TECHEDU: Jurnal Pendidikan Teknologi

Volume 1, Number 1, (2026), p. 25-31
Available online at:
https://ojs.ucp.ac.id/index.php/pti/index

Strengthening Computational Thinking in Information Technology Education: A Systematic Literature Review on the Development of Higher-Order Thinking Skills

Herianti[1]*, Ayu Widya Listari[2], Andi Nurqamal Ady Putra[3], Andi Gagah Purnama[4], Andi Hasri Tri Ulandari[5]
[1,2,3,4,5] **Universitas Cahaya Prima**
***Email: anthysuhardi@gmail.com**

ARTICLE INFO
Article History
Received: 19-02-2026
Accepted: 26-02-2026

Keywords:
computational thinking, higher-order thinking skills, information technology education, systematic literature review

ABSTRACT

The advancement of digital technology necessitates the strengthening of higher-order thinking skills in Information Technology Education (ITE). One approach considered particularly relevant is Computational Thinking (CT), which encompasses processes such as decomposition, pattern recognition, abstraction, and algorithm design. Although numerous studies have examined the implementation of CT in educational contexts, systematic reviews that integrate the relationship between CT and Higher-Order Thinking Skills (HOTS) within the specific context of ITE remain limited and fragmented. This study aims to synthesize the literature on the conceptual construction of CT, analyze its relationship with the development of HOTS, and identify pedagogical implications within ITE learning environments. The method employed is a Systematic Literature Review of national and international articles published between 2015 and 2026, selected through a structured screening process. The findings indicate that CT demonstrates structural alignment with the dimensions of HOTS, particularly in the domains of analysis, evaluation, and creation. However, the effectiveness of CT in enhancing HOTS is conditional and highly dependent on instructional design grounded in authentic problem-solving, explicit reflection, and assessment practices that evaluate thinking processes rather than merely final products. The novelty of this study lies in the integration of CT and HOTS perspectives within a unified conceptual framework specifically tailored to Information Technology Education. These findings provide both theoretical and practical foundations for the development of ITE learning models that are more analytical, reflective, and oriented toward strengthening higher-order thinking skills.

INTRODUCTION

Global digital transformation has reshaped the way individuals think, work, and solve problems across various domains of life. Amid rapid developments in information technology and artificial intelligence, Computational Thinking (CT) has emerged as a fundamental 21st-century competency (Wing, 2006; Grover & Pea, 2013; Kafai & Burke, 2015) . CT is not limited to programming skills; rather, it refers to a systematic mode of thinking that involves problem decomposition, pattern recognition, abstraction, and algorithm design. This capability is considered essential for navigating the complexity of modern digital systems. In addition, the integration of technology in learning environments can improve students' performance and engagement through digital platforms and interactive learning tools (Marsuki et al., 2025).

In educational contexts, CT has increasingly been positioned as a foundational component in informatics and information technology learning (Shute, Sun, & Asbell-Clarke, 2017). The integration of CT into curricula aims to equip learners with structured and logical problem-solving skills that are relevant not only to technological fields but also to other disciplines. In Indonesia, the promotion of computational thinking has begun to be adopted through the implementation of Informatics as a subject within the Merdeka Curriculum (Kemendikbudristek, 2022). This development indicates that CT has become part of the national strategic agenda for educational advancement.

Theoretically, Computational Thinking is closely associated with the development of Higher-Order Thinking Skills (HOTS), which include analysis, evaluation, and creation as described in the revised Bloom's Taxonomy (Anderson & Krathwohl, 2001). CT encourages students to analyze complex problems, generate alternative solutions, and design algorithms as forms of intellectual creation. Several studies have shown that CT-based learning contributes to the enhancement of higher-order thinking and problem-solving skills (Grover & Pea, 2013; Shute et al., 2017).

However, the implementation of CT in Information Technology Education (ITE) still faces several challenges. International studies indicate that CT integration often remains at a technical level, such as programming syntax exercises, without explicitly developing the reflective and evaluative dimensions central to HOTS (Brennan & Resnick, 2012; Weintrop et al., 2016). In Indonesia, several studies suggest that teachers' understanding of CT varies and has not been fully integrated into systematic instructional design (Rahmawati & Nugroho, 2022; Suryadi & Santoso, 2021). Moreover, the assessment of CT and HOTS frequently lacks standardized instruments (Lestari, 2023).

Most major studies on Computational Thinking focus on the effectiveness of specific program implementations or the development of particular instructional media (Grover & Pea, 2013; Shute et al., 2017). Meanwhile, research that specifically synthesizes the relationship between strengthening CT and developing Higher-Order Thinking Skills within the context of Information Technology Education remains relatively limited, particularly in Indonesia. Existing studies tend to be fragmented and have not comprehensively integrated higher-order thinking theory with the CT framework.

Although previous studies have explored Computational Thinking and Higher-Order Thinking Skills separately, limited research has systematically reviewed how CT contributes to the development of HOTS specifically in Information Technology Education. Therefore, this study provides a comprehensive synthesis of existing literature to clarify the relationship between CT dimensions and HOTS development within technology-based learning contexts.

This gap highlights the need for a systematic literature review capable of integrating empirical findings and conceptual perspectives regarding the relationship between CT and HOTS in ITE. Without a structured synthesis, the implementation of CT risks becoming merely a pedagogical trend lacking a strong theoretical foundation for developing higher-order thinking skills.

Based on this background, this study seeks to answer the following questions: How does the literature define and construct Computational Thinking within the context of technology education? What is the relationship between strengthening CT and the development of Higher-Order Thinking Skills as explained in previous research? What are the pedagogical implications of integrating CT into ITE learning to enhance HOTS?

This study aims to conduct a systematic literature review to synthesize the conceptual framework of Computational Thinking, analyze its relationship with Higher-Order Thinking Skills, and examine the pedagogical implications of empowering CT in ITE learning.

The novelty of this research lies in integrating Computational Thinking perspectives with Higher-Order Thinking Skills theory into a single conceptual synthesis specifically contextualized within Information Technology Education. Unlike previous studies that focus primarily on the technical implementation of CT, this study positions CT as a strategic approach to strengthening higher-order thinking skills within the pedagogical framework of ITE. Therefore, this research is expected to provide both theoretical and practical contributions to the development of more reflective, analytical, and creative information technology education.

METHOD

This study employs a Systematic Literature Review (SLR) approach to identify, examine, and systematically synthesize relevant research on strengthening Computational Thinking (CT) in Information Technology Education (ITE) and its relationship with the development of Higher-Order Thinking Skills (HOTS). The SLR approach was selected because it enables a structured, transparent, and replicable review process, resulting in a more comprehensive conceptual synthesis compared to conventional narrative reviews (Tranfield, Denyer, & Smart, 2003; Kitchenham & Charters, 2007). This method is also aligned with the study’s objective of integrating diverse theoretical perspectives and empirical findings into a unified analytical framework.

The literature search was conducted through several international and national academic databases to ensure representation of both global and Indonesian contexts. The databases included Scopus, Web of Science, Google Scholar, as well as nationally indexed journal portals such as SINTA and Garuda. The publication period was limited to 2015–2026 to capture recent developments in research on CT and HOTS, particularly following the growing global emphasis on computational thinking in education.

The search strategy utilized combinations of keywords in both English and Indonesian, including “computational thinking,” “higher-order thinking skills,” “computational thinking in education,” “computational thinking in computer science education,” “HOTS in informatics,” and “Information Technology Education.” These keywords were combined using Boolean operators (AND, OR) to broaden and refine the search results in accordance with the study’s objective, namely to identify literature explicitly addressing the relationship between CT and HOTS within the context of information technology learning.

The inclusion criteria comprised peer-reviewed scholarly articles published within the specified timeframe that explicitly addressed one or more of the following three aspects: (1) definitions and conceptual frameworks of Computational Thinking; (2) the development of Higher-Order Thinking Skills in technology- or informatics-based learning; and (3) the relationship or integration between CT and HOTS in educational practice. Articles that focused solely on technical programming without addressing computational thinking or higher-order thinking dimensions were excluded. Popular opinion pieces and non-academic reports were also excluded from the analysis.

The literature selection process was conducted in stages, following the principles of transparency and accountability recommended by the PRISMA guidelines (Moher et al., 2009). The first stage involved identifying all articles retrieved through keyword searches in the selected databases. The second stage consisted of screening titles and abstracts to assess their alignment with the research focus. The third stage involved full-text review of articles that met the preliminary criteria to ensure substantive relevance to the study's objectives.

Data analysis was conducted using a thematic analysis approach aligned with the research questions. The selected articles were classified into three main categories. The first category concerned the conceptual construction of Computational Thinking, including its definitions, dimensions (decomposition, pattern recognition, abstraction, and algorithm design), and models of implementation in education. The second category focused on the development of Higher-Order Thinking Skills within technology-based learning contexts. The third category examined the integration or relationship between CT and HOTS in instructional practice.

During the synthesis process, cross-study comparisons were conducted to identify consistent patterns, variations in approaches, and existing research gaps. The analysis was not merely descriptive but also critical, taking into account research contexts, methodological designs, and the limitations of each study. Through this approach, the study aims to produce a comprehensive conceptual synthesis explaining how strengthening Computational Thinking can contribute to the development of Higher-Order Thinking Skills in Information Technology Education.

Through this methodological procedure, the study seeks to establish a strong theoretical foundation for the development of ITE learning models oriented toward the systematic and sustainable enhancement of higher-order thinking skills.

RESULT AND DISCUSSION

The synthesis of the literature indicates that Computational Thinking (CT) in Information Technology Education (ITE) is constructed as a systematic cognitive framework that serves as a foundation for the development of higher-order cognitive abilities. Nearly all major studies refer to the four core components of CT—decomposition, pattern recognition, abstraction, and algorithm design (Wing, 2006; Grover & Pea, 2013). However, a key finding of this study is that the literature does not merely position CT as a technical computational skill, but rather as a cognitive mechanism that supports analytical and creative thinking processes.

Conceptual Construction of CT and Its Relevance to HOTS

From the perspective of the revised Bloom's Taxonomy (Anderson & Krathwohl, 2001), Higher-Order Thinking Skills (HOTS) encompass the three highest cognitive levels: analysis, evaluation, and creation. The literature synthesis reveals structural alignment between CT dimensions and HOTS.

Problem decomposition in CT directly relates to analytical ability, as students must break down complex systems into simpler, structured components. This process requires identifying variables, relationships, and system constraints. Pattern recognition and abstraction are closely associated with evaluation skills, as students must compare representations of solutions, assess efficiency, and consider complexity without losing essential meaning. Meanwhile, algorithm design represents a concrete form of creation, as students construct systematic procedures to solve problems.

These findings suggest that, theoretically, CT is not only compatible with HOTS but inherently embodies higher-order thinking elements. In other words, CT can be positioned as a pedagogical pathway for developing HOTS within ITE.

However, the literature also indicates that this relationship does not occur automatically in instructional practice. Without explicit instructional design that integrates CT with reflection and evaluation, students tend to remain at a procedural level of understanding.

Patterns of CT Implementation in ITE

The analysis identifies three main patterns of CT implementation in ITE. First, the procedural-technical approach. In this model, CT is taught primarily through programming exercises focused on syntax and basic logic. Although students learn to construct algorithms, instruction often does not require analysis of alternative solutions or evaluation of efficiency. This approach tends to develop technical skills but does not maximally strengthen HOTS.

Second, the problem-based and project-based learning approach. The literature shows that when CT is integrated into project-based learning or case studies, significant improvements occur in students' analytical and reflective abilities (Shute et al., 2017). In the ITE context, projects involving application development or system simulations enable students to decompose real-world problems, generate solutions, and design optimal algorithms.

Third, the reflective-critical approach. Although less frequently implemented, this approach demonstrates the strongest impact on HOTS. In this model, students are not only required to solve problems but also to explain the rationale behind their solutions, compare alternative approaches, and reflect on the system they have developed. This approach aligns with critical digital literacy and extends CT into evaluative domains.

These findings indicate that the effectiveness of CT in enhancing HOTS is largely determined by the pedagogical strategies employed.

Moderating Factors: Teachers, Assessment, and Curriculum

The literature synthesis also identifies several moderating factors influencing the relationship between CT and HOTS.

First, teacher competence. Several national studies indicate that teachers' understanding of CT varies (Rahmawati & Nugroho, 2022). When teachers interpret CT merely as coding, the integration of HOTS becomes limited. Conversely, teachers who conceptualize CT as a cognitive framework tend to design more reflective and analytical learning activities.

Second, assessment. The measurement of CT and HOTS is often separated. Many assessment instruments evaluate only the final product of programming tasks rather than students' thinking processes. However, HOTS develops through analysis and evaluation processes, not solely through final outcomes. The absence of integrated assessment instruments poses a barrier to comprehensively strengthening CT.

Third, curriculum structure. Content-heavy curricula that emphasize completion of material tend to limit the deep exploration required for HOTS development. Effective CT implementation requires space for exploration, reflection, and iterative problem-solving.

Based on the synthesis of the literature, the relationship between Computational Thinking (CT) and Higher-Order Thinking Skills (HOTS) can be understood as mediational and conditional. CT provides a structured cognitive framework through decomposition, pattern recognition, abstraction, and algorithm design, which potentially supports the development of analytical, evaluative, and creative abilities. However, the strengthening of HOTS does not occur automatically through CT instruction alone.

Higher-order thinking emerges when CT is implemented within authentic problem-based learning contexts that demand deep exploration of solutions, accompanied by explicit reflection and evaluation of strategies used. Moreover, the effectiveness of CT in developing HOTS is strongly influenced by the assessment system applied. If evaluation focuses solely on final products—such as whether a program runs without errors—the analytical and evaluative dimensions of students' thinking remain under-assessed. In contrast, assessments that examine reasoning processes, justification of solutions, and comparison of alternative algorithms are better able to capture HOTS development.

Thus, CT can be positioned as an operational means for strengthening HOTS in Information Technology Education, but its effectiveness depends heavily on the quality of pedagogical design and evaluation systems that systematically support reflective and critical thinking processes.

CONCLUSION

This systematic literature review demonstrates that Computational Thinking (CT) has a strong conceptual relationship with the development of Higher-Order Thinking Skills (HOTS) in Information Technology Education. The dimensions of CT—decomposition, pattern recognition, abstraction, and algorithm design—are structurally aligned with the cognitive processes of analysis, evaluation, and creation in the revised Bloom's Taxonomy. Therefore, CT has the potential to function as an operational framework for strengthening higher-order thinking skills among learners.

However, the synthesis also confirms that strengthening HOTS through CT does not occur automatically. Its effectiveness depends significantly on the pedagogical design employed. Authentic problem-based learning, explicit reflective activities, and assessment systems that evaluate thinking processes are key factors in optimizing CT's contribution to HOTS. Without such integration, CT implementation tends to remain limited to procedural mastery and technical skills.

The primary implication of this study is the need to develop ITE learning models that position CT not merely as technical content but as a cognitive strategy to promote deep analysis, evaluation of alternative solutions, and reflective algorithmic creation. Through a systematic and integrated approach, CT can serve as a crucial foundation for cultivating higher-order thinking competencies relevant to the demands of the digital era.

REFERENCES

Anderson, L. W., & Krathwohl, D. R. (2001). *A taxonomy for learning, teaching, and assessing: A revision of Bloom's taxonomy of educational objectives*. Longman.

Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. In *Proceedings of the 2012 annual meeting of the American Educational Research Association (AERA)*.

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

Kafai, Y. B., & Burke, Q. (2015). Constructionist gaming: Understanding the benefits of making games for learning. *Educational Psychologist*, *50*(4), 313–334. https://doi.org/10.1080/00461520.2015.1124022

Kemendikbudristek. (2022). *Kurikulum Merdeka dan implementasi mata pelajaran Informatika*. Kementerian Pendidikan, Kebudayaan, Riset, dan Teknologi Republik Indonesia.

Kitchenham, B., & Charters, S. (2007). *Guidelines for performing systematic literature reviews in software engineering*. EBSE Technical Report. Keele University & Durham University.

Lestari, D. (2023). Pengembangan instrumen asesmen computational thinking dalam pembelajaran informatika. *Jurnal Pendidikan Teknologi Informasi*, 7(2), 115–128.

Marsuki, A., Risal, A. W., Aprilo, I., Sulfa, M., & Ali, F. A. (2025). Improving Athlete Performance Through Technology-Based Sports Learning Innovation. *Jurnal Ilmu Keolahragaan*, *1*(1), 27–35. https://doi.org/10.65943/02dw5210

Moher, D., Liberati, A., Tetzlaff, J., & Altman, D. G. (2009). Preferred reporting items for systematic reviews and meta-analyses: The PRISMA statement. *PLoS Medicine, 6*(7), e1000097. https://doi.org/10.1371/journal.pmed.1000097

Rahmawati, Y., & Nugroho, A. (2022). Integrasi computational thinking dalam pembelajaran informatika di sekolah menengah. *Jurnal Pendidikan Informatika Indonesia, 6*(1), 45–56.

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

Suryadi, B., & Santoso, H. (2021). Tantangan implementasi computational thinking dalam kurikulum informatika di Indonesia. *Jurnal Teknologi Pendidikan*, 23(3), 201–214.

Tranfield, D., Denyer, D., & Smart, P. (2003). Towards a methodology for developing evidence-informed management knowledge by means of systematic review. *British Journal of Management, 14*(3), 207–222. https://doi.org/10.1111/1467-8551.00375

Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & 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