ITEJ
Indonesian Technology and Education Journal
E-ISSN: 2985-8216; P-ISSN: 2985-8208
https://journal.diginus.id/index.php/ITEJ

Integration of Coding Education into Elementary School Curriculum: An Analysis Through a Systematic Literature Review (SLR)

Rita Puspita Sri Dewi[1], Trimurtini[2]
[1,2] *Universitas Negeri Semarang, Semarang, Indonesia*

Corresponding e-mai : ritanta83@gmail.com[1], trimurtinipgsd@mail.unnes.ac.id[2]

ARTICLE INFO

Keywords:

Coding Education; Computational Thinking; Curriculum Integration; Elementary Education; Systematic Literature Review.

Article History

Received: April 15, 2026
Revised: July 04, 2026
Accepted: August 07, 2026

ABSTRACT

The rapid advancement of digital technology has increased the need to equip elementary school students with computational thinking, creativity, problem-solving, and digital literacy skills. This study systematically analyzes the integration of coding education into the Indonesian elementary school curriculum, focusing on implementation models, instructional approaches, educational outcomes, and implementation challenges. A Systematic Literature Review (SLR) was conducted following PRISMA 2020 guidelines. Relevant peer-reviewed studies published between 2021 and 2025 were identified through database searches, screened using predetermined eligibility criteria, critically appraised using appropriate CASP checklists, and synthesized thematically. Twelve studies met the inclusion criteria. The findings show that coding education has been implemented through subject-based, thematic, project-based, and extracurricular models using both plugged and unplugged approaches. Frequently used platforms and strategies included Scratch, Scratch Jr, Code.org, game-based learning, logic games, simulations, algorithmic activities, and contextual projects. Across the reviewed studies, coding education was associated with improvements in computational and algorithmic thinking, creativity, problem-solving, digital competence, collaboration, motivation, confidence, and student engagement. However, implementation remains constrained by teacher readiness, limited infrastructure, insufficient instructional resources, and variations in school conditions. The evidence also indicates that unplugged activities offer a practical alternative for schools with limited access to technology. Overall, coding integration has strong potential to support future-ready competencies in elementary education, provided that implementation is accompanied by teacher professional development, contextual learning materials, flexible pedagogical approaches, and equitable access to supporting infrastructure.

This is an open access article under the CC BY-SA license

To cite this article : Dewi, R. P. S, & Trimurtini. (2026). Integration of Coding Education into Elementary School Curriculum: An Analysis Through a Systematic Literature Review (SLR). *Indonesian Technology and Education Journal, 4*(2), 350-364. Doi. 10.61255/itej.v4i2.1722

INTRODUCTION

The rapid advancement of digital technologies has fundamentally transformed the competencies required of future generations. In response, education systems worldwide are increasingly expected to equip students with twenty-first-century skills, including computational thinking, digital literacy, creativity, collaboration, and problem-solving abilities (Kominfo, 2021; Zuhri et al., 2024). Among these competencies, coding education has emerged as an effective pedagogical approach for fostering logical reasoning and computational thinking from an early age. Rather than merely teaching programming languages, coding education encourages students to decompose complex problems, identify patterns, develop algorithms, and create digital solutions through meaningful learning experiences. Consequently, many countries have begun integrating coding into elementary school curricula as part of broader educational reforms aimed at preparing digitally competent citizens (Setiani & Novita, 2021).

Indonesia has recently demonstrated a strong commitment to strengthening digital competencies in basic education. Through the enactment of the Ministry of Primary and Secondary Education Regulation (Permendikdasmen) No. 13 of 2025, coding and artificial intelligence have been formally introduced into the national curriculum as learning content to support students' digital competencies. This policy reflects the government's recognition that coding education contributes not only to technological literacy but also to the development of higher-order thinking skills, creativity, collaboration, and character education. Nevertheless, implementing coding education in Indonesian elementary schools remains challenging due to disparities in teacher readiness, infrastructure availability, curriculum implementation, and access to technological resources. These contextual issues indicate that curriculum integration requires comprehensive planning extending beyond the adoption of digital tools (Sakti, 2025).

Previous studies have reported various approaches to implementing coding education in elementary schools. These include project-based learning, Scratch-based programming, unplugged coding activities, game-based learning, thematic instruction, computational thinking integration, and extracurricular coding programs. The reported outcomes generally indicate improvements in students' computational thinking, creativity, problem-solving skills, motivation, and digital literacy. However, most existing studies have focused on evaluating specific instructional interventions, coding platforms, or classroom activities. Likewise, Nishom et al. (2023), previous literature reviews have primarily discussed computational thinking, Scratch-based learning, or coding implementation in specific educational contexts rather than providing a comprehensive synthesis of how coding education is systematically integrated into the elementary school curriculum. Consequently, current evidence remains fragmented, making it difficult for educators and policymakers to identify effective curriculum integration models, implementation strategies, enabling factors, and persistent challenges across different educational settings (Putro & Astuti, 2022; Supriadi, 2024).

These limitations reveal an important research gap. Existing literature has not comprehensively synthesized evidence regarding the integration of coding education into Indonesia's elementary school curriculum by simultaneously examining curriculum models, instructional approaches, teacher readiness, infrastructure, implementation challenges, and reported learning outcomes (Prihatin, 2025; Sucipto et al., 2024). Furthermore, previous reviews rarely employ transparent systematic review procedures that include explicit search strategies, study selection criteria, and structured evidence synthesis. Therefore, this study offers a novel contribution by conducting a systematic literature review that synthesizes recent empirical evidence on coding education integration in Indonesian elementary schools (Dewi et al., 2025). Rather than merely summarizing previous studies, this review identifies emerging themes, compares implementation approaches, highlights consistent findings and contradictions, and maps future research directions that may support curriculum development and educational policy.

This review Suryana et al (2022), is informed by constructivist and connectivist learning theories. Constructivism, developed by Piaget and Vygotsky, explains that meaningful learning occurs when students actively construct knowledge through exploration, collaboration, and authentic learning experiences. Coding education reflects these principles because students learn through experimentation, debugging, project development, and collaborative problem-solving (Gunawan et al., 2025). Meanwhile, Siemens' theory of connectivism emphasizes that learning in the digital era occurs through networks of information, digital technologies, and collaborative knowledge construction (Sukisnawati & Wicaksana, 2025). These theoretical perspectives

E-ISSN: 2985-8216
P-ISSN: 2985-8208

provide an appropriate conceptual framework for interpreting how different coding education approaches facilitate computational thinking, digital literacy, and twenty-first-century competencies within elementary education.

Based on the identified research gap, this study aims to systematically synthesize recent evidence regarding the integration of coding education into Indonesia's elementary school curriculum. Specifically, this review addresses the following research questions: (RQ1) How has coding education been integrated into Indonesia's elementary school curriculum? (RQ2) What instructional approaches and learning strategies are most frequently employed? (RQ3) What educational outcomes have been reported following the implementation of coding education? (RQ4) What challenges and enabling factors influence successful curriculum integration? By answering these questions, this systematic literature review is expected to contribute theoretically by expanding the body of knowledge concerning coding education and computational thinking while providing practical recommendations for teachers, curriculum developers, school leaders, and policymakers responsible for implementing coding education in Indonesian elementary schools.

METHOD

Research Design

This study employed a Systematic Literature Review (SLR) to systematically identify, evaluate, and synthesize empirical evidence regarding the integration of coding education into the elementary school curriculum in Indonesia. The review was conducted following the Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA 2020) guidelines to ensure transparency, reproducibility, and methodological rigor throughout the review process. The SLR approach was selected because it enables researchers to critically synthesize findings from multiple studies, identify research trends, evaluate implementation practices, and reveal research gaps related to coding education in elementary schools. The review process consisted of five sequential stages: (1) formulation of research questions, (2) literature search, (3) study screening and selection, (4) data extraction and quality appraisal, and (5) thematic synthesis and interpretation of findings.

Literature Search Strategy

The literature search was conducted using three electronic databases: Crossref, Google Scholar, and Garuda, because these databases comprehensively index Indonesian educational research relevant to the implementation of coding education in elementary schools. The search focused on studies published between 2021 and 2025, reflecting the recent development of coding education following the implementation of the Merdeka Curriculum and national policies concerning coding and artificial intelligence. The literature search employed combinations of Boolean operators using keywords related to coding education and elementary schools. The primary search string was: ("coding education" OR coding OR "computational thinking" OR "unplugged coding") AND ("elementary school" OR "primary school") AND (Indonesia OR curriculum). Equivalent Indonesian keywords were also applied during database searching, including:

1) "coding sekolah dasar"
2) "pembelajaran coding"
3) "computational thinking"
4) "unplugged coding"
5) "integrasi coding"
6) "kurikulum sekolah dasar"

The search was limited to peer-reviewed journal articles published in Indonesian or English.

E-ISSN: 2985-8216
P-ISSN: 2985-8208

Eligibility Criteria

The eligibility criteria were established before the screening process to ensure consistency in article selection.

Table 1. Inclusion and Exclusion Criteria

Criteria	Inclusion	Exclusion
Publication type	Peer-reviewed journal articles	Proceedings, books, theses, dissertations, editorials
Publication year	2021–2025	Before 2021
Language	English or Indonesian	Other languages
Educational level	Elementary school	Secondary school, higher education, early childhood
Topic	Coding education, computational thinking, coding curriculum integration	ICT studies unrelated to coding
Study design	Quantitative, qualitative, mixed methods, case study, community-based educational implementation	Opinion papers or conceptual articles without empirical data

Study Selection

This systematic literature review (SLR) was conducted following the Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA) 2020 guidelines to ensure a transparent and systematic process of identification, screening, eligibility assessment, and inclusion of relevant studies (Page et al., 2021). The review focused on empirical studies examining coding education, computational thinking, and the integration of coding into elementary school curricula. The literature search was conducted using Google Scholar, ERIC, and DOAJ. The search strategy combined keywords related to coding education, computational thinking, elementary education, curriculum integration, and coding implementation. The search covered studies published between 2021 and 2025 in English or Indonesian.

Following the PRISMA 2020 framework, the study selection process consisted of four stages: identification, screening, eligibility assessment, and inclusion. During the identification stage, records retrieved from the selected databases were collected and screened for relevance. Duplicate or clearly irrelevant records were removed before the screening process. Titles and abstracts were then screened based on the predetermined inclusion and exclusion criteria. Full-text articles were subsequently assessed for eligibility. Studies that met all eligibility criteria were included in the final synthesis. The overall study selection process is presented in the PRISMA 2020 flow diagram (Figure 1) (Page et al., 2021).

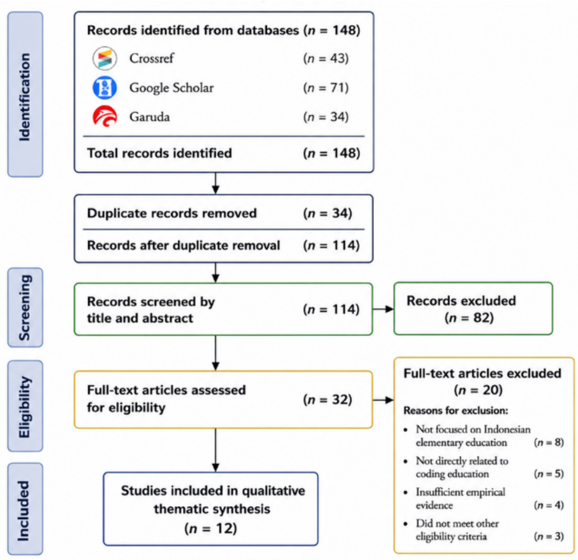

Figure 1. Diagram Prisma

Quality Appraisal

The methodological quality of the included studies was assessed using the Critical Appraisal Skills Programme (CASP) critical appraisal checklists. The appraisal was conducted to examine the methodological rigor, credibility, and relevance of the evidence included in the review. Because the included studies employed different research designs, the checklist was selected according to the methodological characteristics of each study rather than applying a single checklist to all studies. CASP provides different checklists for different study designs, including qualitative studies, randomized controlled trials, cohort studies, cross-sectional studies, and other designs.

For qualitative and case study research, the CASP Qualitative Studies Checklist was used to assess the clarity of the research aims, appropriateness of the qualitative methodology and research design, recruitment strategy, data collection, researcher–participant relationship, ethical considerations, data analysis, clarity of findings, and overall value of the research. For quantitative studies, the CASP checklist corresponding to the reported study design was applied, with attention to the appropriateness of the study design, participant selection, measurement procedures, validity of the findings, precision of the results, and applicability to the review question. For community-based educational implementation studies, the appraisal checklist was selected according to the underlying methodological design reported in the original study.

Each included study was critically appraised against the relevant CASP criteria. Responses were recorded as Yes, No, or Can't tell, following the CASP approach. The appraisal focused on the methodological strengths and limitations of each study rather than producing a numerical quality score, as CASP does not recommend assigning numerical scores to its checklists.

The appraisal results were used to inform the interpretation and synthesis of the evidence. Studies were not automatically excluded solely because they used different research designs or had methodological limitations. Instead, identified strengths and limitations were considered when interpreting the consistency, credibility, and applicability of the findings across the included studies.

Table 2. Quality Appraisal of the Included Studies

No.	Study	Study Design	CASP Checklist	Appraisal Result
1	Istiqomah & Novika (2024)	Community-based educational activity	CASP checklist appropriate to reported design	High
2	Maziyah & Wakhyudin (2025)	Elementary school implementation	CASP checklist appropriate to reported design	High
3	Santoso & Churniawan (2022)	Extracurricular/community-based implementation	CASP checklist appropriate to reported design	Moderate
4	Alindra et al. (2024)	Case study	CASP Qualitative Studies Checklist	High
5	Cuhanazriansyah et al. (2025)	Educational implementation	CASP checklist appropriate to reported design	High
6	Rati et al. (2025)	Elementary science education	CASP checklist appropriate to reported design	Moderate
7	Hilmiyah et al. (2025)	Case study	CASP Qualitative Studies Checklist	High
8	Ahmad et al. (2023)	Classroom implementation	CASP checklist appropriate to reported design	High
9	Syamsiah et al. (2024)	Educational intervention	CASP checklist appropriate to reported design	Moderate
10	Anjani et al. (2025)	Community-based educational implementation	CASP checklist appropriate to reported design	High
11	Kasiono et al. (2025)	Educational intervention	CASP checklist appropriate to reported design	High
12	Sondakh et al. (2024)	Community-based educational implementation	CASP checklist appropriate to reported design	Moderate

Data Extraction

A standardized data extraction form was developed to ensure consistency during data collection. The following information was extracted from each study:

1) Author(s)
2) Publication year
3) Research design
4) Educational setting
5) Sample characteristics
6) Coding platform or learning media
7) Plugged or unplugged approach
8) Curriculum integration model
9) Learning strategy
10) Reported outcomes
11) Major findings
12) Research limitations

The extracted information was subsequently organized into summary tables to facilitate comparison across studies.

Data Synthesis

The extracted data were analyzed using thematic synthesis. Initially, findings from each study were coded according to their similarities and differences. These codes were then grouped into broader themes representing patterns of coding education implementation in elementary schools.

The synthesis generated several major themes, including:

1. Models of curriculum integration;
2. Plugged and unplugged coding approaches;
3. Learning strategies for coding education;
4. Student learning outcomes;
5. Teacher readiness and professional development;
6. Infrastructure and implementation challenges.

Rather than merely describing individual studies, the thematic synthesis compared evidence across studies to identify recurring findings, differences, and research gaps concerning coding education implementation in Indonesian elementary schools.

RESULTS AND DISCUSSION

The systematic literature review identified 12 studies that met the predetermined eligibility criteria. The included studies examined coding education in Indonesian elementary school contexts through different implementation models, including digital or plugged coding, unplugged coding, thematic learning, subject-based integration, project-based learning, and extracurricular activities. The findings were synthesized thematically according to the four research questions concerning curriculum integration, instructional approaches, educational outcomes, and implementation challenges. The results of the review of these articles are shown in Table 3 below.

Table 2. Characteristics of the Included Studies

No.	Study	Design	Coding Approach	Platform /Media	Curriculum Context	Main Outcomes	Main Findings
1	Istiqomah & Novika (2024)	Community-based educational activity; elementary students	Plugged	Scratch	Coding/game development	Computational thinking, algorithmic logic, motivation	Visual and contextual coding through game development supported computational thinking, algorithmic logic, and learning motivation.

No.	Study	Design	Coding Approach	Platform /Media	Curriculum Context	Main Outcomes	Main Findings
2	Maziyah & Wakhyudin (2025)	Elementary school implementation	Plugged	Code.org	Coding instruction	Algorithmic understanding, critical thinking, problem-solving	Game-based coding activities using puzzles and interactive challenges supported students' understanding of programming logic and problem-solving.
3	Santoso & Churniawan (2022)	Extracurricular program	Plugged	Scratch Jr and Scratch	Extracurricular coding	Digital competence, collaboration, creativity, confidence	Hands-on practice and mentoring supported digital competence, collaboration, creativity, and students' confidence in using technology.
4	Alindra et al. (2024)	Case study	Plugged	Coding/project media	Merdeka Curriculum	Digital skills, independence, responsibility, curiosity	Project-based coding activities were linked to learning themes and supported digital skills and selected character competencies.
5	Cuhanazriansyah et al. (2025)	Elementary school implementation	Plugged	Scratch	Thematic coding	Logical thinking, creativity, confidence, digital literacy	Thematic coding connected programming with science and mathematics and was associated with improvements in logical and creative thinking.
6	Rati et al. (2025)	Elementary science education	Unplugged	Physical activities/ learning materials	Science education	Computational thinking, environmental awareness	Unplugged coding combined with Tri Hita Karana supported computational thinking and environmental awareness.
7	Hilmiyah et al. (2025)	Elementary education	Unplugged and block-based	Worksheets, modules, Scratch	Thematic learning	Computational thinking, creativity	Simulations, logic puzzles, and simple projects supported computational thinking and creative problem-solving.
8	Ahmad et al. (2023)	Elementary Indonesian-language instruction	Unplugged	Algorithmic activities	Indonesian language; procedural text	Systematic thinking, text-structure understanding	Procedural-text activities were organized algorithmically, demonstrating that computational thinking concepts can be integrated without computers.
9	Syamsiah et al. (2024)	Elementary school-aged children	Unplugged	Physical activities and logic games	Coding/computational-thinking education	Computational thinking, problem-solving, engagement	Physical activities, role-playing, and logic games supported computational thinking, problem-solving, and active engagement.
10	Anjani et al. (2025)	Elementary education	Plugged	Scratch-based visual programming	Coding/problem-solving learning	Higher-order thinking, collaboration, digital readiness	Mini-projects using visual programming supported logic, creativity, collaboration, and problem-solving.
11	Kasiono et al. (2025)	Elementary education	Unplugged	Logic games, algorithm cards, simulations	Coding/computational-thinking education	Computational thinking, motivation, engagement	Unplugged activities supported decomposition, pattern recognition, abstraction, and algorithmic thinking; teacher understanding and teaching resources remained challenges.
12	Sondakh et al. (2024)	Elementary school	Unplugged	CS Unplugged activities	Algorithmic thinking	Algorithmic thinking, interest in computer science	Sorting Network and Move It, Move It activities supported algorithmic thinking and interest in computer science despite facility and time constraints.

Curriculum Integration of Coding Education

The 12 reviewed studies indicate that coding education has been integrated into elementary education through several models, including subject-based, thematic, project-based, and extracurricular learning. Rather than being implemented exclusively as a separate subject, coding concepts were connected with existing learning content and broader educational objectives. Subject-based integration was demonstrated by Ahmad et al. (2023), who incorporated computational-thinking concepts into Indonesian language instruction through procedural-text activities. Rati et al. (2025) similarly connected unplugged coding with science education and environmental awareness, while Cuhanazriansyah et al. (2025) integrated coding with science and mathematics through thematic learning. These findings demonstrate that computational-thinking concepts can be introduced through existing subjects without requiring coding to function as an independent subject.

Project-based and extracurricular models represented another pattern of curriculum integration. Istiqomah and Novika (2024) introduced coding through game development using Scratch, while Santoso and Churniawan (2022) implemented basic programming through extracurricular activities using Scratch Jr and Scratch. Alindra et al. (2024) connected coding projects with learning themes within the Merdeka Curriculum. These studies indicate that coding can also be introduced through enrichment and project-oriented activities that allow students to create products and apply computational concepts in practical contexts. Overall, the evidence indicates that coding integration occurs along a continuum, ranging from extracurricular activities to direct integration into classroom subjects and thematic learning. However, the reviewed studies were generally conducted within specific schools, programs, or learning activities. Therefore, the current evidence provides stronger support for the feasibility of localized coding-integration models than for sustained and system-wide implementation across Indonesian elementary schools.

Instructional Approaches and Learning Strategies

Two major instructional approaches were identified across the reviewed studies: plugged coding and unplugged coding. Plugged coding involves the use of digital programming environments, whereas unplugged coding introduces computational and algorithmic concepts through physical activities, games, simulations, worksheets, and other non-digital learning resources. Plugged coding was commonly implemented through visual programming platforms and game-based activities. Istiqomah and Novika (2024) used Scratch for game development, while Maziyah and Wakhyudin (2025) implemented coding activities through Code.org using puzzles and interactive challenges. Santoso and Churniawan (2022) used Scratch Jr and Scratch in extracurricular learning. These studies demonstrate that visual programming platforms can provide students with opportunities to practice programming logic while creating or manipulating digital products.

Project-based and thematic strategies were also identified within plugged coding activities. Alindra et al. (2024) employed project-based coding activities connected with learning themes, while Cuhanazriansyah et al. (2025) used thematic coding to connect programming with science and mathematics. Hilmiyah et al. (2025) combined unplugged and block-based activities through modules, worksheets, simulations, logic puzzles, and simple Scratch projects. These approaches indicate that coding activities can be adapted to interdisciplinary and contextual learning rather than being limited to technical programming exercises. Unplugged coding represented an important alternative for introducing computational concepts without continuous access to digital devices. Rati et al. (2025) used physical activities in science education, while Ahmad et al. (2023) introduced algorithmic thinking through procedural-text activities. Syamsiah et al. (2024) employed role-playing and logic games, Kasiono et al. (2025) used logic games, algorithm cards, and movement simulations, and Sondakh et al. (2024) implemented Computer Science Unplugged activities through Sorting Network and Move It, Move It.

The cross-study evidence therefore indicates that plugged and unplugged approaches serve complementary purposes. Plugged approaches provide direct experience with digital programming environments, whereas unplugged approaches enable students to develop foundational computational and algorithmic concepts through accessible physical and contextual activities. The choice of approach appears to depend on the learning objectives, available resources, instructional context, and students' learning characteristics.

Educational Outcomes of Coding Education

The reviewed studies reported a range of educational outcomes associated with coding and computational-thinking activities. These outcomes can be grouped into four broad categories: computational and algorithmic thinking, problem-solving and creativity, digital competence and collaboration, and motivation and engagement. Computational and algorithmic thinking were prominent outcomes across the reviewed studies. Istiqomah and Novika (2024) reported outcomes related to computational thinking and algorithmic logic through Scratch-based game development. Rati et al. (2025), Ahmad et al. (2023), Syamsiah et al. (2024), Kasiono et al. (2025), and Sondakh et al. (2024) also emphasized algorithmic or computational-thinking development through unplugged and contextual activities. The recurring emphasis on sequencing, decomposition, pattern recognition, abstraction, and algorithmic reasoning suggests that coding education can provide elementary students with opportunities to practice fundamental computational-thinking concepts.

Problem-solving and creativity were also reported across several studies. Maziyah and Wakhyudin (2025) associated game-based coding activities with algorithmic understanding, critical thinking, and problem-solving. Hilmiyah et al. (2025) reported computational thinking and creative problem-solving through simulations, logic puzzles, and simple projects. Anjani et al. (2025) emphasized logic, creativity, collaboration, and higher-order thinking through Scratch-based mini-projects. These findings indicate that coding activities may provide opportunities for students to construct solutions, test ideas, and revise their work. Several studies also reported outcomes related to digital competence and collaboration. Santoso and Churniawan (2022) reported improvements in digital competence, collaboration, creativity, and confidence through hands-on coding activities and mentoring. Alindra et al. (2024) connected coding projects with digital skills as well as independence, responsibility, and curiosity. Cuhanazriansyah et al. (2025) also linked thematic coding with digital literacy, logical thinking, creativity, and confidence.

Motivation, interest, and engagement were additional outcomes reported in the reviewed literature. Istiqomah and Novika (2024) reported learning motivation in the context of game development, while Kasiono et al. (2025) reported motivation and engagement through unplugged activities. Sondakh et al. (2024) identified increased interest in computer science following Computer Science Unplugged activities. These findings suggest that interactive, contextual, and hands-on coding activities may support students' affective engagement with computational learning. Overall, the 12 studies indicate that coding education is associated with multiple educational outcomes extending beyond programming knowledge. The most consistent outcomes concern computational and algorithmic thinking, while creativity, problem-solving, digital competence, collaboration, motivation, and engagement were also reported. Nevertheless, the studies differ in research design, implementation context, and outcome measures. Therefore, these findings should be interpreted as reported outcomes within individual study contexts rather than as evidence of a uniform causal effect of coding education across elementary schools.

Challenges and Supporting Factors

The reviewed studies identified several factors that influence the implementation of coding education in elementary schools. These factors can be grouped into teacher readiness, learning resources and infrastructure, instructional time, and contextual adaptation. Teacher readiness was identified as an important consideration. Cuhanazriansyah et al. (2025) incorporated teacher coding training before the implementation of thematic coding activities, indicating the importance of teacher preparation. Kasiono et al. (2025) identified teachers' limited understanding of computational-thinking concepts and the lack of contextual teaching materials as challenges. These findings indicate that coding integration requires teachers to have sufficient conceptual understanding as well as appropriate instructional resources.

Infrastructure was another important factor, particularly for plugged coding. Digital platforms such as Scratch, Scratch Jr, and Code.org require access to suitable devices and learning environments. In contrast, Rati et al. (2025), Syamsiah et al. (2024), Kasiono et al. (2025), and Sondakh et al. (2024) demonstrate that unplugged approaches can provide alternatives when access to ICT facilities is limited. Time and contextual adaptation also influence implementation. Sondakh et al. (2024) identified limitations related to ICT facilities and instructional time, while Kasiono et al. (2025) emphasized the need for appropriate instructional guides and contextual teaching materials. These findings suggest that coding activities need to be adapted to school conditions rather than implemented through a single standardized model.

Across the studies, several supporting factors were evident, including teacher training, hands-on learning, mentoring, contextual learning, project-based activities, games, and the availability of appropriate learning materials. These factors can facilitate students' engagement with coding concepts while allowing teachers to adapt activities to available resources. Taken together, the evidence indicates that successful coding integration depends not only on the availability of technology but also on teacher competence, instructional support, appropriate learning materials, sufficient learning time, and contextual adaptation. Unplugged coding can serve as a practical alternative in resource-constrained settings, while plugged approaches can provide opportunities for direct interaction with digital programming environments.

Discussion

The findings of this systematic literature review indicate that coding education in Indonesian elementary schools has been implemented through diverse approaches rather than a single standardized model. Across the 12 reviewed studies, coding was integrated through subject-based learning, thematic learning, project-based activities, and extracurricular programs using both plugged and unplugged approaches. The findings further indicate that coding education is associated with the development of computational and algorithmic thinking, problem-solving, creativity, digital competence, collaboration, motivation, and student engagement. However, the implementation of coding education remains influenced by teacher readiness, infrastructure, instructional time, and the availability of contextual learning resources.

One important finding is that coding education can be integrated into existing elementary school learning rather than being treated exclusively as an independent subject. The reviewed studies demonstrate integration with Indonesian language, science, mathematics, thematic learning, and extracurricular activities. This integration allows students to encounter coding concepts through familiar learning contexts and existing curriculum objectives. Ahmad et al. (2023), for example, demonstrated that algorithmic thinking can be introduced through procedural-text activities, while Rati et al. (2025) connected unplugged coding with science learning and environmental awareness. Cuhanazriansyah et al. (2025) also demonstrated the integration of coding with science and mathematics through thematic activities. These findings suggest that the educational value of coding extends beyond programming syntax and can support the development of transferable ways of thinking across subject areas.

The integration of coding into different subjects is particularly relevant in elementary education because students are still developing foundational cognitive and problem-solving abilities. When computational concepts are embedded in familiar subjects and real-world situations, students can understand coding-related concepts through meaningful learning experiences. This finding supports the use of interdisciplinary and contextual learning approaches in which coding becomes a means of developing computational thinking rather than an isolated technical skill. However, the reviewed studies were generally conducted within specific schools or programs. Therefore, the existing evidence provides stronger support for the feasibility of localized and contextual coding integration than for sustained, system-wide implementation across Indonesian elementary schools.

The coexistence of plugged and unplugged approaches is another important finding. Plugged approaches using Scratch, Scratch Jr, and Code.org provide students with direct experience of digital programming environments. Through game development, visual programming, and project-based activities, students can construct digital products, test solutions, identify errors, and revise their work. These experiences can provide an authentic introduction to programming while simultaneously supporting creativity and digital literacy. In contrast, unplugged approaches demonstrate that fundamental computational concepts can be introduced without continuous access to computers. Physical activities, algorithm cards, role-playing, simulations, and logic games can be used to introduce sequencing, decomposition, pattern recognition, abstraction, and algorithmic thinking.

The findings therefore suggest that plugged and unplugged approaches should not be considered competing models but rather complementary approaches. Unplugged activities can provide an accessible introduction to computational concepts, particularly in schools with limited technological infrastructure, while plugged activities can provide opportunities for students to apply those concepts through digital programming environments. The combination of the two approaches may provide greater flexibility for schools because implementation can be adjusted according to learning objectives, student characteristics, teacher competence, and available resources. This is particularly important in the Indonesian elementary school context, where differences in infrastructure and technological access may influence the feasibility of technology-dependent learning.

The reviewed studies also indicate that coding education is associated with educational outcomes extending beyond programming knowledge. Computational and algorithmic thinking were among the most frequently reported outcomes, while problem-solving, creativity, collaboration, digital competence, motivation, confidence, and engagement were also identified. Coding activities require students to organize procedures, decompose problems, identify patterns, develop solutions, test their ideas, detect errors, and revise their work. These activities provide opportunities for students to practice systematic and reflective problem-solving. Project-based and game-development activities also allow students to create products and explore alternative solutions, thereby creating opportunities for computational and creative thinking to develop together.

The findings concerning creativity and collaboration are also important because coding learning does not necessarily have to be conducted as an individual technical activity. When students work together to design projects, discuss solutions, identify errors, and modify their products, coding activities can become collaborative learning experiences. Santoso and Churniawan (2022), for example, reported outcomes related to digital competence, collaboration, creativity, and confidence, while Alindra et al. (2024) connected coding projects with independence, responsibility, and curiosity. These findings indicate that the pedagogical design of coding activities is important in determining whether coding is used only to teach programming skills or also to develop broader 21st-century competencies.

Nevertheless, the findings should be interpreted cautiously because the 12 reviewed studies differ in research design, participants, implementation duration, learning contexts, and outcome measures. The current evidence therefore demonstrates recurring educational outcomes reported across the literature but does not establish a uniform causal effect of coding education across Indonesian elementary schools. Future empirical studies using comparative designs, larger and more diverse samples, longer implementation periods, and validated assessment instruments are needed to provide stronger evidence regarding the effectiveness and sustainability of coding education.

Teacher readiness emerged as another critical factor in the implementation of coding education. Coding integration requires teachers to understand fundamental computational concepts and to translate those concepts into age-appropriate learning activities. Teachers must also be able to connect coding with curriculum objectives, facilitate student problem-solving, manage digital or unplugged activities, and respond to students' learning difficulties. Cuhanazriansyah et al. (2025) incorporated teacher coding training before implementing thematic coding activities, while Kasiono et al. (2025) identified limited teacher understanding of computational-thinking concepts and a lack of contextual teaching materials as challenges. These findings indicate that teacher professional development should accompany coding curriculum implementation.

Teacher training should therefore not focus solely on technical programming skills. It should also provide teachers with pedagogical strategies for introducing computational thinking, designing unplugged activities, integrating coding with existing subjects, and adapting learning activities to students' developmental characteristics. This approach would enable teachers to view coding not simply as a technological subject but as a learning strategy that can support reasoning, creativity, and problem-solving. Infrastructure is another factor that needs to be considered. Plugged coding requires access to suitable devices, software, connectivity, and learning environments. Differences in school facilities may therefore create unequal opportunities for students to participate in technology-based coding activities. The evidence from unplugged studies provides an important alternative because computational concepts can be introduced through physical activities, logic games, simulations, and other low-resource learning materials. Consequently, coding education should not be designed around a technology-dependent model alone. A flexible combination of plugged and unplugged activities may provide a more equitable approach to implementation.

The findings have several practical implications for elementary education. Coding can be introduced gradually, beginning with unplugged activities that develop basic concepts such as sequencing, pattern recognition, decomposition, and algorithms. Once students and teachers are sufficiently prepared, visual programming platforms can be introduced to support digital creation and project development. Coding can also be connected with existing subjects and thematic projects so that computational thinking becomes part of everyday learning rather than an additional isolated subject. At the school level, teacher professional development, appropriate instructional materials, infrastructure, and sufficient instructional time should be considered together rather than separately.

The review also suggests that the implementation of coding education should be contextual and flexible. Schools should be able to select learning activities according to their infrastructure, teacher capacity, student characteristics, curriculum requirements, and local context. The use of local contexts and culturally relevant activities, as demonstrated by Rati et al. (2025), may further increase the meaningfulness of coding education for elementary students. Such contextualization can help ensure that coding is not introduced merely as a technological trend but as a meaningful educational approach that responds to students' learning needs.

Overall, the evidence suggests that coding education has considerable potential to support computational thinking and broader 21st-century competencies in Indonesian elementary schools. However, the literature currently provides stronger evidence regarding the feasibility, diversity, and reported benefits of coding activities than regarding their long-term effectiveness at a national scale. The successful integration of coding therefore requires a combination of appropriate pedagogy, teacher readiness, contextual learning materials, infrastructure, and flexible implementation models. Further research should strengthen the evidence base through more rigorous comparative designs, broader samples, longitudinal implementation, and consistent assessment of computational thinking and related learning outcomes.

CONCLUSIONS

The findings from the sources reviewed indicate that integrating coding education into elementary schools has a positive impact on the development of students' computational thinking, creativity, and problem-solving skills. Various approaches including both "plugged" and "unplugged" activities have proven effective in helping students understand algorithmic logic in a fun and accessible way. Other key challenges such as teacher readiness, limited facilities, and varying school conditions require attention to ensure the equitable implementation of coding education. Overall, this literature review confirms that coding holds great potential as part of the elementary school curriculum in preparing a technologically literate generation, provided it is supported by appropriate policies, teacher training, and infrastructure.

Based on these findings, strategic steps are needed to ensure that the implementation of coding aligns with the identified potential. First, improving teacher competencies is a fundamental aspect that must be prioritized. Second, the provision of resources and access to technology must be strengthened to reduce disparities among schools. Third, schools need technical guidelines that are practical and easy to implement. Fourth, collaboration among stakeholders is crucial to ensuring the sustainable implementation of coding. Fifth, further research is needed, focusing on exploring the effectiveness of various coding learning models, measuring regional readiness, and assessing the long-term impact on students' learning processes. To make coding instruction more effective, teachers can start with simple projects that are relevant to students' lives, conduct these sessions regularly even if they are brief and make use of the tools and learning resources available at school. Theoretically, coding instruction should be developed through a more contextual approach, such as by linking it to local wisdom so that students can more easily understand the material and feel a connection to it. Further research should also focus on developing coding teaching materials, such as instructional modules, worksheets, and interactive learning media, as these resources are essential for teachers in preparing structured lessons tailored to classroom conditions.

ACKNOWLEDGMENT

The authors would like to express their gratitude to all individuals and institutions who contributed to the completion of this study. The authors also acknowledge the support and valuable feedback provided during the preparation and revision of the manuscript.

AUTHOR CONTRIBUTION STATEMENT

All authors contributed to the development of this study. The authors were involved in the conceptualization of the study, literature search and selection, data extraction and analysis, interpretation of the findings, and preparation and revision of the manuscript. All authors have read and approved the final version of the manuscript and agreed to be accountable for all aspects of the work.

AI DISCLOSURE STATEMENT

During the preparation of this manuscript, the authors used ChatGPT (OpenAI) solely to improve the language, grammar, and readability of the manuscript. The authors carefully reviewed, verified, and edited all AI-assisted content and take full responsibility for the accuracy, integrity, and final content of the manuscript.

REFERENCES

Ahmad, Miskliana, D., & Aprinastuti, C. (2023). Integrasi Computational Thinking pada Mata Pelajaran Bahasa Indonesia Materi Teks Prosedur. *Pedadidaktika: Jurnal Ilmiah Pendidikan Guru Sekolah Dasar, 10*(2), 158–169. https://doi.org/10.17509/pedadidaktika.v10i2.62782

Alindra, A. L., Nafira, A., Khaerunnisa, H., Ayu, P., Sari, K., Anggia, Y., & Nurhaliza, Y. (2024). Studi Kasus Pembelajaran Berbasis Koding Guna Memperkuat Kurikulum Merdeka di Era Digital. *Jurnal Pendidikan Tambusai*, *8*(1), 3171–3183.

Anjani, D., Ali, P. R., & Novianti, D. (2025). Menguasai Keterampilan Abad 21: Pengenalan Problem Solving dan Coding pada Sekolah Dasar. *Smart Dedication: Jurnal Pengabdian Masyarakat*, *2*(1), 39–48. https://doi.org/10.70427/smartdedication.v2i1.167

Awaluddin, & Muhamad Sofian Hadi. (2025). Integrasi Pembelajaran Coding Dan Kecerdasan Buatan Di Sekolah Dasar: Tantangan Dan Peluang. *Pendas: Jurnal Ilmiah Pendidikan Dasar*, *10*(1), 1081–1086. https://journal.unpas.ac.id/index.php/pendas/article/view/21753/11028

Bakti, A., Irawan, W. D., Sukaesih, A., Melati, E., Sudarso, H., & Irawan. (2025). Peran Literasi Digital dalam Meningkatkan Kemampuan Berpikir Kritis Mahasiswa di Era Society 5.0. *Jurnal Pengabdian Masyarakat Dan Riset Pendidikan*, *4*(2), 8283–8291. https://doi.org/10.31004/jerkin.v4i2.3166

Carayannis, E. G., & Morawska, J. (2022). The Future of Europe: Society 5.0 and Industry as Driving Forces of Future Universities. *Journal of the Knowledge Economy*, 3445–3471. https://doi.org/10.1007/s13132-021-00854-2

Critical Appraisal Skills Programme (CASP). (2024). *CASP checklist: Qualitative research*. Critical Appraisal Skills Programme. https://casp-uk.net/casp-checklists/CASP-checklist-qualitative-2024.pdf

Cuhanazriansyah, M. R., Amir, D. R., Irhadtanto, B., Amin, A. K., Zuhad, Z., Malyana, M., & Saputra, L. K. (2025). Membangun Sekolah Ramah Teknologi Dengan Penerapan Kelas Coding Tematik Di Sdn Dolokgede. *Jurnal Pengabdian Kolaborasi Dan Inovasi IPTEKS*, *3*(3), 774–780. https://doi.org/10.59407/jpki2.v3i3.2585

Dewi, M., Ballianie, N., Astuti, M., Fatimahi, S., & Sari, G. P. I. (2025). Tantangan Implementasi Kurikulum Di Era Digital : Kesiapan Guru dan Infrastruktur. *Innovative: Journal Of Social Science Research*, *5*(4), 10732–10741. https://doi.org/10.31004/innovative.v5i4.20987

Fikriyah, N., Wiryanto, & Istiq'faroh, N. (2024). Systematik Literature Review: Penggunaan Media Interaktif Scratch pada Peningkatan Kemampuan Numerasi Siswa SD. *Pendas: Jurnal Ilmiah Pendidkan Dasar*, *09*(4). https://journal.unpas.ac.id/index.php/pendas/article/view/20251

Gunawan, A. I., Rizaldi, F., Afifah, N., Aryanti, A. N., & Utama, D. H. (2025). Relevansi Teori Belajar Konektivisme dalam Disrupsi Teknologi dan Artificial Intelligence Era. *Kamil: Journal of Education*, *1*(1), 10–26. https://doi.org/10.65065/wdmyzw79

Hilmiyah, Muhith, A., & Bahri, S. (2025). Coding for Young Learners: Enhancing Computational Thinking and Creativity in Elementary Education. *Al-Adzka: Jurnal Ilmiah Pendidikan Guru Madrasah Ibtidaiyah*, *15*(1),96–114. https://jurnal.uinantasari.ac.id/index.php/adzka. https://doi.org/10.18592/aladzkapgmi.v15i1.16103

Istiqomah, N., & Novika, F. (2024). Pengenalan Coding Membuat Game pada Siswa Sekolah Dasar menggunakan Scratch. *JURPIKAT (Jurnal Pengabdian Kepada Masyarakat)*, *5*(3), 925–938. https://doi.org/10.37339/jurpikat.v5i3.1827

Kasiono, B., Pradhana, G., Huda, N., Utami, S., & Haerussaleh. (2025). Unlocking Young Minds : The Impact of Unplugged Coding on Elementary Students' Computational Thinking. *Jurnal Paedagogy: Jurnal Penelitian Dan Pengembangan Pendidikan*, *12*(3), 754–763. https://doi.org/10.33394/jp.v12i3.15650

Kemendikdasmen. (2025). *Permendikdasmen Nomor 13 Tentang Perubahan Atas Peraturan Menteri Pendidikan, Kebudayaan, Riset, dan Teknologi Nomor 12 Tahun 2024 Tentang Kurikulum Pada Pendidikan Anak Usia Dini, Jenjang Pendidkan Dasar dan Jenjang Pendidikan Menengah*.

Kominfo. (2021). *Status Literasi Digital di Indonesia Ringkasan Eksekutif*. Kominfo.

Maziyah, F. A., & Wakhyudin, H. (2025). Pelaksanaan Pembelajaran Coding SDN Pandean Lamper 05. *Journal Of MISTER (Multidiscilinary Inquiry in Science Technology and Educational Research)*, *2*(1), 2571–2575. https://doi.org/10.32672/mister.v2i1b.2963

Nishom, M., Sasmito, G. W., & Wibowo, D. S. (2023). Pengenalan dan Pemanfaatan Ilmu Coding untuk Melatih Kemampuan Berpikir bagi Guru dan Peserta Didik. *Jurnal Pengabdian Dan Penerapan IPTEK.*, *7*, 63–70. https://doi.org/10.31284/j.jpp-iptek.2023.v7i1.3310

Page, M. J., McKenzie, J. E., Bossuyt, P. M., Boutron, I., Hoffmann, T. C., Mulrow, C. D., Shamseer, L., Tetzlaff, J. M., Akl, E. A., Brennan, S. E., Chou, R., Glanville, J., Grimshaw, J. M., Hróbjartsson, A., Lalu, M. M., Li, T., Loder, E. W., Mayo-Wilson, E., McDonald, S., McGuinness, L. A., Stewart, L. A., Thomas, J., Tricco, A. C., Welch, V. A., Whiting, P., & Moher, D. (2021). The PRISMA 2020 statement: An updated guideline for reporting systematic reviews. *BMJ, 372*, n71. https://doi.org/10.1136/bmj.n71

Prihatin, M. R. (2025). Koding dan AI di Sekolah: Kajian Literatur Terhadap Kesiapan Kurikulum dan Pembelajaran di SD/SMP. *Jurnal P4I: Jurnal Inovasi Strategi Dan Model Pembelajaran*, *5*(3), 219–231. https://doi.org/10.51878/strategi.v5i3.6022

Putro, Y. T. M., & Astuti, R. (2022). Penerapan Scratch dalam Pembelajaran Coding Siswa Sekolah Dasar. *Emergent: Journal of Educational Discoveries and Lifelong Learning*, *1*(4), 1–21. https://doi.org/10.47134/emergent.v1i4.37

Rati, N. W., Sudatha, I. G. W., Dwiarwati, K. A., Lesmana, K. Y. P., Esaputra, I. N. T., & Chiu, C. (2025). Unplugged Coding and Tri Hita Karana in Fostering Computational Thinking and Environmental Awareness. *Jurnal Pendidikan IPA Indonesia*, *14*(2), 314–323. https://doi.org/10.15294/jpii.v14i2.24212

Sakti, B. P. (2025). Pembelajaran Coding di Sekolah Dasar. *Jurnal Penelitian Ilmu Pendidikan Indonesia*, *4*(1), 62–68. https://doi.org/10.31004/jpion.v4i1.326

Santoso, R., & Churniawan, A. D. (2022). Pendampingan Program Ekstrakurikuler Pembelajaran Coding di SD Anak Panah Kenjeran. *Jurnal Layanan Masyarakat (Jornal of Public Service)*, *6*(1), 109–118. https://doi.org/10.20473/jlm.v6i1.2022.109-118

Setiani, N. N., & Novita, B. (2021). Urgensi Literasi Digital dalam Menyongsong Siswa Sekolah Dasar menuju Generasi Emas Tahun 2045. *Prosiding SEMAI: Seminar Nasional PGMI 2021*, 411–427.

Sondakh, D. E., Pungus, S. R., Maringka, R. C., & Tangka, G. M. W. (2024). Memperkenalkan Algoritma dan Berpikir Algoritmik kepada Siswa Sekolah Dasar Menggunakan CS Unplugged. *Servitium Smart Journal*, *2*(2), 134–140. https://doi.org/10.31154/servitium.v2i2.29

Sucipto, Sukri, M., Patras, Y. E., & Novita, L. (2024). Tantangan Implementasi Kurikulum Merdeka si Sekolah Dasar: Systematic Literature Review. *Kalam Cendekia: Jurnal Ilmiah Kependidikan*, *12*(1). https://doi.org/10.20961/jkc.v12i1.84353

Sukisnawati, & Wicaksana, D. (2025). Pengembangan Pembelajaran Coding Berbasis PBL bertema Keberlanjutan Berorientasi pada SDGs di Sekolah Dasar. *JERCS: Journal of Educational Research and Community Service*, *01*(4), 300–304. https://journal.nabaedukasi.com/index.php/jercs/article/view/52

Supriadi, I. G. I. (2024). Scratch Berdiferensiasi Untuk Meningkatkan Kemampuan Coding Siswa Sekolah Dasar. *Seminar Nasional Pendidikan Guru Sekolah Dasar*, *1*, 11–19.

Suryana, E., Aprina, M. P., & Harto, K. (2022). Teori Konstruktivistik dan Implikasinya dalam Pembelajaran. *JIIP (Jurnal Ilmiah Ilmu Pendidikan)*, *5*(7), 2070–2080. https://doi.org/10.54371/jiip.v5i7.666

Suryaningsih, N. M. A., Sudatha, I. G. W., Suartama, I. K., & Santosa, M. H. (2025). Implementasi Pembelajaran STEAM Terhadap Kemampuan Literasi Anak Usia Dini: Systematic Literature Review. *Aulad: Journal on Early Childhood*, *8*(3), 1226–1235. https://doi.org/10.31004/aulad.v8i3.1280

Syamsiah, N. O., Firmansyah, Y., Mustika, Y., Burhanudin, B., Gani, A., & Firiani, N. (2024). Pengaruh Edukasi Unplugged Coding Terhadap Kemampuan Computational Thinking Anak Usia Sekolah Dasar. *Vox Edukasi: Jurnal Ilmiah Ilmu Pendidikan*, *15*(2), 365–373. https://doi.org/10.31932/ve.v15i2.4048

Zuhri, S., Suwindia, I. G., & Winangun, I. M. A. (2024). Literasi digital dan Kecakapan abad ke-21: Analisis komprehensif dari Literatur Terkini. *Journal IICET: Education and Social Sciences Review*, *5*(2), 149–155. https://doi.org/10.29210/07essr500300