

Scratch and Computational Thinking in Elementary School: A Meta-analysis

Maulana Malik Ibrahim¹, Eveline Siregar², Uwes Anis Chaeruman³

¹ Universitas Negeri Jakarta, Jakarta, Indonesia; maulanamalikibrohim_9901819016@mhs.unj.ac.id

² Universitas Negeri Jakarta, Jakarta, Indonesia; esiregar@unj.ac.id

³ Universitas Negeri Jakarta, Jakarta, Indonesia; uwes@unj.ac.id

ARTICLE INFO

Keywords:

Scratch;
Computational Thinking;
Programming;
Elementary School

Article history:

Received 2022-06-24

Revised 2022-09-19

Accepted 2023-07-19

ABSTRACT

Computational thinking has become a fundamental skill that all students must possess. Learning to program with Scratch is considered the most effective way to improve students' computational thinking skills. Many studies prove its effectiveness in learning in elementary schools. However, no study discusses the systematic aggregation of the results of these studies, especially in the context of learning programming in elementary schools. Therefore, this meta-analysis aims to calculate the cumulative effect of twenty-two data sets obtained from fourteen relevant quantitative research results. The data collection procedure was carried out using PRISMA and analyzed using the JASP statistical test software. The calculation of effect size shows that the use of Scratch has significant effects on the computational thinking skills of elementary school students, both as a whole and in each of its dimensions. The dimensions of computational thinking skills that are most affected are computational concepts and practices. Differences also influence these skills in student age, where students aged 9-12 years are most affected by the use of Scratch in learning. However, it was not found that there was a significant effect of differences in the duration of learning programming using Scratch on computational thinking skills in all dimensions.

This is an open-access article under the [CC BY-NC-SA](https://creativecommons.org/licenses/by-nc-sa/4.0/) license.



Corresponding Author:

Maulana Malik Ibrahim

Universitas Negeri Jakarta; maulanamalikibrohim_9901819016@mhs.unj.ac.id

1. INTRODUCTION

Computational thinking has become a fundamental skill that not only must be possessed by computer scientists but must be possessed by everyone (Jiang & Wong, 2019) and 21st-century students (Shute, Sun, & Asbell-Clarke, 2017). OECD (Organization for Economic Co-operation and Development) has included this computational thinking skill as an aspect of mathematics assessment

in the 2021 PISA framework. Computer Science Teacher Associate (CSTE) and International Society for Technology in Education (ISTE) define computational thinking in Operational Definition of Computational Thinking for K-12 Education as a problem-solving process that includes abstraction, decomposition, algorithm design, data collection, and analysis with tasks such as pattern recognition, automation, and generalization (Güven & Gulbahar, 2020). Even some countries in the world, including Malaysia (Chongo, Osman, & Nayan, 2021) and Thailand (Threekunprapa & Yasri, 2020) already integrated these skills into their national curriculum. In Indonesia, the need for students' computational thinking skills has begun to be accommodated through plans to include computational thinking as one of the competencies in the Indonesian children's learning system. This step has already been implemented by Google through the Google Educator Group and Bebras Indonesia (Zahid, M, *et al.* 2021).

Since programming is something that can be learned by everyone, regardless of age (Ciftci & Bildiren, 2020), it is commonly utilised to develop elementary school kids' computational thinking skills (Wong & Cheung, 2020). Programming is so important that it is taught in elementary schools in several nations. While the United Kingdom, Estonia, Finland, and Sweden incorporate programming into their regular curriculum, other countries including, Australia, New Zealand, South Korea, Norway, the United States of America, and Macedonia make programming a new subject in elementary schools (Mason & Rich, 2020; Pérez-Marín *et al.*, 2020).

Many programming applications are used in learning, such as micro: bit (Carlborg, Tyrén, Heath, & Eriksson, 2019), Pencil Code (Deng, Pi, Lei, Zhou, & Zhang, 2020), Alice (Allsop, 2019), and ScratchJr (Pao Nan Chou, 2020). In the context of elementary schools, the most widely used and proven most effective application is Scratch (Tikva & Tambouris, 2021). Scratch is considered to have many advantages and is suitable for use in learning programming for elementary school students because of its ease of use and access (Fidai, Capraro, & Capraro, 2020). The ease of use of Scratch makes learning programming possible for students to learn from an early age (Zhang & Nouri, 2019).

Various studies have been conducted to measure the effectiveness of learning programming using Scratch. Some of these studies refer to a framework developed by Brennan & Resnick (2012), a framework designed to study and assess the computational thinking skills of K-12 education students through learning programming. This framework divides various computational thinking skills into three dimensions (concepts, practices and perspectives). Researches examine the effect of using Scratch on various dimensions of computational thinking, such as the dimension of concepts (Grover & Basu, 2017; Kafai & Vasudevan, 2015; Lee, Kim, & Lee, 2017; Tsukamoto *et al.*, 2017; Von Wangenheim, Alves, Rodrigues, & Hauck, 2017; Weng & Wong, 2017), dimension of practices (Hoover *et al.*, 2016; Statter & Armoni, 2017; Strawhacker *et al.*, 2018), and the dimension of perspectives (Falloon, 2016; Jun, Han, & Kim, 2017; Sáez-López & Sevillano-García, 2017).

These various studies have shown positive results, where the use of Scratch applications in programming learning is proven to improve student's skills in each dimension as well as overall computational thinking. Although many studies prove that the use of Scratch has been successful, there has been no study that discusses the systematic aggregation of the results of these studies, especially in the context of learning programming in elementary schools. Therefore, the researcher tried to do a meta-analysis of the effect of using Scratch on the computational thinking skills of elementary school students.

This meta-analysis aims to synthesize the accumulated results of various studies that have been conducted regarding the effect of using Scratch on the computational thinking skills of elementary school students. Where this research specifically examines: (a) how significance the effect of using Scratch on the computational thinking skills of elementary school students; (b) how significant is the use of Scratch has on each dimension of computational thinking skills (concepts, practices, perspectives); and (c) moderator effect on computational thinking skills of elementary school students who are learning to program using Scratch.

2. METHODS

The method used in this study is a meta-analysis by calculating the cumulative effect sizes from several quantitative studies (Pigott, 2012). Various research articles were collected from Publish or Perish (PoP) 7 with Crossref, PubMed, and Scopus source data sets, and also several databases, including ScienceDirect, ERIC dan EBSCOhost by search strings “Scratch” and “Computational Thinking” in the keywords and titles, as well as “Elementary OR Primary OR K-6 OR K6”. The time span of publication in the search process is limited between 2007 – 2021 because Scratch was first launched in 2007, so the researcher assumes that there will be no research discussing the use of Scratch before 2007.

2.1. Articles Selection Criteria

The various articles collected are then selected based on the criteria: (a) in English; (b) published between 2007 – 2021; (c) elementary school student as a research subject; (d) the research context focuses on the effect of Scratch on computational thinking skills; (e) published in a format that can be accessed online; dan (f) presents statistical data that can be used to calculate cumulative effect sizes. Inaccessible research articles (not available online or in paid versions) were categorized as not meeting the criteria and were not included in this meta-analysis.

2.2. Articles Submission Procedure

Systematically, the literature search procedure is carried out using the PRISMA guidelines (Preferred Reporting Items for Systematic Reviews and Meta-Analyses) by Moher et al. (2009). To determine the relevant research results, we focus on quantitative studies on two sets of artifacts in the form of journals and proceedings with a systematic process flow using the PRISMA framework. (See **Figure 1**).

From the search results, fourteen articles were collected, consisting of eleven journal articles and three conference proceedings. The researcher then identified the articles and summarized them based on the author, year of publication, type of publication, number of participants, age of the subject, and duration of the study. (See **Table 1**.) These studies used Scratch in learning programming to improve students' computational thinking skills, except for two studies that used the ScratchJr application (Pao Nan Chou, 2020; Rose, Habgood, & Jay, 2017). We still include the results of both studies in this meta-analysis because ScratchJr has the same programming language and environment as Scratch. ScratchJr is another version of Scratch developed by the same company, namely MIT Lab (Massachusetts Institute of Technology).

2.3. Coding Procedure

For grouping data sets, we use the framework Brennan & Resnick's framework (2012), which was later further developed by (Zhang & Nouri, 2019). Where computational thinking skills as a dependent variable on various research results collected are divided into three dimensions, we group these skills into 1) concepts, if the research results measure the skills of sequences, loops, events, parallelism, conditionals, operators, data, input and output; 2) practices dimension, if the research results measure incremental and interactive skills, testing and debugging, reusing and remixing, abstracting and modularizing, multimodal design, predictive thinking, reading-interpreting-communicating; dan 3) perspectives dimension, if the research results measure expressing, connecting, questioning, user interaction skills.

Based on the results of these groupings, twenty-two data sets were collected from fourteen research results in this meta-analysis. Fourteen data sets are grouped as computational concepts, three data sets as computational practices, and five data sets as computational perspectives. All data sets provide statistical results that give rise to means and standard deviations so that cumulative effect sizes and standardized mean differences are calculated using the effect sizes calculation by Cohen's *d* (1988).

After obtaining the effect size for all data sets, the standard error of each effect size is calculated. All the effects size and standard error data collected are then entered into the statistical test application.

The statistical test application that the researcher uses in this meta-analysis is JASP 0.14.1. The mean effect sizes of all data sets are combined to measure the cumulative effect size. An analysis of the Funnel Plot and Fail-Safe N visualizations resulted from statistical tests at JASP was conducted to identify the potential for publication bias. The researcher also tested the heterogeneity between the collected data sets to identify the potential for moderating variables that affect the relationship between the use of Scratch and students' computational thinking skills in learning programming. Test the heterogeneity of variance between data sets using a Q-test and analysis of the Forest Plot using a random-effects model.

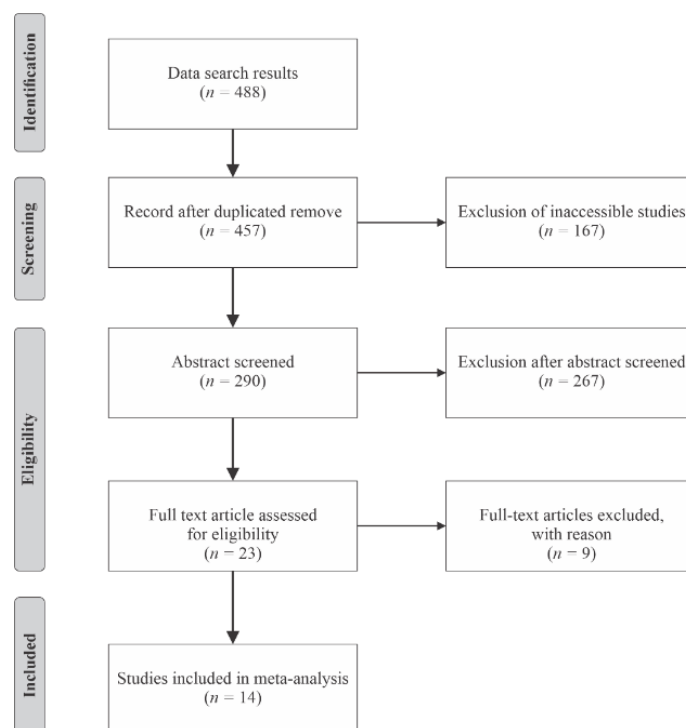


Figure 1. Article selection flowchart with PRISMA

Table 1. Summary of research result included in the meta-analysis

Author, Years	Publication	Age ¹	Duration ²	n ³
Kong, 2020	Proceedings	9 – 12	1	30
Jiang, 2021	Journal	9 – 12	5	336
Ince & Koc, 2021	Journal	9 – 12	2	17
Kong & Wang, 2019	Proceedings	9 – 12	24	1678
Mouza et al., 2020	Journal	9 – 12	9	138
Chou, 2020	Journal	5 – 9	8	12
Pérez-Marín et al., 2020	Journal	5 – 9	6	132
Rodríguez-Martínez et al., 2020	Journal	9 – 12	1	47
Durak, 2020	Journal	9 – 12	8	110
Sáez-López et al., 2016	Journal	9 – 12	82	107
Van Es et al., 2017	Proceedings	9 – 12	5	129
Kert et al., 2020	Journal	9 – 12	10	78
Wei et al., 2021	Journal	9 – 12	17	171
Rose et al., 2017	Journal	5 – 9	1	40

¹(in years); ²(in weeks); ³n=participants number

The researcher developed an initial hypothesis regarding the heterogeneity of effect sizes in data sets, in which the data sets had heterogeneous variance. If this hypothesis is accepted, it indicates the potential for moderating variables that affect the relationship between the use of Scratch and students' computational thinking skills in learning programming. The researcher divided the moderator variable into two variables, namely the age of the students and the duration of learning. Although the results of the studies included in this meta-analysis were limited to the school context, the age range of the subjects in these studies varied considerably. Therefore, the researcher grouped them based on the developed framework by Zhang & Nouri (2019). However, we only use the age grouping of 5 – 9 years and 9 – 12 years because the age group of 12 – 15 years is already high school age. In addition, the duration of the learning carried out in the study varied from one week to eighty-two weeks. Therefore, we divided the learning duration into two groups, i.e., more than or less than nine weeks. The hypothesis of each moderator variable is accepted if it affects students' computational thinking skills. This is evidenced by the test results, which show that there is a difference between the effect sizes between groups. Hypothesis testing is carried out on each variable on all dimensions of computational thinking skills.

3. FINDINGS AND DISCUSSION

3.1 Findings

The calculation result of the cumulative effect sizes of the use of Scratch on the computational thinking skills of elementary school students as a whole shows a significant effect based on the significance criteria of the effect size of Cohen (1988). Where the effect size is categorized as: (a) small : $0.10 \leq r \leq 0.30$; (b) middle: $0.30 \leq r \leq 0.50$; dan (c) big : $r \geq 0.50$. The results of calculating the cumulative effect sizes of using Scratch on students' computational thinking skills $r = 0.15$ ($CI = [0.06, 0.23]$). Of the twenty-two data sets, eleven of them have a significant effect size, while the remaining eleven are in the zero areas with a 95% confidence interval (See Figure 2).

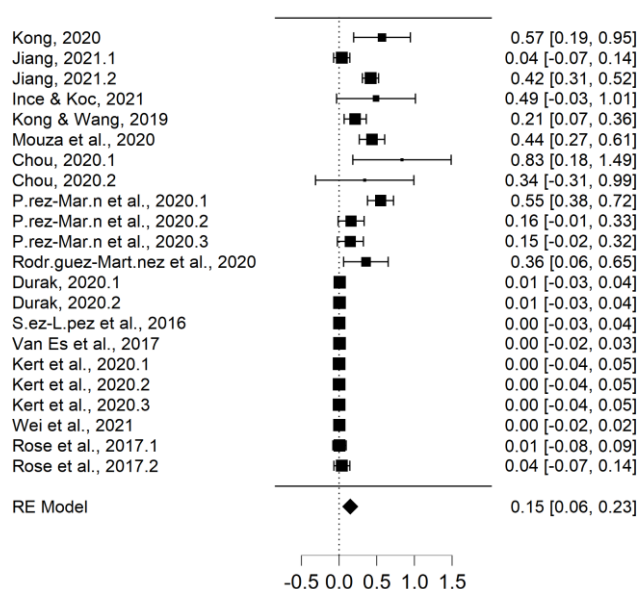


Figure 2. Forest plot the effect of Scratch on computational thinking skills

From twenty-two data sets, eleven of them have a significant effect size, while the remaining eleven are in the zero areas with a 95% confidence interval. The results of calculating the effect size of using Scratch on students' computational thinking skills on the dimension of the concept are $r = 0.18$ ($CI = [0.06, 0.30]$). From fourteen data sets, seven of them have a significant effect size, while the

remaining seven are in the zero areas with a 95% confidence interval. (See **Figure 3**). The results of calculating the effect size of using Scratch on students' computational thinking skills on the dimensions of practices are $r = 0.07$ ($CI = [-0.08, 0.22]$). From three data sets, only two data sets have a significant effect size value with a 95% confidence interval (See **Figure 4**). The calculating result of the effect size of using Scratch on students' computational thinking skills on the perspectives dimension is $r = 0.12$ ($CI = [-0.04, 0.27]$). From five data sets, only two data sets have a significant effect size with a 95% confidence interval. (See **Figure 5**). A summary of the cumulative effect sizes of each computational thinking dimension with a 95% confidence interval can be seen in **Table 4**.

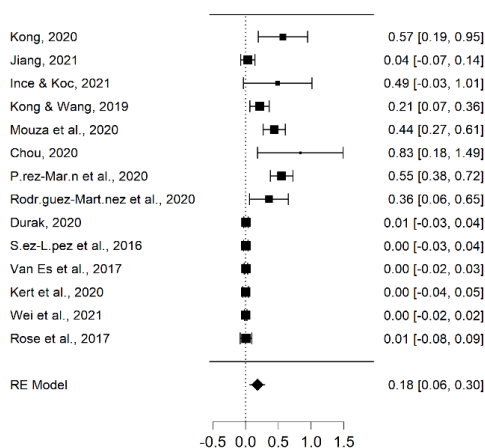


Figure 3. Forest plot the effect of Scratch on computational concepts skills

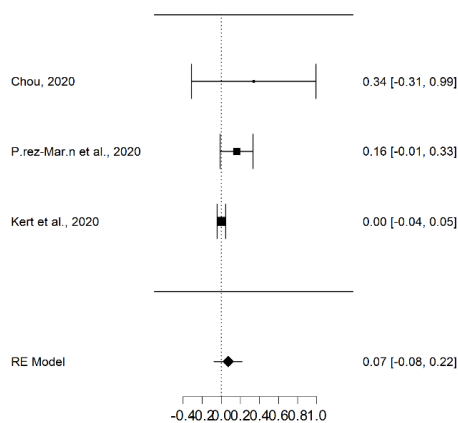


Figure 4. Forest plot the effect of Scratch on computational practices skills

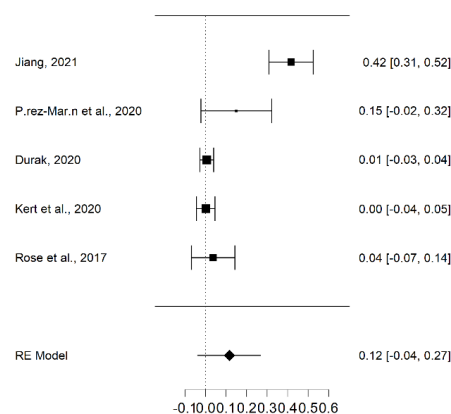


Figure 5. Forest plot the effect of Scratch on computational perspectives skills

The results of the homogeneity test on the overall effect sizes of data sets in the study stated that the homogeneity test hypothesis was rejected, thus accepting the alternative hypothesis that the variance of the effect sizes of the overall data sets was heterogeneous with $Q = 154.751$ ($df=21, p < 0.001$). These results indicate the potential for moderator variables that also influence (See **Table 2**). While the results of the homogeneity test on the effect sizes on the dimensions of computational concepts are $Q = 93.977$ ($df = 13, p < 0.001$), the computational practice dimension gets results $Q = 93.977$ ($df = 2, p < 0.001$), dimensions of skills perspectives get results $Q = 93.977$ ($df = 13, p < 0.001$). The analysis of the homogeneity test results shows that the effect sizes on all data sets have heterogeneous variances, both in terms of effect sizes on the overall computational thinking skills and on each skill dimension. These results indicate that there are moderator variables that influence the use of Scratch on students' computational thinking skills based on the hypothesis that we developed, namely student age and duration of programming learning.

Table 2. Heterogeneity test effect sizes data set results

	Q	df	p
Omnibus test of Model Coefficients	12.293	1	< .001
Test of Residual Heterogeneity	154.751	21	< .001

Note: *p*-values are approximate

Table 3. File-Safe N of overall effects of scratch in computational thinking skills.

	Fail-safe N	Target Significance	Observed Significance
Rosenthal	544.000	0.050	< .001

The results of the homogeneity test and File-Safe N analysis show that all data sets are heterogeneous, and there is no publication bias. Based on the results of the previous tests and calculations, then analysis was carried out to examine the effect size on the variables of student age and duration of programming learning on each dimension of computational thinking skills.

Three data sets show that the dimensions of students' concepts of skills in students aged 5 – 9 years have effect sizes with the acquisition scores of $d = 0.40$ ($CI = -0.07, 0.87$). Only two data sets have a significant effect, but overall they show a statistically significant effect. Effect size on each data set is not homogeneous with the value of variance $I^2 = 93.991\%$ ($p < 0.001$). Effect size on each data set is not homogeneous with the acquisition scores $d = 0.12$ ($CI = 0.02, 0.24$). Only five data sets have a significant effect, but overall they show a statistically significant effect. Effect size on each data set is not homogeneous with the value of variance $I^2 = 97.974\%$ ($p < 0.001$). The different tests of the two groups showed that the age difference of the students had a significant moderating effect on the dimensions of students' concepts skills with a $Qb = 8.08$ ($df = 1, p = 0.003$). These results prove that there is a significant difference in the dimensions of concepts skills between students in the 5 – 9 year and 9 – 12 year age groups.

Two data sets show that the dimensions of student practice skills in students aged 5 – 9 years have effect sizes with the acquisition scores $d = 0.17$ ($CI = 0.00, 0.34$). One data set shows that the dimensions of students' practice skills in students aged 9-12 years have effect sizes with the acquisition scores $d = 0.00$ ($CI = -0.04, 0.05$). The different test results of the two groups show that the age difference of students does not have a significant moderating effect on the dimensions of skills practices with a $Qb = 3.753$ ($df = 1, p = 0.053$). These results prove that there is no significant difference in the dimensions of practical skills between students in the 5 – 9 year and 9 – 12 year age groups.

Two data sets show that the dimensions of students' perspective skills in students aged 5-9 years have effect sizes with the acquisition scores $d = 0.07$ ($CI = -0.03, 0.17$). Both data sets have a significant effect. The effect size on each data set is not homogeneous with the value of variance $I^2 = 93.991\%$ ($p < 0.001$). The three data sets show that the dimensions of students' perspectives skills on students aged 9-

12 years have effect sizes with the acquisition scores $d = 0.19$ ($CI = -0.06, 0.43$). Only two data sets have a significant effect, but overall, they show a statistically significant effect. The effect size on each data set is not homogeneous, with the value of variance $I^2 = 94.457\%$ ($p < 0.001$). The different test results of the two groups showed that the age difference of the students had a significant moderating effect on the dimensions of skills concepts with $Qb = 52.563$ ($df = 1, p = 0.000$). These results prove that there is a significant difference in the dimensions of perspective skills between students in the age group of 5 – 9 years and 9 – 12 years.

Nine data sets show that the dimensions of the concepts skills of students who attend lessons for less than nine weeks have effect sizes with the acquisition scores $d = 0.24$ ($CI = 0.06, 0.43$). Only five data sets have a significant effect, but overall they show a statistically significant effect. Effect size on each data set is not homogeneous with the value of variance $I^2 = 97.779\%$ ($p < 0.001$). Five data sets show that the dimensions of the concept skills of students who take lessons for more than nine weeks have effect sizes with the acquisition scores $d = 0.12$ ($CI = -0.04, 0.28$). Only two data sets have a significant effect, but overall, they show a statistically significant effect. The effect size on each data set is not homogeneous, with the value of variance $I^2 = 98.428\%$ ($p < 0.001$). The different test results of the two groups show that the difference in the duration of programming learning has a significant moderating effect on the dimensions of concepts skills with $Qb = 0.636$ ($df = 1, p = 0.425$). These results prove that there is a significant difference in the dimensions of concepts skills between students who take programming lessons for less than nine weeks and those who are more than nine weeks.

Two data sets show that the dimensions of the practical skills of students who attend lessons for less than nine weeks have effect sizes with the acquisition scores $d = 0.17$ ($CI = 0.00, 0.34$). One data set shows that the dimensions of the practical skills of students who take lessons for more than nine weeks have effect sizes with the acquisition scores $d = 0.00$ ($CI = -0.04, 0.05$). The different tests of the two groups show that the difference in the duration of programming learning does not have a significant moderating effect on the dimensions of students' practice skills with $Qb = 3.753$ ($df = 1, p = 0.053$). These results prove that there is no significant difference in the dimensions of practical skills between students who take programming lessons for less than nine weeks and those who are more than nine weeks.

The four data sets show that the dimensions of skills in perspectives of students who attend lessons for less than nine weeks have effect sizes with the acquisition scores $d = 0.15$ ($CI = -0.04, 0.34$). Only two data sets have a significant effect, but overall, they show a statistically significant effect. The effect size on each data set is not homogeneous with the value of variance $I^2 = 93.964\%$ ($p < 0.001$). One data set shows that the dimensions of the skills of perspectives of students who take lessons for more than nine weeks have effect sizes with the acquisition of scores $d = 0.00$ ($CI = -0.04, 0.05$). The different test results of the two groups showed that the difference in the duration of programming learning did not have a significant moderating effect on the dimensions of skills concepts with $Qb = 2.503$ ($df = 1, p = 0.114$). These results prove that there is no significant difference in the dimensions of skills perspectives between students who take programming lessons for less than nine weeks and those who are more than nine weeks.

Table 4. This is a table. Tables should be placed in the main text near to the first time they are cited.

Dimension	#Effect Sizes	Mean Effect Sizes	Standard Error	95% CI
CT Concepts	14	0.18	0.062	[0.06, 0.30]
CT Practices	3	0.07	0.076	[-0.08, 0.22]
CT Perspectives	5	0.12	0.079	[-0.04, 0.27]
CT Overall	22	0.15	0.042	[0.06, 0.23]

3.2 Discussion

3.2.1 Signification of Scratch on elementary school student computational thinking skills

The results of the calculation of cumulative effect sizes in this meta-analysis indicate that programming learning using Scratch affects the computational thinking skills of elementary school students as a whole with a small significance level based on Cohen's criteria (1988). These findings demonstrate that teaching primary school pupils to code in Scratch does, in fact, increase their computational thinking skills. In addition, elementary school pupils' problem-solving abilities can benefit from using Scratch for programming education. (Rodríguez-Martínez et al., 2020; Sáez-López et al., 2016), reflective thinking skills (Durak, 2020), creative, cooperative and critical thinking (Jiang, 2021). In addition, the use of Scratch in programming learning can also increase students' learning motivation (Chou, 2020; Sáez-López et al., 2016), make students focus better on following learning (Kong & Wang, 2019), and make students more able to enjoy the learning process (Pérez-Marín et al., 2020).

3.2.2 Signification of Scratch on Each Computational Thinking Skill Dimension

The results of this meta-analysis show that the use of Scratch in programming learning has a different effect on each dimension of computational thinking skills. Several research results show that the use of Scratch in learning programming does not only show a positive effect on one skill dimension. There are several dimensions of computational thinking skills that are affected simultaneously through the use of Scratch in learning programming. The use of Scratch learning has the most influence on the dimensions of skills, concepts and perspectives, although the significance of the influence is included in the small category based on the criteria for its significance. This is in accordance with the results of research from (Chou, 2020; Wei et al., 2021). However, the use of Scratch also has a positive effect on the dimensions of skills practices, but it is not statistically significant. Sequences, loops, and parallelism skills are the skills that are most affected by the increase in the use of the Scratch programming language (Kong, 2020; Mouza et al., 2020; Pérez-Marín et al., 2020). This is due to the use of block-based in the Scratch programming environment so that it makes accessibility and use easy (Fidai et al., 2020). This is in accordance with the statement by Çiftci & Bildiren (2020), which states that a programming environment like this makes programming learning accessible to anyone. Meanwhile, from a domain perspective, the various collaborative environments presented by Scratch make the acquisition of computational thinking skills of elementary school students high. Various perspectives such as expressing and connecting are very well facilitated by the Scratch Online Community (Brennan & Resnick, 2012; Chiocciariello & Freina, 2019). The Scratch Online Community allows collaboration and pair programming activities (Wei et al., 2021).

3.2.3 Moderator effect on computational thinking skills of elementary school students

The results of the hypothesis analysis on moderator factors reflect some interesting findings from this meta-analysis. There is no attenuating effect of time spent learning Scratch programming on overall or granular measures of computational thinking ability. Pérez-Marn et al. (2020) discovered that students may be taught fundamental programming concepts like conditionals, loops, and memory in as little as a few weeks. Students' overall computational thinking skills and the dimensions of computational concepts and perceptions are moderated by their age group, but students' computational practises skills are unaffected by their age. Nine- to twelve-year-olds outperform five- to nine-year-olds on the concept and viewpoint dimensions, with effect sizes that are twice as great.

4. CONCLUSION

The foregoing discussion leads to the conclusion that teaching programming using Scratch applications has been shown to greatly increase primary school students' computational thinking skills. Using Scratch to learn programming has a different effect on each skill dimension, with skills concepts being the most strongly influenced ($d = 0.181$), perspectives skills coming in at a close second ($d = 0.117$), and practise skills coming in last ($d = 0.072$) with a value that is statistically insignificant according to the Cohen's criterion ($d < 0.1$). There was no correlation with how long it took to learn something. Meanwhile, there is a correlation between students' ages and their levels of proficiency in computational thinking. Students in the 9–12 age range benefited more from Scratch programming courses than those in the 5–9 age range, even if they achieved equivalent results on the dimension of computational practises skills. This meta-analysis has significant drawbacks. This study's findings, however, can serve as a summary and extension of the existing literature on Scratch's usefulness for teaching programming and its impact on students' computational thinking abilities. The identification of publications that do not include research results from the Web of Science and PsycNet databases severely limits the research sample used in this study. Moreover, this meta-analysis only includes research data from publications that were published in English and are available online to researchers. More studies of this nature need to broaden the readership and accessibility of published findings, including those that make use of local contexts. Researchers in the future should also pay more attention to the aspects of practise for teaching elementary school students to code using Scratch.

REFERENCES

- Allsop, Y. (2019). Assessing computational thinking process using a multiple evaluation approach. *International Journal of Child-Computer Interaction*, 19, 30–55. <https://doi.org/10.1016/j.ijcci.2018.10.004>
- Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. *Proceedings of Teh 2012 Annual Meeting of the Americans Educational Research Association*.
- Carlborg, N., Tyrén, M., Heath, C., & Eriksson, E. (2019). The scope of autonomy when teaching computational thinking in primary school. *International Journal of Child-Computer Interaction*, 21, 130–139. <https://doi.org/10.1016/j.ijcci.2019.06.005>
- Chiocciariello, A., & Freina, L. (2019). Programming to learn in primary schools: Including scratch activities in the curriculum. *Proceedings of the European Conference on Games-Based Learning, 2019-October*, 143–150. <https://doi.org/10.34190/GBL.19.070>
- Chongo, S., Osman, K., & Nayan, N. A. (2021). *Impact of the Plugged-in and Unplugged Chemistry Computational Thinking Modules on Achievement in Chemistry*. 17(4).
- Chou, P N. (2020). Using ScratchJr to Foster Young Children's Computational Thinking Competence: A Case Study in a Third-Grade Computer Class. *Journal of Educational Computing Research*, 58(3), 570–595. <https://doi.org/10.1177/0735633119872908>
- Chou, Pao Nan. (2020). Using ScratchJr to Foster Young Children's Computational Thinking Competence: A Case Study in a Third-Grade Computer Class. *Journal of Educational Computing Research*, 58(3), 570–595. <https://doi.org/10.1177/0735633119872908>
- Çiftci, S., & Bildiren, A. (2020). The effect of coding courses on the cognitive abilities and problem-solving skills of preschool children. *Computer Science Education*, 30(1), 3–21. <https://doi.org/10.1080/08993408.2019.1696169>
- Cohen, J. (1988). *Statistical Power Analysis for the Behavioral Sciences* (Second). Hillsdale, NJ: Lawrence Erlbaum.
- Deng, W., Pi, Z., Lei, W., Zhou, Q., & Zhang, W. (2020). Pencil Code improves learners' computational thinking and computer learning attitude. *Computer Applications in Engineering Education*, 28(1), 90–

104. Retrieved from <http://10.0.3.234/cae.22177>
- Durak, H. Y. (2020). The Effects of Using Different Tools in Programming Teaching of Secondary School Students on Engagement, Computational Thinking and Reflective Thinking Skills for Problem Solving. *Technology, Knowledge and Learning*, 25(1), 179–195. <https://doi.org/10.1007/s10758-018-9391-y>
- Falloon, G. (2016). An analysis of young students' thinking when completing basic coding tasks using Scratch Jnr. On the iPad. *Journal of Computer Assisted Learning*, 32(6), 576–593. <https://doi.org/10.1111/jcal.12155>
- Fidai, A., Capraro, M. M., & Capraro, R. M. (2020). "Scratch"-ing computational thinking with Arduino: A meta-analysis. *Thinking Skills and Creativity*, 38(September), 100726. <https://doi.org/10.1016/j.tsc.2020.100726>
- Grover, S., & Basu, S. (2017). *Measuring Student Learning in Introductory Block-Based Programming*. 267–272. <https://doi.org/10.1145/3017680.3017723>
- Güven, I., & Gulbahar, Y. (2020). Integrating Computational Thinking into Social Studies. *The Social Studies*, 111(5), 234–248. <https://doi.org/10.1080/00377996.2020.1749017>
- Hoover, A. K., Puttick, G., Barnes, J., Tucker-Raymond, E., Fatehi, B., Harteveld, C., & Moreno-León, J. (2016). Assessing computational thinking in students' game designs. *CHI PLAY 2016 - Proceedings of the Annual Symposium on Computer-Human Interaction in Play Companion*, 173–179. <https://doi.org/10.1145/2968120.2987750>
- Iskrenovic-Momcilovic, O. (2019). Pair programming with scratch. *Education and Information Technologies*, 24(5), 2943–2952. <https://doi.org/10.1007/s10639-019-09905-3>
- Jiang, B. (2021). Effect of Scratch on computational thinking skills of Chinese primary school students. *Journal of Computers in Education*. <https://doi.org/10.1007/s40692-021-00190-z>
- Jiang, S., & Wong, G. K. W. (2019). Primary school students' intrinsic motivation to plugged and unplugged approaches to develop computational thinking. *International Journal of Mobile Learning and Organisation*, 13(4), 336–351. <https://doi.org/10.1504/IJMLO.2019.102540>
- Jun, S. J., Han, S. K., & Kim, S. H. (2017). Effect of design-based learning on improving computational thinking. *Behaviour and Information Technology*, 36(1), 43–53. <https://doi.org/10.1080/0144929X.2016.1188415>
- Kafai, Y., & Vasudevan, V. (2015). Hi-Lo tech games: Crafting, coding and collaboration of augmented board games by high school youth. *Proceedings of IDC 2015: The 14th International Conference on Interaction Design and Children*, 130–139. <https://doi.org/10.1145/2771839.2771853>
- Kert, S. B., Erkoç, M. F., & Yeni, S. (2020). The effect of robotics on six graders' academic achievement, computational thinking skills and conceptual knowledge levels. *Thinking Skills and Creativity*, 38(August), 100714. <https://doi.org/10.1016/j.tsc.2020.100714>
- Kong, S C. (2020). Development of computational thinking concepts in scratch programming. *ICCE 2020 - 28th International Conference on Computers in Education, Proceedings*, Vol. 1, pp. 652–657. Retrieved from <https://www.scopus.com/inward/record.uri?partnerID=HzOxMe3b&scp=85099460722&origin=inward>
- Kong, Siu Cheung, & Wang, Y. Q. (2019). Assessing programming concepts in the visual block-based programming course for primary school students. *Proceedings of the European Conference on E-Learning, ECEL, 2019-Novem*, 294–302. <https://doi.org/10.34190/EEL.19.035>
- Lee, S. J., Kim, J. M., & Lee, W. G. (2017). Analysis of Factors Affecting Achievement in Maker Programming Education in the Age of Wireless Communication. *Wireless Personal Communications*, 93(1), 187–209. <https://doi.org/10.1007/s11277-016-3450-2>
- Mason, S. L., & Rich, P. J. (2020). Development and analysis of the Elementary Student Coding Attitudes Survey. *Computers and Education*, 153, 103898. <https://doi.org/10.1016/j.compedu.2020.103898>
- Moher, D., Liberati, A., Tetzlaff, J., & Altman, D. G. (2009). Preferred Reporting Items for Systematic Reviews and Meta-Analyses: The PRISMA Statement. *PLoS Medicine*, 6(7).

- <https://doi.org/https://doi.org/10.1371/journal.pmed.1000097>
- Mouza, C., Pan, Y. C., Yang, H., & Pollock, L. (2020). A Multiyear Investigation of Student Computational Thinking Concepts, Practices, and Perspectives in an After-School Computing Program. *Journal of Educational Computing Research*, 58(5), 1029–1056. <https://doi.org/10.1177/0735633120905605>
- Pérez-Marín, D., Hijón-Neira, R., Bacelo, A., & Pizarro, C. (2020). Can computational thinking be improved by using a methodology based on metaphors and scratch to teach computer programming to children? *Computers in Human Behavior*, 105. <https://doi.org/10.1016/j.chb.2018.12.027>
- Pigott, T. D. (2012). Advances in meta-analysis. In *Advances in Meta-Analysis*. <https://doi.org/10.1007/978-1-4614-2278-5>
- Rodríguez-Martínez, J. A., González-Calero, J. A., & Sáez-López, J. M. (2020). Computational thinking and mathematics using Scratch: an experiment with sixth-grade students. *Interactive Learning Environments*, 28(3), 316–327. <https://doi.org/10.1080/10494820.2019.1612448>
- Rose, S. P., Habgood, M. P. J., & Jay, T. (2017). An Exploration of the Role of Visual Programming Tools in the Development of Young Children’s Computational Thinking. *Electronic Journal of E-Learning*, Vol. 15, pp. 297–309.
- Sáez-López, J. M., Román-González, M., & Vázquez-Cano, E. (2016). Visual programming languages integrated across the curriculum in elementary school: A two year case study using “scratch” in five schools. *Computers and Education*, 97, 129–141. <https://doi.org/10.1016/j.compedu.2016.03.003>
- Sáez-López, J. M., & Sevillano-García, M. L. (2017). Sensores, programación y dispositivos en sesiones de Educación Artística. Un caso en el contexto de Educación Primaria. *Cultura y Educacion*, 29(2), 350–384. <https://doi.org/10.1080/11356405.2017.1305075>
- Shute, V. J., Sun, C., & Asbell-Clarke, J. (2017). Demystifying computational thinking. *Educational Research Review*, 22, 142–158. <https://doi.org/10.1016/j.edurev.2017.09.003>
- Statter, D., & Armoni, M. (2017). Learning abstraction in computer science: A gender perspective. *ACM International Conference Proceeding Series*, 5–14. <https://doi.org/10.1145/3137065.3137081>
- Strawhacker, A., Lee, M., & Bers, M. U. (2018). Teaching tools, teachers’ rules: exploring the impact of teaching styles on young children’s programming knowledge in ScratchJr. *International Journal of Technology and Design Education*, 28(2), 347–376. <https://doi.org/10.1007/s10798-017-9400-9>
- Threekunprapa, A., & Yasri, P. (2020). Unplugged coding using flowblocks for promoting computational thinking and programming among secondary school students. *International Journal of Instruction*, 13(3), 207–222. <https://doi.org/10.29333/iji.2020.13314a>
- Tikva, C., & Tambouris, E. (2021). Mapping computational thinking through programming in K-12 education: A conceptual model based on a systematic literature Review. *Computers and Education*, 162, 104083. <https://doi.org/10.1016/j.compedu.2020.104083>
- Tsukamoto, H., Oomori, Y., Nagumo, H., Takemura, Y., Monden, A., & Matsumoto, K. ichi. (2017). Evaluating algorithmic thinking ability of primary schoolchildren who learn computer programming. *Proceedings - Frontiers in Education Conference, FIE, 2017-October*, 1–8. <https://doi.org/10.1109/FIE.2017.8190609>
- Van Es, N., Jeuring, J., & Es, N. Van. (2017). Designing and comparing two scratch-based teaching approaches for students aged 10-12 years. *ACM International Conference Proceeding Series*, 178–182. <https://doi.org/10.1145/3141880.3141883>
- Von Wangenheim, C. G., Alves, N. C., Rodrigues, P. E., & Hauck, J. C. (2017). Teaching Computing in a Multidisciplinary Way in Social Studies Classes in School “A Case Study. *International Journal of Computer Science Education in Schools*, 1(2), 3. <https://doi.org/10.21585/ijcses.v1i2.9>
- Wei, X., Lin, L., Meng, N., Tan, W., Kong, S.-C., & Kinshuk. (2021a). The effectiveness of partial pair programming on elementary school students’ Computational Thinking skills and self-efficacy. *Computers & Education*, 160, 104023. <https://doi.org/https://doi.org/10.1016/j.compedu.2020.104023>
- Wei, X., Lin, L., Meng, N., Tan, W., Kong, S. C., & Kinshuk. (2021b). The effectiveness of partial pair

- programming on elementary school students' Computational Thinking skills and self-efficacy. *Computers and Education*, 160, 104023. <https://doi.org/10.1016/j.compedu.2020.104023>
- Weng, X., & Wong, G. K. W. (2017). Integrating computational thinking into english dialogue learning through graphical programming tool. *Proceedings of 2017 IEEE International Conference on Teaching, Assessment and Learning for Engineering, TALE 2017*, Vol. 2018, pp. 320–325. <https://doi.org/10.1109/TALE.2017.8252356>
- Wong, G. K. W., & Cheung, H. Y. (2020). Exploring children's perceptions of developing twenty-first century skills through computational thinking and programming. *Interactive Learning Environments*, 28(4), 438–450. <https://doi.org/10.1080/10494820.2018.1534245>
- Yilmaz Ince, E., & Koc, M. (2021). The consequences of robotics programming education on computational thinking skills: An intervention of the Young Engineer's Workshop (YEW). *Computer Applications in Engineering Education*, 29(1), 191–208. <https://doi.org/10.1002/cae.22321>
- Zahid, M. Z., Dewi, N. R., Asih, T. S. N., Winarti, E. R., Putri, T. U. K., & Susilo, B. E. (2021, February). Scratch Coding for Kids: upaya memperkenalkan mathematical thinking dan computational thinking pada siswa sekolah dasar. In *PRISMA, Prosiding Seminar Nasional Matematika* (Vol. 4, pp. 476-486)
- Zhang, L. C., & Nouri, J. (2019). A systematic review of learning computational thinking through Scratch in K-9. *Computers and Education*, 141, 103607. <https://doi.org/10.1016/j.compedu.2019.103607>