Journal of Authentic Research
https://journal-center.litpam.com/index.php/jar/index

August 2026 Vol. 5, No. 3
e-ISSN: 2828-3724
pp. 5430-5348

# The Effect of the SSCS Model on Computational Thinking Skills in Linear Motion

**[1*] Anggita Jasma Apriyola, [1*] Arif Hidayat , [2] Winny Liliawati**

[1] Department of Physics Education, Indonesia University of Education, Indonesia

*Corresponding Author e-mail: arifhidayat@upi.edu

*Received: May 2026; Revised: June 2026; Published: July 2026*

***Abstract***

*This study aims to examine the effect of the SSCS (Search, Solve, Create, and Share) learning model on improving computational thinking skills in the linear motion topic. This study employed a mixed-methods design, combining quantitative research with qualitative data from learning video transcript analysis. The research participants were 32 experimental class students and 34 control class students at a high school in Cimahi, selected through a purposive sampling technique based on their enrollment in physics and the linear motion topic. The participants were divided into an experimental class taught using the SSCS model and a control class taught using a discovery learning model. Data collection was carried out using test instruments to measure the improvement in computational thinking skills and learning video transcripts. Quantitative analysis using N-gain and t' test (due to normally distributed non-homogeneous data) and qualitative data were analyzed by providing coding on learning transcripts. The hypothesis test results showed that the application of the SSCS model had a significant effect on improving computational thinking skills compared to the control class, with t~count~ = 6.46 > t~table~ = 2.00. The N-gain score for the experimental class reached 0.54 (medium category), while the control class only reached 0.22 (low category). The highest improvement in the experimental class occurred in the abstraction and decomposition indicators, whereas the control class experienced a decline in the algorithmic skills indicator. The qualitative analysis supports these findings, demonstrating that SSCS syntax fosters computational thinking skills during learning, particularly in the create phase, which extensively trains algorithmic thinking and evaluation.*

***Keywords:*** Computational thinking; SSCS model; Linear motion.

**How to Cite:** Apriyola, A. J., Hidayat, A., & Liliawati, W. (2026). The Effect of the SSCS Model on Computational Thinking Skills in Linear Motion. *Journal of Authentic Research*, *5*(3), 5430–173. https://doi.org/10.36312/jar.v5i3.6009

https://doi.org/10.36312/jar.v5i3.6009



## INTRODUCTION

In the 21st century, students are required to continue to develop skills related to science and technology at a very rapid pace. One of these essential competencies is computational thinking skills. The computational thinking skill was first introduced by Seymour Papert in 1980 and popularized by Jeanette Wing in 2006 (Peracaula-Bosch & González-Martínez, 2024). According to Wing, computational thinking is a process of thinking in formulating problems and solutions (Malik et al., 2019). similarly, the Computer Science Teacher Association (CSTA) defines computational thinking is a process of formulating problems using various tools to get solutions to these problems (Maharani et al., 2020; Malik et al., 2019).

According to Selby and Woollard (2013), computational thinking involves problem solving activities that require a thorough understanding of conditions through decomposition, abstraction, algorithm design, pattern generalization, and evaluation (Selby & Woollard, 2010). Decomposition is the skill of identifying and breaking down complex problem into simple, manageable parts. Abstraction is the ability to filter out irrelevant details and focus only on essential information. Algorithm design involves formulating a step-by-step logical sequence to achieve a solution. Pattern generalization is the ability to identify recurring patterns and apply previous solution to solve new, similar problems. Evaluation is the skill off assessing the generated solution to ensure it effectively resolves the problem (Ihsan et al., 2021; Latifah et al., 2022; Maharani et al., 2024; Rozady & Koten, 2021; Selby & Woollard, 2010; Boshrabadi & Hosseini, 2021; Woollard, 2013).

However, many students' computational thinking skills remain relatively low. This is because there is still a lot of learning in schools that is conventional and not problem based. So that students are not used to thinking systematically. For instance, research conducted by Kamil in 2020 to analyze the mathematical thinking skills of students of number pattern material can be seen as a result that as many as 48% of students were in the low category, 16% are in the fair category, and 36% are in the good category (Kamil, 2021). In addition, another study conducted by Hapizah in 2026 obtained the results of computational thinking skills with a decomposition indicator of 33.33%, pattern recognition was 39.47%, algorithm 30.26%, pattern generalization 49.12%. Although it is quite good, the results of students' computational thinking skills have not received satisfactory results, so they need to be improved again (Hapizah et al., 2026)

One learning model capable of training computational thinking skills is the SSCS (search, solve, create, and share) model. The SSCS model provides students with structured opportunities to solve problems systematically. The implementation of the SSCS learning model is expected to improve students' computational thinking skills because it systematically teaches the step-by-step process of problem solving (Batul et al., 2022; Saâ et al., 2021). Several previous studies have investigated the effect of this model for instance, research by Islami demonstrated that SSCS learning positively impacted computational thinking skills within the topic of two variable linear equation system among junior high school students (Islami et al., 2023).

The SSCS model proposed by Edward Pizzini consist of four syntactic phases: search (identifying the problem), solve (planning the problem solving), create (generating the solution),and share (presenting the result) (Kamaruddin et al., 2023; Pizzini et al., 1989; Kinanti et al., 2025; Yuliana et al., 2024; Hidayat et al., 2021) . in the search phase, student identify problem and collect relevant facts. In the solve phase, students apply their skill to analyze these facts and determine potential solutions. During the create phase, students implement their chosen solutions, this phase also allows learners to continually evaluate their thinking process. This final phase, share, students communicate their solution, processes, and result with their peers (Pizzini et al., 1989).

While previous research has predominantly highlighted improvements in computational thinking skills within junior high school mathematics, there is a need to explore this effect within physics education. Physics learning involves numerous activities, such as practicums and complex problem solving, that are highly conducive to training computational thinking skills. Specifically, the linear motion topic provides an excellent context for problem analysis and resolution because it is a fundamental and highly systematic Subject. Through this topic, computational thinking skills indicators can be explicitly trained during the learning process, for example by decomposing problem into simpler parts, recognizing pattern of magnitude changes in linear motion, developing algorithms, filtering essential information through abstraction, and evaluating proposed solutions. Furthermore, linear motion concepts are closely related to students' daily lives, making the problem-solving process more contextual and effective.

Based on this description, this study focuses on looking at the influence of the SSCS (Search, Solve, Create, and Share) model on improving computational thinking skills in linear motion topic.

## METHOD

The study employed a mixed-methods design, primarily utilizing a quantitative approach supported by qualitative data. The quantitative approach was utilized to examine the effect of the SSCS (search, solve, create, and share) model on the improvement of students' computational thinking skills. Concurrently, a qualitative approach was applied to observe how the SSCS model facilitates computational thinking practice through the analysis of six learning video transcripts. Comprising three videos from the experimental class and three from the control class.

The participants of this study were eleventh-grade students from a senior high school in the city of Cimahi. The samples were selected using a purposive sampling technique, which was chosen rather than random assignment due to specific criteria aligned with the research objectives. The criteria for sample selection included students who were actively enrolled in physics classes covering the linear motion topic according to the school curriculum, and who were willing to participate in the research activities. These intact groups were designated as an experimental class consisting of 32 students and a control class of 34 students. To maintain objectivity and ensure equivalence, both classes were facilitated by the same teacher and received an equal instructional duration. The learning process for both classes was conducted over three meetings: the first and third meetings were allocated 3 instructional periods, while the second meeting was allocated 2 instructional periods. Furthermore, extraneous variables were strictly controlled. Both classes utilized equivalent learning materials and evaluation instruments, whereas the students' worksheets were specifically adapted to align with the learning model applied in each respective class. The treatment in both groups was configured to be identical in all other aspects, differing exclusively in the application of the SSCS learning model within the experimental class. Regarding the learning materials, the first meeting discussed uniform linear motion, the second meeting uniformly accelerated linear motion, and the third meeting focused on vertical motion.

The instrument used to measure computational thinking skills was a multiple-choice test comprising 19 items. This final number of test items was determined after conducting statistical test for empirical validity and reliability. Before the empirical testing phase, the instrument underwent expert judgment by two educational practitioners (teacher) and one academician (lecturer) to review its content validity and feasibility. The development of this instrument was grounded in computational thinking indicators, which encompass abstraction, decomposition, pattern generalization, algorithm design, and evaluation.

The quantitative data analysis was preceded by prerequisite testing, which included a normality test and a homogeneity of variance test. The prerequisite analysis results indicated that the data were normally distributed but non-homogeneous. Therefore, hypothesis testing was conducted using the t'-test (t-test for unequal variances) to determine the significance of the improvement in computational thinking skills. For the qualitative analysis, data derived from the learning video transcripts were analyzed through a coding process for each computational thinking skills indicator identified during the instructional sessions. The coding process was conducted independently by two researchers to ensure inter-rater reliability. Any discrepancies in coding were resolved through discussion to reach a consensus. This qualitative analysis aimed to complement and contextualize the quantitative findings.

## RESULTS AND DISCUSSION

Before the treatment was administered, a pre-test was conducted to determine the students' initial abilities. Following the learning process, a post-test utilizing the same test instrument was administered to evaluate the improvement in their computational thinking skills. The pre-test and post-test results for the experimental class (taught using the SSCS model) and the control class (taught using the discovery learning model) are presented in Table 1.

**Table 1.** Comparison of pre-test and post-test score between classes

| | **Experimental Class** | | **Control Class** | |
|---|---|---|---|---|
| | **Pre-test** | **Post-test** | **Pre-test** | **Post-test** |
| Lowest Score | 15.79 | 42.11 | 42.11 | 36.84 |
| Highest Score | 68.42 | 94.74 | 52.63 | 94.74 |
| Average | 34.8 | 69.41 | 40.25 | 54.18 |

As shown in Table 1, the mean pre-test score of the control class was higher than that of the experimental class, indicating that the control group possessed slightly better initial skills. However, after receiving their respective treatments, the experimental class demonstrated a substantially higher average post-test score. The magnitude of this improvement in computational thinking skills can be further analyzed using the N-gain score for both classes.

**Table 2.** N-gain Computational Thinking Skills

| | **Experiments** | **Control** |
|---|---|---|
| Highest | 0.92 | 0.91 |
| Lowest | 0.27 | -0.33 |
| Average | 0.54 | 0.22 |
| Categories | Medium | Low |

Table 2 shows that the mean N-gain score for experimental class was 0,54 which falls into the medium category, whereas the control class achieved a mean N-gain of 0,22 placing it in the low category. In the control class, the lowest N-gain score in the control class was negative (-0,33), indicating that some students in this group experienced a decline in their computational thinking skills after learning process. This proves that the treatment given in the experimental class has a positive impact on computational thinking skills not only by increasing the average but also by maintaining the improvement of computational thinking skills.

To evaluate the effect of the SSCS model on improving students' computational thinking skills, a hypothesis was conducted. Before conducting this test, prerequisite tests were performed which indicated that the data normally distributed but non-homogeneous. Consequently, a t′-test was utilized for the analysis. The statistical test was performed using Microsoft Excel based on the following hypotheses $H_0$: There was no significant difference in improvement between the experimental class and the control class and $H_1$: There was a significant difference in the improvement between the experimental class and the control class. The results of the test are presented in Table 3.

**Table 3.** Test t'

| | |
|---|---|
| t Stat | 6,46150865 |
| P(T<=t) one-tail | 1.09559E-08 |
| t Critical one-tail | 1,671093032 |
| P(T<=t) two-tail | 2.19118E-08 |
| t Critical two-tail | 2,000995378 |

Based on Table 3 , the hypothesis test obtained a calculated t of 6.46 and a critical t-value (t-table) of 2.00. The value $t_{count} > t_{table}$ (6.46 > 2.00), the null hypothesis ( $H_0$) is rejected and the alternative hypothesis ($H_1$) is accepted. It can be concluded that there is a significant difference in the improvement of computational thinking skills between the experimental class, which was taught using the SSCS model, and the control class, which was taught using the discovery learning model.

The improvement of computational thinking skills was analyzed again specifically on each indicator of computational thinking skills through the N-gain calculation presented in Figure 1.

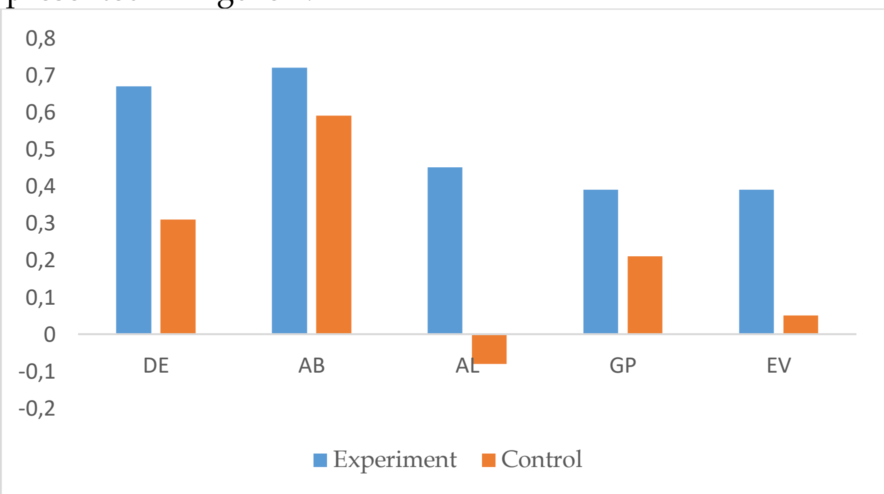


**Figure 1.** N-gain comparison per Experimental and control class indicator

As illustrated in Figure 1, the experimental class consistently achieved higher N-gain scores than the control class across all computational thinking indicators. The most significant improvements within the experimental class were observed in the abstraction (AB) and algorithm (AL) indicators. Conversely, the control class experienced a decline in the algorithm (AL) indicator, as evidenced by negative N-gain score. To complete the quantitative findings, a qualitative analysis was conducted on the learning video transcript. The analysis involved coding the observed events based on the emerging computational thinking skills indicators: DE (decomposition), AB (Abstraction), AL (Algorithm), GP (Pattern Generalization), and EV (Evaluation). After assigning a specific code to each event demonstrating computational thinking skills, the frequency of these indicators at each meeting was calculated. The results of this frequency analysis are presented in Figure 2.

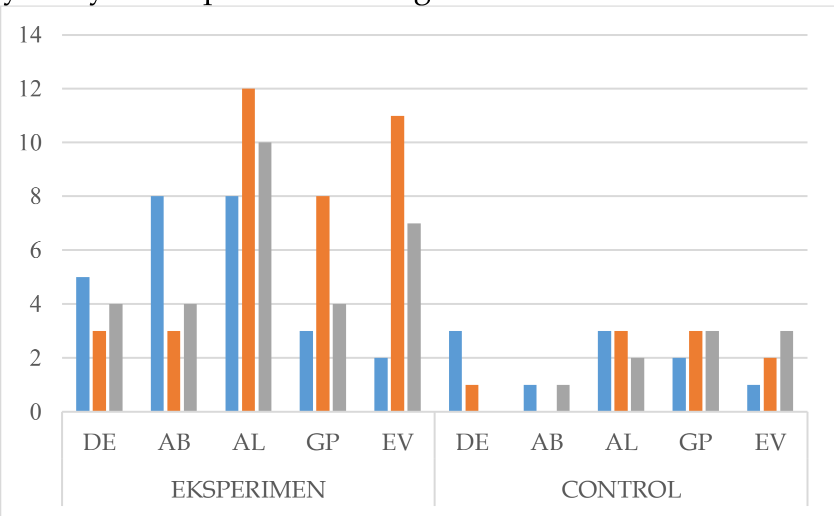


**Figure 2.** Frequency of CT indicators during learning

Figure 2 illustrates a stark contrast in the frequency of activities reflecting computational thinking skills between the experimental and control classes. The experimental class exhibited a higher frequency of computational thinking indicators, particularly during the second meeting. The most prominent indicators observed were algorithm (AL) and evaluation (EV), both occurring with very high frequency (exceeding 10 instances during the second meeting). Conversely, the appearance of these indicators in the control class was substantially lower. Furthermore, certain indicators were completely absent in the control class. Specifically, the abstraction (AB) indicator during the second meeting and the decomposition (DE) indicator during the third meeting.

During the learning process, students are trained to investigate problems consciously by identifying essential information and discarding irrelevant data. This process directly cultivates student' abstraction skills. Furthermore, students' decomposition skills enable them to break down complex problems and delineate solution trajectories independently, reducing reliance on teacher guidance. The algorithm indicator emerged as the most frequently observed skill. During the learning process, algorithmic thinking was calculated by requiring students to design problem solving strategies, create flowchart, and conduct practicums. Although teacher guidance was prominent during the first meeting, students demonstrated significant autonomy in subsequent sessions. The evaluation indicator also showed a notable increase by the second meeting. At this phase, student became proficient at identifying calculation and data collection errors, and they were able to verify whether their proposed solutions effectively resolved the given problem. Furthermore, regarding the pattern generalization indicator, student demonstrated the ability to make prediction based on established patterns and determine relationships between variables using the acquired data.

The discrepancy between the quantitative and qualitative results of the experimental and control classes stems from the fact that the SSCS model syntax is inherently aligned with computational thinking skills. During search and solve, students encounter physics problem related to linear motion. Here, students must search for and select pertinent information, which aligns with the abstraction indicator. Furthermore, students break the problems down into simpler components, reflecting decomposition. During solve, student design problem solving strategies and are required to create flowchart, demonstrating the algorithm indicator. (Maharani et al., 2020). Create directly reinforces algorithmic thinking. This alignment explains the high frequency of the algorithm indicator observed in the experimental class (figure 2) and the correspondingly high N-gain (figure 1). Create requires students to formulate systematic steps, evaluate their calculation, and design accurate procedures, culminating in a high evaluation indicator. In addition to algorithms and evaluation, this phase also gives rise to a pattern generalization (GP) indicator in looking at the relationship between variables and chart analysis. In the final phase, the share at the stage of students evaluating the results of problem solving from other groups.

The findings of this study align with the standards by the international society for technology in education (ISTE), which emphasizes that computational thinking skills include the ability to analyze data, think abstractly and algorithmically and break down problems into simpler ones(Maharani et al., 2020). Fundamentally, the SSCS model is a problem-solving learning approach. This is consistent with research conducted by Islami which concluded that the SSCS model has an influence on computational thinking skills which are classified as moderate. Therefore, this study not only confirms existing literature but also expands the empirical evidence demonstrating that the SSCS (Search, Solve, Create, and Share) model is an effective model for solving physics problems, especially in linear motion topic. However, this study has several limitations, including its implementation in only one school, focus on a single topic (linear motion), and relatively limited duration of treatment. Future research should address these limitations by expanding the scope of the study to multiple schools, diverse physics topics, and longer intervention periods to further validate the effectiveness of the SSCS model.

## CONCLUSIONS

In conclusion, the SSCS (search, solve, create, and share) learning model significantly improves students' computational thinking skills within the linear motion topic. The SSCS model effectively facilitates activities that train computational skills. Which is particularly evident in the abstraction, decomposition, and algorithm indicators. The inherent syntax of the SSCS model seamlessly guides students to deconstruct physics problems, design problem solving strategies, evaluate their solutions, and collaboratively share thir findings. Furthermore, this study demonstrates that SSCS model is substantially more effective than the discovery learning model in improving computational thinking skills. This is because during the learning process with the SSCS model, all computational thinking skills are trained, which can be seen in the results of video transcript analysis.

## RECOMENDATION

Based on the results of the research on the effect of the use of the SSCS (Search, Solve, Create, and Share) model on computational thinking skills through learning physics on linear motion topic, the researcher has suggestions in collecting data in the form of learning videos, make sure the device used is in good condition and make sure the audio is clearly sounded to make it easier to make a learning video transcript. Additionally, future research should explore the application of the SSCS model across other physics topics to further validate its broader effectiveness in fostering computational thinking skills.

## REFERENCES

Batul, F. A., Pambudi, D. S., & Prihandoko, A. C. (2022). Pengembangan Perangkat Pembelajaran Model SSCS dengan Pendekatan RME dan Pengaruhnya terhadap Kemampuan Berpikir Komputasional. *AKSIOMA: Jurnal Program Studi Pendidikan Matematika*, *11*(2), 1282. https://doi.org/10.24127/ajpm.v11i2.5074

Boshrabadi, A. M., & Hosseini, M. R. (2021). Designing collaborative problem solving assessment tasks in engineering: an evaluative judgement perspective. *Assessment & Evaluation in Higher Education*, *46*(6), 913-927. https://doi.org/10.1080/02602938.2020.1836122

Hapizah, H., Mulyono, B., & Cahyawati, D. (2026). Fostering students' computational thinking: Student worksheets on integer material. *Infinity Journal*, *15*(1), 19-36. https://doi.org/10.22460/infinity.v15i1.p19-36

Hidayat, S., Nufus, H., & Muhandaz, R. (2021). THE EFFECT OF APPLICATION OF SEARCH, SOLVE, CREATE, AND SHARE (SSCS) LEARNING MODEL ON STUDENTS'MATHEMATICS COMMUNICATION ABILITY BASED ON PREVIOUS KNOWLEDGE IN MATHEMATICS IN JUNIOR HIGH SCHOOL. *Jurnal Prinsip Pendidikan Matematika*, *3*(2), 34-42. https://doi.org/10.33578/prinsip.v3i2.88

Ihsan, H., Sutamrin, & Zaki, A. (2021). *Keterampilan Berpikir Komputasi pada Matematika*. Nas Media Pustaka.

Islami, A., Fatra, M., & Diwidian, F. (2023). Model Search, Solve, Create, and Share untuk Meningkatkan Kemampuan Berpikir Komputasi Matematis Siswa Berdasarkan Self Efficacy. *Plusminus: Jurnal Pendidikan Matematika*, *3*(3), 453–468. https://doi.org/10.31980/plusminus.v3i3.1508

Kamaruddin, I., Irvan, I., Boari, Y., Sakti, B. P., & Manoppo, Y. (2023). The Effectiveness Analysis Of Search, Solve, Create And Share (Sscs) Learning Model Implementation On Students' Cognitive Ability. *Mudir: Jurnal Manajemen Pendidikan*, *5*(1), 178-183. https://doi.org/10.55352/mudir.v5i1.348

Kamil, M. R. (2021). Analisis kemampuan berpikir komputasional matematis Siswa Kelas IX SMP Negeri 1 Cikampek pada materi pola bilangan. *AKSIOMA: Jurnal Matematika dan Pendidikan Matematika*, *12*(2), 259-270. https://doi.org/10.26877/aks.v12i2.8447

Kinanti, W., Rahmi, A., Unaida, R., Fitriani, H., & Alvina, S. (2025). EFFECT OF SEARCH, SOLVE, CREATE, AND SHARE (SSCS) LEARNING MODEL ON STUDENTS'HIGHER ORDER THINKING SKILLS AND PSYCHOMOTOR SKILLS. *Chimica Didactica Acta: Journal of Chemistry & Chemistry Education*, *13*(2). https://doi.org/10.24815/jcd.v13i2.55

Latifah, S., Diani, R., Lusiana, S., & Malik, M. (2022). Jurnal Penelitian dan Pengembangan Pendidikan Fisika. *JPPPF*, *8*(2). https://doi.org/10.21009/1

Maharani, P. P., Juandi, D., & Nurlaelah, E. (2024). Analisis kemampuan computational thinking peserta didik SMP dalam menyelesaikan masalah matematika ditinjau dari mathematical habits of mind. *Sigma Didaktika: Jurnal Pendidikan Matematika*, *12*(1), 1-20.

Maharani, S., Nusantara, T., As'ari, A. R., & Qohar, A. (2020). *COMPUTITONAL THINKING Pemecahan Masalah di Abad Ke-2*. BuatBuku. com.

Malik, S., Prabawa, H. W., & Rusnayati, H. (2019). Peningkatan kemampuan berpikir komputasi siswa melalui multimedia interaktif berbasis model quantum teaching and learning. *International Journal of Computer Science Education in Schools*, *8*(1), 41.

Peracaula-Bosch, M., & González-Martínez, J. (2024). Towards a hermeneutics of computational thinking in Wing's approximations. *Journal of Educational Computing Research*, *61*(8), 1675-1694. https://doi.org/10.1177/07356331231193142

Pizzini, E. L. (1989). A rationale for and the development of a problem solving model of instruction in science education. *Science Education*, *73*(5), 523-34.

Rozady, M. P., & KOTEN, Y. P. (2021). SCRATCH SEBAGAI PROBLEM SOLVING COMPUTATIONAL THINKING DALAM KURIKULUM PROTOTIPE. *Increate-Inovasi Dan Kreasi Dalam Teknologi Informasi*, *8*(1), 11-17.

Saâ, F. N., Mania, S., & Suharti, S. (2021). Pengembangan instrumen tes untuk mengukur kemampuan berpikir komputasi siswa. *JPMI (jurnal pembelajaran matematika inovatif)*, *4*(1), 17-26. https://doi.org/10.22460/jpmi.v4i1.17-26

Selby, C. C., & Woollard, J. (2010). *Computational Thinking: The Developing Definition*.

Selby, C., & Woollard, J. (2013). The Developing Concept of Computational Thinking. *Informatics in Education, October 2018*, 1-3.

Yuliana, Y., Zuhdi, M., & Gunada, I. W. (2024). Pengaruh model pembelajaran SSCS (search, solve, create, and share) berbantuan e-LKPD terhadap penguasaan konsep fisika peserta didik. *Jurnal Penelitian dan Pembelajaran Fisika Indonesia*, *6*(1). https://doi.org/10.29303/jppfi.v6i1.238