



CODING WORKSHEETS (LKPD) IN IPAS LEARNING TO FOSTER COMPUTATIONAL THINKING AMONG MADRASAH IBTIDAIYAH STUDENTS

Rahmatun Nisa¹, *Aulia Rohmawati², Muhamad Ilman Nafi'an³, Maziyyatul Muslimah⁴

^{1,2,3} Faculty of Education and Teacher Training, UIN Syekh Wasil Kediri, Indonesia

⁴ School of Education, The University of Queensland
aulia@uinkediri.ac.id

Received: July, 2026, Accepted: September, 2026, Published: September, 2026

ABSTRACT

Computational thinking is an important skill for elementary students, enabling them to solve problems logically and systematically. This study aimed to examine changes in students' computational thinking skills following the use of Coding Worksheets (LKPD) in IPAS learning at Madrasah Ibtidaiyah. A quantitative approach with a one-group pretest–posttest design was employed, involving 30 fifth-grade students at MIN 5 Nganjuk. The worksheets introduced basic coding concepts through structured activities without computer-based programming, focusing on decomposition, pattern recognition, abstraction, and algorithmic thinking. Data were collected using computational thinking tests before and after the intervention. The data were analyzed using descriptive statistics, a paired-samples t-test, and normalized gain (N-Gain) analysis. The mean score increased from 61.67 on the pretest to 84.33 on the posttest. The paired-samples t-test showed a statistically significant difference between the scores ($t(29) = -12.030, p < .001$). The mean N-Gain was 0.5967, categorized as moderate. These findings indicate an improvement in students' computational thinking scores following the implementation of Coding Worksheets in IPAS learning. The findings suggest that Coding Worksheets have potential to support computational thinking development among Madrasah Ibtidaiyah students.

Keywords: Computational Thinking, Coding Worksheets, IPAS, Madrasah Ibtidaiyah, Unplugged Coding.

1. Introduction

Primary education does not merely equip students with factual knowledge. Schools also help students understand problems, sort through information, and arrive at solutions through coherent reasoning. Technological development, globalization, and social change have made higher-order thinking skills, such as problem-solving, critical thinking, communication, collaboration, and adaptability, increasingly necessary (Astuti et al. 2025). It is in this context that computational thinking is worth developing from the primary level, because it offers students a systematic way to approach problems that seem complex.

The importance of computational thinking cannot be judged simply by how often children come into contact with technology. What matters more is whether they can reason



Copyright:

© 2026 by the author(s).

This open-access article is distributed under the terms and conditions of the Creative Commons Attribution-ShareAlike (CC BY-SA) license (<https://creativecommons.org/licenses/by-sa/4.0/>).

in a structured way when facing a problem, rather than merely using devices to look up information. The case for developing it is therefore better grounded in the need for learning that trains reasoning and systematic problem-solving (Morales-Teherán, Gómez-Jaramillo, and Parada-Trujillo 2026). Digital literacy, reading literacy, mathematics, science, and computational thinking are related, but they are not the same construct. Accordingly, survey results such as PISA 2022 should not be read directly as a portrait of students' computational thinking.

Conceptually, Jeannette Wing views computational thinking as a way of solving problems using fundamental concepts from computer science, one that can be applied beyond programming activities (Bers, Strawhacker, and Sullivan 2022). In other words, computational thinking does not stop at the ability to write code or operate digital devices. It includes formulating problems, choosing suitable representations, identifying relevant information, and devising solution procedures that can be followed logically. Wu et al. (2024) similarly describe it as a cognitive skill for formulating and solving problems through systematic processes. From this perspective, programming is only one means of training computational thinking, whereas computational thinking itself is a reasoning process that can appear in many learning contexts.

Critical thinking and computational thinking overlap because both require problem analysis, logical reasoning, and judgment about the soundness of a solution (Candraningtyas and Khusna 2023). Critical thinking emphasizes examining information, evaluating arguments, and drawing well-founded conclusions. Computational thinking focuses more on breaking problems down, recognizing patterns, filtering information, and arranging solution steps. The two can support each other, for example when students judge the relevance of information or check whether the steps they have arranged are correct. Even so, this relationship does not mean that critical thinking is the only foundation of computational thinking. They share some processes, but their emphases differ.

Computational thinking is also closely related to science learning and other fields that require students to understand relationships among components, interpret information, and solve problems based on particular sequences or patterns. Several studies have discussed its links with critical thinking and STEM learning (Morales-Teherán et al. 2026). In classroom practice, these links can be realized through tasks that ask students to understand a problem before answering it, rather than merely recalling information. Computational thinking should therefore be trained through learning activities that give students room to try out problem-solving strategies for themselves.

This study focuses its measurement on four indicators of computational thinking: decomposition, pattern recognition, abstraction, and algorithmic thinking. Decomposition is the ability to break a complex problem into simpler parts. Pattern recognition is the ability to find similarities, regularities, or relationships that help in understanding a problem. Abstraction means selecting relevant information and setting aside unnecessary details. Algorithmic thinking is the ability to arrange steps or instructions in a coherent sequence. These four are consistent with the principles of computational thinking described by Fuadah, Mukti, and Firmansyah (2025). Keeping the indicators consistent is important so that the learning activities and the test instrument measure the same skills.

These four indicators can be trained through activities that ask students to break problems down, find regularities, select important information, and determine the order of a solution. Some computational thinking frameworks include other processes, such as debugging or generalization, as discussed by Christi and Rajiman (2023). However, this study deliberately limits its focus to the four indicators above. This limitation does not imply

that other processes are less important; it is intended to keep the conceptual foundation, the learning activities, and the measurement of outcomes aligned.

In everyday life, computational thinking helps a person organize the steps of an activity, weigh information, and make more structured decisions, for instance in managing time or solving problems. In primary school, such benefits need to be translated into learning experiences appropriate to students' developmental stage. Introducing the term "computational thinking" alone is not enough. Students need the opportunity to practice its reasoning process through tasks that are clear and can be completed step by step.

This need is consistent with Integrated Science and Social Studies (Ilmu Pengetahuan Alam dan Sosial, IPAS) learning at Madrasah Ibtidaiyah. IPAS invites students to understand natural and social phenomena, the relationships among events, and the information used to explain a situation. With suitable topics, students can practice breaking a problem into simpler components, recognizing relationships or sequences, sorting out important information, and arranging solution steps. In this way, computational thinking grows as part of the process of understanding the material, rather than as a separate skill attached to the lesson. Of course, this can only be realized when the tasks are truly aligned with the IPAS material being studied.

The learning conditions at MIN 5 Nganjuk form the empirical basis of this study. An interview with the Grade V-A1 teacher showed that IPAS learning is still dominated by conventional, teacher-centered methods. The teacher tends to follow the available textbooks and student worksheets (LKS), and most formative and summative assessments use questions taken from those books. Project-based or problem-solving learning is not yet frequently implemented, although the teacher had tried a simple project on the human respiratory system. This condition indicates room to expand learning experiences that train students to solve problems in a structured way. These interview findings describe a single class and a single site, so they cannot be directly generalized to all Madrasah Ibtidaiyah.

This picture is reinforced by a preliminary assessment of 30 Grade V students at MIN 5 Nganjuk. The test instrument was based on the four indicators of computational thinking: decomposition, pattern recognition, abstraction, and algorithmic thinking. As a result, only 7 students (23.3%) met the Learning Objective Achievement Criteria (Kriteria Ketercapaian Tujuan Pembelajaran, KKTP), with the greatest difficulties in pattern recognition and algorithmic thinking. This preliminary finding suggests that most students still need learning experiences that help them recognize problem patterns and arrange solution steps logically. This figure describes the initial condition of the research group, not the condition of all Madrasah Ibtidaiyah students.

One way to respond to this need is to use Student Worksheets (Lembar Kerja Peserta Didik, LKPD) designed to guide students in solving problems. According to Prastowo (as cited in Syamsiah et al. 2024), an LKPD can contain materials, summaries, instructions, and tasks that help students learn in a guided way. In this study, the value of the LKPD lies not in its form as a printed material but in its ability to organize learning activities into tasks aligned with the indicators of computational thinking. An LKPD that is consistent with the learning objectives can direct students' attention to the solution process, not just the final answer. However, this benefit depends on how the tasks are designed and carried out, so it cannot be assumed merely because an LKPD is used.

The development of LKPDs has been examined through various approaches. Masjudin et al. (2022), for example, discuss a STEM-based LKPD that encourages students to connect several aspects when understanding a problem. This example shows that LKPDs can be designed to support learning that links concepts with reasoning processes. Even so, studies

of STEM-based LKPDs do not in themselves show that an unplugged coding LKPD will change computational thinking skills. The relationship among LKPD design, material, student activities, and the skills measured needs to be explained based on the intervention actually implemented.

Another approach appears in the Search, Solve, Create, and Share (SSCS) model. Hapsari and Masduki (2024) applied it in an e-worksheet with Islamic elements, using the stages of searching, solving, creating, and sharing results. This model illustrates how learning activities can be organized, but an e-worksheet and a printed LKPD based on unplugged coding differ in the character of their media and activities. For this reason, SSCS is used here to show the range of LKPD designs, not to equate its learning mechanism with the intervention in this study. This distinction keeps the choice of the Coding LKPD grounded in students' needs and the activities actually used.

Miftah, Kurniawati, and Musa (2024) developed an LKPD based on Interactive Case-Based Learning (ICBL) using the Cublend App for prospective mathematics teachers. This study shows that LKPDs can be built around cases and can make use of digital media. However, its subjects, field of study, and use of technology differ from the Grade V IPAS learning that is the focus here. The study therefore provides context on the variety of LKPD development, but it does not yet address how coding activities without digital devices can train the four indicators of computational thinking among Madrasah Ibtidaiyah students.

Overall, studies of STEM-based, SSCS, and ICBL LKPDs show many ways of organizing learning. However, this diversity needs to be distinguished from evidence that specifically supports a Coding LKPD based on unplugged coding. For this study, the central question is not which LKPD approach is superior, but how to design and examine a worksheet that fits the needs of IPAS learning at the research site. Classroom context, activity type, material, and computational thinking indicators are therefore treated as interrelated parts of a single whole.

Coding can be one way to train computational thinking, because students work directly with instructions and problem-solving strategies. However, coding is not the same as computational thinking, and coding activities do not automatically develop all indicators unless they are designed to match the learning objectives. Syamsiah et al. (2024) describe coding as the activity of arranging instructions, which can be done digitally or non-digitally. One form is unplugged coding, an activity without computers that can be carried out through structured activities or worksheets (Dwijayani 2022). This approach is worth examining because it allows students to practice reasoning through structured activities without relying on digital devices.

In this study, the Coding LKPD is integrated into IPAS learning using an unplugged coding approach. The activities refer to the four indicators of computational thinking and are applied to the human respiratory and digestive systems. In decomposition, students break a problem into simpler parts, for example by identifying the components related to a body system. In pattern recognition, they identify patterns or relationships, such as the sequence in which air or food passes through the body's organs. In abstraction, they select the information needed to solve a case-based problem. Meanwhile, algorithmic thinking is trained by arranging a sequence of steps based on the available information or instructions. These examples show the conceptual fit between the LKPD activities and the computational thinking indicators.

Previous research provides a basis for examining coding as an approach to developing computational thinking. Munir, Muhith, and Arifin (2025) reported that Scratch-based learning among Grade V students in Indonesia improved coding and computational thinking

skills, with an effect size of Cohen's $d = 1.85$ in the experimental group compared with the control group. A review by Listiowati et al. (2025) of 19 articles published between 2020 and 2025 concluded that coding instruction, both plugged and unplugged, is associated with the development of logical reasoning, pattern recognition, and problem-solving among elementary school students. Both sources support the relevance of coding as a learning approach, but differences in media, subject matter, and research design should be kept in mind when their results are compared with this study.

Based on this review, this study seeks to provide contextual evidence on changes in computational thinking skills after Grade V students at MIN 5 Nganjuk learned IPAS using a Coding LKPD based on unplugged coding. Its distinctiveness lies in the combination of a printed LKPD, coding activities without digital devices, IPAS material on the human respiratory and digestive systems, and measurement focused on decomposition, pattern recognition, abstraction, and algorithmic thinking. This study does not claim that such an approach has never been used in other research, but limits its contribution to the context and design examined. Because the design is a one-group pretest–posttest without a control group, changes in scores can be analyzed, but it cannot be concluded that the Coding LKPD is the sole cause of those changes.

Therefore, this study aims to analyze the differences in computational thinking skills of Grade V students at MIN 5 Nganjuk before and after IPAS learning using a Coding LKPD based on unplugged coding. This objective is consistent with the one-group pretest–posttest design and is not intended to measure perceptions of the importance of computational thinking or to compare the Coding LKPD with other learning approaches.

2. Method

This study employed a quantitative approach using a one-group pretest–posttest design. This design involved a single group of students who were assessed before and after the intervention, without a control group. The comparison between pretest and posttest scores was used to analyze changes in computational thinking skills following instruction using Coding Worksheets. However, because the study did not include a control group, the observed changes cannot be attributed solely to the use of Coding Worksheets. Other factors, such as learning experiences during the study, teacher guidance, and time-related factors, may also have contributed to the results.

The study was conducted at MIN 5 Nganjuk and involved 30 fifth-grade students in Phase C, consisting of 13 male students and 17 female students. The participants were drawn from one class.

The intervention was conducted over three sessions in the IPAS subject using Coding Worksheets. The instructional content covered the human respiratory and digestive systems. The procedure consisted of administering a pretest, conducting IPAS instruction using Coding Worksheets, and administering a posttest. The pretest assessed students' initial computational thinking skills, while the posttest assessed their scores following the instructional intervention.

The research instrument consisted of a computational thinking test administered before and after the intervention. Each test comprised 20 items and was administered to the 30 participants. The assessment focused on four computational thinking indicators: decomposition, pattern recognition, abstraction, and algorithmic thinking.

The research data were analyzed using descriptive statistics, a normality test, a paired-sample t-test, and N-Gain analysis. Descriptive statistics were used to summarize the pretest and posttest scores, including the mean and relevant measures of dispersion. The normality test was used to examine the distributional assumptions required for the planned parametric

analysis. For the paired-sample *t*-test, the relevant normality assumption concerns the distribution of the difference scores between the posttest and pretest, rather than merely the separate distributions of the pretest and posttest scores. The mean difference, 95% confidence interval, degrees of freedom, and *p*-value were reported. N-Gain analysis was used to describe the normalized change in scores.

3. Results and Discussions

The study was conducted in one fifth-grade class at MIN 5 Nganjuk over three sessions in May 2026. The first session involved a pretest to assess students' initial computational thinking skills, the second involved instruction using Coding Worksheets for Students (LKPD) in IPAS learning, and the third involved a posttest to assess students' performance after the intervention. The results are presented through descriptive statistics, a paired-samples *t*-test, and normalized gain (N-Gain) analysis.

3.1. Pretest and Posttest Results

The pretest and posttest scores of the 30 fifth-grade students were analyzed to examine changes in overall computational thinking performance. The following results present the normality assessment, paired-samples *t*-test, and N-Gain analysis.

Table 1 shows the Shapiro–Wilk results for the pretest and posttest scores. The pretest produced a test statistic of 0.970 and a significance value of 0.526, based on 30 scores. The posttest produced a statistic of 0.953 and a significance value of 0.209, also based on 30 scores. Both significance values exceed 0.05. Thus, the Shapiro–Wilk tests did not identify a statistically significant departure from normality in either set of scores. Table 1 reports these tests separately for the pretest and posttest; it does not report a normality test for the difference between each student's two scores.

Table 1 Shapiro Wilk Normality Test Results

	Shapiro-Wilk		
	Statistic	df	Sig.
<i>Pretest</i>	.970	30	.526
<i>Posttest</i>	.953	30	.209

Table 2 presents the results of the paired-sample *t*-test, which compares the two measurements from the same students. In this table, the paired difference was calculated as the pretest score minus the posttest score. Its mean was -23.000 , with a standard deviation of 10.472 and a standard error of 1.912. The negative mean indicates that, on average, posttest scores were higher than pretest scores. The standard deviation describes the variation in these paired differences; it is not the standard deviation of the pretest or posttest scores considered separately.

The test yielded $t(29) = -12.030$ and a two-tailed significance value reported in the table as 0.000. Because a reported value of 0.000 does not mean that the probability is exactly zero, the result is expressed as $p < 0.001$. The 95% confidence interval for the mean difference was -26.910 to -19.090 . Both limits are negative, consistent with the direction of the mean difference. These results indicate a statistically significant difference between the group's pretest and posttest scores.

Table 2 Paired Sample t-test Results

	Paired Differences					t	df	Sig. (2-tailed)
	Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference				
				Lower	Upper			
Pair 1 <i>Pretest-posttest</i>	-23.000	10.472	1.912	-26.910	-19.090	-12.030	29	.000

The N-Gain analysis in Table 3 describes the increase in scores relative to the possible gain. Results were available for all 30 students. The lowest reported N-Gain score was 0.25, or 25.00%, while the highest was 1.00, or 100.00%. The mean N-Gain was 0.5967, reported in percentage form as 59.6738%. Its standard deviation was 0.18599, equivalent to 18.59947 percentage points in the table's percentage presentation. The minimum, maximum, and standard deviation indicate that the measured gains varied among students. Under the N-Gain classification used in this study, the mean value of 0.5967 belongs to the moderate category.

Table 3 N-Gain Results of Pretest and Posttest Data

	N	Minimum	Maximum	Mean	Std. Deviation
NGain_Skor	30	.25	1.000	.5967	.18599
NGain_Persen	30	25.00	100.00	59.6738	18.59947
Valid N (listwise)	30				

In summary, the paired-samples t-test showed a statistically significant difference between pretest and posttest scores, with posttest scores being higher on average. The mean N-Gain was 0.5967, which falls within the moderate category according to the classification used in this study. These findings describe changes in overall computational thinking scores among the 30 students. The analyses reported here do not provide separate results for decomposition, pattern recognition, abstraction, or algorithmic thinking.

3.2. Discussion

Computational thinking is one of the essential skills that students need to acquire, as it relates not only to the use of technology but also to how a person understands and solves problems in a logical, systematic, and structured manner. Computational thinking can be applied in various everyday activities, such as determining the sequence of steps when cooking, grouping items by type, finding the most efficient travel route, scheduling activities, or solving a problem by breaking it down into simpler parts. Thus, computational thinking is not limited to programming activities using computers but is a thinking skill that can be applied in various situations to find effective solutions. UNESCO also identifies the ability to organize and analyze information, use abstraction, outline solution steps, and identify effective solutions as key components of computational thinking.

In elementary education, computational thinking can be integrated into subject learning through contextual tasks that encourage students to solve problems systematically. One approach is unplugged coding, which introduces coding concepts through structured activities without requiring computers or digital programming tools. In this study, Coding Worksheets for Students (LKPD) were used in fifth-grade IPAS learning to connect computational thinking with content on the human respiratory and digestive systems. The worksheet activities were designed to engage students in four computational thinking indicators: decomposition, by breaking down biological processes into simpler components; pattern recognition, by identifying regularities in coded information; abstraction, by selecting relevant information about digestive disorders and their causes; and algorithmic thinking, by arranging stages of human growth in a logical sequence. By embedding these tasks in IPAS content, the worksheets provided a context for practicing computational thinking within subject learning rather than treating coding as a separate programming activity.

The results showed an increase in fifth-grade students' computational thinking scores following the implementation of Coding Worksheets in IPAS learning. The mean score increased from 61.67 on the pretest to 84.33 on the posttest. The learning activities were designed to encourage students to identify problems, analyze information, and organize solution steps systematically. However, because this study used a one-group pretest–posttest design, the observed improvement cannot be attributed solely to the worksheets.

The observed increase in computational thinking scores can be discussed in relation to the activities included in the Coding Worksheets, particularly those involving the human digestive system and growth. These activities were designed to provide concrete opportunities for students to practice the four computational thinking indicators. Each activity is designed to provide a learning experience that concretely represents the stages of computational thinking. In the decomposition stage, students break down the digestive process into simpler parts and identify the sequence of organs involved. This activity helps students understand that a complex problem or process can be solved by breaking it down into several parts that are easier to understand. For the pattern recognition indicator, the “Identifying Patterns in Food Substance Codes” activity invites students to discover regularities through the codes assigned to various types of food substances. Students must observe the information, compare the characteristics that emerge, and then recognize patterns to determine the appropriate answer. Furthermore, the abstraction indicator is implemented through the activity “Body Factory Rescue Mission,” which guides students to select important information related to disorders of the digestive system and their causes, while disregarding unnecessary information in completing the task. Meanwhile, the algorithmic thinking indicator is applied through the “Tracking the Human Growth Code” activity, which requires students to follow and arrange the stages of growth in sequence based on the codes and instructions provided. These four activities demonstrate that the concept of computational thinking is not only presented theoretically but is also applied through tasks that practice the processes of breaking down problems, identifying patterns, filtering information, and systematically organizing solution steps in the IPAS curriculum.

The findings can be discussed in relation to Piaget's theory of cognitive development, particularly the concrete operational stage, which is commonly associated with the development of logical reasoning about concrete situations (Babullah 2022). This perspective provides a theoretical rationale for incorporating contextual and sequential problem-solving activities into fifth-grade IPAS learning. However, students' developmental characteristics may vary, and the present study did not directly assess their

cognitive developmental stages. Therefore, Piaget's theory serves as a framework for interpreting the instructional approach rather than as direct evidence explaining the observed changes in computational thinking scores.

The findings also highlight the potential of integrating unplugged coding into subject-specific learning at the Madrasah Ibtidaiyah level. Unlike computer-based programming, the Coding Worksheets for Students (LKPD) used in this study incorporated the four computational thinking components—decomposition, pattern recognition, abstraction, and algorithmic thinking—into IPAS content related to the human respiratory and digestive systems. The increase in pretest and posttest scores suggests that this instructional approach may support students' computational thinking performance in a subject-learning context. Nevertheless, the study employed a one-group pretest–posttest design and combined the worksheets with broader instructional activities; therefore, the observed improvement cannot be attributed exclusively to unplugged coding. The contribution of this study lies in examining the integration of computational thinking activities into IPAS learning among fifth-grade students at a Madrasah Ibtidaiyah, rather than teaching coding as a separate subject.

The pretest results indicated that students' initial computational thinking performance was relatively low in pattern recognition and algorithmic thinking. These results suggest that students may have experienced difficulties in identifying patterns and arranging problem-solving steps systematically. However, the pretest scores alone do not establish the causes of these difficulties or demonstrate the characteristics of the students' previous instruction.

During the intervention, instruction using the Coding Worksheets was supported by instructional modules incorporating the Deep Learning approach. The instructional planning was designed to create a student-centered learning experience through activities that encouraged active engagement in constructing knowledge. In this study, the Coding Worksheets were used for unplugged coding activities that introduced coding concepts without computers, requiring students to formulate instructions, sequence steps, and solve problems systematically.

Students not only work on practice problems but also engage in a series of activities that require them to observe contextual problems, identify relevant information, organize concepts, discover relationships between pieces of information, and develop systematic solution strategies. Through these unplugged activities, students are trained to develop an understanding of decomposition, recognize patterns, practice abstraction, and formulate steps. During the learning process, the teacher acts as a facilitator who provides guidance, prompting questions, and feedback to help students reflect on their thought processes without directly providing answers. This approach gives students the opportunity to build their understanding independently through active, in-depth, and meaningful learning experiences. The increase in posttest scores occurred following an intervention that combined Coding Worksheets with Deep Learning-based instructional activities. Because these components were implemented together, their individual contributions to the observed change cannot be determined from this study. Unplugged coding activities in IPAS instruction provide a more concrete learning experience, allowing students to understand the concepts of computational thinking through hands-on activities, rather than just explanations. In this process, the deep learning approach strengthens student engagement because they do not merely receive information but also process and connect the knowledge they acquire.

This is in line with the concept of Deep Learning, which emphasizes meaningful learning through students' active engagement in understanding concepts, establishing connections between pieces of information, and applying knowledge to solve problems in

various contexts. This approach is not only oriented toward mastery of the subject matter but also toward the development of higher-order thinking skills through the processes of exploration, reflection, and problem-solving. Therefore, the integration of Deep Learning-based instructional modules with the Coding Worksheet provides a learning environment that supports the development of students' computational thinking skills (Hidayat 2026).

This is reflected in the activities in the Coding Worksheet, which encourage students to identify components of the respiratory and digestive systems, categorize information based on specific characteristics, analyze the relationship between organ structure and function, and logically sequence scientific processes. These activities provide opportunities for students to apply the stages of computational thinking in the context of IPAS learning so that the concepts studied are not only understood conceptually but also through a systematic reasoning process.

The findings indicate that students' computational thinking scores increased following the implementation of the Coding Worksheets for Students (LKPD) in IPAS learning. This result suggests that the worksheets, as part of the broader instructional activities, may support the development of computational thinking skills among fifth-grade students, although the one-group pretest–posttest design does not establish a causal relationship. In terms of product features, the LKPD is presented in a printed-book format, making it portable and usable without digital devices. The increase in computational thinking scores observed in this study is consistent with the potential of coding-based instructional media to support computational thinking development (Syamsiah et al., 2024).

The findings of this study have implications for the implementation of IPAS instruction in Madrasah Ibtidaiyah, particularly in efforts to develop students' computational thinking skills. The use of the Coding Worksheet demonstrates that unplugged coding activities can be integrated into instruction without relying on digital devices, allowing teachers to provide learning experiences that still engage students in computational thinking through structured activities. This integration also shows that the development of computational thinking skills need not be treated as a standalone subject but can be fostered through IPAS content that is closely tied to students' experiences and daily lives. The Coding Worksheets in this study provide an alternative learning approach that combines IPAS content with activities involving decomposition, pattern recognition, abstraction, and algorithmic thinking. Thus, this study contributes to the understanding that worksheets based on unplugged coding can be utilized as a learning tool that supports the development of students' computational thinking skills at the Madrasah Ibtidaiyah level.

4. Conclusions

This study found a statistically significant difference between pretest and posttest computational thinking scores among fifth-grade students following the implementation of Coding Worksheets in IPAS learning. The mean score increased from 61.67 to 84.33, with a mean N-Gain of 0.5967, categorized as moderate. These findings indicate that integrating unplugged coding activities into IPAS instruction has potential to support students' computational thinking development. However, because the study involved only 30 students in a one-group pretest–posttest design with a three-session intervention and no control group, the observed improvement cannot be attributed solely to the Coding Worksheets. Future studies should involve larger and more diverse samples, comparison groups, and longer intervention periods to examine the consistency of these findings.

5. Acknowledgements

The author would like to express his deepest gratitude to all those who have provided support, assistance, and contributions toward the completion of this study. Special thanks go

to the principal, classroom teachers, and fifth-grade students at MIN 5 Nganjuk who participated in this study. The author also extends gratitude to others who provided guidance, feedback, and support throughout the process of writing this article. May all the assistance provided be considered a charitable deed and be rewarded many times over.

References

- Astuti, S., Budiardjo, E. K., Hidayat, A., Wahyudin, A., Jihad, A., Alhapip, L., Nugroho, S. E., Jayanti, B. A. W. W., Aji, I. B., & Safitri, D. (2025). Panduan mata pelajaran koding dan kecerdasan artifisial: Fase C, D, E, dan F. Badan Standar, Kurikulum, dan Asesmen Pendidikan.
- Babullah, R. (2022). Teori Perkembangan Kognitif Jean Piaget dan Penerapannya dalam Pembelajaran: Jean Piaget's Theory of Cognitive Development and Its Application in Learning. *Epistemic: Jurnal Ilmiah Pendidikan*, 1(2), 131–152. <https://doi.org/10.70287/epistemic.v1i2.10>
- Bers, M. U., Strawhacker, A., & Sullivan, A. (2022). The state of the field of computational thinking in early childhood education (OECD Education Working Papers No. 274). OECD Publishing. <https://doi.org/10.1787/3354387a-en>
- Candraningtyas, S. R., & Khusna, H. (2023). Computational thinking ability becomes a predictor of mathematical critical thinking ability. *Alifmatika: Jurnal Pendidikan dan Pembelajaran Matematika*, 5(2), 247–263. <https://doi.org/10.35316/alifmatika.2023.v5i2.247-263>
- Christi, S. R. N., & Rajiman, W. (2023). Pentingnya berpikir komputasional dalam pembelajaran matematika. *Journal on Education*, 5(4), 12590–12598. <https://doi.org/10.31004/joe.v5i4.2246>
- Dwijayani, N. M. (2022). Precoding activities to improve student's computational thinking skills. *Widyagogik: Jurnal Pendidikan dan Pembelajaran Sekolah Dasar*, 10(2), 21–37. <https://doi.org/10.21107/widyagogik.v10i1.15671>
- Fuadah, A. S., Mukti, T. S., & Firmansyah, A. (2025). Differences in mathematical problem-solving skills based on computational thinking in problem-based learning and conventional learning. *Jurnal Pengembangan Pembelajaran Matematika (JPPM)*, 7(2), 122–133. <https://doi.org/10.14421/jppm.2025.72.122-133>
- Hapsari, Y. N. M., & Masduki. (2024). Eksplorasi kemampuan berpikir komputasi siswa SMP dalam menyelesaikan soal pola bilangan ditinjau dari gaya belajar. *FIBONACCI: Jurnal Pendidikan Matematika dan Matematika*, 10(1), 21–38. <https://doi.org/10.24853/fbc.10.1.21-38>
- Hidayat, F. (2026). Pembelajaran mendalam (Deep Learning) dalam pendidikan dasar: Analisis filosofis terhadap transformasi pembelajaran di era Artificial Intelligence. *Walada: Journal of Primary Education*, 5(1), 141–156. <https://doi.org/10.61798/wjpe.v5i1.470>
- Listiowati, M. P., Astuti, T., Ellianawati, E., & Subali, B. (2025). Effectiveness of coding in enhancing computational thinking among elementary school students: A literature review 2020–2025. *Edunesia: Jurnal Ilmiah Pendidikan*, 6(3), 1766–1784. <https://doi.org/10.51276/edu.v6i3.1299>
- Masjudin, M., Muliadi, A., Kurniawan, A., Fatwini, F., Nissa, I. C., & Adaminata, A. (2022). Development of STEM-integrated problem-based learning instructional materials to enhance science students' problem-solving ability. *Jurnal Penelitian Pendidikan IPA*, 8(5), 2582–2592. <https://doi.org/10.29303/jppipa.v8i5.13075>

- Miftah, R., Kurniawati, L., & Musa, K. F. (2024). Development of prospective teacher student worksheets through interactive case-based learning model assisted by Cublend App to improve mathematical literacy skills. *Mathematics Teaching Research Journal*, 16(4), 76–93. <https://doi.org/10.31004/mtrj.v16i4.1442>
- Morales-Teherán, M. W., Gómez-Jaramillo, S., & Parada-Trujillo, A. E. (2026). Pensamiento computacional, educación STEM y prácticas de enseñanza: Una revisión sistemática de la bibliografía. *Revista Electrónica Educare*, 30(1), 1–21. <https://doi.org/10.15359/ree.30-1.20137>
- Munir, M. M., Muhith, A., & Arifin, A. S. (2025). The effect of Scratch-based instructional media on Grade 5 elementary students' coding skills: A quasi-experimental study. *Journal of Educational Research and Practice*, 3(3), 493–508. <https://doi.org/10.70376/jerp.v3i3.410>
- Syamsiah, N. O., Firmansyah, Y., Mustika, Y., Burhanudin, B., Gani, A., & Fitriani, N. (2024). Pengaruh edukasi unplugged coding terhadap kemampuan computational thinking anak usia sekolah dasar. *VOX EDUKASI: Jurnal Ilmiah Ilmu Pendidikan*, 15(2), 365–373. <https://doi.org/10.31932/ve.v15i2.4048>
- Wu, T.-T., Asmara, A., Huang, Y.-M., & Hapsari, I. P. (2024). Identification of problem-solving techniques in computational thinking studies: Systematic literature review. *SAGE Open*, 14(2), 1–20. <https://doi.org/10.1177/21582440241249897>