilizing programming traces to explore and model the dimensions of novices' code-writing skill. *Computer Applications in Engineering Education*, *31*(4), 1041–1058.