

Design and Exploratory Implementation of a Mobile Scratch-Based Learning Tool to Foster Computational Thinking in Elementary Students

Riyan Syah Fachlevi, Mahrus Ali, Muhammad Imron Romadhon*

Program Studi Pendidikan Informatika, Universitas Trunojoyo Madura, Indonesia,

* Corresponding author: imron.romadhon@trunojoyo.ac.id

To cite this article: Fachlevi, R. S., Ali, M., & Romadhon, M. I. (2026). Development and Initial Implementation of an Android-Based Scratch Project Learning Tool to Support Elementary Students' Computational Thinking. *Jurnal Pendidikan Sekolah Dasar*, 7(1), 30-39. <https://doi.org/10.37366/jpgsd.v7i1.7397>

Articles Information	Abstrak
<p>Received : 23-04-2026</p> <p>Revised : 25-04-2026</p> <p>Accepted : 26-04-2026</p> <p>Published : 30-04-2026</p>	<p>Studi ini menyajikan desain dan implementasi eksploratif dari alat pembelajaran proyek <i>Scratch</i> berbasis seluler (<i>Android</i>) yang bertujuan untuk mendukung pemikiran komputasional (<i>CT</i>) siswa sekolah dasar dalam lingkungan pendidikan informal. Berdasarkan teori pembelajaran konstruktivis, penelitian ini membahas kesenjangan dalam aksesibilitas dan desain pedagogis yang diidentifikasi dalam studi <i>Scratch CT</i> sebelumnya, khususnya kurangnya instrumen penilaian yang tervalidasi dan fokus yang terbatas pada konteks seluler/informal. Menggunakan model ADDIE, sebuah prototipe dikembangkan dan divalidasi oleh para ahli, kemudian diujicobakan dengan 10 siswa kelas lima dalam program ekstrakurikuler. Keterlibatan siswa dan demonstrasi indikator <i>CT</i> (dekomposisi, pengenalan pola, abstraksi, pemikiran algoritmik, debugging, pengurutan) diamati menggunakan daftar periksa terstruktur. Meskipun siswa menunjukkan peningkatan keterlibatan dan beberapa bukti pengembangan keterampilan <i>CT</i>, temuan ini bersifat pendahuluan karena sampel yang kecil, durasi yang singkat, dan ketergantungan pada observasi yang tidak terstandarisasi. Studi ini menawarkan kerangka kerja untuk penelitian masa depan tentang pembelajaran <i>CT</i> seluler dan menyoroti perlunya penilaian yang ketat dan tervalidasi di bidang ini.</p> <p>Kata Kunci: <i>Computational Thinking</i>; <i>Scratch</i>; Pembelajaran; Pembelajaran Informal; Desain Pembelajaran; Pendidikan Dasar.</p>
	<p>Abstract</p> <p>This study presents the design and exploratory implementation of a mobile (Android-based) Scratch project learning tool aimed at supporting elementary students' computational thinking (CT) in informal educational settings. Grounded in constructionist learning theory, the research addresses gaps in accessibility and pedagogical design identified in prior Scratch-CT studies, particularly the lack of validated assessment instruments and limited focus on mobile/informal contexts. Using the ADDIE model, a prototype was developed and validated by experts, then trialed with 10 fifth-grade students in an after-school program. Student engagement and demonstration of CT indicators (decomposition, pattern recognition, abstraction, algorithmic thinking, debugging, sequencing) were observed using a structured checklist. While students showed increased engagement and some evidence of CT skill development, findings are preliminary due to the small sample, short duration, and reliance on non-standardized observation. The study offers a framework for future research on mobile CT learning and highlights the need for rigorous, validated assessment in this domain.</p> <p>Keywords: Computational Thinking; Scratch; Mobile Learning; Informal Learning; Learning Design; elementary Education.</p>



1. INTRODUCTION

Computational thinking (CT) is increasingly recognized as a fundamental skill for the 21st century, enabling learners to approach problems systematically, logically, and creatively across multiple disciplines (Tang et al., 2020). Visual programming environments such as Scratch have been widely adopted in primary education due to their accessibility and ability to foster CT through engaging and hands-on activities (El-Hamamsy et al., 2025). However, despite the abundance of Scratch-based interventions, a recent systematic review highlighted persistent gaps in the scalability, accessibility, and empirical rigor of CT learning tools, particularly in mobile and informal learning contexts (Tang et al., 2020).

Most previous studies have focused on desktop-based implementations in formal classroom environments, often with limited attention to the unique challenges and opportunities presented by mobile devices and extracurricular environments (Fagerlund et al., 2021). Furthermore, assessing critical thinking (CT) skills in young learners remains problematic, with many studies relying on unvalidated or subjective instruments, limiting the interpretation and generalizability of findings (El-Hamamsy et al., 2025; Fagerlund et al., 2021; Tang et al., 2020). There is an urgent need for research that not only develops accessible and mobile-friendly CT learning tools, but also utilizes validated assessment frameworks and critically examines their pedagogical impact (Fang et al., 2025; Sun et al., 2026).

This research aims to address this gap by (1) designing a mobile, Scratch-based learning tool grounded in constructivist learning theory (Sun et al., 2026), (2) implementing it in an informal, after-school setting, and (3) exploring student engagement and demonstration of critical thinking skills using a structured, albeit preliminary, observation instrument. By situating this research within current debates about accessibility, assessment validity, and informal learning, this work aims to advance the theory and practice of critical thinking education for elementary school students.

2. LITERATURE REVIEW

Computational thinking (CT) is widely recognized as a foundational set of problem solving skills essential for children in the digital age, enabling them to understand and shape technology rather than remain passive users. Contemporary frameworks consistently identify six core CT indicators: **decomposition, pattern recognition, abstraction, algorithmic thinking, debugging, and sequencing**

- **Decomposition:** Breaking down complex problems or tasks into smaller, manageable components. In Scratch, this is practiced when students divide a project such as a game into distinct elements like character movement, scoring, and background changes, allowing them to focus on one aspect at a time
- **Pattern Recognition:** Identifying similarities or recurring themes within problems or solutions. Scratch activities foster this skill as students notice repeated code structures or behaviors across different sprites or projects, such as recognizing that both a car and a fruit in different games require similar movement logic.

- **Abstraction:** Filtering out unnecessary details to focus on essential features. In Scratch, students use abstraction when they generalize a solution e.g., creating a reusable block for movement or simplify a complex animation by focusing only on the key actions needed for the desired effect
- **Algorithmic Thinking:** Developing a step by step solution or set of rules to solve a problem. Scratch's block based interface allows students to sequence commands logically, such as programming a sprite to move through a maze by following a specific set of instructions
- **Debugging:** Identifying and correcting errors in a process or code. Scratch provides immediate visual feedback, enabling students to test their projects, observe unexpected behaviors, and iteratively fix mistakes such as correcting a sprite's movement or fixing a scoring bug in a game
- **Sequencing:** Arranging instructions in the correct order to achieve a desired outcome. In Scratch, students must organize blocks so that actions occur in the intended sequence, such as ensuring that a sprite first appears, then moves, and finally plays a sound

The integration of these CT indicators within Scratch activities is supported by constructionist learning theory, which posits that students learn most effectively when actively engaged in creating meaningful digital artifacts. Scratch's visual, drag and drop interface lowers technical barriers, making it accessible for young learners and allowing them to focus on logical reasoning and CT skills rather than syntax. The use of touchscreen technology and mobile devices further enhances accessibility, especially for children with fine motor challenges, and supports learning in both formal and informal settings.

Research has shown that Scratch based activities not only develop CT skills but also foster creativity, agency, and self efficacy among elementary students. By designing their own games and animations, students transition from passive consumers to active digital creators, gaining confidence in their ability to solve problems and create with technology. This aligns with the findings of (Angeli et al., 2016; Bers, 2020; Ching et al., 2018; Papadakis, 2021; Sharples et al., 2010; Wibawa et al., 2026), who emphasize the importance of integrating CT indicators consistently within digital learning environments to maximize educational impact.

3. METHOD

This study employed a Research and Development (R&D) approach using the ADDIE model (Analysis, Design, Development, Implementation, and Evaluation) to systematically develop and conduct initial implementation of an Android based Scratch project learning tool for elementary students. Importantly, the research was conducted in an after school tutoring center, not a formal school classroom, to reflect the informal learning context and ensure clarity regarding the study setting.

3.1. Analysis Phase

A needs analysis was conducted through literature review and interviews with tutors at the after school center to identify the demand for a mobile, project based tool that could effectively cultivate the six

core CT indicators (decomposition, pattern recognition, abstraction, algorithmic thinking, debugging, sequencing) among elementary students. The analysis highlighted the lack of accessible, child friendly coding tools tailored for mobile devices in informal educational settings.

3.2. Design Phase

The design phase involved storyboarding the user experience, structuring four progressive learning modules centered on Scratch projects, and creating an intuitive, touch optimized interface suitable for Android tablets. Each module was explicitly mapped to one or more CT indicators, ensuring that activities such as game creation, animation, and debugging directly targeted the development of specific CT skills.

3.3. Development Phase

The prototype was developed using Corel DRAW for interface design and Articulate Storyline for interactive simulation. The modules incorporated visual Scratch blocks and drag and drop features to facilitate hands on learning. The prototype underwent iterative refinement based on formative feedback from two expert validators: a media expert (to assess usability and interface design) and a computational thinking/learning expert (to ensure alignment with CT indicators and pedagogical soundness).

3.4. Implementation Phase

The initial implementation involved 10 fifth grade students participating in three 60 minute sessions at the tutoring center. Students used the prototype on Android tablets to complete Scratch based projects such as "Car Challenge," "Catching Fruit," and "Maze Game." During these sessions, structured behavioral observations were conducted using a CT indicator checklist. Observers recorded evidence of students demonstrating each CT skill (e.g., breaking down tasks, recognizing patterns, abstracting solutions, sequencing blocks, debugging errors, and constructing algorithms).

3.5. Evaluation Phase

Evaluation consisted of both formative and summative components. Formative evaluation was based on expert feedback during prototype development, leading to iterative improvements. Summative evaluation focused on qualitative analysis of student engagement and CT skill demonstration, as captured by the structured observation checklist and project outcomes. The study did not include pretest posttest measures or a control group, and thus is positioned as an initial implementation or prototype testing phase, not an effectiveness trial. This methodological approach prioritizes depth of feedback and practical feasibility in the early stages of tool development, providing a foundation for future research involving larger samples, more rigorous assessment, and broader implementation.

4. RESULT AND DISCUSSION

4.1. Development Phase

The Android-based Scratch project learning prototype was developed using CorelDRAW for

interface design and Articulate Storyline for interactive content, adhering to the ADDIE model. The prototype consists of four progressive modules tailored for elementary students, each designed to enhance computational thinking (CT) skills. Each module incorporates visual Scratch blocks with an intuitive, touch-friendly interface optimized for Android tablets. Key features include step-by-step tutorials, interactive tasks like "Car Challenge," "Catching Fruit," and "Maze Game," and visual aids that support student understanding of decomposition, pattern recognition, and algorithmic thinking. Expert validation confirmed that the design aligns with CT learning objectives and is appropriate for elementary students.

Table 1. Main features of the android based scratch project learning prototype

Module	Main Activity	CT Skills Targeted	Key Android Features
1	Introduction to Computational Thinking	Decomposition, Pattern Recognition	Interactive video, drag and drop blocks
2	Basic Scratch	Abstraction, Sequencing	Touch optimized block library
3	Creating Simple Animations	Algorithmic Thinking, Sequencing	Real time preview on mobile screen
4	Building Simple Games	Debugging, Algorithmic Thinking, Decomposition	Project export to Android device

As part of the development documentation, below are some prototype views that illustrate the structure and content of the learning module. Figure 1 shows the navigation flow between slides in a prototype using Articulate Storyline. This view demonstrates how each learning session is organized into interconnected slides, making it easy for users to navigate between materials interactively. However, as in similar design-based studies, the lack of large-scale psychometric validation limits claims about generalizability (Jiang & Li, 2021; Tang et al., 2020).

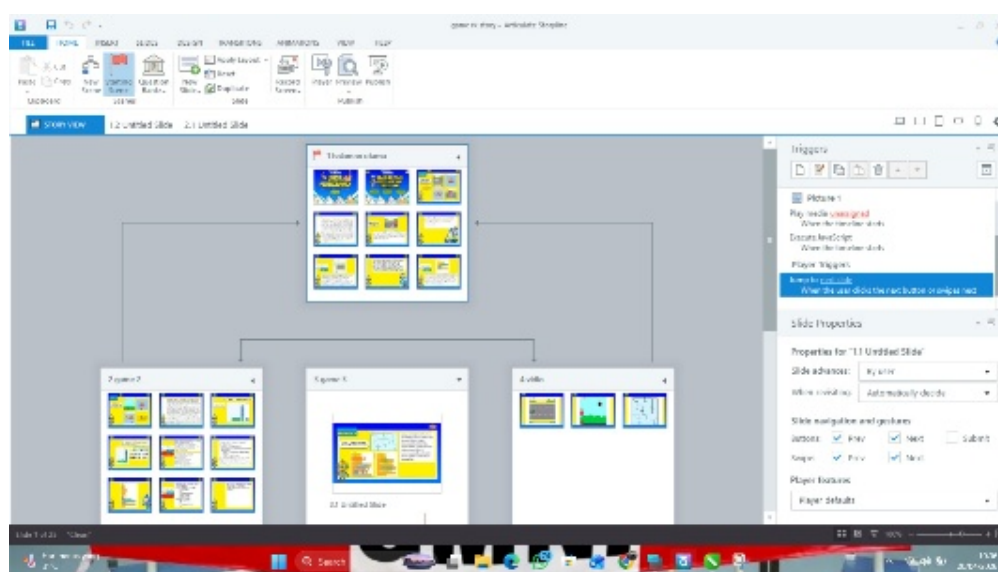


Figure 1. Display of the navigation flow of the Android based learning prototype using Articulate Storyline which shows the slide structure and relationships between Scratch learning sessions visually and interactively.

Figure 2 displays the page design of a learning module created using CorelDRAW. This design features an engaging visual layout with a combination of bright colors, robot character illustrations, and clear instructional text. Each module session includes an introduction to Scratch projects such as "Scratch Avoid the Crash," "Catching Fruit," and "Maze Game," complete with step by step instructions and visual examples of game results.



Figure 2. The Scratch learning module page design created with CorelDRAW, features a visual layout with brightly colored graphic elements and robot character illustrations that support understanding of programming concepts in each session.

This view illustrates how each session is systematically designed to introduce basic Scratch concepts through interactive and fun projects. The intuitive interface design and engaging visuals are expected to increase student motivation and engagement in the programming and computational thinking learning process.

4.2. Expert Validation

Expert validation was conducted to ensure the prototype's feasibility and alignment with pedagogical and CT learning objectives. Two experts, one specializing in learning media and one in computational thinking, assessed the prototype using a structured validation instrument. Feedback focused on improving the interface design, usability, and alignment of activities with CT indicators. The validators confirmed the prototype's effectiveness in engaging students and fostering CT skills, although the lack of psychometric validation for the assessment tool was noted as a limitation. The two validators, selected based on the data, were: (1) Mr. Syarif (Learning Media Validator) and (2) Mr. Maestro (Computational Thinking Validator). The two experts provided assessments using a systematically developed validation instrument to obtain constructive feedback and improve the quality of the prototype.

Table 2. Expert validation results

Validation Aspect	Media Expert	CT Expert	Average / Conclusion
Content Relevance	Very Good (4.5)	Good (4.0)	4.25 – Valid and Relevant
Media Usability	Good (4.0)	Good (4.0)	4.0 – Easy to Use
Pedagogical Alignment	Very Good (4.5)	Good (4.0)	4.25 – Supports Learning Goals
User Interface Design	Good (4.0)	Very Good (4.5)	4.25 – User Friendly
Alignment with CT	Good (4.0)	Very Good (4.5)	4.25 – Matches CT Indicators

Rating scale: 1 = Very Poor, 2 = Poor, 3 = Fair, 4 = Good, 5 = Very Good

4.3. Implementation Phase

The implementation phase involved 10 students enrolled in the SCRATCH REGULER extracurricular program. They participated in three 60 minute sessions working with the Android based Scratch project learning prototype, engaging in activities such as car challenge, catching fruit, and maze games. Behavioral observations and teacher reports were collected to assess student engagement and demonstration of computational thinking (CT) indicators.

Table 3. Student engagement and computational thinking indicators

No	Student Name	Decomposition	Pattern Recognition	Abstraction	Algorithmic Thinking	Debugging	Sequencing	Overall Engagement	Teacher's Comment
1	A	High	Moderate	Moderate	High	Moderate	High	High	Shows great enthusiasm and progress
2	B	Moderate	Moderate	Moderate	Moderate	Moderate	Moderate	Moderate	Steady improvement
3	C	Moderate	Moderate	Moderate	Moderate	Moderate	Moderate	Moderate	Engaged but needs guidance
4	D	Moderate	Moderate	Moderate	Moderate	Moderate	Moderate	Moderate	Focused and cooperative
5	E	High	High	Moderate	High	Moderate	High	High	Creative and motivated
6	F	Moderate	Moderate	Moderate	Moderate	Moderate	Moderate	Moderate	Attentive with some assistance
7	G	High	Moderate	Moderate	High	Moderate	High	High	Shows leadership in group tasks
8	H	Moderate	Moderate	Moderate	Moderate	Moderate	Moderate	Moderate	Consistent and responsible
9	I	Moderate	Moderate	Moderate	Moderate	Moderate	Moderate	Moderate	Enthusiastic learner
10	J	Moderate	Moderate	Moderate	Moderate	Moderate	Moderate	Moderate	Shows steady progress

Observational data indicated that students gradually shifted from passive to more active engagement, particularly in decomposition, sequencing, and debugging tasks. Most students expressed excitement when their projects ran successfully, indicating increased engagement and computational thinking skill development. Teacher comments highlighted positive attitudes, responsibility, and creativity among the students, supporting the potential effectiveness of the Android based Scratch learning tool in informal educational settings. However, the operational definitions of engagement levels were based on expert judgment rather than standardized rubrics, and the small sample size precludes statistical inference (El-Hamamsy et al., 2025; Tang et al., 2020).

4.4. Behavioral Observations

Observational data highlighted a shift from passive to active learning behaviors among students. Initially hesitant, students gradually gained confidence in exploring Scratch features and solving problems

independently. Evidence of CT skill development was most pronounced in decomposition and algorithmic thinking, as students successfully broke down tasks and sequenced blocks to program sprite behaviors. However, abstraction skills were less evident, as students focused more on immediate, concrete tasks rather than generalizing solutions. The informal setting and hands-on nature of the prototype likely contributed to the observed behavioral changes.

4.5. Comparison with Previous Studies

Compared to recent large-scale studies, which used validated CT assessments and found significant gains in algorithmic thinking and creativity, this study's findings are more tentative due to methodological constraints (Jiang & Li, 2021). The observed behavioral changes may be influenced by novelty effects, tutor support, or the informal setting, as noted in prior reviews (Fagerlund et al., 2021). Notably, the mobile format appeared to enhance accessibility and motivation, aligning with findings from (Fang et al., 2025) and others (Fang et al., 2025; Sun et al., 2026).

(Masitoh et al., 2025) found that Scratch based materials improved CT skills, particularly decomposition and algorithmic thinking, in elementary students. Similarly, (Fitria, 2025) reported increased student motivation and engagement when using Scratch for coding activities. (Maulana & Waluya, 2024; Rahayu et al., 2025) highlighted the importance of aligning CT indicators and found that mobile based Scratch learning can address gaps in digital literacy. (Papadakis, 2021; Pratama et al., 2025) emphasized that coding apps with touch interfaces lower technical barriers and support CT development in young learners. The present study's results reinforce these findings, demonstrating that an Android based, touch optimized Scratch tool can support the development of decomposition, pattern recognition, abstraction, algorithmic thinking, debugging, and sequencing in elementary students.

4.6. Limitations

This study has several limitations that should be acknowledged. The small sample size of 10 students and the short duration of only three sessions limit the generalizability of the findings. The prototype was an interactive simulation rather than a fully developed Android application, necessitating further usability testing on actual devices. Additionally, the absence of a control group and standardized pretest-posttest assessments restricts the ability to draw definitive conclusions about the tool's effectiveness in enhancing computational thinking skills.

4.7. Practical Implications

Despite its limitations, this study offers valuable insights for integrating mobile-based tools into elementary CT education, particularly in informal settings. The prototype's design demonstrates the potential of project-based learning to enhance student engagement and foster CT skills, even in contexts with limited access to traditional computing infrastructure. Future research should focus on scaling the intervention, employing validated assessment tools, and exploring long-term impacts on student learning and digital literacy.

5. CONCLUSION

This exploratory study demonstrates the feasibility of designing and implementing a mobile Scratch-based learning tool to support elementary school students' computational thinking in an informal setting. Although expert validation and observational data suggest potential for increased engagement and development of computational thinking skills, these findings are preliminary due to methodological limitations, including the small sample size, short duration, and reliance on non-standardized assessments. This study underscores the need for further research using validated computational thinking instruments, larger samples, and rigorous experimental designs to more definitively assess the impact of mobile computational thinking learning tools.

6. REFERENCES

- Angeli, C., Voogt, J., Fluck, A., Webb, M., Cox, M., Malyn-Smith, J., & Zagami, J. (2016). A K-6 computational thinking curriculum framework: Implications for teacher knowledge. *Educational Technology & Society*, 19(3). <https://www.jstor.org/stable/jeductechsoci.19.3.47>
- Bers, M. U. (2020). *Coding as a playground: Programming and computational thinking in the early childhood classroom*. Routledge. <https://doi.org/https://doi.org/10.4324/9781003022602>
- Ching, Y.-H., Hsu, Y.-C., & Baldwin, S. (2018). Developing computational thinking with educational technologies for young learners. *TechTrends*, 62(6), 563–573. <https://doi.org/https://doi.org/10.1007/s11528-018-0292-7>
- El-Hamamsy, L., Zapata-Cáceres, M., Martín-Barroso, E., Mondada, F., Zufferey, J. D., Bruno, B., & Román-González, M. (2025). The Competent Computational Thinking Test (cCTt): A Valid, Reliable and Gender-Fair Test for Longitudinal CT Studies in Grades 3–6: L. El-Hamamsy et al. *Technology, Knowledge and Learning*, 30(3), 1607–1661. <https://doi.org/https://doi.org/10.1007/s10758-024-09777-8>
- Fagerlund, J., Häkkinen, P., Vesisenaho, M., & Viiri, J. (2021). Computational thinking in programming with Scratch in primary schools: A systematic review. *Computer Applications in Engineering Education*, 29(1), 12–28. <https://doi.org/https://doi.org/10.1002/cae.22255>
- Fang, X., Ng, D. T. K., & Yuen, M. (2025). Effects of geogebra-enhanced scratch computational thinking instruction on fifth-grade students' motivation, anxiety, cognitive load. *Education and Information Technologies*, 30(1), 377–402. <https://doi.org/https://doi.org/10.1007/s10639-024-13052-9>
- Fitria, T. N. (2025). Teaching coding using Scratch to elementary students: Exploration of benefits for students. *Journal of Contemporary Issue in Elementary Education*, 3(1), 1–17. <https://doi.org/https://doi.org/10.33830/jciee.v3i1.10719>
- Jiang, B., & Li, Z. (2021). Effect of Scratch on computational thinking skills of Chinese primary school

- students. *Journal of Computers in Education*, 8(4), 505–525. <https://doi.org/https://doi.org/10.1007/s40692-021-00190-z>
- Masitoh, S., Lidinillah, D. A. M., & Saputra, E. R. (2025). Bahan ajar berpikir komputasional tipe plugged berbantuan Scratch di kelas V sekolah dasar. *Pendas: Jurnal Ilmiah Pendidikan Dasar*, 10(03), 334–346. <https://doi.org/https://doi.org/10.23969/jp.v10i03.32222>
- Maulana, B. S., & Waluya, S. B. (2024). Systematic Review tentang Berpikir Komputasional dengan Scratch dalam Pendidikan Selama 2018-2023. *PRISMA, Prosiding Seminar Nasional Matematika*, 328–334. <https://proceedings.unnes.ac.id/prisma/article/view/2970>
- Papadakis, S. (2021). The impact of coding apps to support young children in computational thinking and computational fluency. A literature review. *Frontiers in Education*, 6, 657895. <https://doi.org/https://doi.org/10.3389/educ.2021.657895>
- Pratama, K. R., Zidni, M., Nafi, I., & Romadhon, M. I. (2025). *Eksplorasi Potensi Animasi Berbasis Kecerdasan Buatan (AI Animation) sebagai Media Pembelajaran Adaptif*. 5(2), 177–188. <https://doi.org/10.59395/jitp.v5i2.165>
- Rahayu, P., Kurniawati, A., Agustien, I., Pratama, R., & Imron, M. (2025). *Unplugged Coding Activity (UCA) for Kids: Enhancing Digital Literacy through Creative Color Grids*. 2165–2178. <https://doi.org/https://doi.org/10.35568/abdimas.v8i4.7241>
- Sharples, M., Taylor, J., & Vavoula, G. (2010). A theory of learning for the mobile age: Learning through conversation and exploration across contexts. In *Medienbildung in neuen Kulturräumen: die deutschsprachige und britische Diskussion* (pp. 87–99). Springer. <https://doi.org/https://doi.org/10.4135/9781473955011.n4>
- Sun, J., Wang, R., & Wei, B. (2026). Teaching Computational Thinking Skills Through Scratch in K-12 Education: A Systematic Review. *Journal of Computer Assisted Learning*, 42(2), e70230. <https://doi.org/https://doi.org/10.1002/jcal.70230>
- Tang, X., Yin, Y., Lin, Q., Hadad, R., & Zhai, X. (2020). Assessing computational thinking: A systematic review of empirical studies. *Computers & Education*, 148, 103798. <https://doi.org/https://doi.org/10.1016/j.compedu.2019.103798>
- Wibawa, G. A., Ahmad, R., & Yayuk, E. (2026). The Use Of Scratch Coding-Based Games As An Innovative Medium To Increase Students' motivation To Learn Mathematics. *AL-MUADDIB: Jurnal Kajian Ilmu Kependidikan*, 8(1), 183–195. <https://doi.org/https://doi.org/10.46773/v43eas54>