TunasEdu: Journal of Primary Education Development
Vol 1, No 1, 2026

Basic Coding Learning for Elementary School Students Using Block-Based Programming

Amanda Putri Nazilla[1], Suprayuandi Pratama[2], Yuniansyah[3]
[1-3]Prodi Rekayasa Perangkat Lunak, Universitas Lembah Dempo, Indonesia

Article Info

Article history:

Received 02 03, 2026
Revised 11 04, 2026
Accepted 30 04, 2026

Keywords:

coding education,
computational thinking,
elementary education,
block-based programming,
digital literacy

ABSTRACT

Coding education has become an important component in developing computational thinking, creativity, and problem-solving skills among students. However, many elementary school students still have limited exposure to basic programming concepts due to the lack of simple and engaging learning materials. This study aims to develop a basic coding learning module designed specifically for elementary school students using visual and block-based programming approaches. The instructional material is presented through illustrated learning content that introduces fundamental coding concepts such as algorithms, sequences, loops, conditional statements, and variables. The development process involves the design of structured learning materials that combine simple explanations, visual illustrations, and practical examples to help students understand coding concepts more easily. The learning module is organized into several sections, including an introduction to coding, algorithmic thinking, block coding activities, mini projects, and practice exercises. The use of visual representations and interactive examples is intended to improve students' engagement and facilitate conceptual understanding. The results of this study indicate that the developed learning material provides a clear and accessible introduction to coding for elementary school students. The structured presentation and visual approach help learners understand basic programming logic in an enjoyable and meaningful way. Furthermore, the module encourages students to develop creativity and logical thinking through simple programming activities. Therefore, the proposed learning material can serve as an effective educational resource to support the integration of coding education in elementary school learning environments.

Corresponding Author:

Amanda Putri Nazilla
amandaputrinazilla@gmail.com
Prodi Rekayasa Perangkat Lunak, Universitas Lembah Dempo, Indonesia, Indonesia

1. INTRODUCTION

The rapid development of digital technology has significantly influenced various aspects of education, including the need to introduce computational thinking and coding skills at an early stage of learning. Coding education is increasingly recognized as an essential component in preparing students to face the challenges of the digital era. Through coding activities, students can develop logical thinking, creativity, and problem-solving abilities that are important for understanding how technology works in everyday life. Introducing basic programming concepts in elementary school can help students build a strong foundation in digital literacy and computational thinking.

However, many elementary schools still face challenges in implementing coding education effectively. One of the main obstacles is the limited availability of learning materials that are appropriate for young learners. Coding concepts are often perceived as complex and difficult to understand by elementary school students. Therefore, learning materials need to be designed in a simple, interactive, and visually engaging manner so that students can understand the concepts more easily. Educational approaches that integrate visual illustrations and practical examples can help students grasp abstract programming concepts in a more concrete way.

Previous studies have emphasized the importance of integrating coding education in early schooling to improve students' computational thinking and problem-solving skills. Early exposure to coding activities allows students to understand logical sequences, patterns, and structured thinking processes. Furthermore, coding learning activities can encourage creativity and collaboration among students when solving simple programming tasks. These learning experiences can also foster students' interest in science, technology, engineering, and mathematics education.

In response to these challenges, this study focuses on developing basic coding learning materials designed specifically for elementary school students. The proposed learning content introduces fundamental programming concepts such as algorithms, sequences, loops, conditional statements, and variables through simple explanations and visual illustrations. The learning materials are organized systematically, starting from the introduction of coding concepts to simple exercises and mini projects that allow students to practice coding logic.

The main contribution of this study is the development of structured and visually supported coding learning materials that are suitable for elementary school students. By presenting coding concepts in a simple and engaging way, the proposed learning materials are expected to support the integration of coding education in elementary schools and enhance students' computational thinking and digital literacy skills.

2. METHOD

2.1. Research Design

This study employed a developmental research design aimed at producing basic coding learning materials for elementary school students. Developmental research focuses on designing, developing, and evaluating educational products to improve learning effectiveness. The research was conducted through several stages, including needs analysis, design of learning materials, development of instructional content, and evaluation of the developed materials. This approach allows researchers to systematically develop educational materials that are suitable for learners and aligned with instructional objectives (Li & Zheng, 2018; Ocak & Yamaç, 2013).

2.2 Research Procedure

The research procedure was carried out in several stages. First, a needs analysis was conducted to identify the challenges faced by elementary school students in understanding basic programming concepts. This stage involved reviewing relevant literature and identifying suitable coding concepts for young learners. Second, the design stage focused on organizing the learning materials and determining the structure of the coding module. The learning content was structured into several sections, including introduction to coding, algorithms, sequences, loops, conditional statements, and simple programming exercises.

Third, the development stage involved creating visual-based learning materials that combine explanations, illustrations, and simple coding activities. The use of visual learning approaches aims to improve students' engagement and facilitate understanding of abstract programming concepts. Finally, the evaluation stage was conducted to examine the clarity, structure, and usability of the developed learning materials in supporting coding education for elementary school students (Roick & Ringeisen, 2018).

2.3 Learning Design Algorithm

The learning flow of the coding module is presented in the following algorithm:

Algorithm 1. Coding Learning Process
Start
Introduce basic concept of coding
Explain algorithm and sequence concept
Provide visual examples of coding instructions
Guide students to practice simple block-based coding
Introduce loops and conditional statements
Give students simple coding exercises
Evaluate students' understanding
End

This structured learning process allows students to gradually understand basic programming logic and computational thinking.

2.4 Data Collection

Data collection in this study was conducted through documentation and evaluation of the developed learning materials. The data were obtained from the development process, including instructional design documents, coding learning modules, and visual learning materials. These data were used to evaluate the structure, clarity, and effectiveness of the learning content in supporting coding education for elementary school students.

2.5 Data Analysis

The data analysis was conducted using descriptive analysis to evaluate the developed learning materials. The analysis focused on examining the suitability of the coding concepts, clarity of instructional explanations, and the effectiveness of visual learning elements in supporting students' understanding of basic programming concepts. Descriptive analysis is commonly used in educational research to interpret the effectiveness of instructional materials and learning designs (Pintrich et al., 1991; Zimmerman & Moylan, 2009).

3. RESULTS AND DISCUSSION (10 PT)

This section presents the results of developing basic coding learning materials for elementary school students. The results include the structure of the learning module, examples of coding concepts introduced in the module, and the implementation of coding activities designed to improve students' computational thinking skills.

3.1 Structure of the Coding Learning Module

The developed learning module is organized systematically to help elementary school students understand basic coding concepts gradually. The module consists of several sections including introduction to coding, algorithm concepts, block-based programming, loops and conditions, and simple coding projects. The learning materials are designed using visual explanations and simple instructions to make the concepts easier for young learners to understand.

The module begins with an introduction to coding and computational thinking. Students are first introduced to the concept of algorithms as step-by-step instructions used to solve problems. This foundational concept helps students understand how computers execute commands sequentially.

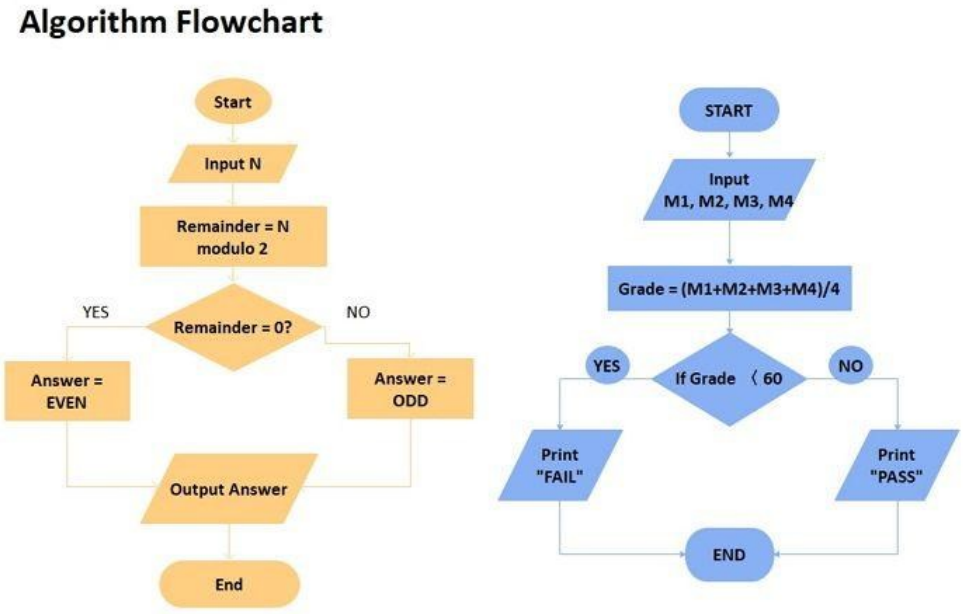

Figure 1. algorithm flowchart

Figure 1 illustrates a simple algorithm representation using a flowchart. The diagram demonstrates how a sequence of instructions begins with an input process, followed by decision-making steps, and ends with an output. Flowcharts help students visualize problem-solving processes and understand the logical flow of programming instructions.

Using visual algorithm representations helps students understand that programming is essentially a structured process of solving problems step by step. This understanding forms the basis for learning more advanced coding concepts such as loops and conditions.

3.2 Introduction to Block-Based Programming

After learning about algorithms, students are introduced to block-based programming environments. Block programming allows learners to create programs by combining visual blocks representing commands. This approach reduces syntax errors and enables students to focus on logic and problem-solving rather than programming syntax.

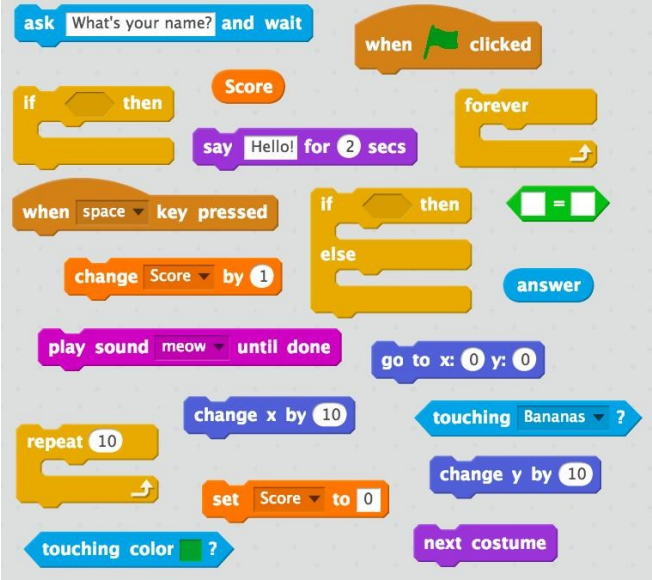

Figure 2. Example of block-based coding interface

Figure 2 shows examples of block-based programming environments commonly used in coding education, such as visual programming platforms for beginners. In this environment, commands are represented as colored blocks that can be dragged and connected like puzzle pieces. Each block represents a specific instruction, such as movement, sound, or event triggers.

Block-based programming is widely used in elementary education because it simplifies the learning process and encourages creativity. Students can easily create animations, simple games, and interactive stories by arranging blocks logically.

3.3 Implementation of Loop and Conditional Concepts

The coding module also introduces looping and conditional structures, which are essential concepts in programming. Loops allow instructions to be repeated multiple times, while conditional statements allow programs to make decisions based on specific conditions.

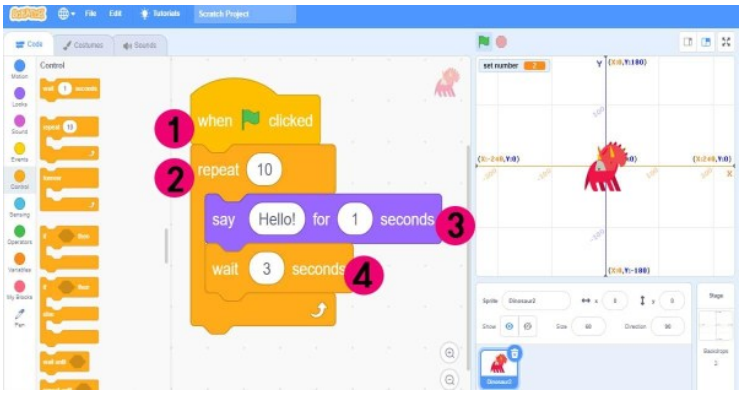

Figure 3. loop and conditional structures in block coding

Figure 3 presents examples of loop structures in block-based programming. The repeat block allows students to execute commands multiple times, enabling them to create repetitive actions such as animation movement or repeated sound effects. Conditional blocks allow students to create programs that respond to specific events or user interactions.

Through these coding activities, students begin to understand how logical structures are used in programming to control program behavior. These concepts help students develop computational thinking and logical reasoning skills.

3.4 Coding Learning Activities in the Classroom

The developed coding learning materials are designed to be implemented in classroom environments through interactive activities. Students are encouraged to experiment with simple coding tasks, such as creating animations, moving characters, or designing simple interactive games.

figure 4. Coding learning activities in elementary school classrooms

Figure 4 illustrates examples of coding learning activities conducted in classroom environments. Students interact with computers and programming tools while teachers provide guidance during the learning process. Collaborative learning activities also encourage students to share ideas and solve problems together.

These classroom activities demonstrate that coding learning can be engaging and interactive for elementary school students. The integration of visual learning materials, block-based programming environments, and collaborative activities helps create a positive learning experience.

3.5 Discussion

The results of this study indicate that the developed coding learning materials provide an effective introduction to programming concepts for elementary school students. The structured learning approach allows students to gradually understand complex concepts through simple explanations and visual representations.

The use of block-based programming environments helps students focus on problem-solving and logical thinking rather than syntax memorization. This approach aligns with modern educational practices that emphasize computational thinking as an essential skill in the digital era.

Furthermore, the integration of visual learning materials and hands-on coding activities enhances students' engagement during the learning process. Students become more motivated to explore programming concepts and create simple digital projects. The collaborative nature of coding activities also supports communication and teamwork among students.

Overall, the developed coding learning module can serve as an effective educational resource for introducing coding concepts in elementary education. The findings suggest that integrating coding education into early learning environments can support the development of digital literacy, creativity, and problem-solving skills among students.

4. CONCLUSION

This study aimed to develop basic coding learning materials designed for elementary school students to support the introduction of computational thinking and programming concepts at an early stage of education. The developed learning module presents fundamental coding concepts such as algorithms, sequences, loops, and conditional statements through visual explanations, simple instructions, and block-based programming activities. The structured learning design allows students to gradually understand coding concepts while engaging in interactive learning experiences.

The results indicate that the coding learning materials provide an accessible and engaging approach for introducing programming to young learners. The use of visual illustrations and block-based programming environments helps students understand abstract concepts more easily while encouraging creativity and logical reasoning. In addition, hands-on coding activities and simple projects motivate students to actively participate in the learning process and apply problem-solving skills in practical situations.

Furthermore, the developed learning module supports the integration of digital literacy and computational thinking in elementary education. By providing structured and visually supported learning materials, teachers can introduce coding concepts more effectively in classroom environments. The findings suggest that coding education at the elementary level can contribute to the development of students' analytical thinking, creativity, and technological awareness.

Future studies may further evaluate the effectiveness of the developed coding module through classroom experiments involving larger student groups. Such studies can provide deeper insights into the impact of coding education on students' learning outcomes and computational thinking skills.

ACKNOWLEDGMENTS

The authors would like to express their sincere appreciation to colleagues and educators who provided valuable insights and suggestions during the development of this study. Their constructive feedback helped improve the clarity and structure of the coding learning materials designed for elementary school students. The authors also acknowledge the support of educational practitioners who shared their experiences and perspectives regarding the implementation of coding education in primary school environments. All individuals mentioned in this acknowledgment have given their consent to be included.

REFERENCES

Bandura, A. (1985). Social foundations of thought and action: A social cognitive theory. Prentice-Hall.

Cleary, T. J., & Kitsantas, A. (2017). Motivation and self-regulated learning influences on middle school mathematics achievement. School Psychology Review, 46(1), 88–107.

Dent, A. L., & Koenka, A. C. (2015). The relation between self-regulated learning and academic achievement across childhood and adolescence. Educational Psychology Review, 28(3), 425–474.

Li, X., & Zheng, Y. (2018). The effects of coding education on computational thinking development. Journal of Educational Computing Research, 56(5), 1–15.

Ocak, G., & Yamaç, A. (2013). The effects of learning environments on students' problem-solving skills. Educational Sciences: Theory & Practice, 13(2), 987–993.

Pintrich, P. R. (2000). The role of goal orientation in self-regulated learning. In M. Boekaerts, P. Pintrich, & M. Zeidner (Eds.), Handbook of self-regulation (pp. 451–502). Academic Press.

Pintrich, P. R., Smith, D., Garcia, T., & McKeachie, W. (1991). A manual for the use of the motivated strategies for learning questionnaire (MSLQ). University of Michigan.

Pressley, M., & McCormick, C. (1995). Advanced educational psychology for educators, researchers, and policymakers. HarperCollins.

Reichert, F., Chen, J., & Torney-Purta, J. (2020). Profiles of adolescents' civic engagement and citizenship norms. Youth & Society, 52(6), 1020–1045.

Roick, J., & Ringeisen, T. (2018). Students' self-regulated learning in digital environments. Computers in Human Behavior, 78, 358–365.

Vonkova, H., & Hrabak, J. (2015). The impact of digital technology on learning outcomes in schools. Educational Technology Research and Development, 63(3), 377–392.

Wing, J. M. (2006). Computational thinking. Communications of the ACM, 49(3), 33–35.

Papert, S. (1980). Mindstorms: Children, computers, and powerful ideas. Basic Books.

Resnick, M., Maloney, J., Monroy-Hernández, A., et al. (2009). Scratch: Programming for all. Communications of the ACM, 52(11), 60–67.

Grover, S., & Pea, R. (2013). Computational thinking in K-12 education. Educational Researcher, 42(1), 38–43.

Brennan, K., & Resnick, M. (2012). New frameworks for studying computational thinking. Proceedings of the American Educational Research Association, 1–25.

Lye, S. Y., & Koh, J. H. (2014). Review on teaching and learning of computational thinking through programming. Computers in Human Behavior, 41, 51–61.

Shute, V., Sun, C., & Asbell-Clarke, J. (2017). Demystifying computational thinking. Educational Research Review, 22, 142–158.

Weintrop, D., Beheshti, E., Horn, M., et al. (2016). Defining computational thinking for mathematics and science classrooms. Journal of Science Education and Technology, 25(1), 127–147.

Kafai, Y., & Burke, Q. (2014). Connected code: Why children need to learn programming. MIT Press.