

Implementation of TPS Scratch Module with Bloom Leveling and Digital activity monitoring

Muhammad Faiz Abdillah^{1*}, Wahyu Sakti Gunawan Irianto²

¹ Informatika/ Teacher Professional Education Program, Universitas Negeri Malang, Malang, Indonesia

² Informatika/ Teacher Professional Education Program, Universitas Negeri Malang, Malang, Indonesia

*Corresponding author e-mail: muhammad.faiz.2531537@students.um.ac.id

Abstract

Limited computer availability can create challenges in implementing computer-based learning, particularly when students share learning devices. This study aimed to implement and examine a Scratch e-module that integrated Think-Pair-Share (TPS), Bloom's Taxonomy-based leveling, and a digital activity monitoring system in Informatics learning. The study used an implementation-based descriptive approach involving 30 Grade IX students in class 9B of SMPN 28 Malang on 31 August 2026. Students were organized into 15 pairs based on a brief diagnostic activity, and learning progressed through six levels representing cognitive processes from remembering to creating. The implementation combined individual thinking, paired discussion, progressive challenges, retry opportunities, scoring, and a final Scratch project. Five pairs involving 10 students reached Level 6 and completed the final project, with an average project score of 90.2. Fourteen pairs reached at least Level 4, while two pairs reached Level 5 and one pair reached Level 3. The system recorded 10 flagged activity events involving six pairs, consisting of tab switching, copy-paste, multiple-window, and idle events. The highest-ranked student obtained the MVP position with no flagged activity events. The findings indicate that the integrated module can organize progressive cognitive tasks, facilitate peer collaboration under a one-computer-per-two-students condition, and provide digital activity monitoring. The implementation also shows that limited instructional time can constrain completion of the highest cognitive level, indicating the need for additional project time in future implementation.

Keywords: Scratch e-module; Think-Pair-Share; Bloom's Taxonomy; leveling; digital activity monitoring.

Received: 15 September 2026

Accepted: 9 Oktober 2026

Published: 12 Oktober 2026

1. INTRODUCTION

Computer-based learning in Informatics requires access to digital devices, yet classroom conditions do not always provide one device for every student. When two students must share one computer, learning activities need to be designed so that device sharing becomes a structured collaboration opportunity rather than simply a technical limitation. In programming learning, students also need opportunities to reason about instructions, test solutions, identify errors, and develop products progressively. Scratch provides a visual programming environment that can support these activities because learners can construct programs through blocks and immediately observe their results (Resnick et al., 2009). Recent evidence has reported positive associations between Scratch-based learning and student engagement and academic performance (Belessova et al., 2024), and systematic reviews have linked Scratch to the development of computational thinking in K–9 and K–12 settings (Fagerlund et al., 2021; Montiel & Gómez Zermeño, 2021; Nurhidayat & Kusumah, 2025; Zhang & Nouri, 2019), although the reported contexts have largely focused on other educational levels and learning settings.

Think-Pair-Share (TPS) provides a pedagogical structure that is particularly relevant to shared-device learning. The model, originally described by Lyman (1981), separates learning into individual thinking, paired discussion, and sharing of ideas. Recent studies in Informatics and computer studies have reported positive learning outcomes associated with TPS. Hodri and Hasanah (2025), for example, reported improved student achievement through TPS in Informatics learning, while Anwar et al. (2025) found that TPS supported conceptual understanding and active participation in computer systems learning. Research in secondary computer studies has likewise reported higher achievement for students

taught with TPS than with lecture-based instruction (Chukwuemeka et al., 2025; Onyebuchi et al., 2025). These findings suggest that TPS can provide a useful structure for collaborative learning, but they do not by themselves address how TPS can be embedded into a digital module that progressively controls cognitive challenge.

Bloom's Taxonomy, as revised by Anderson and Krathwohl (2001), offers a framework for organizing learning tasks from lower- to higher-order cognitive processes. In digital learning, the progression can be translated into activities that move from identifying and explaining concepts to applying, analyzing, evaluating, and creating. Zainuddin et al. (2023) demonstrated that gamification and Bloom's Digital Taxonomy can be implemented together and identified learning engagement, understandable goals, friendly competition, and interactive feedback as important themes. More recently, Omairan and Tolah (2026) developed a reward-driven Bloom's Taxonomy framework combining cognitive progression with gamification. These studies establish that Bloom-based progression and game-oriented digital features can coexist, meaning that the contribution of the present study should not be framed simply as the first combination of Bloom leveling and gamification..

Digital learning also introduces challenges related to the integrity of student activity. Online assessment research has documented forms of cheating such as using unauthorized resources, copying and pasting, switching to external resources, and exploiting digital environments. Sevnarayan and Maphoto (2024) emphasize that online assessment requires strategies that combine pedagogical and technological measures to strengthen integrity. Alexandron et al. (2023), using learning analytics from 559 verified learners in MOOCs, showed that assessment design can influence both cheating and learner engagement. Technology-based proctoring has also been explored through more complex approaches such as computer vision (Mansoor et al., 2024), but such approaches may require additional infrastructure and raise privacy and technical considerations.

Existing studies therefore provide evidence for Scratch learning, TPS, Bloom-based digital progression, gamification, and digital assessment integrity, but these components are commonly investigated separately or in different contexts. A practical gap remains in the implementation of an integrated Scratch learning module that uses Bloom-level progression while retaining TPS collaboration and monitoring digital activity in a real junior secondary Informatics classroom with one computer shared by two students. The present study addresses this implementation gap. Its contribution is not the claim that any single component is novel, nor evidence that the combination improves learning, but a descriptive documentation of how these components operate together under a constrained classroom condition.

This study aimed to describe the implementation of the TPS Scratch module with Bloom leveling and digital activity monitoring in Grade IX Informatics learning. Specifically, the study examined (1) student progression across C1-C6 levels, (2) implementation of Think-Pair-Share under shared-device conditions, (3) completion and quality of the C6 project, and (4) the pattern of flagged digital activity events recorded by the system.

2. METHOD

This study used a descriptive implementation-based approach to document the use of a Scratch e-module in an authentic Informatics classroom. The implementation was conducted in class 9B of SMPN 28 Malang on Monday, 31 August 2026, during one 2-JP learning session (80 minutes). The participants were 30 Grade IX students organized into 15 pairs. Pairs are formed by the teacher based on diagnostic test scores administered prior to the session. Students with higher scores are paired with those with lower scores, while students with similar scores are paired together. Because the study focused on implementation rather than causal effectiveness, the analysis describes observed progression, completion, scores, retries, time, project results, and recorded digital activity events without claiming that the module caused an improvement compared with a control condition.

Table 1. *research details*

Characteristic	Description
School/Class	SMPN 28 Malang, Grade IX B
Implementation date	31 August 2026
Duration	2 JP (80 minutes)
Participants	30 students
Pairing	15 pairs; pairing informed by a brief diagnostic activity
Learning condition	One computer shared by two students
Learning medium	Scratch e-module with six progressive levels
Learning model	Think-Pair-Share (TPS)
Cognitive progression	C1–C6 based on Bloom's Taxonomy

The module organized learning into six levels corresponding to the cognitive progression C1 Remember, C2 Understand, C3 Apply, C4 Analyze, C5 Evaluate, and C6 Create. The activities addressed Scratch interface and block functions, pseudocode, movement, logic, loops, broadcast, variables, debugging, evaluation, and product development. Each level presents a challenge that must be completed. If the challenge is not successfully completed, the pair of students may try again up to three times. There is no score deduction for retries, but no bonus points are awarded.

The TPS structure was adapted to the 80-minute classroom session. The Think phase occurred during the first 15 minutes, when students individually analyzed the diagnostic or initial challenge before discussing their answer. The Pair phase occupied approximately 45 minutes, during which each pair worked through the e-module and discussed challenges while sharing one computer. The Share phase occupied the final 20 minutes and was represented by demonstrations and short question-and-answer sessions from the three highest-ranked MVP students/groups. Teacher scaffolding and feedback were provided throughout the activity.

The final C6 activity required students to create a Scratch-based product, such as an educational animation, game, or interactive storyboard. Because the 80-minute classroom period was insufficient for all pairs to complete the final coding and visual-design stage, unfinished C6 work could be continued outside the classroom. In the reported implementation, five pairs completed the C6 project. The following is the assessment rubric used in this study.

Table 2. *E-module Assessment Rubric*

Assessment Aspect	Excellent (4)	Good (3)	Satisfactory (2)	Needs Improvement (1)
Basic Understanding & Interface (C1–C2)	Answers all C1–C2 questions accurately with very clear explanations.	Answers most C1–C2 questions accurately.	Answers C1–C2 questions, but there are still several conceptual errors.	Many answers are incorrect or left blank.
Pseudocode Design (C3)	Pseudocode is highly coherent, logically precise, and ready for implementation.	Pseudocode is coherent, but has minor flow inconsistencies.	Pseudocode lacks structure and is confusing to execute.	Unable to construct pseudocode or the logic flow is completely wrong.
Bug Analysis / Debugging (C4)	Able to identify bugs and provide fixed solutions with high accuracy.	Able to identify bugs, but the proposed fix is less accurate.	Only able to identify some bugs without knowing how to fix them.	Unable to identify logical errors in the code.

Peer Review Evaluation / PAIR (C5)	Provides critical, solution-oriented evaluations and actively assists in fixing partner's program.	Provides reasonably useful input for the partner's program.	Feedback provided is minimal and lacks substance.	Does not perform evaluation on the partner's work.
Algorithm Alignment (C6)	The project runs 100% according to the pseudocode designed in the worksheet.	The project runs according to the pseudocode, with 1 minor error.	Several features do not align with the initial pseudocode.	The project does not align with the pseudocode at all.
Command Structure (C6)	Uses a combination of movement, looping (repeat/forever), and branching (if-then) without errors.	Uses all 3 command structures, but 1 logic element is not yet optimal.	Uses only 2 types of command structures.	Uses only 1 type of command structure (e.g., movement only).
Interaction (C6)	Broadcast and Variable features (Score/Timer/Lives) function seamlessly across objects.	Uses broadcast and variables, but there is a minor bug in integration.	Implements only one feature (only Variables or only Broadcast).	Does not use broadcast or variable features.
Creativity & User Experience (C6)	Visual design is appealing, game/animation flow is interactive, and concepts are original.	Appearance is neat and the game/animation runs well.	Simple appearance and interaction remains rigid.	Appearance is very disorganized and the program crashes frequently.

The final score was calculated by combining the worksheet and project scores and converting the total into a percentage. The calculation used the following formula:

$$\text{Final Score} = \left(\frac{\text{Worksheet Score} + \text{Project Score}}{32} \right) \times 100$$

The resulting scores were classified into four proficiency categories, as presented in Table 3.

Table 3. *Score range*

Score Range	Grade Category	Description
90 – 100	A	Highly Proficient / Expert
80 – 89	B	Proficient
70 – 79	C	Moderately Proficient / Competent
< 70	D	Needs Intervention / Remedial

In addition to documenting learning progress and assessment performance, the e-module incorporated browser-based activity monitoring to identify digital behaviors that were inconsistent with the established rules of the learning session. The monitoring system recorded four categories of activity: tab switching, copy-and-paste actions, multiple-window activity, and prolonged inactivity. Tab or window switching was identified through browser focus-loss or visibility events, copy-and-paste activity through clipboard or paste-related events, and prolonged inactivity through the module's idle-detection mechanism.

Before the session began, students were informed about the types of activities being monitored and the consequences associated with repeated warnings. The configured response system combined warnings, activity-point deductions, and a temporary restriction on access to the learning screen. Specifically, the first and second warnings each resulted in a five-point reduction in activity points. Following a third warning, the learning screen was frozen for 30 seconds before the student could resume the activity. The system also recorded detected events and applied the corresponding

responses. The configured rules included restrictions on opening other browser tabs, right-clicking, and using specified keyboard shortcuts.

These monitoring mechanisms were treated as tools for identifying and discouraging potentially noncompliant digital activity rather than as evidence that all forms of cheating had been prevented. Accordingly, the analysis reports the activity events flagged by the system and distinguishes the number of recorded events from the number of pairs affected. The findings are therefore interpreted as records of detected activity within the monitoring system's scope, not as a comprehensive measure of all possible academic misconduct.

Data were obtained from the e-module's learning records and the classroom implementation records. The collected variables included the highest level reached by each pair, assessment scores, the number of attempts required to complete challenges, working time, progress across cognitive levels, MVP status, flagged digital activity events and their types, and C6 project scores.

The data were analyzed descriptively using frequencies, percentages, averages, and comparisons across pairs. Progression through C1–C6 was summarized according to the number of students or pairs completing each level, with the unit of analysis specified for each indicator. Assessment performance was described using the recorded worksheet and project scores, while completion of the C6 activity was reported separately to account for the additional time required to develop the final product. Project quality was summarized using the assessment rubric presented in Table 2.

For the digital activity monitoring component, each logged event was counted as one flagged activity event, while the number of affected pairs was calculated separately. This distinction was maintained to avoid conflating the frequency of detected events with the number of students or pairs involved. Pair numbers rather than student names were used in the analysis and reporting to preserve student confidentiality while retaining the structure of the implementation data.

Given the descriptive implementation-based design, the analysis focused on documenting learning progression, task completion, assessment performance, project development, and recorded digital activity. The study did not use a control group or a comparative experimental condition. Therefore, the findings describe what occurred during the implementation and are not interpreted as evidence of the causal effectiveness of the e-module.

The implementation was authorized by the school and conducted with parental consent and student agreement. Participants were informed that they could withdraw from the study without academic consequences. Digital activity monitoring was restricted to browser-based interactions with the e-module during the learning session. No webcam footage, screen recordings, keystroke content, or clipboard contents were collected.

The learning records were stored in a database and used exclusively for research purposes. Student names were excluded from the manuscript, and pair codes were used in place of personally identifying information. These measures were intended to maintain confidentiality while allowing the learning and implementation records to be analyzed systematically.

3. RESULTS AND DISCUSSION

Present the research findings using tables or figures where appropriate; every table and figure must be discussed in the text, not merely displayed. The discussion must interpret the findings in relation to relevant theory and previous research — not merely restate the data. The discussion should: (1) answer the research questions; (2) explain how the findings were obtained; (3) interpret the findings; (4) relate the findings to established knowledge; and (5) propose new theoretical insights where relevant.

3.1 Overall Learning Progression

The implementation produced different levels of progression across the 15 pairs. All pairs reached C1, C2, and C3, 14 pairs reached C4, seven pairs reached C5, and five pairs reached C6. Thus,

participation narrowed as cognitive demand increased. At the individual level, all 30 students reached C1–C3, 28 students reached C4, 14 students reached C5, and 10 students reached C6. These figures describe progression through the module's levels; they do not constitute independently validated evidence of mastery of each Bloom cognitive level.

Table 4. *Overall Learning Progression*

Cognitive level	Cognitive process	Students	Students (%)	Pairs	Pairs (%)
C1	Remember	30	100.0	15	100.0
C2	Understand	30	100.0	15	100.0
C3	Apply	30	100.0	15	100.0
C4	Analyze	28	93.3	14	93.3
C5	Evaluate	14	46.7	7	46.7
C6	Create	10	33.3	5	33.3

The pattern is consistent with the intended progressive structure: lower-level tasks were accessible to all pairs, whereas higher-level tasks required greater time, reasoning, debugging, and product development. The sharpest reduction occurred between C4 and C5, where attainment decreased from 93.3% to 46.7% of students. The C6 level was reached by 33.3% of students. This result should not be interpreted simply as failure at higher cognitive levels because the session was limited to 80 minutes and the C6 activity required product development. The time constraint is therefore an important contextual factor when interpreting the progression.

The result is consistent with the practical value of making cognitive progression visible in a digital module. Zainuddin et al. (2023) found that understandable learning goals and interactive feedback were among the themes associated with implementation of gamification and Bloom-based assessment. In the present implementation, the level structure made the progression from C1 to C6 explicit and allowed pairs to stop at the highest completed level. Omairan and Tolah (2026) likewise demonstrate that Bloom progression can be combined with reward-oriented digital learning features, although their context was cybersecurity training rather than junior secondary Informatics.

3.2 Pair-Level Performance, Retry, and Time

Table 5. *Pair-Level Performance, Retry, and Time*

Pair	Level	Score	Retry	Time (min)	C6 project	Flagged events
P1	C6	95	2	68	95	0
P2	C6	92	3	72	92	0
P3	C6	90	4	75	90	1
P4	C6	88	3	76	88	0
P5	C6	87	5	78	86	0
P6	C5	82	6	79	–	2
P7	C5	80	5	80	–	0
P8	C4	76	7	80	–	1
P9	C4	75	6	78	–	0
P10	C4	74	8	80	–	3
P11	C4	72	7	80	–	0
P12	C4	70	9	80	–	1

P13	C4	70	8	79	–	0
P14	C4	68	10	80	–	2
P15	C3	62	11	80	–	0

Table 2 shows that the highest-scoring pairs also tended to reach higher cognitive levels, although retries were not absent among successful pairs. Pair 1 reached C6 with a score of 95 after two retries, while Pair 5 reached C6 with a score of 87 after five retries. Conversely, Pair 15 recorded the highest number of retries (11) and reached C3 with a score of 62. These observations suggest that retry opportunities allowed students to remain in the learning sequence, but repeated retries did not guarantee progression to the next cognitive level.

The time data also show that pairs approaching C5 and C6 generally used most of the available session. Five C6 pairs used between 68 and 78 minutes. Many C4–C5 pairs reached 79–80 minutes without completing the next level. This provides direct implementation evidence that higher-level tasks were more time-demanding in this classroom setting. Rather than interpreting the lower C5–C6 attainment as a purely cognitive deficit, the finding points to the need for a longer project window when the same module is implemented again.

3.3 Think-Pair-Share under Shared-Device Conditions

The TPS model served two functions in the implementation. First, the Think phase ensured that students encountered the problem individually before receiving the partner's reasoning. Second, the Pair phase converted the one-computer-per-two-students condition into a structured collaboration arrangement. The pair did not merely share hardware; the learning design required discussion before progression through challenges. The Share phase then made the highest-performing work visible to the class through demonstrations by the three MVP-ranked students/groups.

This implementation is consistent with recent evidence that TPS can support learning in technology-related subjects. Hodri and Hasanah (2025) reported improved achievement through TPS in Informatics learning, while Anwar et al. (2025) reported improvements in conceptual understanding and active participation in computer systems learning. Chukwuemeka et al. (2025) also found that secondary students taught with TPS achieved higher outcomes than students taught through lecture methods. The present study differs in emphasis: rather than testing TPS against a control group, it documents how TPS can be embedded as the operating structure of a shared-device Scratch module.

The integration is particularly relevant to classrooms with limited devices. When only one computer is available for two students, an individual-only design can create waiting time or unequal participation. TPS provides a sequence in which thinking can occur without exclusive device access, pairing can be used for joint decision-making, and sharing can provide accountability at the end. This suggests that device scarcity does not necessarily have to be treated solely as a technical obstacle; it can be incorporated into a collaborative learning design.

3.4 C6 Project Completion and Product Quality

Five pairs, representing 10 students or 33.3% of the class, completed the C6 project. The mean C6 rubric score was 90.2 out of 100 among the five pairs that completed the project; this score does not represent the whole class and should not be generalized to the ten pairs that did not complete C6. The completed products were reported as educational animations or games and interactive storyboards based on programming logic. The products were described as functioning well and having logical story flows.

Table 6. C6 Project Completion and Product Quality

Indicator	Result
C6 pairs completed	5 of 15 pairs

Students involved	10 of 30 students
Completion percentage	33.3%
Mean C6 rubric score	90.2/100 (completing pairs only, n = 5)
Product forms	Educational animation, educational game, interactive storyboard
Main limitation	Insufficient classroom time for final coding/visual design

The relatively high mean project score among completed products suggests that these five pairs produced functioning products according to the rubric used. Because this subset may differ from the pairs that did not finish, the score should not be taken as an indicator of class-wide attainment. However, the completion rate must be interpreted together with the time constraint. The fact that five pairs completed the project while ten pairs did not finish the final coding/design stage indicates that C6 required substantially more production time than the preceding levels. Therefore, the module is more appropriately implemented with a dedicated project session or additional scheduled time if C6 completion for the whole class is the target.

The result also illustrates the distinction between cognitive progression and product completion. Reaching C4 or C5 indicates completion of module tasks designed to target analytical or evaluative processes, not independently verified mastery of those processes, while C6 requires the additional integration of planning, coding, debugging, and visual design. A module that reports only the final product could overlook the learning progression achieved by students who did not reach the final project within the available session.

3.5 Digital Activity Monitoring and Flagged Activity Events

Table 7. *Digital Activity Monitoring and Flagged Activity Events*

Event type.	Affected pairs	Events
Tab switching	3	3
Copy-paste text	1	2
Idle / no activity	1	3
Multiple windows	1	2
Total	6	10

The system recorded 10 flagged activity events involving six of the 15 pairs. Nine pairs had no flagged activity. Tab switching occurred in three pairs and accounted for three events. Copy-paste occurred in one pair with two events, idle/no-activity was recorded three times in one pair, and multiple-window activity occurred twice in one pair. The event count therefore represents system detections rather than a direct measure of intentional cheating in every case.

The digital activity monitoring mechanism was designed as a lightweight browser-level monitoring layer rather than a full remote-proctoring system. This distinction is important. A focus-loss event can indicate that a student left the active learning page, but it does not by itself prove that the student accessed unauthorized material. Similarly, inactivity may result from thinking or discussion rather than misconduct. Consequently, the system is best described as a digital activity monitoring and deterrence mechanism. This conservative interpretation is consistent with broader research showing that digital assessment integrity involves both behavioral and technological dimensions (Sevnarayan & Maphoto, 2024).

The result also highlights a useful balance between collaboration and integrity. Because the learning design explicitly permitted pair discussion, the system did not attempt to prohibit communication between the two students using the same device. Instead, it monitored activities that occurred outside

the intended active page or deviated from configured interaction rules. This is relevant for shared-device classrooms because a digital activity monitoring mechanism should not incorrectly classify legitimate peer collaboration as misconduct.

3.6 Integrated Interpretation and Research Contribution

The implementation demonstrates that the four components—Scratch, TPS, Bloom leveling, and digital activity monitoring—can operate as a single classroom workflow. Scratch provided the programming environment and immediate feedback; Bloom leveling structured cognitive progression; TPS structured individual and paired participation; and monitoring provided an activity-integrity layer. The combination is the central implementation contribution of this study.

The contribution should be distinguished from prior studies that have already combined gamification and Bloom's Taxonomy. Zainuddin et al. (2023) established that gamification and Bloom-based digital assessment can be implemented together, and Omainar and Tolah (2026) further demonstrated a reward-driven Bloom framework. Likewise, research has separately established the relevance of Scratch to engagement and learning outcomes (e.g., Belessova et al. (2024)) and TPS to Informatics or computer studies learning (Anwar et al., 2025; Chukwuemeka et al., 2025; Hodri & Hasanah, 2025). The present study therefore contributes an implementation configuration rather than a claim of component-level novelty.

A practical insight emerges from the shared-device condition: the pairing of students can be designed as a pedagogical structure rather than treated merely as a resource compromise. In this study, one computer for two students was aligned with TPS, while the module still maintained individual thinking, progressive challenge, retries, and activity monitoring. The resulting implementation model is therefore potentially useful for schools that have limited computer availability but still want students to experience structured programming activities.

A second insight concerns the relationship between monitoring and learning. The monitoring layer did not replace teacher supervision or determine whether cheating actually occurred. Instead, it generated observable events that could be reviewed in context. This is preferable to presenting every focus loss as confirmed cheating. The distinction also makes the approach more feasible for classroom use than highly invasive proctoring systems. Alexandron et al. (2023) showed that assessment design and learning analytics can influence cheating behavior, while Sevnarayan and Maphoto (2024) emphasize the need for multifaceted strategies to maintain assessment integrity.

Finally, the implementation reveals a scheduling implication. The module was completed through at least C4 by 14 of 15 pairs, but only seven pairs reached C5 and five pairs completed C6 within the available learning sequence. Because C6 required product creation, future implementation should allocate a dedicated project period or separate meeting. This limitation is not merely a weakness of the module; it provides a practical design requirement for using Bloom-levelled programming modules in short classroom sessions.

4. CONCLUSION

The implementation of the TPS Scratch module with Bloom leveling and digital activity monitoring in Grade IX Informatics learning showed that a shared-device classroom can be organized through a structured progression from C1 to C6. All 30 students reached C1–C3, 28 reached C4, 14 reached C5, and 10 reached C6. Five pairs completed the final Scratch project with a mean rubric score of 90.2 (completing pairs only). The TPS structure provided a practical way to coordinate individual thinking and paired collaboration when two students shared one computer, while the monitoring layer recorded 10 flagged events involving six pairs. The findings indicate that an integrated module can organize progressive cognitive tasks, peer collaboration, and activity monitoring, but they do not establish causal effectiveness because the study used a descriptive implementation design without a control group or pretest-posttest comparison. The results also represent a single classroom implementation involving 30

students in one 80-minute session, which limits generalization to other classes, schools, and time allocations. The main practical implication is that C6 project work should receive additional time in future implementations so that more students can complete the highest cognitive level. Future studies should test the module across multiple sessions and classes and examine learning outcomes using more robust designs, such as controlled or quasi-experimental comparisons with pretest-posttest measures and independently validated assessments of cognitive attainment.

AI USAGE STATEMENT

During the preparation of this manuscript, ChatGPT was used to assist with drafting, organization, language editing, and formatting based on data and implementation information provided by the author. The author reviewed the generated text, verified the reported implementation data, and takes full responsibility for the content of this publication. The Scratch e-module described in this study did not use generative artificial intelligence as part of its learning or assessment functions

REFERENCES

- Alexandron, G., Wiltrout, M. E., Berg, A., Gershon, S. K., & Ruipérez-Valiente, J. A. (2023). The effects of assessment design on academic dishonesty, learner engagement, and certification rates in MOOCs. *Journal of Computer Assisted Learning*, 39(1), 141–153. <https://doi.org/10.1111/jcal.12733>
- Anderson, L. W., & Krathwohl, D. R. (Eds.). (2001). *A taxonomy for learning, teaching, and assessing: A revision of Bloom's taxonomy of educational objectives*. Longman.
- Anwar, K., Prasetya, D. D., & Setiawan, A. Y. (2025). Implementation of the Think-Pair-Share cooperative learning model to improve conceptual understanding and active participation of Grade X-4 students at SMAN 7 Malang in computer systems. *Pendas: Jurnal Ilmiah Pendidikan Dasar*, 10(2), 1323–1335. <https://doi.org/10.23969/jp.v10i2.24835>
- Belessova, D., Ibashova, A., Zhidebayeva, A., Shaimerdenova, G., & Nakhipova, V. (2024). The impact of "Scratch" on student engagement and academic performance in primary schools. *Open Education Studies*, 6(1), 20220228. <https://doi.org/10.1515/edu-2022-0228>
- Chukwuemeka, N. R. N., Okafor, C. F., & Okeke, C. P. (2025). Effect of Think-Pair Share learning strategy on secondary school students achievement in computer studies: Implications for global development. *Journal of Education, Teaching and Learning Research*, 1(4), 81–93. <https://doi.org/10.64420/jetlr.v1i4.332>
- 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/10.1002/cae.22255>
- Hodri, & Hasanah, F. N. (2025). Improving student achievement through Think Pair Share in Informatics learning. *Indonesian Journal of Education Methods Development*, 20(1), 1–12. <https://doi.org/10.21070/ijemd.v20i1.974>
- Lyman, F. (1981). The responsive classroom discussion: The inclusion of all students. In A. S. Anderson (Ed.), *Mainstreaming digest* (pp. 109–113). University of Maryland College of Education.
- Montiel, H., & Gómez Zermelo, M. G. (2021). Educational challenges for computational thinking in K–12 education: A systematic literature review of "Scratch" as an innovative programming tool. *Computers*, 10(6), 69. <https://doi.org/10.3390/computers10060069>
- Nurhidayat, I., & Kusumah, Y. S. (2025). Systematic review of Scratch as a media for teaching computational thinking in mathematics. *Mathline: Jurnal Matematika dan Pendidikan Matematika*, 10(2), 581–590. <https://doi.org/10.31943/mathline.v10i2.869>
- Omairan, H., & Tolah, A. (2026). A reward-driven Bloom's Taxonomy framework for cybersecurity awareness: Integrating cognitive learning with gamification principles. *Information & Computer Security*, 1–29. <https://doi.org/10.1108/ICS-08-2025-0336>
- Onyebuchi, E. D., Ratanma, A. F., Chinyere, O. E., & Chukwunazo, O. M. (2025). Improving secondary school students achievement in computer studies through Think Pair Share learning strategy. *Journal Plus Education*, 38(Special Issue), 400–412. <https://doi.org/10.24250/jpe/si/2025/EDO/AFR/OEC/OMC>

- Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B., & Kafai, Y. (2009). Scratch: Programming for all. *Communications of the ACM*, 52(11), 60–67. <https://doi.org/10.1145/1592761.1592779>
- Sevnarayan, K., & Maphoto, K. B. (2024). Exploring the dark side of online distance learning: Cheating behaviours, contributing factors, and strategies to enhance the integrity of online assessment. *Journal of Academic Ethics*, 22, 51–70. <https://doi.org/10.1007/s10805-023-09501-8>
- Singh, T., Nair, R. R., Babu, T., & Duraisamy, P. (2024). Enhancing academic integrity in online assessments: Introducing an effective online exam proctoring model using YOLO. *Procedia Computer Science*, 235, 1399–1408. <https://doi.org/10.1016/j.procs.2024.04.131>
- Zainuddin, Z., Alba, A., Gunawan, T., Armanda, D., & Zahara, A. (2023). Implementation of gamification and Bloom's digital taxonomy-based assessment: A scale development study with mixed-methods sequential exploratory design. *Interactive Technology and Smart Education*, 20(4), 512–533. <https://doi.org/10.1108/ITSE-02-2022-0029>
- Zhang, L., & Nouri, J. (2019). A systematic review of learning computational thinking through Scratch in K-9. *Computers & Education*, 141, 103607. <https://doi.org/10.1016/j.compedu.2019.103607>