Bigint Computing Journal
Volume 4, Issue 2, 2026, pp. 109-118
DOI: https://doi.org/10.55537/bigint.v4i2.1950
ISSN: 3032-5374
https://journal.aira.or.id/bigint/index



Nearpod-Supported Learning for Algorithms and Programming: A Quasi-Experimental Study of Student Learning Outcomes

Adinda Mursallina Karjaya*, Dodi Ahmad Haerudin

Information and Communication Technology Education Study Program, Faculty of Education, Social Sciences, and Technology, Universitas Muhammadiyah Kuningan, Kuningan, Indonesia

ABSTRACT
Students' learning outcomes in Informatics remain an important concern, particularly when classroom instruction provides limited opportunities for active student participation. This study examined whether the change in learning outcomes differed between students who participated in Nearpod-supported Informatics learning and those who received conventional instruction. A quantitative approach was employed using a quasi-experimental method with a Nonequivalent Control Group Design. The study population comprised 660 Grade X students at SMK Negeri 1 Kuningan during the 2025/2026 academic year. A total of 68 students were selected through purposive sampling, consisting of 34 students in the experimental group and 34 students in the control group. Data were collected using a 10-item multiple-choice achievement test selected from an initial set of 20 items based on validity, reliability, difficulty, and discrimination analyses. Data were analysed using descriptive statistics, the Shapiro–Wilk normality test, Levene's test, and the Independent Samples t-Test. Because the two groups had different initial pretest scores, the analysis compared change scores calculated as posttest scores minus pretest scores. The results showed a statistically significant difference in learning outcome changes between the two groups, $t(66) = -3.322$, $p = 0.001$. The experimental group showed a mean increase of 46.47 points, compared with 30.00 points in the control group. The standardised mean difference was Cohen's $d = 0.81$. These findings indicate that students who participated in Nearpod-supported learning showed a greater change in learning outcomes than students who received conventional instruction. The findings provide empirical evidence regarding the use of Nearpod as an interactive learning medium in vocational Informatics education, particularly in Algorithms and Programming learning involving C and C++.
KEYWORDS: Nearpod; interactive learning media; learning outcomes; algorithms and programming; informatics education

*Corresponding author: Adinda Mursallina Karjaya, 221223100@mhs.upmk.ac.id

Article History: Received: 2026-08-21; Accepted: 2026-10-07

1. INTRODUCTION

The development of digital technology has influenced the implementation of learning by providing opportunities for teachers to combine instructional materials with interactive activities. In technology-supported learning, the availability of digital tools alone does not necessarily ensure active student participation. The way technology is integrated into learning activities can influence how students interact with learning content and participate in the learning process. Research on Nearpod has reported its use in supporting student participation and learning outcomes in different educational contexts [1], [2].

Informatics education requires students to develop not only conceptual knowledge but also logical reasoning, computational thinking, and problem-solving abilities. These competencies are particularly relevant when students learn programming because they need to understand relationships between concepts, instructions, syntax, and program outputs. A study on programming education reported that a technology-supported collaborative programming approach could improve students' learning achievement and computational thinking tendencies [3]. In addition, research on computational thinking integrated into school subjects has highlighted the importance of learning activities that allow students to apply systematic approaches to problem solving [4].

The use of appropriate multimedia representations can also support students in processing instructional information. Mayer's cognitive theory of multimedia learning explains that learners process information through different cognitive channels and that instructional materials need to be organised appropriately to support

© 2026 The Author(s). Published by Ali Institute of Research and Publication (AIRA) – Ali Bersaudara Sejahtera Foundation. This is an Open Access article distributed under the terms of the Creative Commons Attribution-ShareAlike 4.0 International License (CC BY-SA 4.0).

meaningful learning [5]. Therefore, learning media should not only function as a means of delivering information but should also provide opportunities for students to process and respond to the learning content.

These considerations are relevant to Informatics learning, particularly Algorithms and Programming. Programming materials require students to understand programming language concepts, syntax, program structures, sequences of instructions, program output, and programming errors. Students therefore need opportunities to examine examples, respond to questions, practice concepts, and receive feedback during learning.

One digital platform that can integrate instructional materials and interactive activities is Nearpod. Nearpod is an interactive learning platform that allows teachers to combine learning materials, student-response activities, and formative assessment within a digital learning environment. Previous research has reported that the interactive and gamified features of Nearpod can support students' learning motivation, attention, and engagement [6]. Nearpod has also been investigated as a medium for promoting student engagement in English learning [7] and in Foreign Language for Specific Purposes classes [8].

Several studies have also examined the relationship between Nearpod and learning outcomes. Violita and Irawan examined the use of Nearpod in Indonesian language learning [9]. Bintoro et al. investigated the influence of Nearpod on motivation and academic achievement [10]. Hidayat and Effendi examined Nearpod-based interactive learning media in biology learning [11]. Rahman et al. investigated Nearpod within a Discovery Learning model and examined students' learning outcomes in chemistry [12].

Nearpod has also been investigated in vocational education. Panduwinata and Az-Zahro examined the use of Nearpod among Grade XI OTKP students at SMKN 4 Surabaya using a Nonequivalent Control Group Design. Their study reported a statistically significant difference in learning outcomes between the groups [13]. Anandita et al. examined Nearpod within a Discovery Learning model and investigated its relationship with students' motivation and learning outcomes [14]. Nurmailia et al. examined a Nearpod interactive edugame and its relationship with learning retention [15].

Other studies have examined Nearpod-based interactive learning media in science and secondary education. Pinawadhani et al. used a quasi-experimental design to investigate Nearpod-based interactive media and learning outcomes in an excretory system topic [16]. Hasan et al. investigated Nearpod-based interactive learning media and students' economic literacy at the high school level [17]. Arsyad et al. examined Nearpod-based interactive learning media using a Nonequivalent Control Group Design and investigated its relationship with IPAS learning outcomes [18]. Perlawanan et al. also investigated Nearpod within a Discovery Learning model and examined its relationship with learning outcomes [19]. Sari et al. also examined the use of Nearpod in flipped learning for primary mathematics and reported improvements in students' digital learning outcomes [20].

Although previous studies have examined Nearpod across various subjects and educational levels, research on Nearpod-supported learning in vocational Informatics education, particularly in Algorithms and Programming, remains limited. This study extends the existing research by examining how interactive Nearpod activities support programming-related learning tasks, including understanding programming concepts and syntax, interpreting program structures and outputs, developing simple programs, and identifying errors in code. Thus, the contribution of this study lies not only in examining Nearpod as an interactive learning medium but also in investigating its use within the specific context of Algorithms and Programming education.

At SMK Negeri 1 Kuningan, initial observation indicated that Informatics learning was still largely conducted through conventional instruction, particularly teacher explanations supported by instructional videos. Students had limited opportunities to participate in discussions, express their understanding, and interact directly with learning materials. This condition indicates the need for an interactive learning medium that can combine instructional content with activities requiring student responses.

In the present study, Nearpod was implemented through several interactive activities, namely Open-Ended Question, Collaborate Board, Matching Pairs, Memory Test, Fill in the Blanks, and Time to Climb. These activities provided opportunities for students to respond to questions, express understanding, match concepts, recall information, complete learning tasks, and participate in interactive quizzes. The implementation was intended to provide a more participatory learning environment during Informatics instruction.

Based on this research gap, the present study examines the use of Nearpod in Grade X Informatics learning at SMK Negeri 1 Kuningan using a quasi-experimental design. The experimental group received Nearpod-supported learning, while the control group received conventional instruction. Both groups completed pretests and posttests.

Therefore, this study aims to determine whether there is a statistically significant difference in changes in learning outcomes between Grade X students who participated in Nearpod-supported Informatics learning and those who received conventional instruction at SMK Negeri 1 Kuningan. The study focuses specifically on Algorithms and Programming materials involving the C and C++ programming languages.

2. METHODOLOGY

2.1. Research Design

This study employed a quantitative approach using a quasi-experimental method with a Nonequivalent Control Group Design. The design was selected because the research involved two pre-existing classes and did not involve random assignment of individual students. The use of a quasi-experimental design is appropriate when researchers compare existing groups under different learning treatments [21].

The experimental group received Informatics learning supported by Nearpod, while the control group received conventional learning using the instructional media and learning approach normally employed by the teacher. Both groups were administered a pretest before the learning activities and a posttest after the learning activities.

The study was conducted at SMK Negeri 1 Kuningan during the second semester of the 2025/2026 academic year. The research focused on Informatics, particularly the Algorithms and Programming topic covering the C and C++ programming languages.

The learning competencies included understanding programming language concepts, identifying the history and differences between C and C++, understanding basic program structures and syntax, determining program output, understanding the steps for creating simple programs, understanding the installation of an Integrated Development Environment (IDE), and identifying errors in program code. The research design is presented in Table 1.

Table 1. Nonequivalent Control Group Design

Group	Pretest	Treatment	Posttest
Control	O_1	X_1	O_2
Experimental	O_3	X_2	O_4

Note:
O_1 = Pretest of the control group
O_2 = Posttest of the control group
O_3 = Pretest of the experimental group
O_4 = Posttest of the experimental group
X_1 = Conventional learning
X_2 = Learning using Nearpod

2.2. Intervention Procedure

The intervention was conducted over two meetings within a two-week period, with each meeting lasting approximately 160 minutes. Both the experimental and control groups received the same Algorithms and Programming learning content and learning objectives. The difference between the two groups was the learning approach and instructional activities used during the intervention.

In the experimental group, Nearpod was used to present learning materials and support interactive learning activities. The Nearpod activities used during the intervention included Open-Ended Question, Collaborate Board, Matching Pairs, Memory Test, Fill in the Blanks, and Time to Climb. These activities engaged students in answering questions, participating in discussions, matching related concepts, completing learning tasks, and responding to interactive quizzes.

During the first meeting, the teacher delivered the Algorithms and Programming learning materials using Nearpod and guided students through the interactive activities. Students followed the presented materials and participated in the Nearpod activities under the teacher's guidance. During the second meeting, Nearpod was primarily used to present the learning materials through slides. The learning activity was then continued with practical programming activities in the computer laboratory. Students installed Dev-C++ and practiced creating a simple C++ program that required them to input their name and year of birth and display their calculated age as the output. The teacher guided students during the installation and programming practice and provided assistance when students encountered difficulties.

In the control group, the same learning content and learning objectives were delivered using the conventional learning approach and instructional media normally used by the teacher. Students followed the teacher's explanations, studied the learning materials, and participated in the planned learning activities. Programming practice was also conducted in the computer laboratory according to the learning activities planned by the teacher.

To maintain treatment fidelity, both groups received the same learning content, learning objectives, and intervention period. The main difference between the groups was that the experimental group received Nearpod-supported interactive activities, while the control group followed the conventional learning approach and instructional media normally used by the teacher.

2.3. Population and Sample

The population consisted of 660 Grade X students at SMK Negeri 1 Kuningan during the 2025/2026 academic year. The sample was selected using purposive sampling based on specific considerations related to the implementation of the research. Grade X APHP 2 was selected as the experimental group, while Grade X TKR 1 was selected as the control group based on the Informatics teacher's recommendation. Both classes were taught by the same teacher, followed the same curriculum, studied the same Algorithms and Programming content, and had not yet received the selected learning material before the intervention. The two classes also had access to the same computer laboratory and comparable learning facilities. The selection of the classes was therefore based on their suitability for receiving the planned learning treatment under comparable instructional conditions.

Because the classes were pre-existing and were not randomly assigned, the sampling procedure may have introduced selection bias. To reduce the potential influence of differences between the groups, the study used the

same teacher, curriculum, learning content, computer laboratory, and comparable learning facilities for both groups. Pretest scores were also collected to identify the students' initial learning outcomes before the intervention.

The initial sample consisted of 71 students. However, three students did not complete the entire research process. Therefore, 68 students were included in the final analysis, consisting of 34 students in the experimental group and 34 students in the control group.

2.4. Research Procedure

The research procedure began with the determination of the experimental and control groups using purposive sampling. After the groups were determined, both groups were administered a pretest to identify their initial learning outcomes.

The experimental group subsequently participated in learning activities supported by Nearpod, while the control group participated in conventional learning using the learning media and instructional approach normally employed by the teacher.

After the learning activities were completed, both groups were administered the same posttest. The pretest and posttest data were subsequently analyzed statistically to determine whether there was a difference in changes in learning outcomes between the two groups.

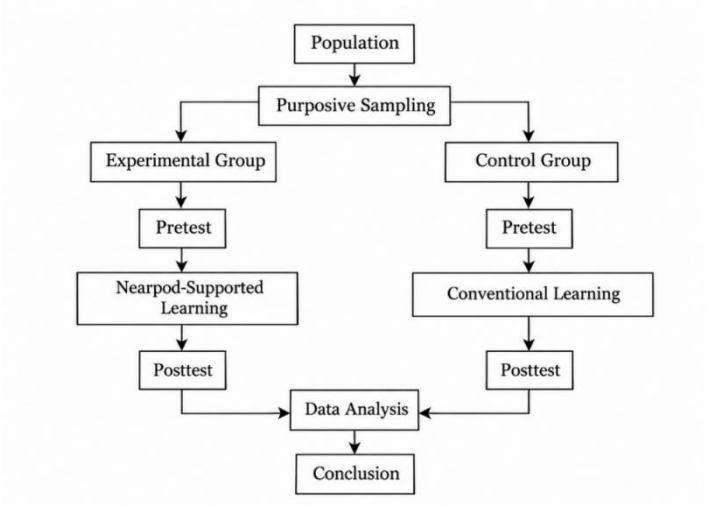


Figure 1. Research Procedure

2.5. Data Collection and Research Instruments

Data were collected through tests, observations, and documentation.

The primary instrument used to measure student learning outcomes was a 10-item multiple-choice achievement test administered as both the pretest and posttest. The 10 items were selected from an initial set of 20 items based on the results of item analysis.

Before the main research, the Nearpod-based learning medium was validated by a media expert to assess its suitability as an interactive learning medium. The learning achievement test was reviewed by a subject-matter expert to determine the correspondence between the learning indicators, instructional materials, and test items.

The initial 20-item achievement test was then administered to students outside the research sample to examine item characteristics, including validity, reliability, difficulty level, and discrimination index.

Table 2. Research Variables and Data Collection Techniques

Variable	Instrument	Data Collection Technique	Data Source
Use of Nearpod	Learning implementation observation sheet	Observation	Teacher and students
Learning outcomes	Multiple-choice test (pretest–posttest)	Test	Students
Research documentation	Activity documentation	Documentation	School

2.6. Learning Achievement Test Blueprint

The learning achievement test was developed based on the competencies and learning achievement indicators for Algorithms and Programming, particularly the C and C++ programming language subtopic.

Table 3. Blueprint of the Informatics Learning Achievement Test

Competency	Learning Achievement Indicator	Item Number
Understanding programming language concepts	Explaining the definition of a programming language	1
Understanding the history of C and C++	Identifying the history of C and C++	2
Understanding the differences between C and C++	Analysing the differences between C and C++	3–4
Understanding the basic structure of C++ programs	Identifying the basic structure of a C++ program	5–6
Understanding the use of basic syntax	Explaining the functions of basic syntax, such as cout, cin, and others	7–9
Understanding program development procedures	Arranging the steps for developing a simple program	10–11
Analysing program output	Determining the output of a simple program	12–14

Competency	Learning Achievement Indicator	Item Number
Understanding the IDE installation process	Explaining the steps for installing Dev-C++	15
Understanding simple program development	Identifying the steps for creating a simple program	16–17
Analyzing programming errors	Identifying errors in program code	18–20

2.7. Data Analysis

Data analysis was conducted using IBM SPSS Statistics version 25.

Descriptive statistics were first used to describe the learning outcomes of the experimental and control groups. The analysis included the minimum score, maximum score, mean, and standard deviation.

Because the two groups had different initial pretest scores, the analysis focused on the change in learning outcomes from pretest to posttest. For each student, a change score was calculated as the posttest score minus the pretest score. The change scores were then compared between the experimental and control groups to examine whether the improvement in learning outcomes differed between the two groups.

The normality of the change scores was examined using the Shapiro–Wilk test [22]. A significance value greater than 0.05 was interpreted as indicating that the data did not significantly deviate from a normal distribution.

Homogeneity of variance was examined using Levene's test. Because the change scores in both groups were normally distributed and the variances were homogeneous, an Independent Samples t-Test was used to compare the change scores between the experimental and control groups. Statistical significance was evaluated at $\alpha = 0.05$.

To describe the magnitude of the standardised difference between the two groups, Cohen's d was calculated based on the difference between the group mean change scores and the pooled standard deviation [23].

3. RESULTS AND DISCUSSION

3.1. Instrument Analysis

Before being used to collect the main research data, the learning achievement instrument was analysed to determine its quality and suitability for measuring students' learning outcomes. The pilot test was conducted with 23 students who were not included in the main research sample. The initial instrument consisted of 20 multiple-choice items covering the Algorithms and Programming material, particularly the C and C++ programming languages.

The validity of the 20 dichotomously scored multiple-choice items was examined using the Pearson Product Moment correlation with the assistance of IBM SPSS Statistics. Because the items were scored dichotomously, with responses coded as 0 and 1, the item-level Pearson correlation provides a point-biserial form of item-total association when the item score is related to a continuous total score. The Pearson correlation coefficient and its Sig. (2-tailed) value were examined at the item level. An item was considered valid when the Sig. (2-tailed) value was below 0.05.

The analysis showed that 10 of the 20 items met the validity criterion, namely items 1, 2, 3, 4, 5, 7, 8, 14, 15, and 18. The Pearson correlation coefficients of the retained items ranged from 0.429 to 0.801, with significance values ranging from 0.000 to 0.041. These 10 items were retained for the final instrument.

Table 4. Item Validity Analysis

Item	Pearson Correlation	Sig. (2-tailed)	Decision
1	0.592	0.003	Valid
2	0.619	0.002	Valid
3	0.686	<0.001	Valid
4	0.539	0.008	Valid
5	0.801	<0.001	Valid
6	0.389	0.067	Not Valid
7	0.755	<0.001	Valid
8	0.674	<0.001	Valid
9	0.027	0.901	Not Valid
10	0.271	0.211	Not Valid
11	0.137	0.532	Not Valid
12	-	-	Not Valid
13	0.201	0.357	Not Valid
14	0.568	0.005	Valid
15	0.744	<0.001	Valid
16	-0.025	0.911	Not Valid
17	0.272	0.208	Not Valid
18	0.429	0.041	Valid
19	0.220	0.313	Not Valid
20	0.385	0.070	Not Valid

*Item 12 could not be analysed because the response data were constant.

The reliability analysis was subsequently performed after the invalid items had been removed. Thus, Cronbach's Alpha was calculated using the 10 retained items. The analysis produced a Cronbach's Alpha coefficient of 0.887 (N = 10), indicating high internal consistency of the final achievement-test instrument.

To provide additional item-total evidence, the Corrected Item-Total Correlation values were also examined. The retained items showed corrected item-total correlation values ranging from 0.319 to 0.746, providing additional evidence of the consistency of the retained items with the overall test.

The retained 10 items represented several aspects of the intended Algorithms and Programming content. Item 1 assessed the basic concept of programming languages; items 2 and 3 addressed the history and development

of C and C++; item 4 addressed the identification of C and C++ usage; item 5 assessed basic program structure; items 7 and 8 assessed input and output functions; item 14 assessed program output interpretation; item 15 assessed the use of an Integrated Development Environment (IDE); and item 18 assessed the identification of errors in program code. Thus, the retained items represented several of the main content areas covered in the learning activities.

The difficulty analysis was conducted on all 20 items using responses from the 23 students in the pilot test. The results showed four very easy items (20%), eight easy items (40%), five moderate items (25%), and three difficult items (15%).

The discrimination analysis using Corrected Item-Total Correlation showed nine very good items (45%), two good items (10%), one sufficient item (5%), six poor items (30%), and two very poor items (10%).

Table 5. Summary of Instrument Analysis

Analysis	Result	Interpretation
Validity	10 of 20 items were valid (50%)	Items 1, 2, 3, 4, 5, 7, 8, 14, 15, and 18 were retained
Reliability	Cronbach's Alpha = 0.887 (10 items)	High internal consistency
Difficulty	4 very easy (20%), 8 easy (40%), 5 moderate (25%), 3 difficult (15%)	Most items were easy to very easy
Discrimination	9 very good (45%), 2 good (10%), 1 sufficient (5%), 6 poor (30%), 2 very poor (10%)	Items that did not meet the validity criterion were excluded from the final instrument.

Overall, the item analysis resulted in 10 valid and reliable items that were subsequently used as the final achievement test instrument for the pretest and posttest.

3.2. Descriptive Statistics of Learning Outcomes

Descriptive statistics were used to describe students' learning outcomes before and after the learning activities.

Table 6. Descriptive Statistics of Students' Learning Outcomes

Group	Test	N	Min	Max	Mean	SD
Control	Pretest	34	10	90	47.35	18.799
Control	Posttest	34	40	100	77.35	11.886
Experimental	Pretest	34	10	90	37.06	16.055
Experimental	Posttest	34	50	100	83.53	14.330

Both groups demonstrated an increase in their mean learning outcomes.

The control group's mean score increased from 47.35 in the pretest to 77.35 in the posttest, representing an increase of 30.00 points.

The experimental group's mean score increased from 37.06 in the pretest to 83.53 in the posttest, representing an increase of 46.47 points.

Thus, the experimental group showed a greater increase in mean score than the control group. The experimental group also achieved a higher posttest mean, with a difference of 6.18 points compared with the control group.

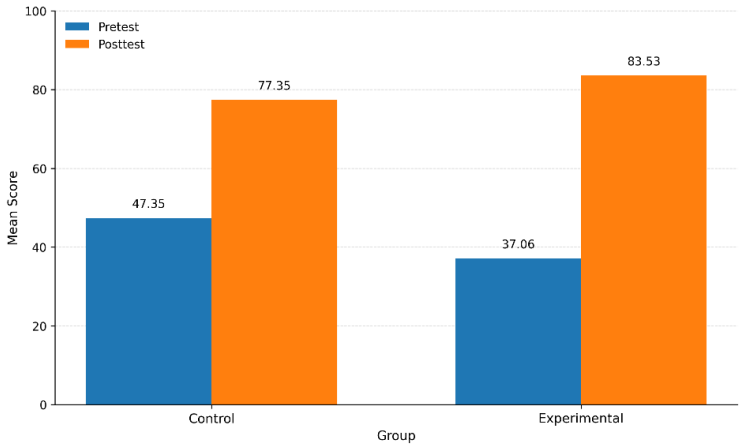


Figure 2. Comparison of Mean Pretest and Posttest Scores

3.3. Normality Test of Change Scores

Because the main comparison focused on the change in learning outcomes, the normality test was conducted on the change scores calculated as posttest minus pretest for each student. The Shapiro–Wilk test was used to examine the distribution of the change scores [22].

Table 7. Results of the Shapiro–Wilk Normality Test for Change Scores

Group	N	Shapiro–Wilk Statistic	Sig.	Interpretation
Control	34	0.952	0.140	Normal
Experimental	34	0.949	0.113	Normal

The Shapiro–Wilk results showed significance values of 0.140 for the control group and 0.113 for the experimental group. Both values were greater than 0.05, indicating that the change scores in both groups were normally distributed. Therefore, the normality assumption for the Independent Samples t-Test was satisfied.

3.4. Hypothesis Testing

After the normality assumption was satisfied, homogeneity of variance was examined using Levene's test.

Table 8. Levene's Test and Independent Samples t-Test

Test	Statistic	df	Sig.
Levene's Test	3.018	1, 66	0.087
Independent Samples t-Test	−3.322	66	0.001

The Levene's test produced a significance value of 0.087, which was greater than 0.05. Therefore, the variance of the change scores was considered homogeneous, and the "Equal variances assumed" row was used for interpreting the Independent Samples t-Test.

Table 9. Group Statistics of Change Scores

Group	N	Mean Change	Std. Deviation	Std. Error Mean
Control	34	30.00	17.753	3.045
Experimental	34	46.47	22.814	3.913

The Independent Samples t-Test produced a t-value of −3.322 with 66 degrees of freedom and a significance value of 0.001. Because the significance value was below 0.05, there was a statistically significant difference in the change in learning outcomes between the experimental and control groups.

The mean difference was −16.471, with a 95% confidence interval ranging from −26.369 to −6.572. The negative sign reflects the order in which the groups were entered into SPSS; descriptively, the experimental group showed a greater mean increase than the control group.

The experimental group showed a mean change of 46.47 points, whereas the control group showed a mean change of 30.00 points. The standardised mean difference was calculated using Cohen's d and resulted in d = 0.81. This indicates a substantial standardised difference in change scores between the two groups [23].

Because the research used pre-existing classes without random assignment and the groups differed in their initial pretest means, the statistically significant difference should be interpreted as a difference in observed change under the respective learning conditions rather than as definitive evidence of a purely causal effect of Nearpod.

3.5. Discussion

The results show that both the experimental and control groups experienced improvements in learning outcomes after the learning activities. The experimental group increased from a pretest mean of 37.06 to a posttest mean of 83.53, representing a mean increase of 46.47 points. Meanwhile, the control group increased from a pretest mean of 47.35 to a posttest mean of 77.35, representing a mean increase of 30.00 points. The experimental group therefore showed a greater increase in learning outcomes than the control group.

The comparison of change scores showed a statistically significant difference between the two groups. The Independent Samples t-Test produced $t(66) = -3.322$ with $p = 0.001$. The experimental group had a mean change of 46.47 points, compared with 30.00 points in the control group. The standardised mean difference was Cohen's $d = 0.81$. These findings indicate that the two groups experienced significantly different changes in learning outcomes under the conditions of the study.

The findings are consistent with previous research reporting positive learning outcomes or participation associated with Nearpod-based learning. Anggaretta et al. [1] reported improvements in learning outcomes and student participation following the implementation of Nearpod-based interactive learning media. Maharani [2] also examined Nearpod among Grade X vocational students using a quasi-experimental design and reported a significant difference in learning motivation between the experimental and control groups. Although these studies examined different outcomes, they provide relevant evidence that Nearpod can be incorporated into structured learning activities involving existing classroom groups.

The present study extends this body of research to Algorithms and Programming education in vocational Informatics learning. Unlike studies that examine Nearpod primarily in general subjects or in relation to motivation and engagement, the present study involved learning tasks related to programming concepts, syntax, program structures, program outputs, and identification of errors in C and C++ code. This context is important because programming learning requires students to understand relationships between instructions, structures, and outputs rather than only recall factual information.

The possible contribution of Nearpod in this context can be understood from the characteristics of the interactive activities used during the intervention. Students participated in Open-Ended Question, Collaborate Board, Matching Pairs, Memory Test, Fill in the Blanks, and Time to Climb. These activities required students to respond to questions, recall information, connect related concepts, complete learning tasks, and communicate their understanding. Therefore, the Nearpod implementation involved active responses during the learning process rather than only presenting digital learning materials.

The interactive activities may also support programming learning by providing repeated opportunities to process and retrieve programming-related information. For example, Matching Pairs and Memory Test required students to recall and associate related concepts, while Fill in the Blanks required students to retrieve specific information. Open-Ended Question and Collaborate Board provided opportunities for students to express their understanding and respond to programming-related questions. These activities can be relevant to learning programming concepts because students need to distinguish syntax, recognize structures, and understand the relationship between programming instructions and their expected outputs.

The use of interactive activities also provided opportunities for formative responses during the learning process. Student responses could help the teacher identify concepts that were not yet understood and provide clarification or feedback. In this way, the interactive features were not only used to make the learning process more participatory but also provided information that could support instructional responses during the lesson. This is relevant to multimedia-supported learning because students interacted with presented information while simultaneously responding to learning tasks [5].

The relevance of active response and structured learning activities is particularly important in programming education. Programming requires students to repeatedly apply concepts and interpret the consequences of instructions. When students respond to questions about syntax, structures, or program outputs, they are required to process the information rather than merely observe it. Recent research in programming education has examined how instructional approaches and learning environments can support programming performance and computational thinking [24]–[27]. Zhan et al. [24], for example, investigated task-driven collaborative learning in a high-school programming course and reported differences in programming performance and computational thinking. Sun et al. [25] examined different programming modalities and their relationship with programming behaviours, syntactical errors, debugging processes, and knowledge performance. Aksakal and Kucuk [26] examined problem-based programming education among secondary school students and reported improvements in computational thinking performance. Chen et al. [27] investigated self-explanation approaches in secondary school programming and found differences in computational thinking and programming behaviours. These studies demonstrate that programming education involves specific learning processes that extend beyond general content delivery.

The present study did not directly measure computational thinking, algorithmic reasoning, or debugging performance as separate variables. Therefore, the findings cannot be interpreted as evidence that Nearpod directly improved these abilities. However, the Algorithms and Programming learning activities provided a context in which students had to understand logical sequences, programming syntax, program structures, and outputs. The interactive activities used in Nearpod may therefore be relevant as instructional support for these programming-related learning processes by requiring students to actively process and respond to programming concepts.

The findings are also consistent with previous Nearpod research across different educational contexts. Several studies have reported associations between Nearpod-based learning and student engagement, participation, learning motivation, retention, or learning outcomes [6]–[20]. For example, Abdullah et al. [6] examined gamification and interactivity in Nearpod, while Efendi [7] and Metaria et al. [8] examined student engagement. Other studies have examined Nearpod in Indonesian language, biology, chemistry, economic literacy, IPAS, and other learning contexts [9]–[20]. These studies provide broader evidence regarding the use of Nearpod as an interactive learning medium. However, the present study adds a programming-specific context by examining learning outcomes in Algorithms and Programming among vocational Informatics students.

The findings should nevertheless be interpreted in light of the research design. The experimental group had a lower pretest mean (37.06) than the control group (47.35). Although the experimental group showed a greater mean increase and a higher posttest mean, the difference in initial scores indicates that the groups were not equivalent at baseline. The use of purposive sampling and intact classes also means that students were not randomly assigned to the two treatments. Consequently, factors other than the learning medium may have contributed to the observed difference.

For this reason, the statistically significant difference in change scores should be interpreted as evidence that the two groups experienced different changes in learning outcomes under the respective learning conditions, rather than as definitive evidence that Nearpod alone caused the difference. The change-score analysis provides a comparison of improvement from pretest to posttest, but it does not eliminate all possible effects of baseline differences or other characteristics associated with the pre-existing classes.

Within these limitations, the findings provide empirical evidence concerning the use of Nearpod-supported learning in Algorithms and Programming education. The greater change observed in the experimental group, together with the standardised mean difference of 0.81, indicates a substantial standardised difference in change scores under the conditions of this study. However, this effect size should be interpreted together with the baseline difference and the nonrandomized design.

The study has several limitations. First, the research involved only 68 students from two existing classes at one vocational high school. Second, the study focused on Algorithms and Programming involving C and C++, so the findings may not represent other Informatics topics. Third, purposive sampling and the use of intact classes limit the generalisability of the findings. Fourth, the experimental group had a lower pretest mean than the control group, which should be considered when interpreting the differences in learning outcomes. Finally, computational thinking, algorithmic reasoning, engagement, and other related constructs were not measured as separate outcome variables in this study.

Future research could involve larger samples, more classes, and multiple vocational schools to improve the generalisability of the findings. Further studies could also examine Nearpod in other Informatics topics and use specific measures of programming performance, computational thinking, algorithmic reasoning, debugging ability, motivation, or learning engagement. Research using random assignment where feasible, stronger baseline-adjustment procedures, or repeated measurements could provide additional evidence regarding the contribution of interactive digital media to programming education.

4. CONCLUSION

This study found a statistically significant difference in the change in learning outcomes between Grade X students who participated in Nearpod-supported Informatics learning and those who received conventional instruction. The experimental group showed a mean increase of 46.47 points, from a pretest mean of 37.06 to a posttest mean of 83.53, while the control group showed a mean increase of 30.00 points, from a pretest mean of 47.35 to a posttest mean of 77.35.

The Independent Samples t-Test on the change scores showed a statistically significant difference between the two groups, with $t(66) = -3.322$ and $p = 0.001$. The experimental group showed a greater mean change than the control group, and the standardised mean difference was Cohen's $d = 0.81$.

However, the findings should be interpreted cautiously because the study used a Nonequivalent Control Group Design, purposive sampling, and pre-existing classes. The experimental group also had a lower pretest mean than the control group, indicating that the groups were not equivalent at baseline. Therefore, the results should not be interpreted as definitive evidence of a purely causal effect of Nearpod.

Within the conditions of this study, Nearpod-supported learning was associated with a greater change in learning outcomes in Algorithms and Programming education involving C and C++ among Grade X vocational Informatics students. The findings provide empirical evidence regarding the use of Nearpod as an interactive learning medium in this specific programming-learning context.

ACKNOWLEDGMENT

The authors would like to express their gratitude to Universitas Muhammadiyah Kuningan for the academic support provided throughout this research. The authors also sincerely thank the research supervisor for the guidance, suggestions, and academic support provided during the research and preparation of this article. The authors further express their appreciation to SMK Negeri 1 Kuningan, particularly the teachers and Grade X students, for the permission, support, and participation provided during the implementation of the research.

DECLARATIONS

Artificial Intelligence Use

Generative artificial intelligence tools were used to assist with language editing, manuscript organisation, and English-language refinement. The authors reviewed the manuscript and remain responsible for the accuracy, interpretation, and final content of the article.

Author Contributions

Adinda Mursallina Karjaya: Conceptualization, Methodology, Investigation, Data Curation, Formal Analysis, Writing – Original Draft, Writing – Review & Editing. Dodi Ahmad Haerudin: Supervision, Validation, Writing – Review & Editing.

Funding

This research received no specific external funding.

Conflict of Interest

The authors declare that there is no conflict of interest related to this research or its publication.

Ethics Approval / Ethical Clearance

No formal ethical approval was obtained for this study. The research was conducted with permission from the participating teacher as part of an undergraduate academic research project. The research activities consisted of regular learning activities and achievement tests conducted within the instructional setting. No personally identifiable student information was included in the manuscript, and the collected data were used solely for academic research purposes. Participant data were handled confidentially to protect student privacy.

Data Availability Statement

The data supporting the findings of this study are available from the corresponding author upon reasonable request, subject to applicable institutional and participant privacy considerations.

REFERENCES

[1] D. S. Anggaretta, D. M. C. Dewi, D. Hardiyanti, V. A. Ahyuni, N. S. Langit, and J. T. B. Santoso, "The role of Nearpod learning media in increasing participation and learning outcomes," *Economic Education Analysis Journal*, vol. 13, no. 2, pp. 192–199, 2024, doi: 10.15294/eeaj.v13i2.6389.

[2] A. D. Maharani, "The effect of Nearpod learning media on student learning motivation at SMKN 1 Sukoharjo," *Jurnal Informasi dan Komunikasi Administrasi Perkantoran*, vol. 9, no. 1, pp. 1–8, 2025, doi: 10.20961/jikap.v9i1.90972.
[3] W. Li, C.-Y. Liu, and J. C. R. Tseng, "Development of a metacognitive regulation-based collaborative programming system and its effects on students' learning achievements, computational thinking tendency and group metacognition," *British Journal of Educational Technology*, vol. 55, no. 1, pp. 318–339, 2024, doi: 10.1111/bjet.13358.
[4] S. Yeni, J. Nijenhuis-Voogt, M. Saeli, E. Barendsen, and F. Hermans, "Computational thinking integrated in school subjects: A cross-case analysis of students' experiences," *International Journal of Child-Computer Interaction*, vol. 42, Art. no. 100696, 2024, doi: 10.1016/j.ijcci.2024.100696.
[5] R. E. Mayer, "The past, present, and future of the cognitive theory of multimedia learning," *Educational Psychology Review*, vol. 36, no. 1, 2024, doi: 10.1007/s10648-023-09842-1.
[6] M. I. Abdullah, D. Inayati, and N. N. Karyawati, "Nearpod use as a learning platform to improve student learning motivation in an elementary school," *Journal of Education and Learning (EduLearn)*, vol. 16, no. 1, pp. 121–129, 2022, doi: 10.11591/edulearn.v16i1.20421.
[7] T. A. Efendi, "Nearpod as a means of promoting students' engagement in English class," *Borneo Journal of Language and Education*, vol. 4, no. 1, pp. 65–79, 2024, doi: 10.21093/benjole.v4i1.8166.
[8] S. Metaria, S. Muniroh, and N. Zubaidi, "Enhancing students' engagement in FLSP class: The impact of Nearpod," *Celtic: A Journal of Culture, English Language Teaching, Literature and Linguistics*, vol. 11, no. 1, pp. 39–55, 2024, doi: 10.22219/celtic.v11i1.33329.
[9] F. Violita and W. D. Irawan, "The use of Nearpod media to improve Indonesian language learning outcomes in elementary schools," *Jurnal Pedagogi dan Pembelajaran*, vol. 58, no. 2, pp. 365–373, 2025, doi: 10.23887/jpp.v58i2.93690.
[10] W. P. Bintoro, E. Triwahyuni, and A. Z. Emyus, "The influence of digital learning media Nearpod on the motivation and academic achievement of elementary school students," *Journal of Education Technology*, vol. 8, no. 4, pp. 654–661, 2024, doi: 10.23887/jet.v8i4.85613.
[11] R. Hidayat and E. Effendi, "Pemanfaatan media pembelajaran interaktif berbasis Nearpod untuk meningkatkan hasil belajar biologi siswa SMA," *Biodik*, vol. 10, no. 2, pp. 30–36, 2024, doi: 10.22437/biodik.v10i2.27778.
[12] S. Z. D. Rahman, S. Side, and R. Ramdani, "Efektivitas media interaktif Nearpod dalam model Discovery Learning terhadap hasil belajar peserta didik kelas XI MIPA 4 SMA Negeri 18 Makassar," *ChemEdu*, vol. 5, no. 3, pp. 1–14, 2024, doi: 10.35580/chemedu.v5i3.60830.
[13] L. F. Panduwinata and N. F. Az-Zahro, "The effectiveness of using Nearpod learning media on the public relations learning outcomes of class XI OTKP SMKN 4 Surabaya," *Jurnal Ilmu Pendidikan (JIP) STKIP Kusuma Negara*, vol. 16, no. 1, pp. 26–34, 2024, doi: 10.37640/jip.v16i1.1975.
[14] N. Anandita, N. Herawati, and M. J. Djangi, "Effectiveness of the Nearpod platform within the Discovery Learning model to enhance students' learning motivation and learning outcomes on reaction rate topic," *Jambura Journal of Educational Chemistry*, vol. 7, no. 2, pp. 45–52, 2025, doi: 10.37905/jjec.v7i2.33005.
[15] E. Nurmailia, M. Qurbaniah, and H. M. Rahayu, "The effectiveness of Nearpod interactive edugame on learning retention of class XI IPA students of SMAN 1 District of Seventeen," *Qalam: Jurnal Ilmu Kependidikan*, vol. 13, no. 1, pp. 63–71, 2024, doi: 10.33506/jq.v13i1.2702.
[16] R. K. Pinawadhani, K. Ningsih, and R. G. P. Panjaitan, "Efektivitas media interaktif berbasis Nearpod terhadap hasil belajar siswa pada materi sistem ekskresi," *Jurnal Pendidikan: Teori, Penelitian, dan Pengembangan*, vol. 9, no. 10, 2024, doi: 10.17977/2502-471X.1056.
[17] M. Hasan, A. Anugrawati, Nurjannah, A. T. Ampa, and M. I. S. Ahmad, "Interactive learning media through Nearpod to increase the economic literacy of students at the high school level," *Jurnal Edutech Undiksha*, vol. 12, no. 1, pp. 75–88, 2024, doi: 10.23887/jeu.v12i1.76712.
[18] D. F. Arsyad, M. Irfan, and Sayidiman, "Pengaruh penggunaan media pembelajaran interaktif berbasis Nearpod terhadap hasil belajar IPAS siswa kelas V SD Inpres Belaka Kabupaten Gowa," *Pendas: Jurnal Ilmiah Pendidikan Dasar*, vol. 11, no. 2, 2026, doi: 10.23969/jp.v11i02.47493.
[19] A. T. Perlawanan, Jusniar, and A. F. Madjid, "The effect of Nearpod interactive media in the Discovery Learning model on the learning outcomes of class XI MIA MAN 4 Bone students," *UNESA Journal of Chemical Education*, vol. 11, no. 3, pp. 220–226, 2022, doi: 10.26740/ujced.v11n3.p220-226.
[20] R. M. Sari, Rusnilawati, and S. R. Ali, "Flipped learning with Nearpod media: Enhancing digital learning outcomes in primary mathematics," *Profesi Pendidikan Dasar*, vol. 10, no. 2, pp. 159–173, 2023, doi: 10.23917/ppd.v10i2.4511.
[21] J. W. Creswell and J. D. Creswell, *Research Design: Qualitative, Quantitative, and Mixed Methods Approaches*, 5th ed. Thousand Oaks, CA, USA: SAGE Publications, 2018.
[22] S. S. Shapiro and M. B. Wilk, "An analysis of variance test for normality (complete samples)," *Biometrika*, vol. 52, no. 3–4, pp. 591–611, 1965.
[23] J. Cohen, *Statistical Power Analysis for the Behavioral Sciences*, 2nd ed. Hillsdale, NJ, USA: Lawrence Erlbaum Associates, 1988.
[24] Z. Zhan, T. Li, and Y. Ye, "Effect of jigsaw-integrated task-driven learning on students' motivation, computational thinking, collaborative skills, and programming performance in a high-school programming course," *Computer Applications in Engineering Education*, vol. 32, 2024, e22793, doi: 10.1002/cae.22793.
[25] D. Sun, F. Ouyang, Y. Li, C. Zhu, and Y. Zhou, "Using multimodal learning analytics to understand effects of block-based and text-based modalities on computer programming," *Journal of Computer Assisted Learning*, vol. 40, no. 3, pp. 1123–1136, 2024, doi: 10.1111/jcal.12939.
[26] H. Aksakal and S. Kucuk, "Secondary school students' computational thinking skills, group cohesion, and performance in problem-based programming education," *Journal of Systems and Software*, vol. 230, Art. no. 112535, 2025, doi: 10.1016/j.jss.2025.112535.
[27] P. Chen, D. Yang, J. Zhao, S. Yang, and J. Lavonen, "The effects of self-explanation on secondary school students' computational thinking and programming behaviour," *Journal of Computer Assisted Learning*, vol. 41, no. 5, Art. no. e70116, 2025, doi: 10.1111/jcal.70116.