

THE DEVELOPMENT OF COMPUTATIONAL THINKING MODULES FOR GRADE VII STUDENTS TO IMPROVE PROBLEM-SOLVING SKILLS

Beni Utomo^{1*}, Paulina Heruningsih Prima Rosa², Nur Ernawati³,
Muhammad Khanafi Jazuli⁴, Sahrul Ramadhan⁵, Fatwa Ika Widarti⁶,
Nanda Amalia⁷

^{1,2}Universitas Sanata Dharma, Indonesia

³⁻⁷SMP Islam Al Azhar 26 Yogyakarta, Indonesia

*Correspondence: unlinearid@usd.ac.id

Received: 17 January 2026; Accepted: 21 January 2026

Abstract

This study aims to develop a computational thinking module for seventh-grade students. The type of research is research and development. The method used is ADDIE. The research was conducted by analyzing students' needs for computational thinking modules. The needs analysis stage is carried out by collecting data regarding the curriculum used by the school, the curriculum regarding coding and artificial intelligence, and school development in the self-improvement class subject. Based on these needs, the modules were designed and developed according to learning outcomes and school characteristics. The implementation was conducted in SMP Islam Al Azhar 26 Yogyakarta on July 2025.

Keywords: computational thinking, module, problem-solving

Introduction

Schools with special characteristics, whether oriented toward strengthening religious values, national identity, cultural heritage, or specific academic programs, face unique challenges in designing learning experiences that align with the needs of the 21st century. Many of these institutions integrate the national curriculum with institutional or international curricula such as the Cambridge Curriculum to provide students with a comprehensive and globally relevant education. This combination requires instructional approaches that balance academic achievement, character development, and global competencies.

In schools with religious specializations, teaching and learning emphasize not only cognitive development but also moral values, ethical conduct, and the cultivation of disciplined behaviors that reflect specific spiritual traditions. Meanwhile, schools with national character specializations tend to focus on values such as integrity, cooperation, love of country, civic responsibility, and leadership. There are also schools that specialize in technology, science, arts, or adopt unique learning cultures and disciplinary frameworks. These diverse institutional identities indicate that each school requires pedagogical approaches designed to fit its specific characteristics and educational goals.

At the same time, students in schools with special characteristics must develop essential global competencies, including critical thinking, creativity,



innovation, communication, collaboration, and digital literacy. These competencies form the foundation of Higher Order Thinking Skills (HOTS), which are increasingly important in a rapidly changing digital society. Teachers, therefore, need to adopt instructional strategies that do not merely transmit information but challenge students to analyze, problem-solve, and make informed decisions independently.

Computational Thinking (CT) is one instructional approach that aligns well with these needs. CT emphasizes problem-solving through decomposition, abstraction, pattern recognition, and algorithm design. Beyond programming, CT serves as a cognitive framework that can be applied across subject areas to strengthen students' analytical and critical thinking skills. Research demonstrates that CT supports digital literacy, creativity, and logical reasoning—competencies that are vital for navigating global challenges.

Several specialized schools have begun introducing digital learning programs, science and technology classes, or global competency enhancement initiatives to address these evolving demands. However, the implementation of CT remains limited and often inconsistent due to the absence of systematic instructional designs that integrate CT with the school's distinct identity—whether religious, nationalistic, or academically focused.

Therefore, there is a need for a learning design that embeds Computational Thinking in a contextual manner that aligns with the profile and characteristics of schools with specializations. Such a design should provide practical guidance for teachers to implement CT through project-based learning, real-world problem-solving, and meaningful digital activities while preserving each school's unique identity. Effective CT integration has the potential not only to strengthen academic achievement but also to support character development and global competence among students.

Literature Review

Computational thinking (CT) has become a foundational competence in modern education, particularly as schools strive to prepare students for an increasingly digital and interconnected world. Wing (2006) introduced CT as a universal skill that involves problem decomposition, abstraction, algorithmic design, and evaluation. Those skills are essential not only for computer science but also for broader problem-solving across disciplines. Since then, extensive research has highlighted the importance of integrating CT into middle-school curricula to cultivate higher-order thinking, creativity, and analytical reasoning in students (Grover & Pea, 2013). CT is also viewed as a bridge connecting technology-enhanced learning with conventional subject domains such as mathematics and science, generating meaningful learning experiences that reflect real-world reasoning processes (Czerkawski & Lyman, 2015).

Studies further emphasize that effective CT integration requires structured learning environments, authentic tasks, and clear scaffolding. For example, Weintrop et al. (2016) demonstrated that CT can be embedded within STEM activities through modeling, simulation, data practices, and computational problem-solving. These strategies help students understand abstract concepts through hands-on digital exploration. Similarly, Shute et al. (2017) found that CT skills can be developed through game-based environments that encourage students to engage in

experimentation, iteration, and reflection. Their findings underscore that CT is not merely about mastering tools but about cultivating cognitive habits that support systematic thinking.

In the context of schools with particular characteristics, such as those emphasizing religious values or national character formation, CT can strengthen holistic student development. Research shows that CT does not conflict with religious or character-based education; instead, it enhances students' ability to make thoughtful decisions, reason clearly, and collaborate effectively (Mohaghegh & McCauley, 2016). Schools with specialized missions, whether oriented toward religious identity or national character building, often aim to develop moral reasoning, disciplined thinking, and social responsibility. CT supports these goals by providing structured problem-solving frameworks that align with value-based education. For instance, integrating CT into project-based learning environments helps reinforce perseverance, collaboration, and ethical decision-making, competencies aligned with character-oriented educational missions (Lye & Koh, 2014).

Recent studies highlight the need for localized CT implementation models that match school context, student readiness, and institutional vision, demonstrating CT integration aligned with school culture and supported by professional development for teachers. This aligns with the objectives of schools that blend conventional curricula with specialized programs, such as digital literacy, STEM enrichment, or value-based learning. Additionally, computationally enriched science classrooms, like those described by Lee et al. (2020), highlight how CT can enhance inquiry-based learning, promote interdisciplinary understanding, and foster global competencies become key components for schools aiming to prepare students for international academic standards.

Overall, the literature consistently shows that CT integration benefits students not only cognitively but also socially and ethically. When implemented within structured programs, such as digital science classes or specialized improvement tracks, CT can provide pathways for students to strengthen global skills, character values, and readiness for future academic challenges. The body of research also affirms the importance of designing CT lessons that resonate with local school identities and aspirations, ensuring that computational thinking becomes a powerful tool for holistic and culturally aligned education.

Method

This study adopted an ADDIE-based research and development model comprising analysis, design, development, implementation, and evaluation stages. The analysis stage examined instructional needs, learner characteristics, and curriculum requirements related to Computational Thinking. The design stage focused on developing learning objectives, instructional scenarios, and evaluation instruments. In the development stage, CT-based learning materials were developed and revised based on expert validation. The implementation stage involved a limited trial in a real classroom setting. Finally, the evaluation stage assessed the practicality and implementation of the learning materials.

Data were collected using expert validation instruments, practicality questionnaires administered to teachers and students, and observation sheets of learning implementation. The collected data were analyzed descriptively to

determine the validity, practicality, and feasibility of the developed CT-based learning materials.

Findings and Discussion

ADDIE Analyze Stage

The analysis stage of the ADDIE model was conducted to identify the initial conditions of learning, the competency gaps, and the specific needs of students and teachers in implementing Computational Thinking (CT) within a school with particular characteristics. Although the school integrates national, foundation-based, and international curricula and aspires to develop students who are character-driven, high-achieving, and globally competent, classroom practices do not yet fully reflect these expectations. The demand for higher-order thinking skills, digital literacy, and technology-integrated problem solving has not been optimally aligned with the learning experiences currently provided, highlighting the need for a more systematic approach to embedding CT in instruction while maintaining the school's unique values and identity.

From the learner perspective, classroom observations and the researcher's teaching experience indicate that students possess basic digital literacy and show enthusiasm for technology-supported learning. However, many students are not yet accustomed to applying CT strategies such as decomposition, abstraction, pattern recognition, and algorithmic thinking in solving problems. Learning activities tend to be procedural and outcome-oriented, offering limited opportunities for students to engage in systematic, reflective, and exploratory thinking. This condition demonstrates a clear need for instructional guidance that supports students' development of structured thinking through project-based activities, simulations, and exploratory tasks.

Curricular analysis and interviews with teachers also reveal that existing lesson plans do not explicitly incorporate CT elements. Teachers require structured models and actionable instructional steps to integrate CT consistently and meaningfully into classroom instruction. Although the school provides adequate technological infrastructure and operates a technology-oriented special program, these resources have not yet been fully utilized to promote CT-based learning. Teachers need pedagogical support that bridges technological tools with CT-oriented learning processes.

Based on these findings, the primary needs include: (1) instructional materials that explicitly guide the integration of CT, (2) learning activities that promote systematic and project-based thinking, (3) enhanced teacher capacity to apply CT-oriented pedagogical strategies, and (4) more meaningful use of technology to support students' computational reasoning. This needs analysis establishes a strong foundation for the subsequent Design stage, ensuring that the instructional model to be developed directly addresses the identified gaps and aligns with both the school's character profile and curriculum expectations.

ADDIE Design Stage

Developing general CT learning scenarios

Table 1. Learning Scenario

The Schedule	Learning Material	Subject collaboration
July- September 2025 (about 3 months)	CT plugged and CT unplugged	Informatics

- a. How students learn: Students will be introduced to CT shortly. Then, students will have some activities.
- b. Learning hours: 35 minutes each (70 minutes in a week)
- c. Introduction: learning objectives, learning methods; core learning: activities and ice breaking; closing: reflection

Planning worksheets containing four foundations/ core of CT

- a. Students will access the learning material using an electronic tablet
- b. The learning support by Google Classroom
- c. Some activities may still need paper as a worksheet
- d. Need teamwork

Simple classroom exercises

- a. Activity for session 1: how to introduce to every foundation of CT by examples
- b. Activity for session 2: learning phenomenon of CT’s foundations
- c. And activities in each session should be finished by each student or team

ADDIE Development Stage

Table 2. General Development of the Media

General CT	Problem Decomposition	Pattern Recognition	Abstraction	Thinking Algorithm
The design	Students identify key components of a real-world problem and divide it into simpler sub-problems using guided questions and worksheets	Students compare multiple problem scenarios to recognize similarities, trends, or repeated structures that support solution development	Students select essential variables and information from contextual problems and represent them using diagrams, tables, or simplified models	Students design solution steps in the form of flowcharts, pseudo-code, or structured procedures to solve the given problems

The examples How people eat watermelon: usually people should slice the water melon become smaller parts to make it easier for the mouth People's daily activities become routine Many ways people get up in the morning order of wearing shoes

Appearance (example of the slides)

Dekomposisi Masalah: Problem Decomposition

- Bagaimana bisa kita makan semangka yang besar? (padahal mulut kita lebih kecil)
- Bagaimana menyelesaikan masalah yang besar: dengan cara menjadikannya masalah-masalah kecil

Pengenalan Pola: Pattern Recognition

- Kegiatan kita setiap pagi merupakan suatu yang rutin atau berpola.
- Merupakan contoh pengenalan pola

Abstraction atau Abstraksi

- Cara bangun tidur bisa berbeda-beda.
- Perbedaan itu tidak masalah
- Menghilangkan perbedaan tersebut, semua Namanya sama yaitu bangun tidur.
- Abstraksi adalah menghilangkan detail untuk mendapatkan hal yang umum

Algoritma Berfikir atau Algorithm

- Bagaimana urutan memakai Sepatu?
 - Kita duduk
 - Ambil kaos kaki
 - Pakai kaos kaki
 - Ambil Sepatu
 - Pakai sepatu

Apakah urutan bisa diubah?

activities

Komponen CT I: Dekomposisi Masalah

- Aku tinggal di Yogyakarta. Saat liburan, aku ingin pergi ke rumah nenek di Semarang. Untuk ke Semarang bisa lewat Magelang tapi bisa juga lewat Klaten.
- Supaya lebih cepat, aku perlu menyampaikan ke ayah supaya lewat mana?

Mari Melihat Pengenalan Pola

Nomor	Fermasalahan	Jawaban
1	1,2,3,4,...	...
2	Satu, Dua, Tiga, Empat,...	...
3	▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲	...
4	● ● ▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲	...
5	●● ●●● ●●●● ●●●●● ●●●●●● ●●●●●●● ●●●●●●●● ●●●●●●●●● ●●●●●●●●●●	...

Selanjutnya...

- Kita akan belajar abstraksi

Mengendarai Sepeda

memanfaatkan abstraksi — menganggap sepeda sebagai alat yang bisa dikendalikan dengan kontrol sederhana tanpa harus memahami semua mekanismenya.

- Untuk bisa naik sepeda, kita tidak perlu tahu detail mengenai:
 - Warna sepeda, harga sepeda
 - Bagaimana memasang ban
 - Berapa berat sepeda
 - Siapa yang membuat sepeda?
 - Bagaimana memasang gigi atau gear

Apa yang akan kita pelajari hari ini?

- Memahami Input, Proses, Output
- Memahami cara memasukkan data ke dalam program Stracth
- Memahami cara menampilkan tampilan dalam program Scratch
- Membuat gim sederhana dengan Scratch

Examples of assessment

6 pertanyaan • 16 Poin Tunjukkan jawaban

1. PILIHAN GANDA • 30 detik • 2 poin Review Edit

Ketika kita memiliki masalah yang besar dan rumit, langkah pertama yang dapat kita lakukan agar masalahnya menjadi lebih mudah diselesaikan adalah

Dekomposisi masalah Pengenalan pola
 Abstraksi Algoritma

2. PILIHAN GANDA • 30 detik • 2 poin

Ketika menyusun langkah-langkah untuk menyelesaikan masalah yang kita hadapi, sebenarnya kita sedang melakukan

Dekomposisi masalah Pengenalan pola
 Abstraksi Algoritma

Aktivitas 7-2: Membuat program penjumlahan

5.	Susun blok untuk meminta pengguna memasukkan angka pertama.	
6.	Tambahkan blok untuk menyimpan angka pertama ke variabel angka1.	
7.	Tambahkan blok untuk: <ul style="list-style-type: none"> - meminta pengguna memasukkan angka kedua - menyimpan angka kedua ke variabel angka2. 	

21

ADDIE Implementation Stage
Implementation Schedule

Table 3. Implementation Schedule

No	Date	Activities	Note
1	July 28, 2025	General CT and Problem Decomposition	Students' activities, homework, and CT unplugged
2	August 4, 2025	Pattern recognition	Students' activities and CT unplugged
3	August 11, 2025	Abstraction	Students' activities and CT unplugged
4	August 25, 2025	Thinking algorithm part 1: algorithm and flowchart	Small project (scratch) and CT plugged
5	September 8, 2025	Thinking algorithm part 2: Scratch programming	CT plugged
6	September 15, 2025	Thinking algorithm part 3: implementation of CT for game development	CT plugged

Students' Responses while Learning Foundation of CT's

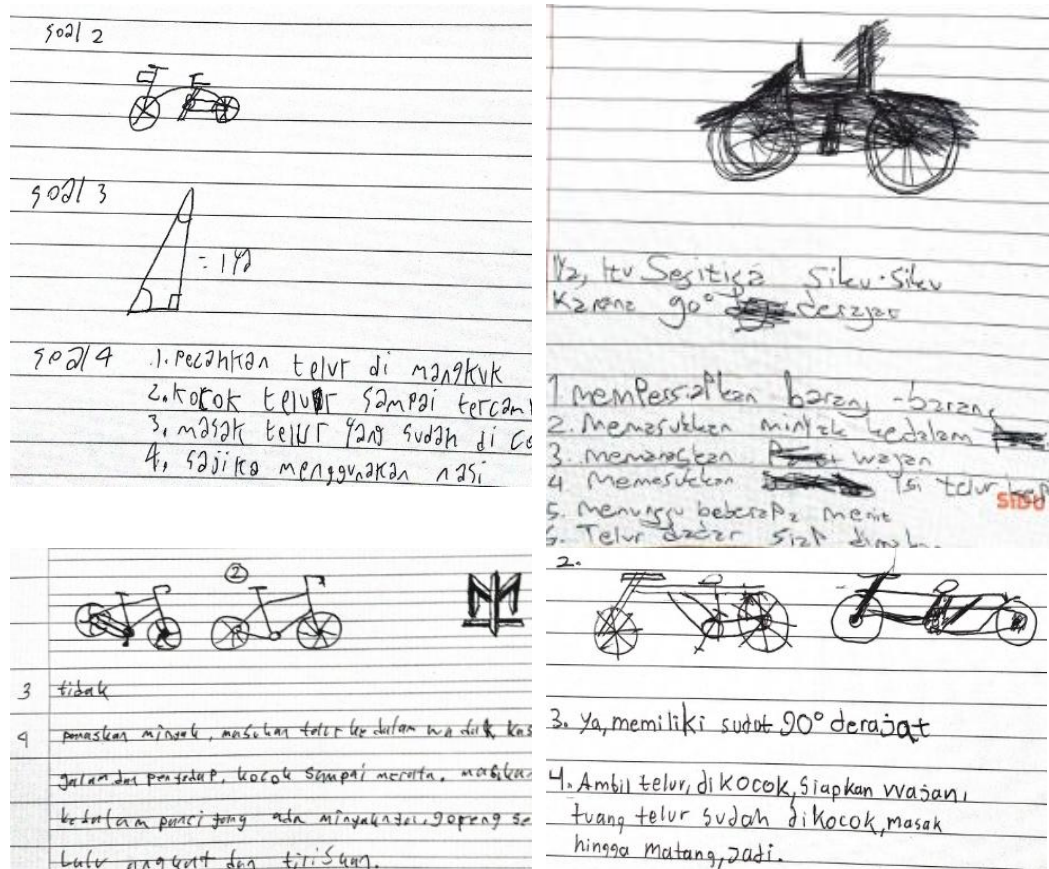


Figure 1. Examples of Students Responses

ADDIE Evaluation Stage

Table 4. The Analysis Stage

No	Practical Analysis	Comments or notes
1	Easy to use for teachers and students	Very practical
2	Feasible to implement in real classroom settings	practical
3	Efficient in terms of time and resources	Practical
4	Support the learning process without significant technical obstacles	Very practical

Conclusion

This study aimed to develop Computational Thinking learning media using the ADDIE instructional design model. The development process systematically followed the five stages of ADDIE, namely Analysis, Design, Development, Implementation, and Evaluation.

The results of the study indicate that the developed Computational Thinking learning media are valid, practical, and effective. Expert validation confirmed that the media met content accuracy, instructional design quality, and usability criteria. Practicality evaluation showed that the media were easy to use and feasible for classroom implementation. Furthermore, summative evaluation demonstrated that

the media effectively supported students' understanding of Computational Thinking concepts, problem-solving skills, collaboration, and communication through project-based learning activities.

These findings suggest that the ADDIE model is an appropriate and systematic framework for developing Computational Thinking learning media that address real classroom needs. The developed media provide meaningful learning experiences and support the development of 21st-century skills. Therefore, the Computational Thinking learning media developed in this study can be used as an alternative instructional resource and further adapted to different learning contexts.

References

- Czerkawski, B. C., & Lyman, E. W. (2015). Exploring issues about computational thinking in higher education. *TechTrends*, 59(2), 57 – 65. <https://doi.org/10.1007/s11528-015-0840-3>
- 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>
- Lee, I., Grover, S., Malyn-Smith, J., Pillai, S., et al. (2020). Computational thinking from a disciplinary perspective: Integrating computational thinking in K-12 science, technology, engineering, and mathematics education. *Journal of Science Education and Technology*, 29(1), 1-8. <https://doi.org/10.1007/s10956-019-09803-w>
- Lye, S. Y., & Koh, J. H. L. (2014). Review of research on computational thinking in education. *Computers in Human Behavior*, 41, 51–61. <https://doi.org/10.1016/j.chb.2014.09.012>
- Mohaghegh, M., & McCauley, M. (2016). Computational thinking: The skill set of the 21st century. *International Journal of Computer Science and Information Technologies*, 7(3), 1524–1530. <https://www.ijcsit.com/docs/Volume%207/vol7issue3/ijcsit20160703104.pdf>
- Shute, V. J., Sun, C., & Asbell-Clarke, J. (2017). Demystifying computational thinking. *Educational Research Review*, 22, 142–158. <https://doi.org/10.1016/j.edurev.2017.09.003>
- Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., & Wilensky, U. (2016). Defining computational thinking for mathematics and science classrooms. *Journal of Science Education and Technology*, 25(1), 127–147. <https://doi.org/10.1007/s10956-015-9581-5>
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33–35. <https://doi.org/10.1145/1118178.1118215>