Hipotenusa : Journal of Mathematical Society
Volume 6 Issue 2 Year 2024
ISSN: 2716-3156
Website: https://ejournal.uinsalatiga.ac.id/index.php/hipotenusa/index

Investigation Injection of Computational Thinking Skills with Scratch on Teachers Performance: A Mixed-Method Study

Saiful Marom[1]*, Stevanus Budi Waluya[2], Scolastika Mariani[3], Bambang Eko Susilo[4], Wardono[5]
[1,2,3,4,5] Universitas Negeri Semarang, Indonesia
*Corresponding Author. Email: saifulmarom2704@students.unnes.ac.id[1]
DOI: 10.18326/hipotenusa.v6i2.2764

Article submitted: October 31, 2024
Article reviewed: November 29, 2024
Article published: December 30, 2024

Abstract

Currently, the injection of computational thinking skills in education is very important because these skills are needed in dealing with technological developments. Every teacher needs to be provided with this ability with the hope that later they will be able to inject it into every student at school. This article presents empirical experiences related to the injection of computational thinking skills for madrasah (Islamic Schools) teachers in Indonesia using the Scratch program for 60 teachers. The research method used is a mix method, namely a quantitative approach (experiment) and a qualitative approach. In quantitative analysis, two classes are used, namely experimental and control classes. In the initial stage, the measurement of computational thinking ability was carried out using 15 questions. The results of this study showed an increase in computational thinking skills in the experimental class, which was also accompanied by impressions given by participants who saw positive developments and significant explorative and practical uses in learning practices.

Keywords: injection, computationl thinking skill, scratch, mix method

INTRODUCTION

Computational thinking ability is a skill in the 21st century that is needed by people in the world, both students and teachers (Tabesh, 2017). Therefore, this ability can be used as a skill in the process of solving problems in real life by using problem-processing agents (the human mind itself, computer assistance, robots, or others) (Aho, 2011). Based on this, the development of computational thinking skills is an important thing that teachers and students need because it is closely related to the development of information technology, which is key in today's life (Del Álamo Venegas et al., 2021).

According to Jeanette M. Wing's research study, computational thinking ability is a set of skills that can be applied widely and can be developed by students in the learning process (J. Wing, 2006). In DiSessa Andrea's book, it has been mentioned the importance of integrating computational thinking skills into the school curriculum, namely by injecting them into each subject, so that there are many developments related to the definition and conceptual framework of computational thinking (Disessa, 2000). The Ministry of Education and Culture in Indonesia includes computational thinking skills in the national curriculum at the elementary school level (Indonesian Ministry of Education and Culture, 2021).

Wing's writing further defines computational thinking skills as thinking skills that involve formulating problems and how to find and solve problems using or without tools to get an effective and efficient solution (J. M. Wing, 2015). According to Aho, computational thinking ability is the ability to think in terms of collecting and using important information to get a satisfactory and optimal solution to a problem (Aho, 2011). Brennan & Resnick also describe the conceptual framework of computational thinking skills with 1) theoretical computing, which includes sequences, repetitions, events, and conditionals; 2) practical computing, which includes testing, debugging, and abstraction; and 3) computational perspectives, which are related to expressing, connecting, and questioning (Brennan & Resnick, 2012). In 2019, Moreno et al. described indicators to formulate computational thinking skills with decomposition, pattern recognition, abstraction (Moreno et al., 2019). In previous research, it has been described that understanding is related to the right pedagogical approach in the process of integrating computational thinking skills in the learning process (Soboleva, 2021). This study uses indicators to describe the computational thinking process with decomposition, pattern recognition, abstraction, algorithms, and mathematical literacy (Marom et al., 2023).

The problem that often arises is the occurrence of obstacles to the integration of students' computational thinking skills in the learning process (Marcia Linn, Alfred V. Aho, 2011). This happens because teachers' understanding of computational thinking is still low and the technique of integrating it into the learning process is not optimal (Reichert, 2020). Furthermore, a solution is needed for developing computational thinking skills for teachers, which can be achieved when adequate teacher training is carried out on computational thinking skills (National Research Council, 2012). The emergence of computational thinking skills in the learning process starts with the learning planning process, so it needs to be understood that there is a need for initial and continuous training related to this for teachers to prepare to help students develop computational thinking skills (Oi-Lam & Zhihao, 2021).

There are several techniques for integrating computational skills in the learning process, including Aminah N (Aminah, 2022). In this research study, it has been stated that the process of solving problems in learning obtained positive results by using Ed-Tech applications. Furthermore, in the research of Nugent et al, it was described that by using robotics and geolocation technology to increase student programming activities in the learning process, geospatial concepts showed a positive impact (Nugent et al., 2009).

Programming activities are indispensable in complementing digital competencies and the use of communication technology in a teacher's learning process (Pérez-Marín, 2020). One of the most popular programming languages to learn and develop in the learning process is Scratch (Romero et al., 2017). Scratch provides visual programming facilities that allow teachers, especially those in non-STEM subjects, to learn programming by manipulating sequence elements graphically without using textual coding (Molina-Ayuso, 2022). Among the most prominent and popular programming languages that are easy for students and teachers to learn to improve creative, algorithmic, and systematic thinking is Scratch (Rodríguez-Martínez, 2020).

To measure the level of development of computational thinking ability in students and teachers, a computational thinking ability test (CT-Test) is needed (Román-González, 2015). The CT-Test is used to measure computational thinking skills based on basic programming concepts and the use of logical syntax in languages such as functions, variables, loops, and others (Román-González, 2015). Based on the results of research by Dagiena V et al, it is stated that the CT-Test is one of the tools to measure computational thinking skills with a high level of validity and reliability (Dagienė & Sentence, 2016). In addition, Chiazzese G has also mentioned that the CT-Task can be used as an aid to measure computational thinking skills well (Chiazzese, 2019). Based on this, this paper will examine the integration technique of the computational thinking process in subjects using scratch programming language and examine the effectiveness of the integration technique.

Based on the mixed methods approach used, and the research focus to investigate the impact of injecting computational thinking skills using scratch, the research questions are:

1. What is the impact of Scratch in injecting computational thinking skills on teachers in the process of integrating CT in subjects?
2. What are the advantages and disadvantages of using scratch in the process of integrating computational thinking skills in teachers?

These research questions in this study will guide the research in comprehensively examining the effectiveness and teachers' experiences in using scratch in the process of injecting computational thinking skills. The questions include both quantitative measures of academic performance and qualitative esploration of participants perspective measures to provide a well evaluation of the injection computational thinking skill intervation.

METHODS

Research Design

This study begins by analyzing the level of computational thinking skills in madrasah (Islamic Schools) teachers before scratch training. Furthermore, it analyzes perspectives related to the use of scratch in the integration of computational thinking skills in subjects. To analyze aspects related to this, the experience of teachers in the process of scratch training in the process of subject integration has been designed. The research method in this paper uses mixed methods (Johnson & Onwuegbuzie, 2007).

Quantitative analysis in this study was conducted using CT-Test. In this process, the ability to create and solve problems based on basic programming concepts and the use of programming syntax such as loops, variables, and conditionals will be measured (Chan et al., 2020). This paper uses 15 questions from the CT-Test to measure computational thinking skills (Dagiene & Futschek, 2008). The CT-Test also not only focuses on computational thinking skills but also measures the teacher's skills in transferring those skills to everyday problems (Román-González, 2016).

After the quantitative analysis process, this paper is then supported by qualitative data obtained using surveys to obtain participants' impressions, which will help find out the real conditions in the classroom learning process from the perspective of teachers and students. The combination of quantitative and qualitative research methods will provide results from various perspectives to analyze various objects of study. In this paper, the quantitative research method uses a quasi-experimental design with two groups, namely, experimental and control, and in the process of measuring computational thinking skills, pretest and posttest are used (Gerbing, 1984). Furthermore, the qualitative research method is carried out with a questionnaire by collecting any aspects that refer to the development of the practice of integrating computational thinking skills in subjects. The following will give an idea of the flow of the research process in Figure 1:

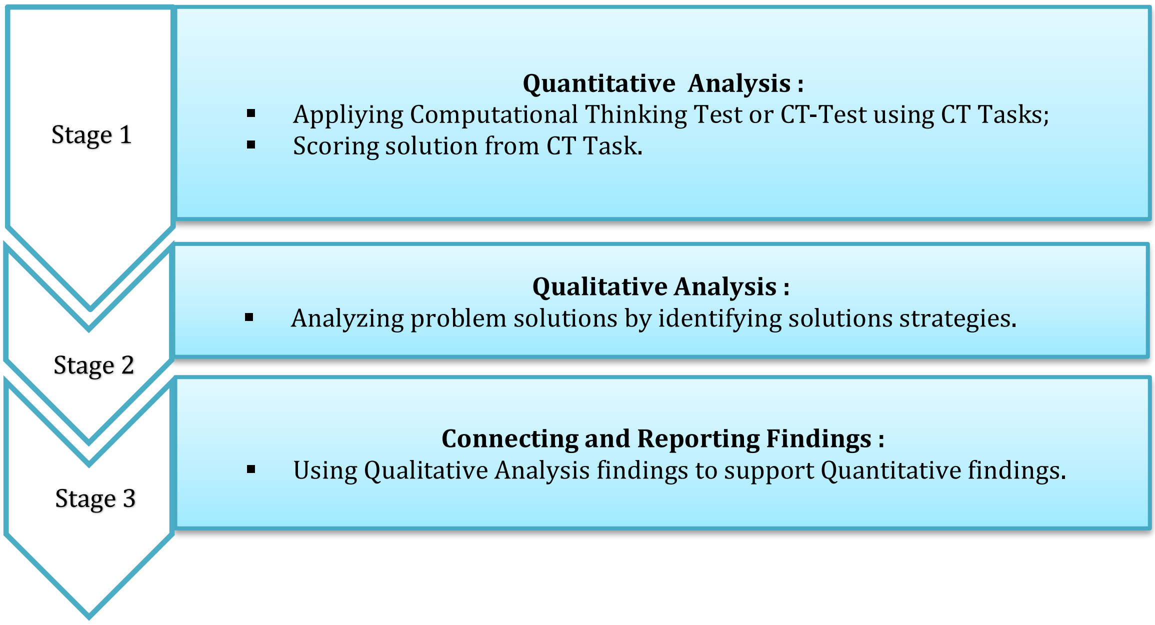

Figure 1. Flow Chart of the Reseacrh Process

Participants

This study was conducted with a sample of 60 teachers of STEM and non-STEM subjects in madrasah in Indonesia who participated in training activities. The sample for the study was selected by the purposive sampling method due to the availability of individuals relevant to this study (Uygun, 2022). The following will give the sample distribution in this research activity in Table 1 :

Table 1. Distribution of Training Participants

	Man	Women	Total
Control Group	18	12	30
Experimental Group	15	15	30
Total	33	27	60

Based on the initial measurement conducted through interviews with participants before the training activities were carried out, it was determined that the training participants, consisting of STEM and non-STEM subject teachers at madrasas in Indonesia, had knowledge about computational thinking in general but did not have specific experience related to computational thinking processes and integration in subjects. The experimental group was trained in five sessions, with the first two lasting 120 minutes and the next three lasting 180 minutes. In the first session, an explanation was given about computational thinking and its importance in the learning process. In the second session, a pretest was conducted to see the ability of computational thinking using the CT-Test measurement tool. In the next three sessions, the participants were given a practicum using scratch.

Practical activities In the third session, participants were given the introduction of tools from scratch with the aim of introducing participants to the worksheet of the scratch program and introducing elements in scratch such as loops, variables, and others. By using the scratch programming approach, the training participants compiled an animation for the learning process; for example, participants from religion teachers made an animation of the prayer process at the prayer service. Furthermore, there are math teacher participants who make animations related to learning about cartesian plane coordinates.

Furthermore, during the practicum activities in the fourth session, participants began to learn to define interactive elements and create programming using concepts such as conditional loops such as if then, repeat until, and others. In the fifth session of practicum activities, trainees create animation programs related to the learning process in each subject they teach. Participants used various codes in the scratch menu to construct learning by solving problems related to the learning process. In addition, in this session, participants created game animations to hunt objects that appear on the screen display by integrating the learning materials of each participant. In the process of making this game, participants are asked to use variables to store the scores obtained, conditional loops, and several other scratch codes.

This practicum activity is focused on overcoming various problems in teaching materials that are contextualized and integrated into the computational thinking process. Empirical experience is used in this training to introduce learning methods and strategies in the process of integrating computational thinking into the subject. In this activity, participants must complete a series of exercises that have been designed, and then evaluation activities will be carried out at the end of the session. The control group conducted the pretest and posttest at the same time as the experimental group.

RESULT AND DISCUSSION

Quantitative Analysis

The results obtained from the experimental and control groups that have been tested can be seen in Figure 2 and Table 2 below:

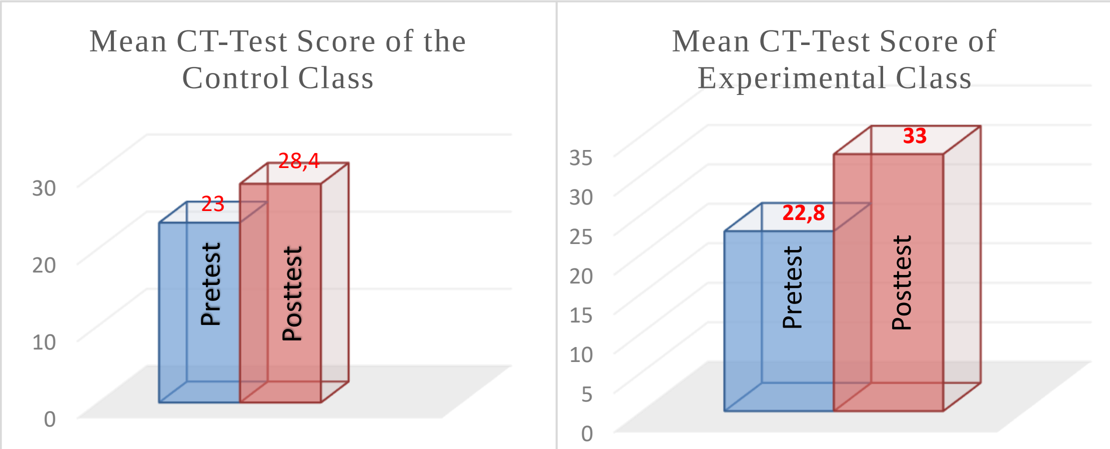

Figure 2. The Mean Scores of The Pretest and Posttest of Computational Thinking Ability in The Control Group and Experimental Group

Table 2. Boxplot Data of Participants' CT Scores

	Experimental Group Data (Pretest)	Experimental Group Data (Posttest)	Control Group Data (Pretest)	Control Group Data (Posttest)
Mean	22,8	33	23	28,4
Highest Scores	28	35	28	32
Lowest Score	20	28	20	23
Quartile 1	21	32	22	27
Quartile 3	24	34	24	31
Median	23	33	23	29
Standard Deviations	2,0	1,7	2,1	2,6

The results shown in Figure 2 and Table 2 show that in the experimental and control groups, there was an increase in the score obtained in the posttest compared to the pretest. In the acquisition of experimental group scores, the increase was more significant. The control group increased during the posttest because participants had the opportunity to explore during the pretest, so there was an increase during the posttest. The difference in the average score at the time of the pretest and posttest in the control group was 5,4 while in the experimental group it was 10,2. So based on the results of descriptive statistical analysis, treatment activities carried out through scratch training have a significant effect on the development of computational thinking skills. The standard deviations of the experimental group data at pretest and posttest were 2,0 and 1,7 respectively. Furthermore, the standard deviations of the control group data at pretest and posttest were

2,1 and 2,6 respectively. Because the standard deviation in the experimental group (pretest and posttest) and control group (pretest and posttest) is smaller than the average of each data set, it can be concluded that the level of data distribution obtained is good.

The computational thinking ability test given at the posttest contains CT-Test questions that can be constructed with the use of directions, loops, and conditionals that allow it to be used to measure computational thinking ability in detail. At the time of the posttest, there were more correct answers in the experimental group compared to the control group. The questions given at the posttest still refer to the questions on the pretest. Furthermore, the posttest questions were also based on the practicum activities in the training activities to improve the indicators of participants' computational thinking skills.

Test Statistical Assumptions

Furthermore, to determine the category of improvement in computational thinking skills, the categorization of N-Gain scores can be seen as follows:

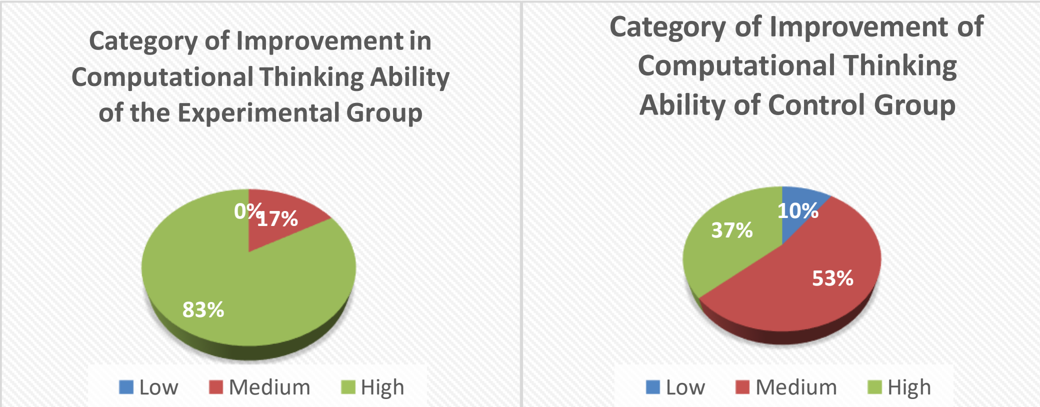

Figure 3. Categorization of the N-Gain Score

Based on Figure 3, it can be seen that in the experimental group there are only two categories of improvement in computational thinking skills, namely the high category (as much as 83%) and the medium category (17%). Meanwhile, in the control group, there are three categories for improving computational thinking skills: the high category (as much as 37%), the medium category (53%), and the low category (as much as 10%). Furthermore, to determine the diversity of data from the two groups, a homogeneity test will be carried out using the independent sample T-test with the help of SPSS statistical software. As shown in Table 3, the significance value for "Based on Mean is 0.111>0.05 (Significance Level). From these data, it can be concluded that the variance of the data results from computational thinking ability being homogeneous (Calfee & Piontkowski, 2016).

Table 3. Test of Homogeneity of Variance

		Levene Statistic	df1	df2	Sig.
Participants' computational thinking skills	Based on Mean	6.923	1	58	.111
	Based on Median	4.679	1	58	.035
	Based on Median and with adjusted df	4.679	1	53.527	.035
	Based on trimmed mean	6.760	1	58	.012

Furthermore, the normality test of the average computational thinking ability in both groups was carried out using the Kolmogorov-Smirnov test using SPSS with a significance level of 0.05 (Sukestiyarno, 2020). The following will be given in the Kolmogorov-Smirnov Test Table:

Table 4. Test of Homogeneity of Variance One Sample Kolmogorov-Smirnov Test and ANOVA

		Participants' computational thinking skills
N		60
Normal Parameters[a,b]	Mean	30.47
	Std. Deviation	3.050
Most Extreme Differences	Absolute	.186
	Positive	.090
	Negative	-.186
Test Statistic		.186
Asymp. Sig. (2-tailed)		.000[c]

Table 5. ANOVA

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	264.600	1	264.600	53.975	.000
Within Groups	284.333	58	4.902		
Total	548.933	59			

Based on Table 4, the asymp sig value is 0.000 <0.05 (significance level), so it can be concluded that the average test results of the participants' computational thinking skills are normally distributed. Furthermore, after the homogeneity test and normality test of the data on the results of the computational thinking ability test after scratch training, to show that there is a significant difference in the average results of the computational thinking ability test after training, one factor analysis of variance, or ANOVA, will be carried out. Based on Table 5, it can be seen that the sig. (significant) in the ANOVA table is 0.000<0.05 (5% significance level), which means that both groups, namely experimental and control, have statistically significant differences in computational thinking ability test results. Furthermore, Table 5 will give the results of the ANOVA test using SPSS.

Qualitative análisis

To support the results of quantitative analysis in this paper, qualitative analysis will be carried out through in-depth questionnaires related to several aspects related to the development of the experience of integrating computational thinking skills in the learning process using Scratch. This combination of analysis is the foundation of mixed research methods (Riba, 2005). This qualitative analysis is used to strengthen the quantitative analysis by analyzing and understanding the condition of the trainees by looking at various aspects and dimensions that are valid and reliable (Anguera et al., 1995). From the questionnaire results of the trainees, we can provide additional information based on the quantitative analysis results in this paper. Most of the results of the questionnaires and in-depth interviews indicated good acceptance of the training activities. This happens because they learn interesting things from this training activity that they have never learned before. Below we will list some of the comments collected from the questionnaire results:

"In this training, it can develop my critical thinking (Participant No. 21) and problem solving process in integrating mathematics subjects (especially in linear equation material)."

"In this activity, it helped me (Participant No. 13) explore the sports subject of soccer material by practicing computational thinking with scratch media."

Furthermore, information related to exploration was found from Participant No. 8, who teaches mathematics, who responded that:

"I found inspiration to make games in the process of learning math by using several tools on Scratch, including tools to move forward to point x from point y. From this game, learning can be built related to the learning process. From this game, learning can be built in relation to the distance at the absolute value of two points".

The development of technology is currently so fast that it is expected that teachers are able to explore the potential of existing resources to assist in injecting the ability to think computationally and integratively into the subjects they teach. In this training, many participants gave various responses regarding the level of difficulty in using Scratch in the process of integrating the subjects taught. The following will be some responses related to training activities:

"At first, it was difficult for me (Participant No. 4) to use the Scratch application because I did not know it and had never used it before. This seemed different after I was introduced to the training activities and practiced it by making projects. After participating in the practical activities for two times, I was finally able to use it and change my views from when I started before the training".

In addition, there are also participants who consider that this training activity is very important because it helps participants who also work as madrasah teachers (STEM and non-STEM teachers) in assisting the integration of computational thinking skills in subjects, namely:

"I (Participant No. 28) consider this training activity very important and fundamental because with this activity I have gained knowledge and technological resources that are suitable and easy to integrate computational thinking skills with the learning process of religion (non-STEM)".

There is the most basic question on the questionnaire related to the training process of integrating computational thinking in subjects using Scratch, namely, whether there is an impact related to the training activities of integrating computational thinking skills. Mention the impact of the activity. Most of the participants answered yes, there was an effect related to this training, and most felt the ease of using Scratch in the process of integrating computational thinking skills in the future learning process. Here are the results of the questionnaire:

"I (Participant No. 2) get enrichment material related to learning media in the subject of linear equations using ICT."

From the results of the previous explanation, the positive effects of the training activities on the use of Scratch to improve computational thinking skills through integration activities into subjects are clearly visible. In general, based on the results of the computational thinking test, computational thinking skills through the Scratch programming language have improved. This can be seen from the difference in CT-Test results on pretest and posttest after practicum activities using scratch in integrating into the subject.

In addition, this empirical study found that using technology assistance in the form of Scratch for STEM and non-STEM teachers can help teachers explore the concepts of their subjects by developing interesting learning media. With this Scratch training activity, teachers are able to integrate technology into their subjects; especially for non-STEM teachers, it is new and easy to accept. The teachers also feel helped by the training activities of using Scratch in integrating it into each lesson. The practical activities in using Scratch can make it easier for teachers to learn the tools in the Scratch program, so that it makes it easier for teachers of both STEM and non-STEM subjects in the integration process into their respective subjects.

This is in line with other research, which shows that training in the use of technology is beneficial to the learning process in schools, namely by including things that can improve computational thinking skills, it can improve teacher resources in the future (Vaca Cárdenas et al., 2016). According to McComas, in his research he has mentioned that teachers need effective competency development so that they can run the curriculum well (Reeves, 2005). In line with the research of Gabriele et al (Gabriele, 2019) conducted this training activity for new teachers with the aim of preparing them and developing 21st century competencies and skills so as to bring positive changes to students. In addition, training activities with more practice will make teachers more interested in development and able to explore innovative and explorative learning methodologies and (Duo-Terron et al., 2022; Moreno-Morilla et al., 2021).

CONCLUSION

From the results of the previous explanation, the positive effects of the training activities on the use of Scratch to improve computational thinking skills through integration activities into subjects are clearly visible. In general, based on the results of the computational thinking test, computational thinking skills through the Scratch programming language have improved. This can be seen from the difference in CT-Test results on pretest and posttest after practicum activities using scratch in integrating into the subject.

In addition, this empirical study found that using technology assistance in the form of Scratch for STEM and non-STEM teachers can help teachers explore the concepts of their subjects by developing interesting learning media. With this Scratch training activity, teachers are able to integrate technology into their subjects; especially for non-STEM teachers, it is new and easy to accept. The teachers also feel helped by the training activities of using Scratch in integrating it into each lesson. The practical activities in using Scratch can make it easier for teachers to learn the tools in the Scratch program, so that it makes it easier for teachers of both STEM and non-STEM subjects in the integration process into their respective subjects.

This is in line with other research, which shows that training in the use of technology is beneficial to the learning process in schools, namely by including things that can improve computational thinking skills, it can improve teacher resources in the future (Vaca Cárdenas et al., 2016). According to McComas, in his research he has mentioned that teachers need effective competency development so that they can run the curriculum well (Reeves, 2005). In line with the research of Gabriele et al (Gabriele, 2019) conducted this training activity for new teachers with the aim of preparing them and developing 21st century competencies and skills so as to bring positive changes to students. In addition, training activities with more practice will make teachers more interested in development and able to explore innovative and explorative learning methodologies and (Duo-Terron et al., 2022; Moreno-Morilla et al., 2021).

REFERENCES

Aho, A. V. (2011). Computation and computational thinking. *Computer Journal*, *55*(7), 833–835. https://doi.org/10.1093/comjnl/bxs074

Aminah, N. (2022). A Teaching Practice Design Based on a Computational Thinking Approach for Prospective Math Teachers Using Ed-Tech Apps. *International Journal of Interactive Mobile Technologies*, *16*(14), 43–62. https://doi.org/10.3991/ijim.v16i14.30463

Anguera, M. T., Arnau, J., Ato, M., Martínez, R., Pascual, J., & Vallejo, G. (1995). Métodos de investigación en psicología. *Madrid: Síntesis*.

Brennan & Resnick. (2012). New frameworks for studying and assessing the development of computational thinking. *Studies in Computational Intelligence*, *727*, 135–160. https://doi.org/10.1007/978-3-319-64051-8_9

Calfee, R., & Piontkowski, D. (2016). Design and analysis of experiments. In *Handbook of Reading Research*. https://doi.org/10.2307/2983009

Chan, S., Looi, C., & Sumintono, B. (2020). Assessing Computational Thinking Abilities Among Singapore Secondary Students: A Rasch Model Measurement Analysis. *Journal of Computers in Education*, *8*(2), 213–236.

Chiazzese, G. (2019). Educational robotics in primary school: Measuring the development of computational thinking skills with the bebras tasks. *Informatics*, *6*(4). https://doi.org/10.3390/informatics6040043

Dagiene, V., & Futschek, G. (2008). Bebras international contest on informatics and computer literacy: Criteria for good tasks. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *5090 LNCS*, 19–30. https://doi.org/10.1007/978-3-540-69924-8_2

Dagienė, V., & Sentence, S. (2016). It’s computational thinking! bebras tasks in the curriculum. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *9973 LNCS*(Bebras), 28–39. https://doi.org/10.1007/978-3-319-46747-4_3

Del Álamo Venegas, J. J., Alonso Díaz, L., Yuste Tosina, R., & López Ramos, V. (2021). La Dimensión Educativa De La Robótica: Del Desarrollo Del Pensamiento Al Pensamiento Computacional En El Aula. *Campo Abierto*, *40*(2), 221–233. https://doi.org/10.17398/0213-9529.40.2.221

Disessa, A. (2000). *Changing minds computer learning and litrecy*.

Duo-Terron, P., Hinojo-Lucena, F. J., Moreno-Guerrero, A. J., & López-Núñez, J. A. (2022). STEAM in Primary Education. Impact on Linguistic and Mathematical Competences in a Disadvantaged Context. *Frontiers in Education*, *7*(June), 1–14. https://doi.org/10.3389/feduc.2022.792656

Gabriele, L. (2019). Lesson planning by computational thinking skills in Italian pre-service teachers. *Informatics in Education*, *18*(1), 69–104. https://doi.org/10.15388/infedu.2019.04

Gerbing, D. W. (1984). Campbell and Stanley for Undergraduates. In *Contemporary Psychology: A Journal of Reviews* (Vol. 29, Issue 4). https://doi.org/10.1037/022808

Indonesian Ministry of Education and Culture. (2021). Decree of the Head of the Research, Development and Bookkeeping Agency Number 028/H/Ku/2021. In *Decree of the Head of the Research and Development and Bookkeeping Agency* (pp. 1–822).

Johnson, R. B., & Onwuegbuzie, A. J. (2007). Toward a Definition of Mixed Methods Research. *Journal of Mixed Methods Research*, *1*(2), 112–133. https://doi.org/10.1177/1558689806298224

Marcia Linn, Alfred V. Aho, E. (2011). Report of a Workshop of Pedagogical Aspects of Computational Thinking. In *The National Academies Press*.

Marom, S., Waluya, S. B., Mariani, S., & Susilo, B. E. (2023). Computational Thinking Patterns In The Mathematical Modeling Process : Self-Efficacy Pre- Service Mathematics Teacher. *The Seybold Report*, *18*(03), 1551–1568. https://doi.org/10.17605/OSF.IO/DVAXT

Molina-Ayuso, Á. (2022). Introduction to Computational Thinking with Scratch for Teacher Training for Spanish Primary School Teachers in Mathematics. *Education Sciences*, *12*(12). https://doi.org/10.3390/educsci12120899

Moreno-Morilla, C., Guzmán-Simón, F., & García-Jiménez, E. (2021). Digital and information literacy inside and outside Spanish primary education schools. *Learning, Culture and Social Interaction*, *28*(September 2020), 0–21. https://doi.org/10.1016/j.lcsi.2020.100455

Moreno, J., Robles, G., Román, M., & Rodríguez, J. D. (2019). No es lo mismo: un análisis de red de texto sobre definiciones de pensamiento computacional para estudiar su relación con la programación informática. *Revista Interuniversitaria de Investigación En Tecnología Educativa*, 26–35. https://doi.org/10.6018/riite.397151

National Research Council. (2012). *A framework for K-12 science education: Practices, crosscutting concepts, and core ideas*.

Nugent, G., Barker, B., Grandgenett, N., & Adamchuk, V. (2009). The use of digital manipulatives in K-12: Robotics, GPS/GIS and programming. *Proceedings - Frontiers in Education Conference, FIE, November*. https://doi.org/10.1109/FIE.2009.5350828

Oi-Lam, N., & Zhihao, C. (2021). Examining primary students' mathematical problem-solving in a programming context: towards computationally enhanced mathematics education. *ZDM - Mathematics Education*, *53*(4), 847–860. https://doi.org/10.1007/s11858-020-01200-7

Pérez-Marín, D. (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

Reeves, C. (2005). The Language of Science. In *The Language of Science*. https://doi.org/10.4324/97802035971255

Reichert, J. T. (2020). Computational thinking in K-12: An analysis with mathematics teachers. *Eurasia Journal of Mathematics, Science and Technology Education*, *16*(6). https://doi.org/10.29333/EJMSTE/7832

Riba, C. (2005). Flick, U.(2004) Introducción a la investigación cualitativa. Madrid: Morata. *Anuario de Psicología/The UB Journal of Psychology*, 127–129.

Rodríguez-Martínez, J. A. (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

Román-González, M. (2015). Computational Thinking Test: Design Guidelines and Content Validation. *Proceedings of the 7th Annual International Conference on Education and New Learning Technologies (EDULEARN 2015), January*, 2436–2444. https://doi.org/10.13140/RG.2.1.4203.4329

Román-González, M. (2016). *Codigoalfabetización y pensamiento computacional en educación primaria y secundaria: validación de un instrumento y evaluación de programas [Code-literacy and Computational Thinking in Primary and Secondary Education:...]*. 720. http://e-spacio.uned.es/fez/view/tesisuned:Educacion-Mroman

Romero, M., Lepage, A., & Lille, B. (2017). Computational thinking development through creative programming in higher education. *International Journal of Educational Technology in Higher Education*, *14*(1). https://doi.org/10.1186/s41239-017-0080-z

Soboleva, E. V. (2021). Formation of Computational Thinking Skills Using Computer Games in Teaching Mathematics. *Eurasia Journal of Mathematics, Science and Technology Education*, *17*(10), 1–16. https://doi.org/10.29333/ejmste/11177

Sukestiyarno, Y. L. (2020). Metode penelitian pendidikan. *Educational Research Methods. UNNES Press*.

Tabesh, Y. (2017). Computational thinking: A 21st century skill. *Olympiads in Informatics*, *11*(Special Issue), 65–70. https://doi.org/10.15388/ioi.2017.special.10

Uygun, T. (2022). Preservice Middle School Mathematics Teachers' Development of Flexibility and Strategy Use by Geometric Thinking in Dynamic Geometry Environments. *Participatory Educational Research*, *9*(5), 183–203.

https://doi.org/10.17275/per.22.110.9.5

Vaca Cárdenas, L. A., Tavernise, A., Bertacchini, F., Gabriele, L., Valenti, A., Pantano, P., & Bilotta, E. (2016). An Educational Coding Laboratory for Elementary Pre-service Teachers: A Qualitative Approach. *International Journal of Engineering Pedagogy (IJEP)*, *6*(1), 11. https://doi.org/10.3991/ijep.v6i1.5364

Wing, J. (2006). Computational thinking. In *Communications of the ACM* (Vol. 49, Issue 3, pp. 33–35). https://doi.org/10.1145/1118178.1118215

Wing, J. M. (2015). *Computational Thinking It represents a universally applicable attitude and skill set everyone, not just computer scientists, would be eager to learn and use. April*.