Global Journal of Mathematics
Vol. 1 No.1 2026 Page 1-8
E-ISSN: XXXX-XXXX, DOI: 10.XXXXX/XXXX

Development of a Computational Thinking-Based Learning Model to Improve Students' Algebraic Thinking Skills

Niken Fauzi[1]*, Rahmat Yunita[2], Anisa Puspitasari[3]
[1,2] Department of Educational Technology, Faculty of Education, Universitas Negeri Malang, Malang, Indonesia
[3] Department of Physical Education, Faculty of Sports Science, Universitas Negeri Jakarta, Jakarta, Indonesia

Article Info

Received : 11 January 2026
Revised : 20 March 2026
Accepted : 12 April 2026

Keyword:

Algebraic thinking;
Computational thinking;
Mathematics learning;
Student reasoning.

ABSTRACT

This study aims to describe the development of a computational thinking-based learning model to improve students' algebraic thinking skills. The study addresses students' difficulties in understanding algebraic structures, recognizing patterns, using variables, and constructing symbolic generalizations. A qualitative case study approach was used to explore the learning process in depth. The research involved Grade VIII students who participated in algebra learning activities designed around decomposition, pattern recognition, abstraction, and algorithmic thinking. Data were collected through semi-structured interviews, classroom observations, documentation, and analysis of students' written work. The findings revealed four main themes. First, decomposition helped students identify known information, unknown elements, and relationships within algebraic problems. Second, pattern recognition supported students in moving from repeated arithmetic procedures to generalization. Third, abstraction helped students understand variables as representations of quantities, patterns, and mathematical relationships. Fourth, algorithmic thinking enabled students to organize solution steps systematically and evaluate errors in their reasoning. The study shows that computational thinking can strengthen algebraic thinking by making students' reasoning processes more visible, structured, and reflective. These findings contribute theoretically to the integration of computational thinking and algebraic thinking in mathematics education. Practically, the model offers teachers a structured approach for designing algebra instruction that emphasizes reasoning, representation, and problem-solving processes. Future studies are recommended to examine the model across different grade levels, school contexts, and algebraic topics.

***) Corresponding Author:**

Niken Fauzi,
Department of Educational Technology, Faculty of Education, Universitas Negeri Malang, Malang, Indonesia
Email: niken.fauzi051@gmail.com

1. Introduction

The evolution of mathematics education in the 21st century calls for learning that emphasizes not only mastery of procedures but also the ability to think logically, systematically, creatively, and reflectively. One skill that is becoming increasingly important in this context is computational thinking, as it helps students break down complex problems, recognize patterns, construct abstractions, and design structured solution steps. Angeli and Giannakos (2020) explain that computational thinking education still faces conceptual and pedagogical challenges because schools often treat it as a technological skill rather than a cross-disciplinary way of thinking. In mathematics learning, this challenge is evident when students can follow examples of problem-solving but are unable to explain the underlying reasons, patterns, and mathematical relationships behind those procedures.

In a global context, the integration of computational thinking into mathematics education is gaining increasing attention because the two are strongly interconnected in problem-solving processes. Tang et al. (2020) demonstrate that assessments of computational thinking must focus on students' thinking processes, not merely on final products or correct answers. This is particularly relevant to algebra instruction because students must not only manipulate symbols but also understand relationships between variables, the structure of expressions, patterns of change, and generalizations. Rich et al. (2020) emphasize that mathematical thinking and computational thinking share significant overlaps, particularly in activities involving representation, decomposition, pattern reasoning, and algorithm design. Thus, algebra instruction can serve as a strategic space for meaningfully developing computational thinking.

In Indonesia, challenges in algebra instruction frequently arise at the levels of conceptual understanding and symbolic reasoning. Many students can mimic the teacher's steps when solving equations, but struggle when the problem format changes, when information is presented in a story context, or when they are asked to explain patterns verbally. Rustika and Rohaeti (2020) found that students' algebraic thinking skills remain at a moderate level, so instruction should be directed toward strengthening interest, understanding, and more structured thinking activities. Similar findings are evident in the study by Permatasari et al. (2021), which indicates that students face challenges in generative activities, particularly when constructing generalizations from patterns and transforming them into symbolic forms.

Theoretically, algebraic thinking encompasses the ability to recognize patterns, make generalizations, use symbols, understand relationships between quantities, and interpret mathematical structures. Pitta-Pantazi et al. (2020) emphasize that algebraic thinking is not monolithic, as students may demonstrate different forms of reasoning when faced with numerical, symbolic, functional, or relational situations. In classroom learning, these differences in reasoning are often overlooked if teachers assess only the final answer. Therefore, computational thinking-based learning is considered relevant because it can guide students to demonstrate their thought processes through the stages of decomposition, pattern recognition, abstraction, and algorithmization. Bråting and Kilhamn (2021) also show that semiotic representations in computational thinking can interact with algebraic thinking through symbols, patterns, and solution structures.

Several recent studies indicate that integrating computational thinking into mathematics education can support student understanding, though its implementation still requires clear pedagogical design. Subramaniam et al. (2022) conclude that programming activities, robotics, and digital media are widely used to foster computational thinking in mathematics education. Ye et al. (2023) found that research on the integration of computational thinking in K-12 mathematics is growing rapidly, but there remains a need to explain how such integration directly supports the learning of mathematical concepts. Ng and Cui (2021) demonstrated that programming contexts can enrich students' mathematical problem-solving if teachers direct computational activities toward clear mathematical objectives. These findings reinforce the need to develop learning models that not only utilize technology but also connect computational activities with algebraic thinking structures.

In algebraic learning practice, the development of computational thinking-based models must consider students' learning experiences in the classroom. Utami et al. (2023) demonstrated that students construct functional thinking through activities involving pattern discovery and explaining linear relationships. Salwadila and Hapizah (2024) found that students' computational thinking abilities in mathematics learning still vary, particularly regarding the indicators of decomposition, pattern recognition, abstraction, and algorithms. Anggraini and Pratiwi (2024) also found that indicators of algebraic thinking in generative, transformational, and meta-global activities do not always emerge evenly in the problem-solving process. This situation indicates that teachers require a learning model capable of guiding students from concrete experiences toward symbolic representations and more mature algebraic reasoning.

Based on the above discussion, this study aims to develop and describe a computational thinking-based learning model to enhance students' algebraic thinking skills. The focus of the study is on how the stages of computational thinking are designed in algebra instruction, how students interpret decomposition, pattern recognition, abstraction, and algorithmization activities, and how these processes help them build generalizations and symbolic representations. This study makes a theoretical contribution by strengthening the relationship between computational thinking and algebraic thinking in mathematics education. Practically, this study provides a foundation for teachers to design more systematic, reflective, and student-centered algebra instruction to address mathematical problems involving patterns, relations, and symbols.

2. Materials and Methods

This study employs a qualitative approach with a case study design. This design was chosen because the research focuses on the development and limited implementation of a computational thinking-based learning model within an algebra classroom context. The case study allows the researcher to deeply understand how students respond to learning stages, how teachers facilitate thinking processes, and how classroom activities shape algebraic thinking skills. The selection of this approach aligns with the research's need to not only assess final outcomes but also explore students' meanings, experiences, strategies, and barriers during the learning process. Ye et al. (2023) emphasize that research on computational thinking in mathematics education still requires studies that clarify the concrete relationship between instructional design, student activities, and mathematics learning outcomes.

The study is designed to be conducted at a public junior high school in Wonosobo Regency, Central Java, during the second semester of the 2025/2026 academic year. The research subjects consist of eighth-grade students currently studying algebraic expressions, linear equations with one variable, and number patterns. Key informants were selected using purposive sampling, considering variations in mathematical ability, learning engagement, and students' ability to explain their thinking processes. Criteria for informants include students who attended the entire learning sequence, completed computational thinking-based assignments, and were willing to participate in interviews. If initial data indicated students with distinctive thinking strategies or those differing from general patterns, snowball sampling was used to add informants through recommendations from mathematics teachers. This strategy helped the researcher obtain richer data on the variations in students' algebraic thinking processes.

Data collection was conducted through semi-structured interviews, classroom observations, documentation, and analysis of student work. Semi-structured interviews were used to explore students' experiences in solving algebraic problems, recognizing patterns, making abstractions, and formulating solution steps. Observations were conducted during the implementation of the learning model to record teacher-student interactions, student responses to instructions, the use of representations, and obstacles that arose during class discussions. Documentation included learning materials, student activity sheets, teacher reflection notes, photos of activities, and student work. Analysis of student work documents was used to trace indicators of algebraic thinking, particularly generative, transformational, and meta-global activities as discussed in the study by Anggraini and Pratiwi (2024).

Data validity was ensured through source triangulation, method triangulation, member checking, and audit trails. Source triangulation is conducted by comparing data from students, teachers, observation results, and learning documents. Method triangulation is conducted by comparing the results of interviews, observations, and analyses of student work. Member checking is conducted by confirming the summary of interview results with informants to ensure that the meaning of students' experiences does not deviate from their statements. An audit trail is maintained by preserving field notes, interview guidelines, coding results, category revisions, and analytical decisions throughout the research. This process is crucial because qualitative studies on computational thinking and algebraic thinking require transparency in interpreting students' thought processes, not merely classifying answers as correct or incorrect.

Data analysis employed thematic analysis using the interactive model by Miles and Huberman, which includes data condensation, data presentation, and drawing conclusions. Interview data, observations, and documents were open-coded based on computational thinking indicators: decomposition, pattern recognition, abstraction, and algorithmization. These codes were then linked to algebraic thinking indicators, namely pattern generalization, symbolic transformation, understanding of variables, and relational reasoning. Subsequently, the researchers identified key themes such as "decomposition as the starting point for understanding algebraic structures," "patterns as a bridge to generalization," and "algorithms as a strategy for symbolic problem-solving." The results of the analysis were used to explain how a computational thinking-based learning model helps students build algebraic thinking skills in a gradual, contextual manner, and can be replicated to a limited extent in classes with similar characteristics.

3. Results and Discussions

The findings show that the computational thinking-based learning model helped students understand algebra in a more gradual and structured way. Based on interviews, classroom observations, and analysis of students' written work, four main themes emerged from the data: problem decomposition in algebra, pattern recognition as a basis for generalization, abstraction in the use of algebraic symbols, and algorithmic thinking as a systematic problem-solving strategy. These themes indicate that students' algebraic thinking skills did not develop instantly. They emerged through repeated learning activities that encouraged students to explain their reasoning, compare strategies, identify errors, and revise their solutions.

The first theme was problem decomposition in algebra. At the beginning of the learning process, many students found it difficult to solve algebraic word problems and pattern-based tasks. They often tried to find a formula immediately without first identifying the information given in the problem. After the teacher guided them to break the problem into smaller parts, students began to identify known information, unknown information, mathematical relationships, and possible solution steps. One student stated, "When I looked at the problem directly, I did not know where to start. But when I separated the numbers, the pattern, and what was being asked, it became easier." This statement shows that decomposition helped students reduce confusion and provided a clearer entry point for understanding algebraic structure.

The second theme was pattern recognition as a bridge to generalization. In activities involving number patterns and algebraic forms, students began to understand that algebra is not only about letters and operations. It also involves repeated relationships and general rules. At the beginning, some students continued patterns by counting one by one. However, after group discussion, they started to identify regularities and formulate general rules. One student explained, "At first, I kept adding the numbers one by one, but after I saw the same distance between them, I could make a faster way." This finding shows that pattern recognition helped students move from arithmetic procedures to algebraic reasoning. Classroom observation also showed that group discussion encouraged students to compare their strategies with those of their peers.

The third theme was abstraction in the use of algebraic symbols. Before the implementation of the computational thinking-based model, several students viewed letters in algebra as confusing symbols. They did not fully understand that symbols could represent unknown numbers, quantities, patterns, or general relationships. After students were guided to move from concrete patterns to tables and then to symbolic forms, they began to understand variables more meaningfully. One student said, "I finally understood that a letter is not only x or y. It can mean an unknown number or the order of a pattern." This response shows a shift in students' understanding of algebraic symbols. Symbols were no longer seen as isolated signs, but as representations of mathematical relationships.

The fourth theme was algorithmic thinking as a systematic problem-solving strategy. Students who previously solved algebraic problems randomly began to organize their solution steps more coherently. Their worksheets showed that they started by reading the problem, identifying variables, constructing relationships, writing equations, and solving them step by step. The teacher also noted that students were more able to explain their solutions when they were asked to write their thinking process in sequence. One student stated, "When the steps are written in order, I can see which part is wrong." This finding suggests that algorithmic thinking helped students not only reach an answer but also evaluate and correct their own reasoning.

Another important finding concerned changes in classroom interaction. Students who were usually passive became more involved in discussion because the learning activities allowed them to explain simple strategies before moving to symbolic forms. The classroom became more dialogic because the teacher did not immediately provide formulas. Instead, the teacher asked guiding questions such as "Which part of the problem can be separated?", "What pattern can you see?", and "How can this rule be written in algebraic form?" These interactions show that the computational thinking-based learning model supported a more reflective learning culture. Students did not only follow the teacher's instructions. They also learned to construct arguments and defend their reasoning.

The documentation of students' work also showed improvement in how they represented algebraic problems. In the initial tasks, many students only wrote final answers without showing their reasoning. In later tasks, students began to use tables, patterns, symbols, and ordered solution steps. This development indicates that visual and symbolic representations played an important role in strengthening students' algebraic thinking. However, not all students developed at the same pace. Some students still struggled to transform verbal patterns into symbolic expressions. This difficulty was more visible among students who had weak basic arithmetic skills and limited experience in writing mathematical explanations.

Overall, the findings indicate that the computational thinking-based learning model supported students' algebraic thinking through a gradual and meaningful learning process. Decomposition helped students understand the structure of problems. Pattern recognition helped them identify regularities and build generalizations. Abstraction helped them transform concrete situations into symbolic representations. Algorithmic thinking helped them arrange solution steps systematically. These four processes were interconnected and formed the basis for improving students' algebraic thinking skills. The findings also suggest that algebra learning becomes more effective when students are given opportunities to display their thinking process, not only to produce final answers.

Discussion

The findings strengthen the view that computational thinking has a close relationship with algebraic thinking. In algebra learning, students need more than procedural fluency. They need to understand patterns, relationships, variables, and mathematical structures. The results show that decomposition, pattern recognition, abstraction, and algorithmic thinking can serve as a learning framework that supports students' movement from concrete understanding to symbolic reasoning. This finding is consistent with Rich et al. (2020), who explain that mathematical thinking and computational thinking intersect in representation, pattern identification, decomposition, and strategy construction.

The findings on decomposition show that students understood algebraic problems more easily when they broke them into smaller parts. This result supports Tang et al. (2020), who argue that the assessment of computational thinking should focus on students' thinking processes, not only on final answers. In this study, decomposition made students' reasoning more visible. The teacher could identify whether students understood the information in the problem, the relationship between data, and the goal of the solution. Therefore, decomposition functioned as an initial step in building more systematic algebraic reasoning.

The findings on pattern recognition show that students began to build generalizations after they were encouraged to compare, classify, and explain regularities. This result is in line with Utami et al. (2023), who found that students developed functional thinking through activities that involved identifying patterns and explaining linear relationships. In this study, pattern recognition helped students understand that algebra is not only a process of manipulating symbols. It is also a way to express general rules. Therefore, pattern-based activities should become an important part of early algebra learning before students move to more formal equations.

The findings on abstraction show that students began to understand variables when they experienced a gradual transition from concrete patterns to tables and symbolic forms. This finding supports Bråting and Kilhamn (2021), who state that the relationship between computational thinking and algebraic thinking can be observed through semiotic representations. In this study, algebraic symbols became more meaningful when students understood how those symbols emerged from contexts and patterns. However, students' difficulty in transforming verbal forms into symbolic expressions shows that abstraction remained the most challenging stage. This finding indicates the need for learning activities that bridge concrete contexts, visual representations, tables, and algebraic symbols.

The findings on algorithmic thinking show that students became more capable of arranging solution steps after they understood the structure of the problem. This finding supports Salwadila and Hapizah (2024), who explain that computational thinking in mathematics includes decomposition, pattern recognition, abstraction, and algorithmic thinking. In algebra, algorithmic thinking helped students view problem-solving as a process that can be checked and improved. When students wrote their solution steps in order, they learned not only to solve problems but also to reflect on their errors. This reflective aspect is important because many algebraic errors occur not only because students forget formulas, but also because they do not understand the relationship between steps.

The findings also expand previous studies on students' difficulties in algebraic thinking. Permatasari et al. (2021) found that students often experience difficulties in generational activities, especially when they need to construct generalizations from patterns. Similar difficulties appeared in this study, but they were reduced through computational thinking-based activities. Students who initially relied on repeated counting began to identify general rules after the teacher provided guiding questions and representational tasks. This indicates that computational thinking can serve as a pedagogical approach to help students overcome difficulties in algebraic generalization.

From a theoretical perspective, this study shows that computational thinking can be positioned as a thinking framework that supports the development of algebraic thinking. Pitta-Pantazi et al. (2020) explain that algebraic thinking consists of different forms, including symbolic, relational, and functional reasoning. The present findings show that these forms of reasoning can be supported through explicit learning stages. Decomposition supports relational understanding. Pattern recognition supports functional reasoning. Abstraction supports symbolic representation. Algorithmic thinking supports systematic problem-solving. Therefore, the computational thinking-based learning model offers a clearer structure for teachers in developing students' algebraic thinking.

From a practical perspective, the findings suggest that mathematics teachers should not begin algebra instruction directly with formulas. Teachers need to start from contextual problems, simple patterns, group discussions, and gradual representations. Classroom activities should also give students opportunities to explain their reasoning, not only to write answers. The computational thinking-based learning model can be implemented through worksheets that contain problem-solving stages, reflective questions, and representation-based tasks. Through this approach, teachers can monitor students' thinking processes more clearly and provide support at the right stage.

However, the implementation of the computational thinking-based learning model requires teacher readiness and sufficient instructional time. Some students still needed intensive guidance when transforming patterns into symbolic expressions. This indicates that the model cannot be applied instantly. Teachers need to build students' systematic thinking habits through repeated activities and varied algebraic contexts. In addition, students' basic mathematical skills need to be strengthened because weaknesses in arithmetic operations may hinder abstraction and algorithmic reasoning.

Critically, the findings show that the success of algebra learning is not determined only by the use of a new learning model. It also depends on the quality of teacher-student interaction during the learning process. The computational thinking-based model became effective when the teacher asked guiding questions, allowed discussion, and helped students connect concrete representations with symbolic forms. Therefore, future research should test this model at different educational levels, in more complex algebra topics, and in schools with diverse characteristics. Further studies may also develop more detailed observation instruments to examine the relationship between computational thinking indicators and the development of students' algebraic thinking skills.

4. Conclusions

This study concludes that the computational thinking-based learning model supports the development of students' algebraic thinking skills through four connected learning processes: decomposition, pattern recognition, abstraction, and algorithmic thinking. The findings show that students understood algebraic problems more effectively when they were guided to break down information, identify regularities, transform concrete patterns into symbolic forms, and arrange solution steps systematically. These processes helped students move from procedural problem-solving toward more meaningful algebraic reasoning. The study contributes to the literature by showing that computational thinking can function as a pedagogical framework for strengthening algebraic thinking, especially in helping students understand variables, patterns, relationships, and symbolic representations.

The findings imply that mathematics teachers need to design algebra instruction that gives students space to explain their reasoning, compare strategies, and reflect on errors. In practice, the model can support classroom activities through contextual problems, structured worksheets, guided questions, group discussion, and gradual representation from concrete forms to algebraic symbols. From a policy perspective, the findings suggest the need to integrate computational thinking more explicitly into mathematics learning design, teacher training, and curriculum development. Future research can test this model in different school contexts, wider participant groups, and more complex algebra topics to examine its consistency, adaptability, and long-term impact on students' mathematical reasoning.

References

Angeli, C., & Giannakos, M. (2020). Computational thinking education: Issues and challenges. *Computers in Human Behavior, 105*, 106185. https://doi.org/10.1016/j.chb.2019.106185

Anggraini, W., & Pratiwi, W. D. (2024). Students' algebraic thinking skills through creative problem solving learning models. *SJME (Supremum Journal of Mathematics Education), 8*(1). https://doi.org/10.35706/sjme.v8i1.10796

Bråting, K., & Kilhamn, C. (2021). Exploring the intersection of algebraic and computational thinking. *Mathematical Thinking and Learning, 23*(2), 170–185. https://doi.org/10.1080/10986065.2020.1779012

Kaufmann, O. T., & Stenseth, B. (2021). Programming in mathematics education. *International Journal of Mathematical Education in Science and Technology, 52*(7), 1029–1048. https://doi.org/10.1080/0020739X.2020.1736349

Muhammad, I., Rusyid, H. K., Maharani, S., & Angraini, L. M. (2024). Computational thinking research in mathematics learning in the last decade: A bibliometric review. *International Journal of Education in Mathematics, Science and Technology, 12*(1), 178–202. https://doi.org/10.46328/ijemst.3086

Ng, O., & Cui, Z. (2021). Examining primary students' mathematical problem-solving in a programming context: Towards computationally enhanced mathematics education. *ZDM Mathematics Education, 53*(4), 847–860. https://doi.org/10.1007/s11858-020-01200-7

Permatasari, D., Azka, R., & Fikriya, H. (2021). Exploring students' algebraic thinking in generational activities and their difficulties. *Beta: Jurnal Tadris Matematika, 14*(1), 53–68. https://doi.org/10.20414/betajtm.v14i1.418

Pitta-Pantazi, D., Chimoni, M., & Christou, C. (2020). Different types of algebraic thinking: An empirical study focusing on middle school students. *International Journal of Science and Mathematics Education, 18*(5), 965–984. https://doi.org/10.1007/s10763-019-10003-6

Rich, K. M., Spaepen, E., Strickland, C., & Moran, C. (2020). Synergies and differences in mathematical and computational thinking: Implications for integrated instruction. *Interactive Learning Environments, 28*(3), 272–283. https://doi.org/10.1080/10494820.2019.1612445

Rodríguez-Martínez, J. A., González-Calero, J. A., & Sáez-López, J. M. (2020). Computational thinking and mathematics using Scratch: An experiment with sixth-grade students. *Interactive Learning Environments, 28*(3), 316–327. https://doi.org/10.1080/10494820.2019.1612448

Rustika, P., & Rohaeti, T. (2020). Algebraic thinking ability and learning interest through social media-based pictorial puzzle in new normal era. *MaPan: Jurnal Matematika dan Pembelajaran, 8*(2), 329–342. https://doi.org/10.24252/mapan.2020v8n2a11

Salwadila, T., & Hapizah, H. (2024). Computational thinking ability in mathematics learning of exponents in grade IX. *Infinity Journal, 13*(2), 441–456. https://doi.org/10.22460/infinity.v13i2.p441-456

Subramaniam, S., Maat, S. M., & Mahmud, M. S. (2022). Computational thinking in mathematics education: A systematic review. *Cypriot Journal of Educational Sciences, 17*(6), 2029–2044. https://doi.org/10.18844/cjes.v17i6.7494

Tang, X., Yin, Y., Lin, Q., Hadad, R., & Zhai, X. (2020). Assessing computational thinking: A systematic review of empirical studies. *Computers & Education, 148*, 103798. https://doi.org/10.1016/j.compedu.2019.103798

Ye, H., Liang, B., Ng, O., & Chai, C. S. (2023). Integration of computational thinking in K-12 mathematics education: A systematic review on CT-based mathematics instruction and student learning. *International Journal of STEM Education, 10*, 3. https://doi.org/10.1186/s40594-023-00396-w