



SCRATCH-BASED INTERACTIVE E-MODULES FOR PROMOTING JUNIOR HIGH SCHOOL STUDENTS' COMPUTATIONAL THINKING SKILLS: A SYSTEMATIC LITERATURE REVIEW

Azzura Qatrunnada¹, Roemintoyo², Budi Tri Cahyono³

^{1,2,3}Universitas Sebelas Maret, Kota Surakarta, Jawa Tengah, Indonesia

Article Info

Keywords:

Computational Thinking,
Interactive E-Module,
Junior High School Students,
Scratch,
Systematic Literature Review,
Visual Programming.

ABSTRACT

This study synthesizes recent evidence on Scratch-based interactive e-modules for supporting junior high school students' computational thinking skills. Using a systematic literature review approach, 20 studies published between 2021 and 2026 were selected through a PRISMA-style screening process and analyzed descriptively and thematically. The findings indicate that Scratch-based learning is most effective when supported by scaffolding, project- or problem-based learning, integration with mathematics or science contexts, and explicit computational thinking assessment. Several Indonesian studies also report that Scratch-assisted e-modules are generally valid, practical, and potentially effective for supporting computational thinking development, although most studies remain limited to small-scale developmental research. This review suggests that interactive e-modules should be designed as contextualized, project-oriented, and scaffolded learning environments that encourage problem solving, reflection, and debugging practices in junior high school learning contexts.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Azzura Qatrunnada
Universitas Sebelas Maret
azzuraqatrunnada@student.uns.ac.id

1. INTRODUCTION

Computational thinking is increasingly recognized as a core competency in 21st-century education, not only in informatics learning but also in problem solving, algorithmic reasoning, abstraction, and situation modeling across various subject areas [1]. Recent studies indicate that the integration of computational thinking into science, mathematics, programming, and interdisciplinary learning contexts has expanded rapidly; however, instructional approaches, assessment practices, and implementation strategies remain highly diverse across educational settings [2]. Computational thinking education has also developed more extensively at the elementary and general K-12 levels than in instructional contexts specifically designed for junior high school students using structured digital learning materials.

The junior high school level represents an important developmental stage for computational thinking instruction because students begin to demonstrate more systematic reasoning, abstract thinking, debugging ability, and algorithmic problem-solving skills. At this stage, students are generally more prepared to engage in structured coding activities, project-oriented learning, and digital problem-solving tasks compared with earlier educational levels. Therefore, junior high school provides a strategically important context for introducing computational thinking through scaffolded and contextualized learning materials.

Among various educational technologies, Scratch has become one of the most widely used tools for supporting computational thinking development in primary and secondary education due to its visual programming environment, low syntactic barriers, and flexibility for project-based and game-based learning activities [3]. Several systematic reviews identify Scratch as a dominant platform for introducing computational thinking concepts such as sequencing, decomposition, debugging, conditionals, and algorithmic reasoning in school settings [4]. However, recent studies also emphasize that Scratch does not automatically improve computational thinking skills. The effectiveness of Scratch-based learning depends heavily on instructional design quality, scaffolding strategies, feedback mechanisms, assessment practices, and the contextual relevance of learning activities. Without appropriate pedagogical support, Scratch activities may remain procedural and fail to promote deeper computational thinking processes [5].

In this context, interactive e-modules may play an important role as structured digital learning materials that organize coding activities progressively and systematically. Interactive e-modules can provide guided exercises, project examples, scaffolding features, reflective prompts, debugging activities, adaptive feedback, and computational thinking assessments within a single learning environment. Such features are particularly relevant for junior high school students, who often require gradual instructional support when learning programming concepts and computational problem-solving strategies.

In the Indonesian context, several studies have reported that Scratch-assisted learning materials, science games, coding modules, and project-based e-modules show promising potential for supporting computational thinking development. Nevertheless, evidence directly combining Scratch, interactive e-modules, and junior high school students remains limited and fragmented. Existing studies are still distributed across different educational levels, instructional designs, and subject contexts. Many studies focus on elementary education, general K-12 programming activities, or small-scale product development research, while relatively few studies specifically examine Scratch-based interactive e-modules designed for junior high school computational thinking instruction.

This article does not report an experimental intervention or the development of a new instructional product. Instead, it presents a systematic literature review synthesizing empirical and conceptual studies related to Scratch, interactive e-modules, and computational thinking in primary and secondary education contexts, particularly those relevant to junior high school students. This review aims to identify major instructional patterns, analyze the consistency of findings across studies, and highlight practical implications for instructional design.

Therefore, this article contributes to the literature by synthesizing recent evidence published between 2021 and 2026 and by formulating evidence-based design recommendations for Scratch-based interactive e-modules intended to support computational thinking development among junior high school students. Based on this objective, the review addresses four research questions: (1) what are the characteristics of studies concerning Scratch, visual programming, interactive e-modules, and computational thinking in primary and secondary education; (2) how consistent is the evidence regarding Scratch-based learning and computational thinking outcomes; (3) which instructional design components are most frequently associated with positive learning outcomes; and (4) what practical implications and future research directions can be identified for developing Scratch-based interactive e-modules in junior high school contexts.

2. RESEARCH METHODS

This study employed a systematic literature review (SLR) approach to synthesize empirical and conceptual studies related to Scratch, interactive e-modules, and computational thinking in primary and secondary education contexts. This article does not report experimental research, product development, or direct classroom intervention. Instead, it focuses on identifying patterns of findings, instructional design characteristics, and research gaps across previously published studies.

The review process adopted a PRISMA-style selection flow to improve transparency in article identification, screening, eligibility assessment, and inclusion procedures. However, this study did not implement the full PRISMA protocol commonly used in medical systematic reviews; rather, it adapted the general PRISMA framework to educational research contexts.

The literature search was conducted using Scopus, ERIC, DOAJ, Google Scholar, Garuda, and official journal or publisher websites. The search focused primarily on article titles, abstracts, and keywords to identify studies related to Scratch, visual programming, interactive e-modules, and computational thinking. The publication period covered studies published between 2021 and 2025, while several studies labeled as 2026 publications were included because they were already available as online first or early access articles at the time of review.

The selection primarily prioritized peer-reviewed journal articles relevant to primary and secondary education, particularly junior high school or equivalent age levels. However, studies at the elementary, K-12, and systematic review levels were also included when they contributed directly to discussions concerning Scratch-based learning design, interactive e-modules, or computational thinking assessment [2].

The core search strings used in this review were as follows:

English:

("Scratch" OR "block-based programming" OR "visual programming" OR "e-module" OR "digital module") AND ("computational thinking" OR "algorithmic thinking") AND ("middle school" OR "junior high school" OR "secondary school" OR "K-12") AND (2021 OR 2022 OR 2023 OR 2024 OR 2025 OR 2026)

The inclusion and exclusion criteria applied in this review are presented in Table 1.

Table 1. Inclusion and Exclusion Criteria.

Aspect	Inclusion Criteria	Exclusion Criteria
Document Type	Peer-reviewed journal articles	Conference proceedings, undergraduate theses, dissertations, books, book chapters
Publication Period	2021-2026	Before 2021
Topic	Scratch, block-based/visual programming, interactive e-modules, and computational thinking	Technical articles that do not report educational implications
Educational Level	Priority on junior high school; elementary/K-12 included if conceptually relevant	Purely higher education contexts without K-12 implications
Outcome Variables	Computational thinking, self-efficacy, problem solving, reflective thinking, motivation, instructional material validity/practicality	Platform descriptions without learning outcomes
Metadata Accessibility	Includes minimum information on methods, sample, and findings	Insufficient or duplicative metadata

The screening process was conducted in three stages. First, duplicate records were removed manually and through database filtering. Second, titles and abstracts were screened to assess relevance to Scratch, computational thinking, interactive e-modules, and school-level education contexts. Third, full-text articles were reviewed to evaluate methodological clarity, educational relevance, outcome reporting, and alignment with the objectives of this review. Articles were excluded when they focused solely on higher education contexts, lacked sufficient methodological information, or did not report educational implications related to computational thinking instruction.

The screening process yielded 86 initial records, of which 18 duplicates were removed. Subsequently, 68 unique articles were screened at the title and abstract stage, 34 articles underwent full-text review, and 20 studies were ultimately included in the final synthesis. The following diagram summarizes the selection process

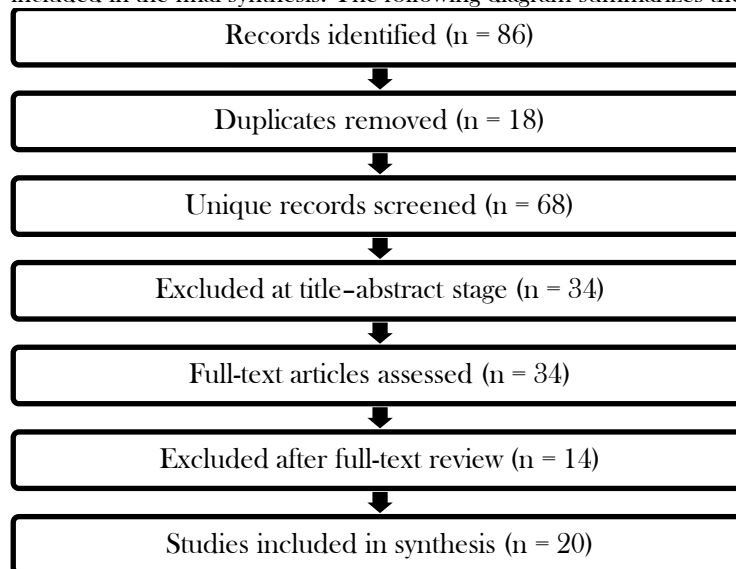


Figure 1. PRISMA-Style Study Selection Flow

The primary reasons for exclusion at the full-text stage included: (1) insufficient alignment with Scratch or Scratch-based e-modules, (2) higher education contexts lacking clear implications for junior high school or K-12 education, and (3) outcome metadata too limited to allow responsible data extraction. Data extracted from each included study covered country of origin, research design, sample/educational level, intervention focus, main findings, and study limitations. The synthesis was conducted descriptively and thematically, complemented by a simple quality appraisal based on five aspects: clarity of research design, sample adequacy, instrument transparency, appropriateness of analytical methods, and reporting of study limitations [7].

3. RESULT AND ANALYSIS

The findings of this review are organized according to the four research questions presented in the Introduction. The first section describes the characteristics of the included studies, followed by evidence concerning Scratch-based learning outcomes, instructional design patterns associated with positive computational thinking outcomes, and research gaps related to Scratch-based interactive e-modules for junior high school students.

A total of 20 studies published between 2021 and early-access 2026 were included in the final synthesis. The publication trend increased gradually between 2021 and 2025, indicating growing academic interest in Scratch, computational thinking, and digital instructional materials in school education contexts.

Based on research focus, the included studies consisted of:

- a. 11 studies on Scratch or block-based programming interventions,
- b. 3 studies specifically related to e-modules or digital instructional materials,
- c. and 6 review, meta-analysis, or scoping review studies.

Geographically, the studies were dominated by Asian countries, particularly Indonesia, China, and Turkey, while several review studies synthesized global datasets. In terms of methodology, quasi-experimental studies, development studies, systematic reviews, and mixed-method approaches were the most common designs. Most studies focused on elementary and general K-12 education contexts, whereas relatively few studies specifically examined junior high school students.

The most frequently measured outcomes included computational thinking performance, problem-solving ability, programming self-efficacy, reflective thinking, motivation, and instructional material validity or practicality. Overall, the evidence concerning Scratch-based learning in general was substantially stronger and more extensive than evidence specifically addressing Scratch-based interactive e-modules for junior high school students.

Table 2. Included Studies

No.	Author(s)	Year	Journal	Country	Method	Sample/Level	Key Findings
1	Montiel & Gomez-Zermeño [4]	2021	Computers	Global	Systematic literature review	30 K-12 articles	Scratch is widely used to promote computational thinking in K-12 education and stands out due to its innovative characteristics; limitation: reliance on a single database and study coverage only up to May 2020.
2	Jiang & Li [1]	2021	Journal of Computers in Education	China	Quasi-experiment, one-group pretest-posttest	336 Grade 5 students	Scratch improved creativity, cooperation, and critical thinking, but did not significantly improve problem-solving and algorithmic thinking; limitation: no control group and conducted in one rural school.
3	Liu & Cavanaugh [6]	2021	Educational Technology	China	Mixed-method pretest-posttest	63 Grade 5 students	Problem-solving-based Scratch

			Research and Development		nonequivalent group design		courses improved computational thinking and self-efficacy, especially among female students; limitation: small sample and non-equivalent groups.
4	Harahap Eliza [2]	& 2022	Jurnal Obsesi	Indonesia	ADDIE R&D	Early childhood learners; coding module	Coding e-module improved computational thinking learning effectiveness from 55% pretest to 88% posttest; limitation: not junior high school context and not specifically Scratch-based.
5	Katchapakirin, Anutariya, & Supnithi [7]	2022	Education and Information Technologies	Thailand	Design and development + school implementation	Elementary students across multiple schools	ScratchThAI, equipped with chatbot support, automated assessment, and dashboards, improved teacher satisfaction, learning performance, and engagement; limitation: technological and curricular context differs from Indonesia.
6	Wang, Shen, & Chao [8]	2022	International Journal of Science and Mathematics Education	Global	Literature review	55 empirical STEM studies	Computational thinking integration most frequently appears in science and mathematics; interdisciplinary designs show strong potential but depend heavily on pedagogy; limitation: not focused specifically on Scratch or e-modules.
7	Tikva & Tambouris [9]	2023	Education and Information Technologies	Greece	Experimental study	57 Grade 7-9 students	Programming games with scaffolding produced higher computational thinking gains than non-scaffolded versions; students with initially low programming attitudes benefited substantially; limitation: only 3

8	Gökçe & Aydoğan Yenmez [10]	2023	Education and Information Technologies	Turkey	Sequential exploratory mixed methods	524 students and 8 teachers	weeks and one school. Scratch learning strengthened reflective thinking toward problem-solving and computational thinking; limitation: mainly elementary-level, not specifically junior high school.
9	Koray & Bilgin [11]	2023	Science Insights Education Frontiers	Turkey	Quasi-experiment	Grade 6 / middle school students	Scratch activities integrated into the 5E science learning model improved computational thinking and programming self-efficacy; limitation: one science topic and one grade level.
10	Yılmaz & İzmirli [12]	2023	Journal of Educational Technology and Online Learning	Turkey	Experimental pretest-posttest control group	Secondary school students	Both unplugged and plugged coding improved computational thinking performance; the Scratch-based plugged group improved performance, though computational thinking perceptions changed little; limitation: no e-module intervention tested.
11	Bilgic & Dogusoy [13]	2023	Education and Information Technologies	Turkey	Action research	Grade 5-6 students, 14 weeks	Block-based programming activities enriched computational thinking experiences and improved several attitudinal dimensions; limitation: local context, action research design, and transition-level participants between elementary and junior high school.
12	Findayani, Suparta, & Sariyasa [5]	2023	Jurnal Pendidikan MIPA	Indonesia	R&D	15 Grade VIII-A and 32 Grade VIII-E junior high	Project-based Scratch-assisted statistics e-module was validated as

						school students	valid, highly practical, and effective, with a problem-solving gain score of 68.93; limitation: one school and focus on mathematical computational thinking.
13	Ibrohim, Siregar, & Chaeruman [14]	2023	Al-Ishlah: Jurnal Pendidikan	Indonesia	Meta-analysis	22 datasets from 14 quantitative studies	Scratch produced significant effects on students' computational thinking, particularly in computational concepts and practices; limitation: primary focus on elementary education, not junior high school.
14	Rao & Bhagat [3]	2024	Educational Technology Research and Development	Global	Systematic review	360 articles	The review demonstrated major advances in computational thinking research, diverse tools, pedagogical strategies, and assessments; however, implementation in non-STEM fields remains limited; limitation: database restricted to Web of Science.
15	Wulandari, Listaji, Sulaiman, Wan Jaafar, & Abdul Rahim [15]	2024	Jurnal Pendidikan Sains Indonesia	Indonesia	Development study	Secondary school / junior high school students	Scratch-based STEM science games were valid and capable of fostering computational thinking among secondary students; limitation: emphasized product validity more than long-term comparative effectiveness.
16	Irawan, Rosjanuardi, & Prabawanto [16]	2024	JTAM	Indonesia / Global dataset	Systematic review	88 Scopus-indexed articles	Scratch was identified as the most widely used tool for developing computational thinking across educational levels; limitation: limited to Scopus articles

17	Woo & Falloon [17]	2025	Computer Science Education	Global	Scoping review	Middle school integrated coding studies	and broad cross-level scope. Cross-curricular coding at the middle school level is promising, but learning quality is strongly influenced by teacher knowledge, pedagogy, and school ecosystem; limitation: number of included studies was not clearly reported in the accessible source.
18	Rohim & Wicaksono [18]	2025	Jurnal Penelitian dan Penilaian Pendidikan	Indonesia	ASSURE R&D	23 Grade VII junior high school students	Scratch-based visual programming e-module received very high expert validation and positive student responses; limitation: small scale and no comparison group.
19	Fang, Ng, & Yuen [19]	2025	Education and Information Technologies	Hong Kong	Comparative intervention	72 Grade 5 students	Scratch combined with GeoGebra improved motivation, perceived ability, and reduced cognitive load compared with Scratch alone; limitation: lower educational level than junior high school and focused on elementary mathematics.
20	Zhang, Li, & Zhang [20]	2026	Educational Technology Research and Development	China	Quasi-experimental mixed methods	31 Grade VII students	Scratch-based probability modules improved computational thinking concepts, computational thinking perspectives, and mathematics achievement, with moderate gender differences but no significant differences based on prior programming experience; limitation: small sample and highly specific topic.

Thematic Synthesis

The strongest finding from the 20 selected studies is that Scratch is most effective when positioned as a constructionist learning environment that enables students to create artifacts, rather than merely follow procedural exercises. In the study by Jiang and Li, improvements were observed in creativity, cooperativeness, and critical thinking, but not automatically across all dimensions of computational thinking (CT). A similar pattern emerged in the studies by Liu and Cavanaugh, Tikva and Tambouris, and Zhang, Li, and Zhang; more stable learning outcomes were achieved when Scratch was framed as a problem-solving tool, embedded in mathematics tasks, or implemented through modules with clear scaffolding structures. The conclusion is that Scratch itself is not a “magic solution”; rather, it becomes effective when instructional design guides students through problem decomposition, algorithm design, debugging, reflection, and artifact revision [1].

The second theme is the importance of scaffolding. The ScratchThAI study demonstrated that dashboards, chatbots, automated assessment, and adaptive tasks help teachers and students overcome limitations in classroom time and address heterogeneous student abilities. The study by Tikva and Tambouris reinforced this finding by showing that scaffolded programming games were statistically superior to programming games without scaffolding features [9]. Likewise, Fang, Ng, and Yuen found that combining Scratch with additional visual support tools reduced students’ cognitive load. From an interactive e-module design perspective, these findings are highly significant: Scratch-based e-modules should not merely consist of instructional content and exercises, but should also incorporate worked examples, graduated hints, reflective checkpoints, and feedback addressing common errors [19].

The third theme is contextual integration with subject matter. Koray and Bilgin demonstrated that Scratch integrated into the 5E instructional model in science education improved both computational thinking and programming self-efficacy, while Zhang, Li, and Zhang showed effective integration within a Grade VII probability module. Wulandari and colleagues further demonstrated that STEM-based science games developed with Scratch have the potential to cultivate computational thinking among secondary school students [20]. This synthesis aligns with the review by Wang, Shen, and Chao, which identified science and mathematics as the most productive domains for computational thinking integration. The implication for this study is clear: in the junior high school context, Scratch-based e-modules will be more powerful when positioned not as isolated “coding lessons,” but as problem-solving tools for concrete mathematics or science topics [8].

The fourth theme concerns the position of interactive e-modules. Evidence from Indonesia suggests a positive direction, although the empirical base remains relatively limited. Findayani et al. found that a Scratch-assisted project-based statistics e-module for Grade VIII students was valid, practical, and effective in improving problem-solving skills. Rohim and Wicaksono reported high expert validation and positive student responses toward a Scratch-based visual programming e-module for Grade VII students. Harahap and Eliza, although working at a different educational level, likewise confirmed that digital coding instructional materials can be effective when systematically designed. However, most Indonesian studies are primarily product development studies, involve small samples, and rarely employ comparative control groups or medium-term follow-up evaluations. Therefore, while there is sufficient empirical grounding to suggest effectiveness for journal publication purposes, such claims should be expressed cautiously as “promising” or “potentially effective”, rather than as universally conclusive findings [18].

The fifth theme is the identification of research gaps. Recent reviews indicate that Scratch remains the most dominant tool for developing computational thinking; however, the distribution of studies remains heavily concentrated in elementary education and general K-12 contexts. The meta-analysis by Ibrohim et al. likewise confirmed the predominance of evidence at the elementary school level. Meanwhile, the scoping review by Woo and Falloon, which specifically focused on middle school, highlighted issues related more to pedagogical strengthening and teacher readiness than to the design of contextualized interactive e-modules. This reveals a research gap that is highly relevant to the focus of this thesis: there remains substantial room for a systematic literature review specifically centered on the design of Scratch-based interactive e-modules for junior high school students, followed by future multi-site empirical testing in Indonesia [16].

Recent studies further reinforce this argument by emphasizing that Scratch-based learning environments become more meaningful when students are engaged in contextualized and project-oriented activities rather than isolated coding drills. Berlian et al. reported that Scratch-supported learning media contributed more effectively to computational thinking and scientific literacy when accompanied by structured pedagogical organization and interactive learning tasks. Similarly, Handayani et al. found that problem-based e-modules assisted with Scratch helped students engage more actively in computational problem solving through guided and progressively organized activities. These findings strengthen the view that Scratch should be understood primarily as a pedagogical environment requiring appropriate instructional support rather than as an automatically effective technological tool [21], [22].

Additional evidence regarding scaffolding was reported by Handayani et al., who observed that Scratch-assisted e-modules became more effective when learning activities were organized through staged guidance, worked examples, and structured problem-solving tasks. These findings support the argument that junior high

school students require gradual instructional assistance when learning computational concepts, debugging strategies, and algorithmic reasoning processes. Consequently, interactive e-modules designed for Scratch learning should integrate adaptive support features rather than relying solely on procedural coding instructions [22].

Further evidence supporting contextual integration was provided by Setiawan et al., who developed a Scratch-assisted project-based geometry e-module that encouraged students to apply computational thinking within mathematical problem-solving activities. Likewise, Yusrina et al. emphasized that Scratch-integrated mathematics instruction became more effective when computational thinking activities were embedded systematically within contextual and scaffolded learning sequences. These findings reinforce the argument that Scratch-based learning environments are more meaningful when connected directly to authentic mathematics or science contexts relevant to junior high school students [23], [24].

Recent scoping reviews also indicate that computational thinking research remains heavily concentrated at the elementary and general K-12 levels, while studies specifically focusing on contextualized instructional designs for junior high school students remain comparatively limited. Santaengracia et al. highlighted that middle-school-specific computational thinking interventions are still underrepresented in the literature, particularly those integrating structured digital learning materials and scaffolded instructional design. This further strengthens the relevance of systematic reviews focusing specifically on Scratch-based interactive e-modules for junior high school contexts [25].

Summary of Study Quality Assessment

A quality assessment was conducted based on five criteria: clarity of research design, adequacy of sample size or article corpus, transparency of instruments/measures, appropriateness of analytical methods, and reporting of study limitations. The maximum score was 5. This assessment was intended as a practical appraisal, developed based on information available in the full articles or accessible abstracts.

Table 3. Summary of Study Quality Assessment

No.	Study	Score	Category	Main Risk of Bias
1	Montiel & Gomez-Zermeño	4	High	Strong review, but limited database coverage
2	Jiang & Li	4	High	No control group; single rural context
3	Liu & Cavanaugh	4	High	Moderate sample size; nonequivalent design
4	Harahap & Eliza	3	Moderate	Educational level not aligned with junior high school; localized R&D context
5	Katchapakirin et al.	4	High	Specific digital ecosystem context in Thailand
6	Wang et al.	4	High	Broad review, but not specifically focused on Scratch
7	Tikva & Tambouris	4	High	Short intervention duration; single school
8	Gökçe & Aydoğan Yenmez	4	High	Greater emphasis on elementary school level
9	Koray & Bilgin	3	Moderate	Specific science topic; limited generalizability
10	Yılmaz & İzmirli	4	High	Did not examine interactive e-modules
11	Bilgic & Dogusoy	3	Moderate	Action research design; localized context
12	Findayani et al.	3	Moderate	Single-school study; focus on mathematical computational thinking
13	Ibrohim et al.	4	High	Strong meta-analysis, but predominantly elementary-level studies
14	Rao & Bhagat	4	High	Single database source: Web of Science

15	Wulandari et al.	3	Moderate	Development study; not longitudinal
16	Irawan et al.	4	High	Large article corpus, but limited to Scopus
17	Woo & Falloon	4	High	Limited corpus details in accessible metadata
18	Rohim & Wicaksono	3	Moderate	Small sample size; no control group
19	Fang et al.	4	High	Educational level below junior high school
20	Zhang et al.	4	High	Small sample size; highly specific probability topic

Overall, the available evidence is sufficiently robust to support an argumentative and analytical systematic literature review article. However, eight studies present a moderate risk of bias due to small-scale samples, single-site developmental designs, or educational levels not fully equivalent to junior high school. This suggests that your article will be stronger if it emphasizes critical synthesis and instructional design recommendations, rather than making overly broad causal claims regarding effectiveness.

Practical Implications and Recommendations

For junior high school teachers, the findings of this review suggest that Scratch-based interactive e-modules should be designed in a progressive and scaffolded format: beginning with an orientation to computational thinking concepts, followed by simple project examples, guided practice, small open-ended projects, and reflection on debugging strategies and algorithmic thinking. The most positive studies generally did not separate coding from meaningful problem contexts; rather, they connected programming activities with mathematics, science, or educational games that were relevant to students' experiences. Therefore, modules that merely explain Scratch blocks without contextualized problems are likely to be less effective than modules that include challenges such as "create a probability simulation," "design an animation of energy transformation," or "develop a simple game to solve a problem."]

For researchers, the future research agenda is relatively clear. First, multi-site experimental or quasi-experimental studies involving Indonesian junior high school students remain urgently needed. Second, computational thinking instruments should be made more explicit, whether through standardized tests, Scratch artifact rubrics, or process analytics. Third, delayed post-tests should be incorporated to determine whether observed gains represent sustained learning rather than short-term treatment effects. Fourth, mediating variables such as self-efficacy, cognitive load, and attitudes toward programming should be measured, as several studies indicate that these dimensions help explain why Scratch interventions succeed or fail. Fifth, teacher readiness and curriculum support must not be overlooked, as both reviews on middle-school integrated coding and teacher support for computational thinking instruction consistently underline the importance of teachers' pedagogical capacity.

Based on this synthesis, a systematic literature review article intended for publication in MORFAI would be strongest if it emphasizes design recommendations for Scratch-based interactive e-modules for junior high school characterized by five core features: (1) project- or problem-based learning, (2) tiered scaffolding, (3) integration with mathematics or science contexts, (4) explicit computational thinking assessment, and (5) opportunities for reflection and debugging. These five features represent the most consistent thread across the literature reviewed from 2021–2026.

Limitations of the Review

This review has several limitations. First, not all accessible sources provided complete bibliographic metadata, such as page numbers or detailed issue information; in the reference list, such entries are marked as unspecified. Second, several included studies were conducted at the elementary school or general K–12 level, because evidence specifically addressing the combination of "interactive e-module + Scratch + junior high school" in journals published between 2021 and 2026 remains limited. Third, several Indonesian studies most closely aligned with the topic employed developmental research designs, which provide weaker causal inference compared with quasi-experimental studies involving comparison groups. Consequently, the findings should be interpreted proportionally: the evidence is sufficiently strong to support the development of a systematic literature review article and an instructional design framework, but not yet sufficient to claim universal effectiveness across all junior high school contexts in Indonesia.

4. CONCLUSION

This systematic literature review synthesized studies published between 2021 and early-access 2026 concerning Scratch, interactive e-modules, and computational thinking in primary and secondary education contexts, particularly those relevant to junior high school students. The review findings indicate that Scratch-based learning environments are frequently associated with positive computational thinking outcomes, especially when supported by scaffolding, project- or problem-based learning, contextual integration with mathematics or science, and explicit computational thinking assessment strategies.

However, the evidence concerning Scratch-based learning in general was substantially stronger and more extensive than the evidence specifically related to Scratch-based interactive e-modules for junior high school students. Existing studies focusing directly on interactive e-modules remain relatively limited and are often characterized by small-scale developmental designs, single-school implementations, and limited comparative evaluation. Therefore, findings regarding Scratch-based interactive e-modules should be interpreted cautiously as promising rather than universally conclusive.

The main contribution of this article lies in its role as an evidence-based synthesis that organizes recent findings from 2021–2026, identifies recurring instructional design patterns associated with positive computational thinking outcomes, and formulates design recommendations for Scratch-based interactive e-modules in junior high school contexts. The review also highlights important research gaps, particularly the limited availability of large-scale and contextually grounded studies focusing on Indonesian junior high school students.

Based on the reviewed evidence, Scratch-based interactive e-modules for junior high school students should be designed as contextualized, scaffolded, project-oriented, and reflective learning environments rather than as procedural coding tutorials alone. Future research should prioritize multi-site empirical studies, clearer computational thinking assessment frameworks, and longer-term evaluation of learning sustainability within Indonesian junior high school contexts.

5. REFERENCES

- [1] B. Jiang and Z. Li, "Effect of Scratch on computational thinking skills of Chinese primary school students," *J. Comput. Educ.*, vol. 8, no. 4, 2021, doi: 10.1007/s40692-021-00190-z.
- [2] M. Harahap and D. Eliza, "E-Modul Pembelajaran Coding Berbasis Pengenalan Budaya Indonesia untuk Meningkatkan Computational Thinking," *J. Obs. J. Pendidik. Anak Usia Dini*, vol. 6, no. 4, 2022, doi: 10.31004/obsesi.v6i4.2323.
- [3] T. S. S. Rao and K. K. Bhagat, "Computational thinking for the digital age: a systematic review of tools, pedagogical strategies, and assessment practices," *Educ. Technol. Res. Dev.*, vol. 72, no. 4, 2024, doi: 10.1007/s11423-024-10364-y.
- [4] H. Montiel and M. G. Gomez-Zermeño, "Educational challenges for computational thinking in k-12 education: A systematic literature review of 'scratch' as an innovative programming tool," *Computers*, vol. 10, no. 6, 2021, doi: 10.3390/computers10060069.
- [5] N. K. Findayani, I. N. Suparta, and S. Sariyasa, "Development of Project based Statistics e-Module with Scratch Assistance to Improve Students' Computational Thinking Skills," *J. Pendidik. MIPA*, vol. 24, no. 3, 2023, doi: 10.23960/jpmipa/v24i3.pp516-527.
- [6] H. Ma, M. Zhao, H. Wang, X. Wan, T. W. Cavanaugh, and J. Liu, "Promoting pupils' computational thinking skills and self-efficacy: a problem-solving instructional approach," *Educ. Technol. Res. Dev.*, vol. 69, no. 3, 2021, doi: 10.1007/s11423-021-10016-5.
- [7] K. Katchapikiran, C. Anutariya, and T. Supnithi, "ScratchThAI: A conversation-based learning support framework for computational thinking development," *Educ. Inf. Technol.*, vol. 27, no. 6, 2022, doi: 10.1007/s10639-021-10870-z.
- [8] C. Wang, J. Shen, and J. Chao, "Integrating Computational Thinking in STEM Education: A Literature Review," *Int. J. Sci. Math. Educ.*, vol. 20, no. 8, 2022, doi: 10.1007/s10763-021-10227-5.
- [9] C. Tikva and E. Tambouris, "The effect of scaffolding programming games and attitudes towards programming on the development of Computational Thinking," *Educ. Inf. Technol.*, vol. 28, no. 6, 2023, doi: 10.1007/s10639-022-11465-y.
- [10] S. Gökçe and A. A. Yenmez, "Ingenuity of scratch programming on reflective thinking towards problem solving and computational thinking," *Educ. Inf. Technol.*, vol. 28, no. 5, 2023, doi: 10.1007/s10639-022-11385-x.
- [11] A. Koray and E. Bilgin, "The Effect of Block Coding (Scratch) Activities Integrated into the 5E Learning Model in Science Teaching on Students' Computational Thinking Skills and Programming Self-Efficacy," *Sci. Insights Educ. Front.*, vol. 18, no. 1, 2023, doi: 10.15354/sief.23.or410.
- [12] T. Yılmaz and S. İzmirlı, "Effect of unplugged and plugged coding activities on secondary school students' computational thinking skills," *J. Educ. Technol. Online Learn.*, vol. 6, no. 4, 2023, doi: 10.31681/jetol.1375335.
- [13] K. Bilgic and B. Dogusoy, "Exploring secondary school students' computational thinking experiences enriched with block-based programming activities: An action research," *Educ. Inf. Technol.*, vol. 28, no. 8, 2023, doi: 10.1007/s10639-023-11583-1.
- [14] M. M. Ibrohim, E. Siregar, and U. A. Chaeruman, "Scratch and Computational Thinking in Elementary School: A Meta-analysis," *AL-ISHLAH J. Pendidik.*, vol. 15, no. 3, 2023, doi: 10.35445/alishlah.v15i3.2326.
- [15] T. D. Wulandari, P. Listaji, T. Sulaiman, W. M. W. Jaafar, and S. S. A. Rahim, "Development of STEM-Based Science Educational Game Using Scratch to Train Computational Thinking Skill for Secondary School Students," *Indones. J. Sci. Educ.*, vol. 12, no. 4, pp. 867-884, 2024, doi: <https://dx.doi.org/10.24815/jpsi.v12i4.39893>.
- [16] E. Irawan, R. Rosjanuardi, and S. Prabawanto, "Promoting Computational Thinking through Programming Trends, Tools, and Educational Approaches: a Systematic Review," *JTAM (Jurnal Teor. dan Apl. Mat.*, vol. 8, no. 4, 2024, doi: 10.31764/jtam.v8i4.26407.
- [17] K. Woo and G. Falloon, "Integrating coding across the curriculum: a scoping review," 2025. doi: 10.1080/08993408.2024.2344402.
- [18] A. Rohim and D. Wicaksono, "Arsitek digital masa depan: revolusi pemahaman coding siswa al bayan melalui E-modul Scratch," *J. Penelit. dan Penilai. Pendidik.*, vol. 7, no. 2, pp. 287-308, 2025, doi: <https://doi.org/10.22236/jppp.v7i2.19227>.
- [19] X. Fang, D. T. K. Ng, and M. Yuen, "Effects of geogebra-enhanced scratch computational thinking instruction on fifth-grade students' motivation, anxiety, cognitive load," *Educ. Inf. Technol.*, vol. 30, no. 1, 2025, doi: 10.1007/s10639-024-13052-9.
- [20] R. Zhang, S. Li, and Q. Zhang, "Fostering students' computational thinking and mathematical learning through a Scratch-based probability module: a quasi-experimental study," *Educ. Tech Res. Dev*, 2026, doi: <https://doi.org/10.1007/s11423-026-10599-x>.

-
- [21] M. Berlian, N. M. Arsad, D. Hardila, Y. Yovita, and D. N. Nasution, "Development of Scratch Learning Media to Improve Scientific Literacy and Computational Thinking in Primary Education in the Society 5.0 Era," *J. Nat. Sci. Integr.*, vol. 7, no. 1, pp. 1-15, 2024, doi: 10.24014/jnsi.v7i1.30904.
- [22] F. Handayani, Y. Fitria, S. Ahmad, and Z. Zen, "Development of E-Module Based on Problem Based Learning Assisted with Scratch Applications to Improve Students' Computational Thinking Skills," *J. Kependidikan*, vol. 9, no. 2, pp. 456-469, 2023, doi: 10.33394/jk.v9i2.7790.
- [23] K. A. Setiawan, I. N. Suparta, and I. G. P. Suharta, "Development of a 2D Shape E-Module Based on Project-Based Learning with Scratch to Improve Computational Thinking Skills," *Int. J. Educ. Manag. Technol.*, vol. 3, no. 1, pp. 242-262, 2025.
- [24] H. Yusrina et al., "CTMath: Enhancing Elementary Mathematics and Computational Thinking through Scratch-Integrated Learning," *Front. Educ.*, vol. 11, 2026, doi: 10.3389/educ.2026.1767445.
- [25] J. J. Santaengracia et al., "Computational Thinking in K-6 Education: A Scoping Review of Current Research Trends," *J. Comput. Educ.*, 2026, doi: 10.1007/s40692-025-00380-z.