# INTEGRATION OF SCRATCH-BASED CODING LEARNING IN IMPROVING COMPUTATIONAL THINKING AND PROBLEM SOLVING OF HIGH SCHOOL STUDENTS

**TIORIA PASARIBU[1], RIZKA PUTRI RAHAYU[2]**

[1]STMIK KAPUTAMA, Jl. Veteran No. 4A-9A Binjai, 20714
E-mail: pasaributioria@gmail.com*

## ABSTRACT

The development of digital technology in the era of the Fourth Industrial Revolution demands that the world of education not only focus on mastery of material but also on the development of higher-order thinking skills relevant to the needs of the 21st century. Computational thinking, or the ability to think systematically about problems using a computational approach, is one of the skills that has gotten a lot of attention around the world. This study aims to empirically analyze the impact of Scratch-based coding learning on the improvement of computational thinking and problem-solving skills among high school students. The research uses a quantitative approach with a quasi-experimental design of the pretest-posttest control group design type. The research sample consisted of 60 eighth-grade students divided into an experimental class and a control class. Data analysis was conducted through validity tests, reliability tests, Kolmogorov-Smirnov normality tests, Levene's homogeneity tests, N-Gain calculations, independent t-tests, and effect size (Cohen's d). The research results show that the experimental class experienced a significant increase with an N-Gain value of 0.71 (high category), while the control class had an N-Gain value of 0.23 (low category). The t-test showed a t_calculated value of 6.54, greater than the t_table value of 2.00 at a significance level of 0.05. An effect size of 1.68 indicates a substantial practical effect. These findings confirm that Scratch-based coding instruction is effective in enhancing students' computational thinking quality and problem-solving abilities.

**Keywords:** Coding, Computational Thinking, Problem Solving, Scratch, Technology Education

## 1. INTRODUCTION

The development of digital technology in recent decades has brought significant changes to the world of education. Schools no longer function merely as places for delivering lesson material but also as spaces for developing 21st-century skills, such as critical thinking, creativity, and problem-solving abilities. Students are required to be able to face various problems logically and systematically, not just memorize information. One of the competencies considered important in facing the challenges of the digital era is computational thinking (CT)[1],[2],[3]. Computational thinking is a way of thinking that emphasizes the ability to solve problems in a structured manner through four main components: decomposition (breaking down problems into smaller parts), pattern recognition, abstraction (selecting relevant information), and algorithm design (systematic steps for resolution). This ability is not only necessary in the field of technology but also in various everyday situations that require logical reasoning.

To develop computational thinking, coding education is starting to be integrated into the curriculum, including at the secondary school level[4]. Coding provides students with hands-on experience in designing solutions in the form of programs. In the process, students learn to organize logical steps, test the program, find errors, and fix them. This process indirectly trains systematic thinking and problem-solving skills[5]. One of the media widely used in coding education is Scratch. Scratch is a visual programming platform designed to help beginners understand the basic concepts of programming. By using command blocks arranged through drag-and-drop, students can create animations, games, or interactive projects without having to understand complex syntax[6]. Through Scratch, students can learn about command sequences, branching, and loops in a simpler and more enjoyable way. However, in practice, coding education in schools often focuses more on the final result in the form of projects created by students. The thinking process underlying program creation has not always been the main focus. In fact, the main goal of integrating coding into education is not only for students to be able to create digital products but also for them to develop computational thinking and problem-solving skills more deeply[7]. Theoretically, computational thinking is closely related to problem solving because both involve the processes of analysis, planning, and evaluation of solutions. However, further research is needed to quantitatively test whether the integration of Scratch-based coding education truly has a significant impact on the improvement of both skills in high school students.

The middle school level was chosen because at this stage, students have more developed abstract thinking abilities. The integration of Scratch-based coding learning is expected to be an effective strategy to strengthen logical, systematic, and structured thinking in problem-solving[8],[9]. Based on this description, this research is conducted to analyze the impact of integrating Scratch-based coding learning on the improvement of computational thinking and problem-solving among high school students. The research results are expected to contribute to the development of more relevant learning in line with the educational needs of the digital era.

## 2. METHODOLOGY

### 2.1 Approach and Research Design

This research uses a quantitative approach with a quasi-experimental method. The design used is a pretest-posttest control group design.

Experiment:

$$O_1 - X - O_2$$

Control :

$$O_1 - \ldots - O_2$$

This design allows researchers to control students' initial abilities thru a pretest so that changes in the posttest can be attributed to the treatment[10].

### 2.2 Population and Sample

The research population consists of all eighth-grade students. A sample of 60 students was selected using purposive sampling based on equivalent academic abilities. The sample was divided into two groups of 30 students each.

### 2.3 Research Instruments

The instrument consists of a problem-solving essay test with 10 questions designed to measure indicators of computational thinking[10],[11],[12],[13].

- **Validity Test**
  Using Pearson Product Moment:

$$r\_xy = (N\Sigma XY - \Sigma X\Sigma Y) / \sqrt{[(N\Sigma X^2 - (\Sigma X)^2)(N\Sigma Y^2 - (\Sigma Y)^2)]}$$

  All items haver_ calculated > r_table (0,361), thus valid.

- **Reliability Test**
  Using Cronbach's Alpha:

$$\alpha = (k/(k-1))(1 - \Sigma\sigma^2\_b/\sigma^2\_t)$$

  The value of α = 0.82 indicates high reliability

### 2.4 Data Analysis Techniques

Data analysis includes:

1. Normality Test (Kolmogorov-Smirnov)
   The normality test is a statistical procedure used to determine whether a dataset comes from a normally distributed population. In parametric statistical analysis, the assumption of normality is one of the important prerequisites because many analysis techniques, such as the t-test and ANOVA, require a data distribution that is close to normal. The Kolmogorov-Smirnov (K-S) test works by comparing the cumulative distribution of the sample data with the theoretical normal distribution. If the difference between the two is not significant, then the data is considered to follow a normal distribution. This test is commonly used on medium to large-sized samples[14].

2. Homogeneity Test (Levene)
   The homogeneity test is used to determine whether the variances of two or more data groups are the same (homogeneous). The equality of variances is an important assumption in parametric statistical analysis, particularly in mean difference tests such as the independent t-test. The Levene test examines the hypothesis that the population variances of several groups are the same. If the variances between groups do not differ significantly, then the data is considered homogeneous and meets one of the conditions for using parametric tests.

3. Calculation of N-Gain
   N-Gain or normalized gain is a measure used to determine the effectiveness of learning outcome improvement after treatment is given. This concept is widely used in educational research to measure the improvement between pretest and posttest scores. The normalized gain takes into account the difference in scores obtained compared to the maximum possible score, thus providing a proportional picture of the level of improvement. Therefore, N-Gain not only looks at the absolute difference but also considers the potential for improvement available[15].

4. Independent t-test
   The independent t-test is a parametric statistical analysis technique used to compare the means of two unrelated groups. The main objective of this test is to determine whether there is a significant difference between two groups based on the variable being studied. This test is based on the t-distribution and considers the mean, variance, and sample size of each group. The use of the independent t-test requires normally distributed data and homogeneous variance.

5. Effect Size (Cohen's d)
   Effect size is a statistical measure used to determine the magnitude of influence or difference that occurs, regardless of its statistical significance. One of the most commonly used effect size measures is Cohen's d. Cohen's d measures the difference in means between two groups in pooled standard deviation units. Unlike significance tests that only indicate whether there is a difference or not, effect size provides information about the practical or substantive magnitude of the difference[16].

## 3. RESULTS AND DISCUSSION

Based on the research results, the average pretest score for the experimental class was 65 and for the control class was 66, indicating that the initial abilities of both classes were relatively equivalent. After the treatment, the average posttest score for the experimental class increased to 85, while the control class increased to 74.

### 3.1 RESULTS

The experimental class experienced an increase of 20 points, while the control class only increased by 8 points. The average posttest difference of 11 points indicates that Scratch-based coding learning provides a higher improvement in learning outcomes compared to conventional learning. Descriptively, this data indicates that the treatment provided is effective in significantly improving students' abilities.

Table 1. Average Pretest and Posttest Data

| Class | Pretest Mean | Posttest Mean | Improvement | Posttest Difference (Experimental–Control) |
|---|---|---|---|---|
| Experimental | 65 | 85 | 20 | 11 |
| Control | 66 | 74 | 8 | — |

### 3.1.1 Normality Test

The average pretest scores of both groups were relatively equal, with 65 in the experimental class and 66 in the control class. This shows that the initial abilities of the students were at a nearly the same level before the treatment was given. After the treatment, the average posttest score of the experimental class increased to 85, while the control class only increased to 74. This improvement indicates a difference in the development of abilities between the groups. Experimental Sig = 0.200, Control Sig = 0.187, Because > 0.05 → normal.

### 3.1.2 Homogeneity Test

The normality test using Kolmogorov-Smirnov resulted in a significance value of 0.200 for the experimental class and 0.187 for the control class. Since both values are greater than 0.05, the data are normally distributed and meet the assumptions of the parametric test. Sig Levene = 0.214 (> 0.05) → homogeneous.

### 3.1.3 N-Gain

The Levene's test produced a significance value of 0.214 (> 0.05), indicating that the variances of both groups are homogeneous. Thus, the homogeneity assumption is met. Experiment = 0.71 (high) and Control = 0.23 (low).

### 3.1.4 T-test

The N-Gain score of the experimental class is 0.71 (high category), while the control class is 0.23 (low). This shows that coding education provides a much greater improvement compared to conventional education.

### 3.1.5 Effect Size

The t-test results show that t_calculated is 6.54, which is greater than t_table 2.00. This means there is a significant difference between the two groups. An effect size of 1.68 indicates a very large practical impact. This means that coding education is not only statistically significant but also has a strong impact in educational practice.

Table 2. Overall test results

| No | Type of Test | Result | Criteria | Interpretation |
|---|---|---|---|---|
| 1 | Normality Test (Kolmogorov-Smirnov) | Sig. > 0.05 | Data are normally distributed if Sig. > 0.05 | The pretest and posttest data of both classes are normally distributed, thus meeting the requirements for parametric testing |
| 2 | Homogeneity Test (Levene’s Test) | Sig. > 0.05 | Variances are homogeneous if Sig. > 0.05 | The variances of both classes are homogeneous, therefore it is appropriate to conduct an independent t-test |
| 3 | N-Gain | 0.71 (Experimental) | ≥ 0.70 = High | The improvement in students' ability in the experimental class is categorized as high |

| | | | | |
|---|---|---|---|---|
| 4 | Independent t-Test | t_calculated = 6.54 > t_table = 2.00 | Reject $H_0$ if t_calculated > t_table | There is a significant difference between the experimental and control classes |
| 5 | Effect Size (Cohen's d) | 1.68 | ≥ 0.80 = Large | The effect of coding-based learning is practically very strong |

To clarify the analysis results, the research data is presented in the form of a graph. This graph displays the N-Gain value (0.71), t-value (6.54), and Effect Size (1.68). The visualization provides an overview of the improvement level, significance of the difference, and the strength of the impact of Scratch-based coding learning on students' abilities.

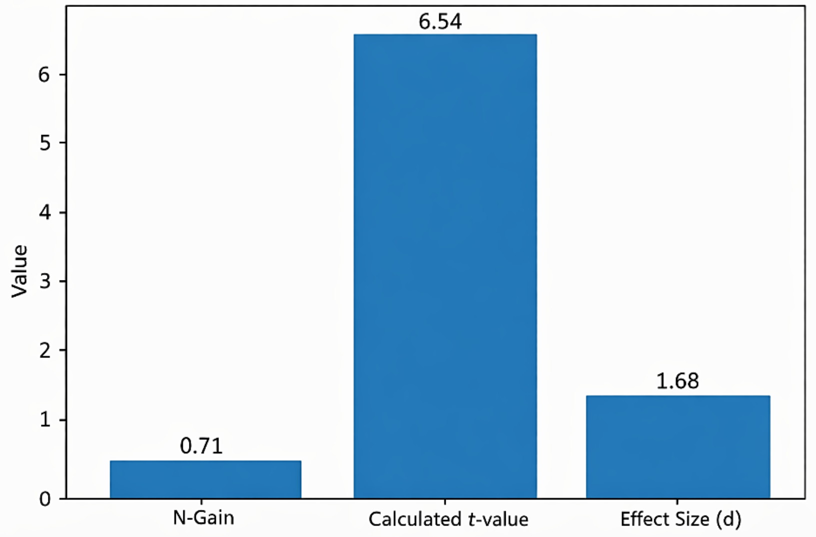


Figure 1. Results of the N-Gain, t_calculated, and Effect Size tests

**3.2 DISCUSSION**

The results of the data analysis show that Scratch-based coding learning has a very significant impact on the improvement of students' computational thinking and problem-solving abilities. This is indicated by an N-Gain value of 0.71, which falls into the high category, as well as an effect size (Cohen's d) value of 1.68, which falls into the very large category. Statistically, the t-test results also show a significant difference between the experimental class and the control class at a significance level of 0.05. These findings affirm that coding education is not only mathematically effective but also has a strong pedagogical impact in enhancing students' thinking quality. The high increase in N-Gain indicates that students experience substantial skill development after participating in Scratch-based learning. This means that the treatment provided is capable of optimally facilitating the computational thinking process, especially in the aspects of problem decomposition, pattern recognition, abstraction, and algorithm design.

The very large effect size further reinforces that the impact of coding education is practically strong, making its implementation highly relevant in the school learning context. These findings align with Piaget's constructivist theory, which emphasizes that knowledge is actively constructed by students thru direct interaction with their learning environment. In coding education, students do not merely receive information, but engage in processes of exploration, experimentation, error correction (debugging), and reflection. These processes enable deeper assimilation and accommodation of concepts.

Additionally, Vygotsky's concept of scaffolding is also evident in the coding learning process.

The teacher plays a role in providing temporary assistance when students encounter difficulties in constructing algorithms or fixing program errors. This support helps students move from the zone of actual development to the zone of proximal development, until they are finally able to complete tasks independently. Thus, coding education can be positioned as a pedagogical strategy that not only improves learning outcomes quantitatively but also supports the strengthening of 21st-century skills, such as critical, creative, collaborative thinking, and adaptability to technological developments. The integration of coding in education becomes a strategic step in preparing students to face the challenges of the digital era more competently and reflectively.

## 4. CONCLUSION

The conclusions of this study are as follows:

1. Scratch-based coding learning is effective in enhancing students' computational thinking abilities, with an increase categorized as high (N-Gain = 0.71) and supported by statistical test results showing a significant difference between the experimental class and the control class at a significance level of 0.05.
2. Coding learning has a very strong impact on students' learning outcomes, with an effect size (Cohen's d) of 1.68 indicating that the influence is not only statistically significant but also practically meaningful in the context of learning.
3. The implementation of coding in education contributes to the development of systematic thinking and 21st-century skills, activities such as algorithm design and debugging processes encourage students to think critically, analytically, creatively, and to be more adaptive to technological advancements.

## 5. SUGGESTIONS

The recommendations from this research are:

1. Schools need to systematically and sustainably integrate coding education into the curriculum. Considering that coding education has proven effective in enhancing computational thinking and has a very strong impact, its implementation should not be limited to Informatics subjects but can also be developed across disciplines to strengthen higher-order thinking skills and 21st-century competencies.
2. Improving teacher competence and technological infrastructure support should be a priority. The success of coding education in developing systematic, critical, and creative thinking highly depends on the pedagogical readiness of teachers and the availability of adequate facilities and infrastructure in schools.
3. Further research is recommended to use a broader sample scope and a more comprehensive design, including longitudinal studies. This is important to strengthen the generalization of the findings and to examine the long-term impact of coding education on students' cognitive development and thinking skills.

## REFERENCES

[1] A. Koray and E. Bilgin, "The Effect of Block Coding (Scratch) Activities Integrated into the 5E Learning Model in Science Teaching on Students' Computational Thinking Skills and Programming Self-Efficacy.," *Sci. Insights Educ. Front.*, vol. 18, no. 1, pp. 2825–2845, 2023, doi: 10.15354/sief.23.or410.

[2] W. H. Stewart and K. Baek, "Analyzing computational thinking studies in Scratch programming: A review of elementary education literature," *Int. J. Comput. Sci. Educ. Sch.*, vol. 6, no. 1, pp. 35–58, Mar. 2023, doi: 10.21585/ijcses.v6i1.156.

[3] N. D. João Piedade, "Effects of Scratch-Based Activities on 4th-Grade Students' Computational Thinking Skills," *Informatics Educ. - An Int. J.*, vol. 22, no. 3, pp. 499–523, 2023.

[4] H. Montiel and M. G. Gomez-Zermeño, "Educational Challenges for Computational Thinking in K–12 Education: A Systematic Literature Review of 'Scratch' as an Innovative Programming Tool," *Comput. 2021, Vol. 10, Page 69*, vol. 10, no. 6, p. 69, May 2021, doi: 10.3390/computers10060069.

[5] M. M. Munir, A. Muhith, and A. S. Arifin, "The Effect of Scratch-Based Instructional Media on Grade 5 Elementary Students' Coding Skills: A Quasi-Experimental Study," *J. Educ. Res. Pract.*, vol. 3, no. 3, pp. 493–508, Nov. 2025, doi: 10.70376/jerp.v3i3.410.

[6] M. Berlian, N. M. Arsad, D. Hardila, Y. Yovita, and D. N. Nasution, "Development of Scratch Learning Media to Improve Scientific Literacy and Computational Thinking in Primary Education

in The Society 5.0 Era," *J. Nat. Sci. Integr.*, vol. 7, no. 1, pp. 94–110, Apr. 2024, doi: 10.24014/jnsi.v7i1.30904.

[7] N. Hermita *et al.*, "Developing Programming Learning Media Using Scratch on the Concept of Buoyancy to Improve Computational Thinking in Primary School," *J. Nat. Sci. Integr.*, vol. 7, no. 2, pp. 274–291, Oct. 2024, doi: 10.24014/jnsi.v7i2.32554.

[8] S. C. KONG and W. Y. KWOK, "From Mathematical Thinking to Computational Thinking: Use Scratch Programming to Teach Concepts of Prime and Composite Numbers," *Int. Conf. Comput. Educ.*, Nov. 2021, Accessed: Mar. 03, 2026. [Online]. Available: https://library.apsce.net/index.php/ICCE/article/view/4199

[9] A. Büyükkarci and E. Taşlıdere, "The Effect Of Coding Education On Students' Efficiency And Scratch Achievements," *J. Educ. Technol.*, vol. 18, no. 2, Accessed: Mar. 03, 2026. [Online]. Available: https://scratch.mit.

[10] J. S. O. Zajić and J. Maksimović, "Quasi-Experimental Research as An Epistemological-Methodological Approach in Education Research," *Int. J. Cogn. Res. Sci. Eng. Educ.*, vol. 10, no. 3, pp. 177–183, 2022, doi: 10.23947/2334-8496-2022-10-3-177-183.

[11] F. Nurfalah, J. A. Utama, and M. Muslim, "Evaluating Complex Problem-Solving Test Instrument Quality in Measuring Computational Thinking Skills in Global Warming: Rasch Model Analysis," *J. Pemberdaya. Masy.*, vol. 4, no. 2, pp. 326–337, Jul. 2025, doi: 10.46843/JPM.V4I2.425.

[12] L. El-Hamamsy, M. Zapata-Cáceres, E. M. Barroso, F. Mondada, J. D. Zufferey, and B. Bruno, "The Competent Computational Thinking Test: Development and Validation of an Unplugged Computational Thinking Test for Upper Primary School," *J. Educ. Comput. Res.*, vol. 60, no. 7, pp. 1818–1866, Dec. 2022, doi: 10.1177/07356331221081753.

[13] R. Hijón-Neira, C. Pizarro, J. French, D. Palacios-Alonso, and E. Çoban, "Computational Thinking Measurement of CS University Students," *Appl. Sci. 2024, Vol. 14, Page 5261*, vol. 14, no. 12, p. 5261, Jun. 2024, doi: 10.3390/app14125261.

[14] S. Septiana, I. Syafei, and S. Sunarto, "The Effectiveness of the Flipped Classroom Learning Model on Improving Students' Critical Thinking Understanding at Taruna Vocational School Bandar Lampung," *Int. J. Adv. Sci. Educ. Relig.*, vol. 8, no. 2, pp. 336–346, May 2025, doi: 10.33648/IJOASER.V8I2.972.

[15] R. Julia, D. Salim Nahdi, and U. Cahyangingsih, "Reconceptualising Mathematical Problem Solving and Higher-Order Thinking: A Comparative Analysis of Lesson Study-Based Instruction and Conventional Teaching," *Int. J. Educ. Innov. Res.*, vol. 4, no. 2, pp. 280–288, 2025, [Online]. Available: https://doi.org/10.31949/ijeir.v4i2.15179

[16] V. Serevina, D. A. Nugroho, and H. F. Lipikuni, "Improving The Quality Of Education Through Effectiveness Of E-Module Based On Android For Improving The Critical Thinking Skills Of Students In Pandemic Era," *MOJEM Malaysian Online J. Educ. Manag.*, vol. 10, no. 1, pp. 1–20, Jan. 2022, Accessed: Mar. 03, 2026. [Online]. Available: https://mjcs.um.edu.my/index.php/MOJEM/article/view/34509