Global Journal of Mathematics
Vol. 1 No.1 2026 Page 19-28
E-ISSN: XXXX-XXXX, DOI: 10.XXXXX/XXXX

Integrating Computational Thinking into Algebra Instruction to Enhance Problem-Solving Skills

Lestari Ningsih[1]*, Rangga Permadi[2], Ratna Hermawan[3]
[1,2] Department of Mathematics Education, Faculty of Teacher Training and Education, Universitas Pendidikan Indonesia, Bandung, Indonesia
[3] Department of Educational Technology, Faculty of Education, Universitas Negeri Malang, Malang, Indonesia

Article Info

Received : 19 January 2026
Revised : 18 March 2026
Accepted : 5 April 2026

Keyword:

Algebra instruction;
Computational thinking;
Problem-solving skills;
Secondary mathematics.

ABSTRACT

This study aims to explore how computational thinking is integrated into algebra instruction to enhance students' problem-solving skills. The study addresses the difficulty students often face in understanding contextual algebra problems, identifying relevant information, constructing symbolic representations, and organizing solution procedures. A qualitative case study design was used to examine the learning process in a junior high school mathematics classroom in Indonesia. Participants consisted of eighth-grade students selected through purposive sampling based on their participation in the learning process, written work, ability to explain their reasoning, and variation in mathematical ability. Data were collected through semi-structured interviews, limited classroom observation, documentation, and analysis of students' written work. The data were analyzed using an interactive qualitative model involving data condensation, data display, and conclusion verification. The findings revealed five main themes: problem decomposition, pattern recognition, abstraction, algorithmic reasoning, and solution evaluation. Students became more able to break down algebra problems, identify repeated structures, translate verbal information into mathematical symbols, arrange solution steps logically, and check the meaning of their answers. Teacher guidance, structured tasks, and classroom discussion played important roles in supporting these processes. The study concludes that computational thinking can strengthen algebra instruction by helping students focus on the logic of problem solving, not only on final answers. The findings offer practical implications for mathematics teachers and theoretical contributions to the integration of computational thinking, algebraic thinking, and constructivist learning.

***) Corresponding Author:**

Lestari Ningsih,
Department of Mathematics Education, Faculty of Teacher Training and Education, Universitas Pendidikan Indonesia, Bandung, Indonesia
Email: lestari.ningsih053@gmail.com

1. Introduction

The rapid development of digital society has changed the orientation of mathematics education. Mathematics learning can no longer focus only on procedural fluency, formula memorization, and final answers. Students need learning experiences that train them to think logically, systematically, and reflectively when solving complex problems. One important competence in this context is computational thinking. Computational thinking refers to the ability to decompose problems, recognize patterns, construct abstractions, design algorithms, and evaluate solutions.

In mathematics education, this competence does not only relate to computer programming. It also supports the way students understand the structure of a problem, identify relevant information, and organize problem-solving steps. Li et al. (2020) argue that computational thinking is more about thinking than computing, which means that it can strengthen mathematical reasoning even in classrooms with limited technology. Ye et al. (2023) also explain that integrating computational thinking into K-12 mathematics instruction can connect learning activities, conceptual understanding, and students' problem-solving strategies.

Algebra is one of the most relevant areas for integrating computational thinking because it requires students to identify patterns, represent relationships, use symbols, generalize structures, and construct systematic procedures. Algebraic problem solving often demands more than the ability to manipulate symbols. Students must interpret the situation, translate verbal information into mathematical expressions, and justify each step of their solution. Bråting and Kilhamn (2021) show that algebraic thinking and computational thinking intersect in students' use of symbols, representations, rules, and structured reasoning. Kallia et al. (2021) also identify abstraction, decomposition, generalization, algorithmic thinking, and evaluation as important dimensions of computational thinking in mathematics education. These dimensions are closely related to the cognitive demands of algebra instruction. Therefore, integrating computational thinking into algebra learning can help students move from mechanical calculation toward deeper problem analysis and more organized solution strategies.

In classroom practice, many students still face difficulties when solving algebra problems, especially contextual and multi-step problems. These difficulties often appear when students fail to identify relevant information, break down a problem into smaller parts, recognize relationships between quantities, or explain the logic behind their solution. Abidi et al. (2023) found that students' computational thinking abilities in solving contextual mathematics problems were uneven across decomposition, pattern recognition, abstraction, and algorithmic thinking. Nurwita et al. (2022) also found that students experienced obstacles in abstraction and strategy construction when solving mathematics problems that required computational thinking. In the Indonesian middle school context, Yana and Wijaya (2025) reported that problem-based learning could support students' computational thinking in solving two-variable linear equation problems, but students with lower mathematical ability still needed structured guidance and clearer teacher prompts. These findings indicate that computational thinking should not be treated as an additional skill outside algebra learning. It needs to be embedded in the learning process through tasks, questions, discussion, and reflection.

This issue is important because algebraic problem-solving skills are closely related to mathematical literacy, digital literacy, and students' readiness to face complex real-life situations. Students who can decompose problems, identify patterns, and construct logical procedures tend to have stronger foundations for reasoning, decision-making, and learning in science, technology, and other disciplines. Nordby et al. (2022) explain that computational thinking in mathematics classrooms can appear through sequencing, decomposition, abstraction, communication, exploration, and student engagement. Falloon (2024) also shows that structured curriculum design can help students build sequencing skills, pattern recognition, debugging ability, and solution construction. In this sense, computational thinking offers a meaningful direction for algebra instruction because it encourages students to focus on how a solution is developed, not only on whether an answer is correct. This focus is consistent with constructivist learning theory, which views students as active meaning-makers who build understanding through experience, interaction, and reflection.

Although research on computational thinking in mathematics education continues to grow, several gaps remain. Many previous studies have focused on program effectiveness, technology-based interventions, curriculum mapping, or quantitative measurement of students' skills. Lv et al. (2023) state that empirical research on the integration of mathematics and computational thinking still needs a deeper explanation of how classroom activities shape students' mathematical understanding. Ersozlu et al. (2023) also show that many studies still emphasize tools, approaches, and literature mapping, while students' lived learning experiences remain less explored. Chytas et al. (2024) add that integrating computational thinking into mathematics instruction requires careful design, classroom observation, and refinement based on students' responses. These studies show the need for qualitative research examining the meanings, experiences, and learning processes that arise when computational thinking is integrated into algebra instruction.

Based on this background, this study aims to explore how computational thinking is integrated into algebra instruction to enhance students' problem-solving skills. The study focuses on how students understand algebra problems, decompose information, recognize patterns, construct abstractions, design solution procedures, and evaluate their answers. It also examines how teacher strategies, task design, classroom interaction, and guiding questions support or hinder students' computational thinking during algebra learning. Theoretically, this study contributes to mathematics education research by connecting constructivist learning theory, algebraic thinking, computational thinking, and problem-solving processes. Practically, the findings can help mathematics teachers design algebra instruction that is more meaningful, systematic, and oriented toward students' thinking processes.

2. Materials and Methods

This study used a qualitative approach with a case study design. The design was selected because the study aimed to explore in depth the process of integrating computational thinking into algebra instruction in a real classroom context. A case study allows the researcher to examine students' experiences, teacher strategies, classroom interaction, learning tasks, and problem-solving processes as connected elements within one learning setting. This approach was suitable because the study did not aim only to measure learning outcomes. It aimed to understand how students used decomposition, pattern recognition, abstraction, algorithmic thinking, and solution evaluation when solving algebra problems. Nordby et al. (2024) emphasize that computational thinking in mathematics classrooms should be understood through the ecology of learning, including tools, tasks, students, teachers, and mathematical goals.

The study was conducted in a junior high school in Indonesia that had implemented problem-solving-oriented mathematics instruction. The school was selected because its mathematics learning context allowed the use of algebra tasks that could support computational thinking activities. The research was carried out over one semester, covering preparation, preliminary observation, learning implementation, interviews, document collection, and data analysis. The research subjects were eighth-grade students who were learning algebraic expressions and systems of linear equations in two variables. This level was chosen because algebra at this stage requires students to identify patterns, translate verbal information into symbols, and develop systematic solution procedures. Lehmann (2025) explains that decomposition, abstraction, and algorithmic thinking interact strongly with mathematical problem-solving heuristics, making algebra a relevant context for observing computational thinking.

Participants were selected using purposive sampling. The criteria included students who participated in the complete learning sequence, produced written work that could be analyzed, were able to explain their thinking orally, and represented high, medium, and low levels of mathematical ability. The mathematics teacher was also included as a supporting informant because the teacher played an important role in designing tasks, giving instructions, asking guiding questions, and responding to students' strategies. Snowball sampling could be used in a limited way when certain students provided richer information or were recommended by the teacher as students who could explain their learning experiences in greater detail. This participant selection strategy was used to capture varied student experiences in applying computational thinking during algebra problem solving.

Data were collected through semi-structured interviews, limited participatory observation, documentation, and analysis of students' written work. Semi-structured interviews were used to explore students' experiences when understanding problems, identifying relevant information, recognizing patterns, designing solution steps, and evaluating answers. Limited participatory observation was conducted during classroom learning to record teacher-student interaction, student responses to algebra tasks, group discussion patterns, and students' explanations of solution strategies. Documentation included lesson modules, student worksheets, teacher notes, classroom activity photos, and learning recordings when permission was granted. Students' written work was analyzed to identify indicators of computational thinking. Salwadila and Hapizah (2024) used written tests and interviews to examine students' computational thinking in mathematics learning, which supports the relevance of combining written and oral data in this study.

The main research instrument was the researcher, supported by an interview guide, observation sheet, documentation guide, and computational thinking analysis rubric. The interview guide was designed to explore students' reasons for choosing certain strategies, the difficulties they experienced, and the ways they revised their answers. The observation sheet was used to record learning behaviors that reflected decomposition, pattern recognition, abstraction, algorithmic thinking, and evaluation. The analysis rubric was used to interpret students' thinking processes based on written and oral data. Decomposition was identified when students broke a problem into smaller and more manageable parts. Pattern recognition was identified when students found relationships among numbers, variables, or problem structures. Abstraction was identified when students selected important information and represented it symbolically. Algorithmic thinking was identified when students arranged solution steps logically. Evaluation was identified when students checked, justified, or corrected their answers.

Data trustworthiness was maintained through source triangulation, method triangulation, member checking, and audit trail. Source triangulation was conducted by comparing data from students, the teacher, students' written work, and learning documents. Method triangulation was conducted by comparing interview data, observation notes, documentation, and learning artifacts. Member checking was carried out by confirming interview summaries and initial interpretations with participants to ensure that the researcher's interpretation reflected their actual experiences. An audit trail was maintained by documenting field notes, interview transcripts, coding results, category changes, and analytical decisions. These procedures were important because qualitative research requires transparency, interpretive accuracy, and clear data traceability.

Data were analyzed using the interactive model of Miles, Huberman, and Saldaña, which consists of data condensation, data display, and conclusion drawing or verification. In the data condensation stage, the researcher selected interview transcripts, observation notes, and students' written work that were relevant to computational thinking indicators and algebraic problem-solving skills. In the data display stage, the researcher developed thematic matrices based on decomposition, pattern recognition, abstraction, algorithmic thinking, solution evaluation, teacher strategies, and student learning obstacles. In the verification stage, patterns across participants and data sources were compared to generate main themes. The analysis involved initial coding, category grouping, theme interpretation, and narrative construction. The final analysis explained how computational thinking integration supported students' algebraic problem-solving skills and identified classroom factors that strengthened or limited the process.

3. Results and Discussions

The findings show that the integration of computational thinking into algebra instruction shaped students' problem-solving processes in five main themes: problem decomposition, pattern recognition, abstraction, algorithmic reasoning, and solution evaluation. These themes emerged from classroom observation, semi-structured interviews, students' written work, and learning documentation. The data indicate that students did not immediately solve algebra problems by applying formulas. Instead, they gradually learned to identify the structure of the problem, separate relevant information, recognize relationships among quantities, and organize solution steps. This process became more visible when the teacher used guiding questions, group discussion, and problem-based algebra tasks.

The first theme concerns students' ability to decompose algebra problems. At the beginning of the learning process, several students tended to read the problem as a whole and directly searched for a formula. This strategy often led to confusion when the problem contained several pieces of information. After the teacher introduced decomposition activities, students began to divide the problem into smaller parts, such as identifying known values, unknown variables, relationships between quantities, and the question being asked. One student explained, "I understood the problem better after I separated what was known, what was asked, and what relationship should be written." This finding shows that decomposition helped students reduce cognitive load and gave them a clearer entry point for solving algebraic problems.

The second theme relates to pattern recognition. Students began to recognize repeated structures in algebra problems after comparing several examples. In group discussions, students identified that different word problems could share the same algebraic structure even when the context was different. For example, problems about prices, ages, and numbers of objects could all be represented using linear equations. One participant stated, "The story is different, but the pattern is similar because there are two unknown values and two relationships." This statement indicates that pattern recognition supported students in moving from surface-level understanding to structural understanding. Observation data also showed that students who recognized patterns more quickly were able to select solution strategies more confidently.

The third theme is abstraction. Students' abstraction skills appeared when they translated verbal information into symbols, variables, and algebraic expressions. At first, some students wrote all information from the problem without distinguishing between relevant and irrelevant details. Through teacher prompts, students learned to select key information and represent it using variables. For instance, one student said, "I used x and y because the problem had two things that were not known yet." This ability marked an important shift from concrete story-based interpretation to symbolic mathematical representation. However, the data also showed that abstraction was one of the most challenging stages. Students with lower mathematical ability often needed examples, visual support, or peer explanation before they could form correct algebraic expressions.

The fourth theme concerns algorithmic reasoning. After students identified information, patterns, and symbolic representations, they began to arrange solution steps in a more systematic order. Students' written work showed clearer sequences, such as defining variables, forming equations, choosing an elimination or substitution method, calculating values, and checking answers. One student explained, "I wrote the steps first so I would not jump directly to the answer." This shows that algorithmic reasoning encouraged students to think about procedure as a logical sequence rather than a set of disconnected calculations. Classroom observation also revealed that students who worked in groups often corrected each other's steps when the order of solution was unclear.

The fifth theme is solution evaluation. Before the integration of computational thinking, many students stopped once they obtained a numerical answer. After the learning intervention, several students began to check whether their answers matched the original problem. They substituted the values back into the equations, reviewed the meaning of the variables, and compared answers with group members. One participant stated, "I checked again because the answer could be right in calculation but wrong in meaning." This finding suggests that evaluation helped students develop more reflective problem-solving habits. It also showed that computational thinking supported not only the process of finding answers but also the process of validating and justifying them.

Teacher support played a central role in the emergence of these five themes. The teacher did not simply explain algebraic procedures. The teacher asked guiding questions such as "What information is important?", "What pattern do you see?", "How can this sentence be written as an equation?", and "How do you know your answer is correct?" These questions encouraged students to explain their reasoning and reflect on their strategies. The classroom environment also supported collaborative meaning-making. Students learned from peer explanations, especially when they compared different solution strategies. This finding indicates that computational thinking developed more effectively when it was embedded in social interaction, guided reflection, and structured algebra tasks.

Discussion

The findings indicate that computational thinking can strengthen algebra instruction by helping students organize their problem-solving processes. This result supports Li et al. (2020), who argue that computational thinking is more about thinking than computing. In this study, students used computational thinking not as a programming skill, but as a reasoning strategy for understanding algebraic problems. Decomposition helped students clarify the structure of problems. Pattern recognition helped them identify similarities across problem types. Abstraction helped them translate real situations into algebraic forms. Algorithmic reasoning helped them arrange solution steps. Evaluation helped them check the accuracy and meaning of their answers. These processes show that computational thinking can function as a bridge between algebraic reasoning and mathematical problem solving.

The role of decomposition in this study is consistent with previous research by Kallia et al. (2021), who identify decomposition as a central element of computational thinking in mathematics education. The students' ability to divide algebra problems into smaller components helped them manage complex information. This finding also supports Ye et al. (2023), who state that computational thinking can support students' mathematical learning when it is connected to meaningful classroom activities. However, this study adds a qualitative perspective by showing how decomposition emerged through teacher prompts and student discussion. Students did not develop this skill automatically. They needed repeated opportunities to identify known information, unknown variables, relationships, and problem goals.

Pattern recognition also became an important finding. Students who recognized similar structures across different algebra problems showed stronger confidence in choosing solution strategies. This finding aligns with Bråting and Kilhamn (2021), who explain that algebraic thinking and computational thinking intersect through symbols, rules, and structured representations. The present study extends this argument by showing that pattern recognition helped students move beyond the surface context of word problems. They learned to see that different stories could share the same mathematical structure. This is important in algebra instruction because many students struggle when they treat every word problem as completely new.

Abstraction was the most challenging process for many students. This finding supports Nurwita et al. (2022), who found that students often experience difficulties in abstraction when solving mathematics problems that require computational thinking. In this study, abstraction required students to decide which information mattered, assign variables, and construct algebraic expressions. Students with lower mathematical ability needed more scaffolding before they could represent verbal information symbolically. This suggests that teachers should not assume that students can move easily from context to symbols. Algebra instruction needs gradual support, such as visual representations, guided questions, examples, and peer discussion.

Algorithmic reasoning in this study appeared when students arranged their solutions in clear and logical steps. This finding is in line with Lehmann (2025), who explains that computational thinking skills interact with mathematical problem-solving heuristics. Students who planned their steps before calculating showed better control over the solution process. This result also supports

Kaufmann and Stenseth (2021), who argue that procedural and programming-related thinking can reveal how students construct mathematical arguments. In the present study, algorithmic reasoning did not reduce mathematics to mechanical procedures. Instead, it helped students explain why each step was needed and how one step connected to the next.

The finding on solution evaluation shows that computational thinking can encourage reflective learning. Students began to check their answers, examine the meaning of variables, and compare their results with the original problem. This finding supports Nordby et al. (2022), who show that computational thinking in mathematics classrooms includes not only technical skills but also communication, exploration, and engagement. Evaluation is important because students often consider a problem finished once they obtain a number. Through computational thinking, students learned that a correct answer must also make sense in the problem context. This process strengthens mathematical reasoning and reduces careless procedural errors.

The study also shows that teacher guidance is crucial in integrating computational thinking into algebra instruction. Students developed stronger problem-solving strategies when the teacher used guiding questions and encouraged discussion. This supports Chytas et al. (2024), who state that computational thinking integration in mathematics requires careful instructional design and classroom refinement. The findings also align with constructivist learning theory because students built understanding through active participation, interaction, and reflection. Computational thinking became meaningful because students used it to make sense of algebra problems, not because the teacher introduced it as a separate concept.

Compared with previous studies that focus on technology-based interventions, this study offers a different perspective. The findings show that computational thinking can be integrated into algebra instruction without relying fully on digital tools. This supports Li et al. (2020), who emphasize that computational thinking is not limited to computing devices. In classrooms with limited technological resources, teachers can still develop computational thinking through problem decomposition, pattern comparison, symbolic representation, step-by-step reasoning, and solution checking. This finding is relevant for Indonesian schools because access to digital infrastructure varies across regions.

Theoretically, this study contributes to the discussion on the relationship between computational thinking, algebraic thinking, and problem-solving skills. It shows that computational thinking can support algebra learning when its indicators are embedded in classroom tasks and teacher questioning. Practically, the findings suggest that mathematics teachers should design algebra tasks that require students to explain their thinking process, not only produce final answers. Teachers can use structured prompts to guide students through decomposition, pattern recognition, abstraction, algorithmic reasoning, and evaluation. Future studies can expand this research by comparing different school contexts, involving more participants, or using classroom-based intervention designs to examine how computational thinking develops over a longer period.

4. Conclusions

This study concludes that integrating computational thinking into algebra instruction helps students develop more structured and reflective problem-solving skills. The findings show that students improved their ability to decompose algebra problems, recognize mathematical patterns, construct abstractions, arrange logical solution procedures, and evaluate the meaning of their answers. These processes indicate that computational thinking supports algebra learning not merely as a technical skill, but as a reasoning framework that helps students understand how problems are structured and how solutions can be developed systematically. This study contributes to the literature by showing that computational thinking can strengthen the connection between algebraic thinking, mathematical reasoning, and problem-solving processes in a qualitative classroom context.

The findings have theoretical, practical, and policy implications. Theoretically, this study reinforces the relevance of constructivist learning theory in explaining how students build mathematical understanding through interaction, reflection, and guided problem solving. Practically, mathematics teachers can use computational thinking indicators as instructional prompts when designing algebra tasks, classroom discussions, and student reflection activities. At the policy level, the findings suggest that computational thinking should be integrated into mathematics learning as part of curriculum development, teacher training, and digital literacy strengthening. Future research can expand this study by involving different grade levels, broader school contexts, or longitudinal designs to examine how students' computational thinking develops over time.

References

Abidi, M. H., Cahyono, H., & Susanti, R. D. (2023). Analysis of students' computational thinking ability in solving contextual problems. *Mathematics Education Journal, 7*(2). https://doi.org/10.22219/mej.v7i2.25041

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

Chytas, C., van Borkulo, S. P., Drijvers, P., Barendsen, E., & Tolboom, J. L. J. (2024). Computational thinking in secondary mathematics education with GeoGebra: Insights from an intervention in calculus lessons. *Digital Experiences in Mathematics Education, 10*(2), 228–259. https://doi.org/10.1007/s40751-024-00141-0

Ersozlu, Z., Wildy, H., & Guldberg, K. (2023). Developing computational thinking through mathematics: An evaluative scientific mapping. *Education Sciences, 13*(4), 422. https://doi.org/10.3390/educsci13040422

Falloon, G. (2024). Advancing young students' computational thinking: An investigation of structured curriculum in early years primary schooling. *Computers & Education, 212*, 105045. https://doi.org/10.1016/j.compedu.2024.105045

Fang, X., Ng, D. T. K., Tam, W. T., & Yuen, M. (2025). Design mobile computational thinking-integrated mathematics lessons based on the 5E instructional model for primary students. *Educational Technology Research and Development, 73*, 2809–2838. https://doi.org/10.1007/s11423-025-10513-x

Fauzi, A. L., Kusumah, Y. S., & Juandi, D. (2024). Computational thinking in mathematics education: A systematic literature review on its implementation and impact on students' learning. *Jurnal Kependidikan: Jurnal Hasil Penelitian dan Kajian Kepustakaan di Bidang Pendidikan, Pengajaran dan Pembelajaran, 10*(2), 640–653. https://doi.org/10.33394/jk.v10i2.11140

Isharyadi, R., & Juandi, D. (2023). A systematics literature review of computational thinking in mathematics education: Benefits and challenges. *Formatif: Jurnal Ilmiah Pendidikan MIPA, 13*(1). https://doi.org/10.30998/formatif.v13i1.15922

Junpho, P., Nilsook, P., & Wannapiroon, P. (2022). Computational thinking in mathematics education: A systematic literature review. *Education and Information Technologies, 27*(5), 6789–6811. https://doi.org/10.1007/s10639-021-10758-9

Kallia, M., van Borkulo, S. P., Drijvers, P., Barendsen, E., & Tolboom, J. L. J. (2021). Characterising computational thinking in mathematics education: A literature-informed Delphi study. *Research in Mathematics Education, 23*(2), 159–187. https://doi.org/10.1080/14794802.2020.1852104

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

Lee, I., & Malyn-Smith, J. (2020). Computational thinking integration patterns along the framework defining computational thinking from a disciplinary perspective. *Journal of Science Education and Technology, 29*, 9–18. https://doi.org/10.1007/s10956-019-09802-x

Lehmann, T. H. (2025). Examining the interaction of computational thinking skills and heuristics in mathematical problem solving. *Research in Mathematics Education, 27*(2), 269–290. https://doi.org/10.1080/14794802.2025.2460460

Li, Y., Schoenfeld, A. H., diSessa, A. A., Graesser, A. C., Benson, L. C., English, L. D., & Duschl, R. A. (2020). Computational thinking is more about thinking than computing. *Journal for STEM Education Research, 3*(1), 1–18. https://doi.org/10.1007/s41979-020-00030-2

Listiaji, P., & Molnár, G. (2025). Integrating computational thinking into teacher education: A systematic literature review. *International Journal of Educational Research, 133*, 102682. https://doi.org/10.1016/j.ijer.2025.102682

Lv, L., Zhong, B., & Liu, X. (2023). A literature review on the empirical studies of the integration of mathematics and computational thinking. *Education and Information Technologies, 28*(7), 8171–8193. https://doi.org/10.1007/s10639-022-11518-2

Mulyono, B., Hapizah, H., & Sukanda, D. C. (2024). Computational thinking skills in mathematics: A study of social arithmetic. *Journal of Honai Math, 7*(3), 401–420. https://doi.org/10.30862/jhm.v7i3.759

Nordby, S. K., Bjerke, A. H., & Mifsud, L. (2022). Computational thinking in the primary mathematics classroom: A systematic review. *Digital Experiences in Mathematics Education, 8*(1), 27–49. https://doi.org/10.1007/s40751-022-00102-5

Nordby, S. K., Mifsud, L., & Bjerke, A. H. (2024). Computational thinking in primary mathematics classroom activities. *Frontiers in Education, 9*, 1414081. https://doi.org/10.3389/feduc.2024.1414081

Nurwita, F., Kusumah, Y. S., & Priatna, N. (2022). Exploring students' mathematical computational thinking ability in solving Pythagorean theorem problems. *Al-Jabar: Jurnal Pendidikan Matematika, 13*(2), 273–287. https://doi.org/10.24042/ajpm.v13i2.12496

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

Su, J., & Yang, W. (2023). A systematic review of integrating computational thinking in early childhood education. *Computers and Education Open, 4*, 100122. https://doi.org/10.1016/j.caeo.2023.100122

Sukowati, B. A., & Masduki, M. (2024). Exploration of students' computational thinking skills in solving fractional number problems judging from learning style. *Desimal: Jurnal Matematika, 7*(2), 163–176. https://doi.org/10.24042/djm.v7i2.21878

Yana, F., & Wijaya, A. (2025). Enhancing computational thinking skills in solving two-variable linear equations through problem-based learning in Indonesian middle schools. *AL-ISHLAH: Jurnal Pendidikan, 17*(1), 614–631. https://doi.org/10.35445/alishlah.v17i1.5869

Global Journal of Mathematics
Vol. 1, No. 1, 2026, pp. 19-28

Ye, H., Liang, B., Ng, O. L., & 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*, Article 3. https://doi.org/10.1186/s40594-023-00396-w