Attractive: Innovative Education Journal
Vol. 7 No. 3, November 2025

From Curriculum to Classroom Practice: A Case Study of a STEAM Learning Ecosystem for Fostering Students' Computational Thinking

Alma Azrel Pamula [1]*, Purnomo [1]
[1] *Universitas Islam Negeri Salatiga, Indonesia*

almakuliah@gmail.com*

ARTICLE INFO
Article history:
Received
August 15, 2025
Revised
October 04, 2025
Accepted
November 28, 2025

ABSTRACT

This study explores how a Science, Technology, Engineering, Arts, and Mathematics (STEAM) learning ecosystem fosters computational thinking among seventh-grade students at Mega Islamic Boarding School Semarang by examining the relationship between curriculum sequencing, classroom practices, and institutional support. A qualitative single-case study was conducted over six weeks. Data were collected through semi-structured interviews with the vice principal for curriculum and one STEAM teacher, two non-participant classroom observations involving 13 students, analysis of the seventh-grade STEAM curriculum, and school documentation. The data were analyzed thematically using deductive and inductive coding, including the mapping of STEAM dimensions across classroom activities and school projects. The findings revealed three interconnected themes. First, the curriculum introduced computational thinking concepts before programming activities. Second, coding practices were facilitated through teacher explanation, independent practice, feedback, peer assistance, and repeated attempts. Third, digital facilities, teacher involvement in curriculum development, and project exhibitions supported the continuity of the STEAM program. The STEAM mapping showed that Technology, Engineering, and Mathematics were more visible in coding activities, whereas Science and Arts appeared more prominently in project-based activities. These findings indicate that computational thinking is facilitated not by coding alone, but through an interconnected ecosystem linking curriculum design, classroom practice, and institutional support.

Keywords: Computational Thinking, Curriculum Sequencing, Junior Secondary Education, STEAM Ecosystem, Institutional Support

Journal Homepage https://attractivejournal.com/index.php/aj/

Published by CV. Creative Tugu Pena

INTRODUCTION

Digital education requires schools to go beyond simply providing devices, internet connections, and learning platforms. Schools need to design learning experiences that help students understand problems, select information, develop strategies, test solutions, and evaluate results systematically. The use of technology that is solely oriented toward information retrieval or completing routine tasks does not necessarily improve the quality of students' thinking (Gardeli & Vosinakis, 2025) . Technology gains pedagogical value when its use is linked to the development of independence, reasoning, and problem-solving skills (Aroonsiwagool et al., 2025; Blanc et al., 2025) .

Computational thinking provides a systematic framework for formulating problems and developing solutions that can be implemented and evaluated. This ability is not synonymous with the skill of writing computer programs, as it encompasses decomposition, pattern recognition, abstraction, algorithm design, and evaluation (Grover & Pea, 2013; Shute et al., 2017; Wing, 2006) . Decomposition helps students break down complex problems into simpler parts, while pattern recognition helps students identify regularities among the components of a problem. Abstraction is used to select relevant information, while algorithms guide the development of logical and structured solution steps (Nurlaelah et al., 2024) . These characteristics enable computational thinking to be applied across various subjects and everyday situations, not just in computer science education.

Junior high school is a critical stage for introducing the foundations of computational thinking. Students at this level begin to encounter material that requires abstract reasoning, the organization of information, and the development of more systematic procedures. Seventh-grade students, in particular, are in a transitional phase from basic learning to activities that demand the ability to connect concepts and develop problem-solving strategies. Gradually structured learning activities can help students progress from recognizing simple patterns to designing algorithms and developing more complex solutions (Acar & Övez, 2022) . Seventh grade is therefore a relevant starting point before students engage in more complex programming activities and projects.

Science, Technology, Engineering, Arts, and Mathematics (STEAM) education offers an integrative framework for developing systematic thinking processes. Science helps students understand phenomena based on evidence; technology provides the means to process information and apply ideas; and engineering guides the process of designing and refining solutions. The arts support creativity, communication, and the representation of ideas, while mathematics reinforces logical reasoning, measurement, and quantitative analysis (Ajibudiarta et al., 2026; Astuti et al., 2024) . The integration of these five disciplines enables students to use knowledge from various fields to understand problems and develop contextually relevant solutions.

The connection between STEAM and computational thinking lies in their shared focus on problem-solving. STEAM learning can provide activities that guide students to identify problems, organize information, develop alternatives, plan steps, and evaluate solutions. This sequence of activities has the potential to involve decomposition, pattern recognition, abstraction, algorithms, and evaluation if teachers explicitly design the learning process to practice these components (Herro et al., 2022) . Integrating STEAM with the design process can also guide students from the stage of understanding a problem toward creating, testing, and refining prototypes (Thepprasit et al., 2026) . The pedagogical value of STEAM is therefore determined not only by the existence of devices or products but also by the thinking processes that accompany their development.

Coding can be used as a medium to facilitate computational thinking. Activities such as using variables, writing instructions, running programs, checking outputs, and correcting errors provide students with opportunities to explore the relationships between commands, processes, and outcomes. However, successfully running a program does not automatically indicate that students understand the problem or are applying all components of computational thinking. Students may complete tasks by copying examples or following instructions without understanding the reasoning behind the steps taken. The pedagogical value of coding depends on task design, scaffolding, feedback, opportunities for debugging, and reflective activities that encourage students to explain and evaluate their problem-solving strategies (Mills et al., 2025; Tikva & Tambouris, 2023) .

Previous research has demonstrated the potential of STEAM to support science literacy, creativity, critical thinking, collaboration, engagement, and problem-solving. STEAM learning has been implemented through project-based learning, inquiry-based learning, design thinking, digital media, simulations, and virtual laboratories (Alves et al., 2026; Fitriani et al., 2025; Panggabean et al., 2025; Simarmata et al., 2026) . These studies show that STEAM can create active and contextual learning experiences. Most studies, however, still treat STEAM as an intervention, model, or tool for measuring changes in academic achievement, creativity, and problem-solving skills (Djam'an et al., 2026; Hariyono et al., 2026; Kuo, 2026) .

The predominance of intervention-based research limits our understanding of how STEAM is designed and sustained as part of school life. The sustainability of STEAM is not determined by a single project, a single tool, or a single teacher. Program implementation is also influenced by curriculum sequencing, classroom practices, teacher competencies, digital facilities, school activities, and institutional support (Isozaki, 2024) . The relationships among these components determine whether STEAM develops into a consistent learning experience or remains merely an incidental activity. Studies on STEAM must therefore consider the pedagogical ecosystem that connects curriculum design, learning activities, resources, stakeholders, and school programs.

Research on computational thinking also remains heavily focused on coding, robotics, computer science, and the use of digital devices. An overly dominant technological orientation can lead to the assumption that computational thinking develops only through programming. In fact, decomposition, pattern recognition, abstraction, algorithms, and evaluation can also be developed through unplugged activities, logic games, modeling, concrete simulations, and visual representations (Aytekin & Topçu, 2026; Wang et al., 2026) . Therefore, an analysis of computational thinking must go beyond merely noting the types of devices or programming languages used. Attention must also be directed toward curriculum sequencing, task design, teacher support, peer interactions, and school conditions that facilitate the thinking process.

Limitations in the literature regarding the relationship between curriculum, learning practices, and institutional support highlight the need for research that frames STEAM as a pedagogical ecosystem. An ecosystem approach enables research to elucidate the connections between planned competencies, implemented learning experiences, and the resources that sustain the program. This approach also prevents a direct equating of the presence of coding with the development of computational thinking. Empirical evidence regarding the mechanisms of intercomponent relationships at the junior high school level still needs to be strengthened, particularly in the early stages when students are first introduced to basic programming.

Mega Islamic Boarding School Semarang provides a relevant empirical setting to examine how a STEAM learning ecosystem is built at the junior high school level. The school's curriculum sequentially introduces computational thinking, decomposition, pattern recognition, abstraction, algorithms, problem-solving, the Bebras Challenge, and basic programming. Learning practices utilize Chromebooks and Google Apps Script, while teachers play a role in curriculum development and provide support during classroom activities. In addition to learning activities, the school also organizes project exhibitions showcasing technology- and engineering-based products. However, the presence of coding activities, digital devices, and technology products does not automatically demonstrate how the STEAM dimensions are integrated or how these processes contribute to the development of students' computational thinking.

Given this gap, this study aims to explore how the STEAM learning ecosystem facilitates the development of computational thinking among seventh-grade students at Mega Islamic Boarding School in Semarang. Specifically, this study focuses on the relationship between curriculum sequencing, in-class coding practices, and institutional support from the school. This study seeks to answer three questions: (1) How is computational thinking positioned within the seventh-grade STEAM curriculum sequence? (2) How are coding learning practices implemented to facilitate students' computational thinking processes? (3) How does school support contribute to the sustainability of the STEAM learning ecosystem? Through this approach, this study does not position STEAM as a single technological activity but as a pedagogical ecosystem that connects curriculum design, classroom learning experiences, and institutional conditions that support its sustainability.

METHOD

This study employs a qualitative approach with a single-case study design to explore the STEAM learning ecosystem in facilitating the development of computational thinking among seventh-grade students at Mega Islamic Boarding School in Semarang. This design was chosen because the study focuses on a single educational context and examines the interconnections between the curriculum, learning practices, technology use, and institutional support in their natural setting (Denzin & Lincoln, 2009; Yin, 2018). The units of analysis include the STEAM curriculum design, classroom learning practices, and school support. The concepts of decomposition, pattern recognition, abstraction, algorithms, and evaluation were used as a framework for interpreting the learning process, not for measuring students' individual abilities (Grover & Pea, 2013; Shute et al., 2017; Wing, 2006).

The study was conducted over six weeks and involved the vice principal for curriculum, one purposefully selected STEAM teacher, and one class of 13 students. Data were collected through semi-structured interviews, non-participant observation, analysis of curriculum documents, and review of the school's digital documentation. Interviews were conducted individually for approximately 40 minutes and recorded with the informants' consent. Observations were conducted twice, each lasting about 80 minutes, to record the teacher's explanations, the use of Chromebooks and Google Apps Script, students' independent work, feedback, peer assistance, and revisions to their work. Curriculum documents were analyzed based on the sequence of material covering computational thinking, problem-solving, Bebras challenges, and basic programming, while social media documentation was used as supporting data regarding STEAM projects and exhibitions.

The data were analyzed using a six-phase thematic analysis process, namely data familiarization, initial coding, theme development, theme review, theme definition and naming, and report writing (Braun & Clarke, 2006). Coding was conducted deductively based on the concepts of computational thinking, problem-solving, and the STEAM ecosystem, as well as inductively to identify patterns emerging from the data (Fereday & Muir-Cochrane, 2006). Learning activities and projects were also mapped according to the dimensions of Science, Technology, Engineering, Arts, and Mathematics. The validity of the findings was strengthened through source triangulation, methodological triangulation, and member checking. The identities of informants and students were anonymized, while the findings were understood contextually and were not intended for statistical generalization.

RESULTS AND DISCUSSION

Analysis of two interview transcripts, two classroom observation notes from a single class of 13 students, the seventh-grade STEAM curriculum documents, and the school's digital documentation yielded three main themes. The first theme indicates that the curriculum introduces computational thinking components before students learn basic programming. The second theme shows that *coding* practices take place through teacher explanations, independent practice, checking results, feedback, learning support, and retrying. The third theme shows that the school supports the program's sustainability through curriculum development, the provision of digital facilities, and STEAM project exhibitions. These three themes demonstrate that seventh-grade STEAM learning is built through the relationship between the curriculum, learning practices, and school support—not merely through the use of devices or the implementation of *coding* activities.

Table 1. Synthesis of Key Findings on the STEAM Learning Ecosystem

Theme	Evidence Sources	Key Findings	Descriptive Meaning
The STEAM curriculum as a pathway for developing ways of thinking	Interviews with the vice principal and STEAM teachers; curriculum documents	The material progresses from computational thinking, decomposition, pattern recognition, abstraction, algorithms, and problem-solving to the Bebras challenges and basic programming	*Coding* is positioned as an application stage following the introduction to thinking processes
Coding practice through exercises and learning support	Classroom observations; interviews with STEAM teachers	Learning includes explanations, independent practice, checking results, feedback, assistance from teachers and peers, and retrying	*Coding* practice takes place as a gradual learning process supported by classroom interaction

Theme	Evidence Sources	Key Findings	Descriptive Meaning
School ecosystem support	Interviews; facility observations; documentation from the school's official social media accounts	The program is supported by curriculum development, Chromebooks, Google Apps Script, and STEAM project exhibitions	Program sustainability involves the relationship between facilities, teachers, the curriculum, and school activities

The STEAM Curriculum as a Pathway for Developing Ways of Thinking

The seventh-grade STEAM curriculum prioritizes the development of ways of thinking over the introduction of more complex programming. The vice principal for curriculum, YSZ, stated, "What we're building isn't the ability to use technology, but the way of thinking." The curriculum document supports this statement by outlining a sequence of topics that begins with computational thinking, decomposition, pattern recognition, abstraction, algorithms, and problem-solving, followed by the Bebras challenges and basic programming. This sequence of material positions *coding* as part of the learning pathway, not as a standalone technical skill. Before using a programming language, students are introduced to how to break down problems, recognize patterns, select relevant information, and devise solution steps. STEAM teacher GS explains, "In 7th grade, I do focus on coding—learning programming languages—since we have the necessary resources." The curriculum document lists JavaScript and Google Apps Script as tools for introducing variables, data types, simple operations, and the use of instructions. The seventh-grade curriculum thus integrates the introduction to computational thinking, problem-solving, structured challenges, and basic programming into a single learning sequence.

***Coding* Practice Through Explanations, Independent Practice, and Learning Support**

Coding practice takes place through a series of teacher explanations, independent practice, result checks, feedback, learning support, and retries. During the observed session, the teacher introduced functions and the use of \`var\`, \`const\`, and \`let\` through Google Apps Script before asking students to write and run instructions using their own Chromebooks. Students demonstrated varying speeds in understanding and completing the task. Students who had finished showed their work to the teacher for review and received immediate feedback. One student who did not yet understand the material asked the teacher for help, then received additional explanations and assistance from a peer who had already completed the task. That student tried again and was eventually able to complete the assignment. Observation notes indicate that Chromebooks served as hands-on devices and Google Apps Script as a medium for applying variable declarations, while task completion was supported by interactions among students, the teacher, and peers. The *coding* practice observed thus consisted not only of writing code but also included the processes of trying, receiving help, checking results, and revising work.

School Support for the Sustainability of the STEAM Program

The school's support for the STEAM program includes institutional orientation, the role of teachers in curriculum development, the provision of digital facilities, and the organization of project exhibitions. The vice principal in charge of curriculum stated, "Computational thinking and problem-solving skills must become ingrained in the school culture." STEAM teachers also serve as curriculum developers and explained, "I am currently formulating a new STEAM-specific curriculum. I'm still developing it and adapting it to the needs here." During the observation, students used Chromebooks to open Google Apps Script, follow instructions, write code, run programs, and demonstrate their work. The availability of devices allowed for both instruction and practice to take place during the same session. The school's support is also demonstrated through project exhibitions. GS explained, "The STEAM exhibition is like a showcase, so we'll present our projects, and parents will come too." A review of the school's official social media accounts revealed projects such as Arduino-based glasses designed to help the visually impaired detect objects, as well as a car equipped with ultrasonic sensors designed to detect obstacles. Digital documentation was used only as supporting data because the researcher did not directly observe the exhibition process or project development. The documentation also could not confirm that both products were created by seventh-grade students. Therefore, the two products are presented as evidence of STEAM activities at the school level, rather than as evidence of student learning outcomes—which were the focus of the observation.

Table 2. Mapping of STEAM Dimensions in School Learning Activities and Projects

STEAM Components	Arduino-Based Glasses	Ultrasonic Sensor Car	*Coding* Activities
Science	Sensor Principles and User Accessibility Issues	The Concept of Ultrasonic Detection	Not explicitly visible
Technology	Arduino, electronic components, and programming	Sensor-based system	Google Apps Script
Engineering	System design and prototype development	Hardware integration and testing	Algorithm implementation
Arts	User-centered product design	Product presentation and visualization	Not explicitly visible
Mathematics	Measurements and logical operations	Distance calculations and sensor interpretation	Variables and computational logic

The mapping in Table 2 shows that the five STEAM dimensions do not appear uniformly in every activity. *Coding* activities more clearly represent Technology, Engineering, and Mathematics through the use of Google Apps Script, the implementation of instructions, variables, algorithms, and computational logic. Science and Arts are more evident in prototype-based projects through the application of sensor principles, product design, visualization, and user-centered design. This distribution indicates that STEAM integration occurs through a combination of classroom learning and school projects. Each activity contributes differently to the STEAM ecosystem, so interdisciplinary integration does not necessarily have to be fully present in a single activity.

Overall, the curriculum provides a sequence for competency development; classroom practices offer students space to apply and refine their work; and teachers, digital facilities, and project exhibitions support the program's sustainability at the school level. The research findings describe the pedagogical processes and institutional conditions that facilitate computational thinking. The research findings are not used to draw conclusions about improvements in individual students' abilities because the study did not involve computational thinking assessments, student interviews, *coding* file analysis, or reviews of individual portfolios.

DISCUSSION

The research findings answer the three research questions by demonstrating that the development of computational thinking in STEAM learning does not rely on a single activity, a single device, or a single discipline. Curriculum sequencing provides a direction for competency development, *coding* practice provides a space to apply and refine problem-solving steps, while school support ensures the program's sustainability. These relationships position STEAM as a pedagogical ecosystem, not merely a collection of technology-based activities. The STEAM framework explains the interconnected disciplines in learning; computational thinking theory explains the guided cognitive processes; problem-solving theory explains the sequence of student actions; while the perspectives *of scaffolding* and the learning ecology explain the social and institutional support that enables the learning process to take place. These findings reinforce the view that the use of new technologies has pedagogical significance when connected to curriculum goals, learning practices, and a supportive institutional environment.

The sequencing of the curriculum—from decomposition, pattern recognition, abstraction, algorithms, and problem-solving toward the Bebras challenges and basic programming—demonstrates the curriculum's role in guiding learning experiences. The STEAM framework highlights the importance of connecting science, technology, engineering, the arts, and mathematics through integrative learning experiences (Yakman & Lee, 2012) , while computational thinking theory positions problem formulation, information organization, and solution development as processes that cannot be equated with coding skills (Grover & Pea, 2013; Shute et al., 2017; Wing, 2006) . The relationship between these two perspectives indicates that disciplinary integration requires a clear pathway of competencies. STEAM without a cognitive focus risks becoming a collection of engaging projects that lack structured skill development. Conversely, computational thinking without an integrative context can become a procedural exercise disconnected from meaningful problems. The curriculum sequencing identified in this study functions as a form of curricular orchestration because it regulates when thinking concepts are introduced, when students face challenges, and when technology is used to implement solutions. This interpretation aligns with the view that STEAM implementation requires a connection between theory, curriculum, and practice so that disciplinary integration does not stop at the program label .

Although the curriculum documents outline a systematic pathway of competencies, the observed classroom practices have not yet demonstrated all dimensions of STEAM and all components of computational thinking with equal strength. The Google Apps Script activities more clearly demonstrate the use of technology, the application of algorithms, variables, computational logic, and the verification of results. Decomposition, pattern recognition, and abstraction have not yet been explicitly evident in students' behavior or work artifacts. This finding supports the view that the presence of programming alone is insufficient to demonstrate the occurrence of the entire computational thinking process (Shute et al., 2017) . STEAM learning also needs to explicitly connect concepts, inquiry, design, collaboration, and problem-solving (Herro et al., 2022) . Based on these findings, the seventh-grade curriculum is better understood as a pedagogical pathway that provides the initial conditions for the development of computational thinking, rather than as evidence that students have mastered all of its components.

Coding practice—which involves explanations, independent practice, review of results, learning support, and retries—demonstrates that programming functions as an iterative learning process. Selby and Woollard's " " framework ((2013)) helps explain the cognitive components that may be involved when students understand instructions, write code, and check results, while Pólya's stages of problem-solving (;(1945)) describe students' progression from understanding the problem, formulating a plan, carrying out steps, to reviewing the results. When students encounter difficulties using `var`, `const`, and `let`, the problem-solving process does not proceed linearly. Students return to the stage of understanding function declarations before trying again. These findings indicate that learning to program involves an iterative cycle of understanding, application, verification, and refinement.

This iterative process is also social in nature, as students receive additional explanations from teachers and assistance from peers. *Scaffolding* enables students to tackle the complexity of tasks without delegating the entire problem-solving process to the helper (Chen et al., 2026) , while support tailored to students' needs can strengthen their engagement in programming learning and computational thinking (Tikva & Tambouris, 2023) . However, successfully completing a task after receiving assistance does not automatically indicate independent understanding. Students may still be following instructions or imitating them without understanding the rationale behind their use. Mills et al.(2025) emphasize that *coding* does not automatically foster computational thinking because its pedagogical value depends on the task design and the reasoning processes accompanying the activity. Therefore, assistance from teachers and peers should be directed toward questions that prompt students to explain difficulties, predict outcomes, select instructions, identify errors, and explain the rationale for corrections. The findings of this study do not support the assumption that the successful execution of code can serve as a direct indicator of the development of computational thinking, but they do show that *coding* provides an environment with the potential to facilitate it when accompanied by processes of reflection and evaluation.

The sustainability of STEAM learning also depends on the relationships between teachers, digital resources, the curriculum, projects, and spaces for communication. The learning ecology perspective views learning as the result of interactions among actors, resources, activities, and the institutional environment. STEAM design needs to connect the use of technology with learning experiences and social interactions (Aguayo et al., 2023) , while integrative learning develops through the interconnectedness of formal environments, technological resources, social participation, and community support (Chatzidaki et al., 2025) . In this study, Chromebooks did not function as an independent pedagogical factor. Their educational value emerges when the devices are used to deliver curriculum content, follow the teacher's explanations, engage in practice, receive feedback, and revise work. Technology is therefore a resource within the ecosystem, not the ecosystem itself.

Teachers play a crucial role as the link between curriculum design and classroom practice. Their role extends beyond simply delivering content to include developing and adapting the curriculum to meet the school's needs. These findings align with research that identifies teacher competence and agency as key factors in STEAM implementation (Kessler et al., 2024; Spyropoulou & Kameas, 2024) . However, the dominant involvement of a single teacher also highlights the program's potential vulnerability. The sustainability of the ecosystem may be compromised when curriculum development and the implementation of activities rely too heavily on a single individual. The school's efforts to foster computational thinking and problem-solving, therefore, cannot yet be equated with an established culture. Research data has not yet shown that these two competencies have been consistently applied across various subjects and school activities. The conditions observed are more accurately understood as an ecosystem currently being developed toward a culture of STEAM and problem-solving.

Mapping the STEAM dimensions reveals that interdisciplinary integration is spread across several activities. Technology, Engineering, and Mathematics are more evident in *coding* activities, while Science and Arts are more prominent in sensor-based products, prototype design, visualization, and user-centered design. This distribution indicates that STEAM integration does not need to be fully present in a single session but can be built through a series of complementary activities. However, the project documentation does not yet show the product development process and does not confirm the involvement of seventh-grade students. The exhibition is also better understood as a space for communicating results to parents and the school community rather than as evidence of external collaboration. Collaboration requires involvement in planning, development, providing feedback, or evaluating the project. These limitations are important so that the product documentation is not interpreted beyond the available information.

Theoretically, this study expands the STEAM discourse from a focus on the effectiveness of a model, tool, or project toward an explanation of the ecosystem mechanisms that sustain the program. This study positions the STEAM ecosystem as a three-layered relationship: the curriculum provides competency direction, classroom practices facilitate implementation and refinement, while school support ensures its sustainability. Each layer has a distinct function but is interdependent. The curriculum does not prove student proficiency; *coding* does not automatically result in computational thinking; Chromebooks do not guarantee meaningful learning; and exhibitions do not, by themselves, prove the integration of all disciplines. The main contribution of this study lies in the explanation that the development of computational thinking is facilitated through the relationships among components, not through the isolated presence of any single component.

The research findings have practical implications at the curriculum, instructional, and institutional levels. The curriculum must sequentially address decomposition, pattern recognition, abstraction, algorithms, and problem-solving before applying them to *coding* or technology projects. Classroom practices must provide students with opportunities to explain their reasoning, predict outcomes, identify errors, receive *scaffolding*, compare strategies, and revise their work. At the school level, digital resources must be linked to teacher competency development, shared program responsibilities, adaptive curricula, project-based activities, and spaces for sharing outcomes. These reinforcements are necessary to ensure that the program's sustainability does not depend on the availability of devices or a single teacher.

Interpretation of the findings must take into account the limitations of the research design. The study was conducted at a single school, involved two interview participants, and included two observation sessions; therefore, the results are contextual in nature. The research did not include student interviews, computational thinking assessments, analysis of *coding* files, or individual project portfolios. Information regarding the exhibition was also obtained through interviews and school documentation, rather than through direct observation of project development. Future research should involve multiple schools, a longer observation period, student perspectives, artifact analysis, and performance-based assessments. Such an approach could elucidate not only how the STEAM ecosystem is constructed but also how the relationships among its components contribute to the development of students' computational thinking over time.

CONCLUSION

This study demonstrates that seventh-grade STEAM learning at Mega Islamic Boarding School in Semarang facilitates the development of computational thinking through the interplay between curriculum sequencing, iterative and social coding practices, and the support of the school ecosystem. The curriculum guides learning from an introduction to decomposition, patterns, abstraction, algorithms, and problem-solving toward the Bebras competition and basic programming; classroom practices provide opportunities for students to apply instructions, receive feedback, get help, and revise their work; while digital facilities, the teacher's role, and exhibition activities support the program's sustainability. These findings confirm that STEAM should not be understood merely as the use of technology or coding activities, but as a pedagogical ecosystem that connects design, implementation, and institutional support. Although this study did not measure improvements in individual students' abilities, the results elucidate mechanisms that have the potential to support the development of computational thinking in a school context.

REFERENCES

Acar, İ. G., & Övez, F. D. (2022). The effect of block-based game development activities on the geometry achievement, computational thinking skills, and opinions of seventh-grade students. *Journal of Educational Technology and Online Learning*, *5*(4), 1106–11 21. https://doi.org/10.31681/jetol.1151170

Aguayo, C., Videla, R., López-Cortés, F., Rossel, S., & Ibacache, C. (2023). Ethical Enactivism for Smart and Inclusive STEAM Learning Design. *Heliyon*, *9* (9), e19205. https://doi.org/10.1016/j.heliyon.2023.e19205

Ajibudiarta, R., Khoiri, N., & Patonah, S. (2026). STEAM-SDGs-Based Physics Learning Design on Renewable Energy Materials to Improve Students' Literacy and Numeracy Skills. *Journal of Innovation in Educational and Cultural Research*, *7* (3), 439–447. https://doi.org/10.46843/jiecr.v7i3.2734

Alves, A. R., Pires, P., Filipe, J., Conceição, T., & Baptista, M. (2026). Design thinking applied to I-STEM education as a trigger for developing creativity: A prototype for monitoring indoor air quality. *International Journal of Technology and Design Education*. https://doi.org/10.1007/s10798-026-10094-7

Aroonsiwagool, A., Tuntiwongwanich, S., Pimdee, P., Meedee, C., & Moto, S. (2025). Assessing Instructors' Perceptions of Critical Skills in Computational Thinking and Block-Based Programming: A Needs Assessment Approach. *International Journal of Instruction*, *18*(2), 245– 260. https://doi.org/10.29333/iji.2025.18214a

Astuti, R., Siswanto, S., & Walid, M. (2024). Innovation in Islamic Education Management: Enhancing Teachers' Professionalism and Techno-Pedagogical Skills. *Academic Journal Research*, *2* (2), 16–23. https://doi.org/10.61796/acjoure.v2i2.231

Aytekin, A., & Topçu, M. S. (2026). How Unplugged Computational Thinking Shapes Students' Creative Thinking: Evidence From the Human Nervous and Endocrine Systems. *Journal of Science Education and Technology*. https://doi.org/10.1007/s10956-026-10306-8

Blanc, S., Conchado, A., Benlloch-Dualde, J. V., Monteiro, A., & Grindei, L. (2025). Digital Competence Development in Schools: A Study on the Association of Problem-Solving with Autonomy and Digital Attitudes. *International Journal of STEM Education*, *12* (1), 13. https://doi.org/10.1186/s40594-025-00534-6

Chatzidaki, E., Giannakos, M., Tasiopoulou, E., Caraghiozov, I., Mentini, L., Laneve, S., Mavromanolakis, G., Koulouris, P., & Liakopoulos, V. (2025). Learning Ecologies as an Inclusive, Impactful Approach to Increasing Science Literacy. *Societal Impacts*, *7* , 100156. https://doi.org/10.1016/j.socimp.2025.100156

Chen, D., Zhang, Y., Luo, H., Gao, Z., Yu, L., & Lin, Y. (2026). Exploring the Impact of Scaffolding in Programming on Students' Computational Thinking: Evidence From a Three-Level Meta-Analysis. *Journal of Educational Computing Research*, *64*(1), 208– 237. https://doi.org/10.1177/07356331251386618

Djam'an, N., Arsyad, N., & Shyaa, F. D. (2026). Emerging Trends and Research Directions of STEAM Integration in Mathematics Education. *Formatif: Scientific Journal of Mathematics and Natural Sciences Education*, *16* (1), 187–200. https://doi.org/10.30998/cy1m8c05

Fitriani, R., Suwono, H., Ibrohim, I., & Lukiati, B. (2025). Enhancing Preservice Teachers' Collaborative Problem Solving through STEM Project-Based Learning. *International Journal of Evaluation and Research in Education (IJERE)*, *14* (3), 2278. https://doi.org/10.11591/ijere.v14i3.27725

Gardeli, A., & Vosinakis, S. (2025). Fostering computational thinking in young students through student-generated challenges in tangible mobile augmented reality games. *Discover Education*, *4* (1), 529. https://doi.org/10.1007/s44217-025-00899-4

Grover, S., & Pea, R. (2013). Computational Thinking in K–12: A Review of the State of the Field. *Educational Researcher*, *42* (1), 38–43. https://doi.org/10.3102/0013189X12463051

Hariyono, E., Rosdiana, L., Budiyanto, M., Satriawan, M., Nabilah, H., & Rizki, I. A. (2026). Factors Determining Socio-Scientific Issues in STEAM Education to Enhance Problem-Solving Skills for Pre-Service Teachers: Development and Validation of a Measurement Model. *Indonesian Journal of Science Education*, *15* (1). https://doi.org/10.15294/jpii.v15i1.40341

Herro, D., Quigley, C., Plank, H., Abimbade, O., & Owens, A. (2022). Instructional practices promoting computational thinking in elementary STEAM classrooms. *Journal of Digital Learning in Teacher Education*, *38* (4), 158–172. https://doi.org/10.1080/21532974.2022.2087125

Isozaki, T. (2024). *Theory and Practice of STEAM Education in Japan* (1st ed.). Routledge. https://doi.org/10.4324/9781003392545

Kessler, T. C., Boice, K. L., Koval, J., Jackson, J. R., Choi, J., Alemdar, M., Grossman, S., Simmons, K., & Usselman, M. (2024). Partnerships in STEAM: How Collaborating with STEAM Experts Impacts K-12 Teachers' Abilities to Implement STEAM Lessons in the Classroom. *Education Sciences*, *14* (6), 666. https://doi.org/10.3390/educsci14060666

Kuo, H.-C. (2026). STEAM PBL as an educational panacea? Investigating its impact on creative thinking and academic achievement across subjects. *Thinking Skills and Creativity*, *60* , 102072. https://doi.org/10.1016/j.tsc.2025.102072

Mills, K. A., Cope, J., Scholes, L., & Rowe, L. (2025). Coding and Computational Thinking Across the Curriculum: A Review of Educational Outcomes. *Review of Educational Research*, *95*(3), 581– 618. https://doi.org/10.3102/00346543241241327

Nurlaelah, E., Pebrianti, A., Taqiyuddin, M., Dahlan, J. A., & Usdiyana, D. (2024). Improving mathematical proof based on computational thinking components for prospective teachers in abstract algebra courses. *Infinity Journal*, *14* (1), 85–108. https://doi.org/10.22460/infinity.v14i1.p85-108

Panggabean, F. T. M., Sutiani, A., Purba, J., Dibyantini, R. E., Hasibuan, M. H. E., & Krisbiantoro, P. A. (2025). Enhancing Higher-Order Thinking Skills in Chemistry

Education: A Validated Canva-IBL-STEM Model for Stoichiometry Learning. *Indonesian Journal of Science Education*, *14* (4). https://doi.org/10.15294/jpii.v14i4.33559

Pólya, G. (1945). *How to Solve It: A New Aspect of Mathematical Method*. Princeton University Press.

Selby, C., & Woollard, J. (2013). *Computational Thinking: The Developing Definition* (pp. 1–6) [Project Report]. University of Southampton. https://eprints.soton.ac.uk/356481/

Shute, V. J., Sun, C., & Asbell-Clarke, J. (2017). Demystifying Computational Thinking. *Educational Research* Review, *22*

Simarmata, J. E., Purnomo, M., & Fallo, K. (2026). Evaluating interactive R-Shiny-based mathematics learning media through motivation and engagement pathways in border-region schools. *Journal on Mathematics Education*, *17*(1), 225– 246. https://doi.org/10.22342/jme.v17i1.pp225-246

Spyropoulou, N., & Kameas, A. (2024). Augmenting the Impact of STEAM Education by Developing a Competence Framework for STEAM Educators for Effective Teaching and Learning. *Education Sciences*, *14* (1), 25. https://doi.org/10.3390/educsci14010025

Thepprasit, W., Pulsawad, W., Thepnurat, M., Aridech, P., Klaewklar, K., & Thepprasit, R. (2026). An innovation-driven pedagogy integrating STEAM and design thinking to enhance computational thinking in middle school. *Discover Education*, *5* (1), 573. https://doi.org/10.1007/s44217-026-01561-3

Tikva, C., & Tambouris, E. (2023). The Effect of Scaffolding Programming Games and Attitudes Toward Programming on the Development of Computational Thinking. *Education and Information Technologies*, *28* (6), 6845–6867. https://doi.org/10.1007/s10639-022-11465-y

Wang, C., Lu, C., Chen, F., Liu, X., & Wang, Q. (2026). Assessing Computational Thinking Beyond Programming: A Scoping Review of Non-- -Programming--K--*Journal of Computer-Assisted Learning*, *42* (1), e70191. https://doi.org/10.1002/jcal.70191

Wing, J. M. (2006). Computational Thinking. *Communications of the ACM*, *49* (3), 33–35. https://doi.org/10.1145/1118178.1118215

Yakman, G., & Lee, H. (2012). Exploring Exemplary STEAM Education in the U.S. as a Practical Educational Framework for Korea. *Journal of the Korean Association for Science Education*, *32*(6), 1072– 1086. https://doi.org/10.14697/jkase.2012.32.6.1072