STUDENT COMPUTATIONAL THINKING IN SOLVING SEQUENCE PROBLEMS BASED ON LEARNING STYLES

Miswanto [1], Toto Nusantara [1], Abdur Rahman As'ari [1], Sukoriyanto [1]
[1] Universitas Negeri Malang, Indonesia
**Corresponding Address*: *toto.nusantara.fmipa@um.ac.id*

Received: 12 March 2022 | Approved: 10 May 2022 | Published: 29 June 2022

Abstract: The purpose of this study is to describe students' computational thinking in solving sequence problems based on learning styles. This research approach is qualitative research with exploratory and descriptive research types. The subjects in this study were students majoring in Mathematics Tadris, Fakultas Tarbiyah dan Ilmu Keguruan UIN SATU Tulungagung semester 5. First, students' computational thinking in solving sequence problems based on assimilation and convergent learning styles, including incoherent, complete, and systematic. Second, students' computational thinking in solving sequence problems based on assimilation and accommodation learning styles, including coherent, complete, and systematic. Third, students' computational thinking in solving sequence problems is based on convergent learning styles and accommodation, including coherent, complete, and unsystematic. Fourth, students' computational thinking in solving sequence problems is based on divergent and convergent learning styles, including incoherent, incomplete, and unsystematic.

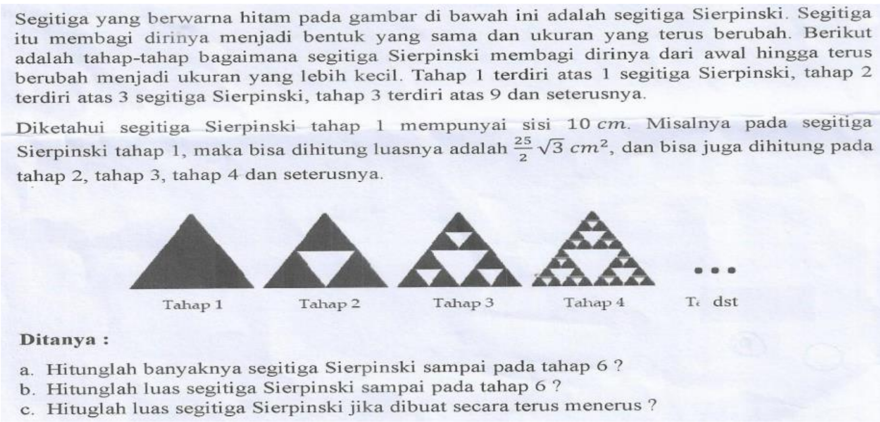
Keywords: Computational Thinking, Problem Solving, Kolb Learning Style

INTRODUCTION

One way to increase cognitive development and learning motivation is to apply computational thinking because improving cognitive development requires direct experience to encourage problem-solving skills (Sung et al., 2017). The main motivation for introducing the practice of computational thinking into mathematics classrooms is the response to increasingly computerized disciplines because these skills can be applied in the world of work later (Gadanidis, 2016; Maharani et al., 2021). In addition, mathematical ability is considered a core factor that predicts students' ability to learn (Grover & Pea, 2013). Some researchers put forward convincing arguments that mathematical thinking plays an important role in computational thinking (Gadanidis, 2017; Rambally, 2017; Son & Lee, 2016) because solving mathematical problems is a construction process (Benakli et al., 2017; Junsay, 2016; Lockwood & Asay, 2016). This construction process requires an analytical perspective to solve problems that are unique and fundamental for students in computational thinking.

Computational thinking is a means to develop students' problem-solving skills and creativity that is integrated with technological developments. Computational thinking is considered a fundamental skill for today's times (Gadanidis, 2017; Maharani et al., 2019). In line with the idea conveyed by the Minister of Education and Culture Nadiem Makarim at the International Symposium in 2019 that 6C competencies, including creativity, collaboration, communication, critical thinking, computational thinking, and compassion, need to be formed in students (Safitri, 2022).

But in fact, the learning applied by the teacher narrows the student space to develop computational thinking skills (Gadanidis et al., 2017; Weintrop et al., 2016). This is in line with the opinion of Tedre & Denning (2016) that the cause of students' computational thinking skills not developing is a lack of innovation in learning. Lecturers emphasize students memorize the procedures used to solve mathematical problems, causing students' computational thinking skills to be low (Angeli & Giannakos, 2020; Gadanidis et al., 2017; Maharani et al., 2020). Another reality is that learning mathematics runs monotonously. This includes teachers presenting material, giving examples, practicing questions, and checking students' answers. As a result, it makes students less interested in developing computational thinking skills, which has an impact on students' computational thinking skills, as a result in which students' computational thinking skills become low (Marchelin et al., 2022; Supiarmo et al., 2021). This is also based on the trial of research questions conducted by researchers by providing tests in the form of 1 computational thinking problem, as in Figure 1 below.



Segitiga yang berwarna hitam pada gambar di bawah ini adalah segitiga Sierpinski. Segitiga itu membagi dirinya menjadi bentuk yang sama dan ukuran yang terus berubah. Berikut adalah tahap-tahap bagaimana segitiga Sierpinski membagi dirinya dari awal hingga terus berubah menjadi ukuran yang lebih kecil. Tahap 1 terdiri atas 1 segitiga Sierpinski, tahap 2 terdiri atas 3 segitiga Sierpinski, tahap 3 terdiri atas 9 dan seterusnya.

Diketahui segitiga Sierpinski tahap 1 mempunyai sisi $10\ cm$. Misalnya pada segitiga Sierpinski tahap 1, maka bisa dihitung luasnya adalah $\frac{25}{2}\sqrt{3}\ cm^2$, dan bisa juga dihitung pada tahap 2, tahap 3, tahap 4 dan seterusnya.

Tahap 1 Tahap 2 Tahap 3 Tahap 4 T. dst

Ditanya :

a. Hitunglah banyaknya segitiga Sierpinski sampai pada tahap 6 ?
b. Hitunglah luas segitiga Sierpinski sampai pada tahap 6 ?
c. Hituglah luas segitiga Sierpinski jika dibuat secara terus menerus ?

Figure 1. Test Question

The test data of research questions, namely test answers, *think aloud* results, and semi-structured interview results of subjects, are seen based on indicators of computational thinking processes, including decomposition, generalization, abstraction, and algorithmic thinking. Students answered one of the test questions, as shown in Figure 2 below.

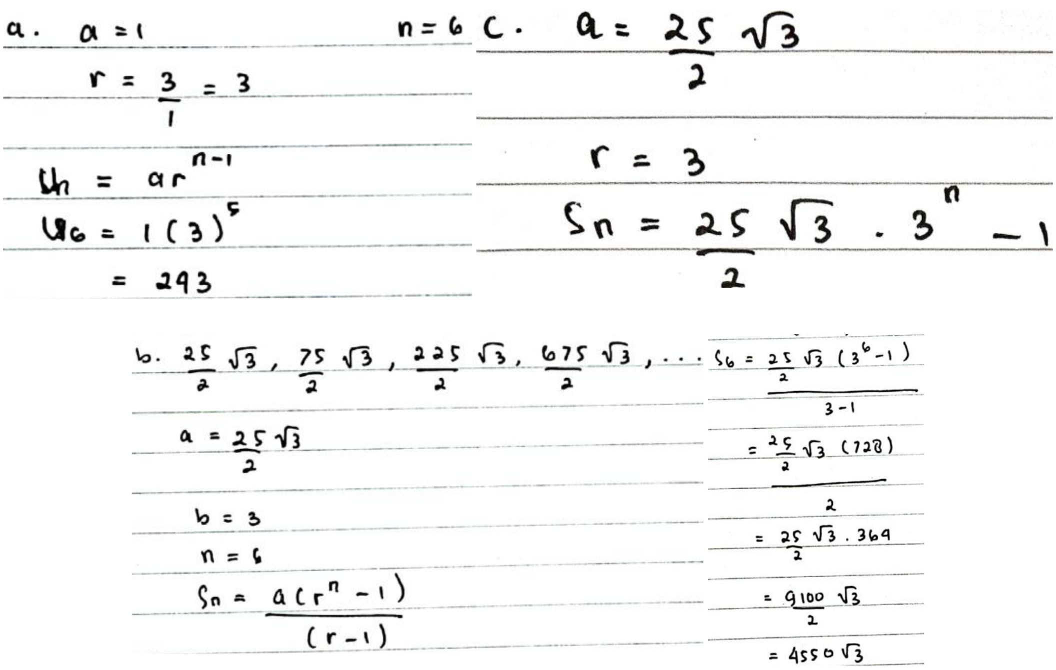


Figure 2. Trial Question Answers

Based on the trial of research test questions, computational thinking is a process that plays a role in formulating problems and their solutions so that the obtained solutions can be represented. It can be said that computational thinking has five skills of computational thinking stages: decomposition, generalization, abstraction, algorithms, and debugging. Through these five stages of computational thinking, students are trained to formulate problems by separating them into small parts to solve them easily (Angeli & Giannakos, 2020). This strategy hones procedural thinking skills that make it easier for students to understand mathematical problems and train students to think creatively (Supiarmo et al., 2021).

Computational thinking skills can help students solve mathematical problems by simplifying complex problems into several aspects, making it easier to understand and solve given problems. Mathematical problems will be easier if students are accustomed to thinking computationally (Angeli & Giannakos, 2020).

Based on the problems and facts above, it is necessary to examine how computational thinking students solve row problems based on learning styles. This is based on the results of observations and interviews of researchers when conducting a research trial on May 22, 2023, on several subjects about computational thinking in solving problems in the review of Kolb's learning style. The results showed that students' computational thinking skills in each learning style had diversity.

The conclusion is that the results of observations and trials conducted on computational thinking with the five indicators above are still low and need to be improved. Therefore, there is a need for computational thinking skills exercises for students based on the knowledge they have. This is because students' thinking processes and initial knowledge are combined in solving mathematical problems so that they can solve problems and think about and arrange problem-solving according to the stages of computational thinking.

According to Kolb (2005), learning style is a person's tendency to receive and process information influenced by the created environment. According to Kolb (2005), the student learning process is influenced by four abilities: concrete experience, reflection observation, abstract conceptualization, and active experimentation (Ramlah, 2014). Most individuals have a combination of 2 of the four abilities described by Kolb. According to Kolb (2005), combining these two abilities forms four learning styles: *convergent, assimilator, divergent,* and *accommodated* (Barra, 2012).

Learning style shows how an individual processes information to learn and apply it. Kolb (2005) uses the term learning style as a whole of three domains, namely the process of cognition and affection for material, mental learning models, and learning orientation. Tedre & Denning (2016) explained that learning orientation is the entire domain containing goals, intentions, motives, expectations, attitudes, and interests about individuals towards the learning process. Learning styles are consistent patterns of behavior to construct knowledge that blends with students' concrete or real-life experiences. According to Bold (2005), the four learning styles are divergent, assimilation, convergent, and accommodation.

Previous research from Syaeful (2016) stated an increase in students' computational thinking skills using interactive multimedia based on Quantum Teaching and Learning with an average upper group of 0.51, middle group of 0.51, and lower group of 0.52. So, it can be concluded that using interactive multimedia learning based on quantum teaching and learning can improve computational thinking skills using interactive multimedia based on quantum teaching and learning models. However, the computational thinking process of students in solving a problem has not been discussed further.

The fact is that computational thinking not only helps acquire computer-related skills but also improves the ability to solve problems in other domains (Schere et al., 2019; Denning., 2016). Several studies have found that computational thinking is beneficial for improving mathematical skills (Kazakoff et al., 2013), reasoning abilities (Boom et al., 2018; Marinus et al., 2018), and creative thinking (DeSchryver & Yadav, 2015; Israel-Fishelson et al., 2021).

The relationship between computational thinking and mathematics has been developed by Papert (1996), who proposed a constructionist framework that computer programming can help realize abstract concepts and manipulations involved in mathematical problems (Papert, 1980). This argument has been supported by numerous empirical studies (Clements et al., 2001; Kazakoff et al., 2013). For example, Clements (2001) found that elementary school students can learn geometric concepts through the participation of computational thinking activities. Kazakoff (2013) uses the CHERP programming language to engage children in computer programming activities and finds learning experiences that significantly improve sequencing skills.

Computational thinking is also related to creative thinking because computational is a creative activity that allows humans to change from consumers of technology to producers (Nagai et al., 2019). Specifically, creative thinking is defined as solving a problem or producing a product in a new, unusual, and useful way (Plucker et al., 2004; Runci & Jaeger, 2012). Recent studies have revealed that creating with other students is one of the main strategies children use in self-directed programming projects (Papert, 1996). Moreover, the importance of computational thinking is realized as fostering the generation of creative behavior in computational thinking. Recently, a positive relationship between computational thinking and creative thinking has been found (DeSchryver & Yadav, 2015; Israel-Fishelson et al., 2021). For example, using computational thinking in classroom activities can improve graduate students' creative thinking (DeSchryver & Yadav, 2015).

Christi (2023), in the journal on education entitled "The importance of computational thinking in mathematics learning," found that computational thinking can increase student creativity in solving mathematical problems. Putra (2022), in the journal Dirasah, titled "Computational Thinking of

Students in Solving Problems Related to Social Arithmetics in Terms of Gender," also states that computational thinking can improve students' understanding of solving problems by reviewing gender.

From the various research results on computational thinking above, no research has reviewed or discussed the description of students' computational thinking characteristics in solving row problems based on learning styles. In this case, the strategy or steps taken is a computational thinking process in solving problems based on Kolb's learning style. So, this study aims to describe students' computational thinking in solving row problems based on learning styles.

METHODS

Types of Research

This research approach is qualitative research with exploratory, descriptive research types. The learning style used in this study is Kolb's, namely divergent learning styles, assimilation, convergence, and accommodation in solving row problems using computational thinking. The research data used was answers to solving algebraic problems with the think-loud process and transcription of think-aloud results.

Research Subjects

The subjects in this study were students majoring in Mathematics tadris, Faculty of Tarbiyah, and Teacher Training UIN SATU Tulungagung semester 5. The choice is because the final year students indicated a much deeper thinking process in solving mathematical problems and have gained more experience in the process of solving mathematical problems. Research subjects are not randomly selected but are selected by considering the computational thinking process and communication so that the computational thinking process can be disclosed well.

The subjects used are students who can solve mathematical problems (research instruments). Sixty students were given the test. Of the 60 students, only a few students met the criteria to be research subjects. We can see the computational thinking process in some of these subjects. The stages of computational thinking will be analyzed based on four learning styles according to Kolb: divergent, assimilation, convergent, and accommodation. The stages correspond to Figure 3 below.

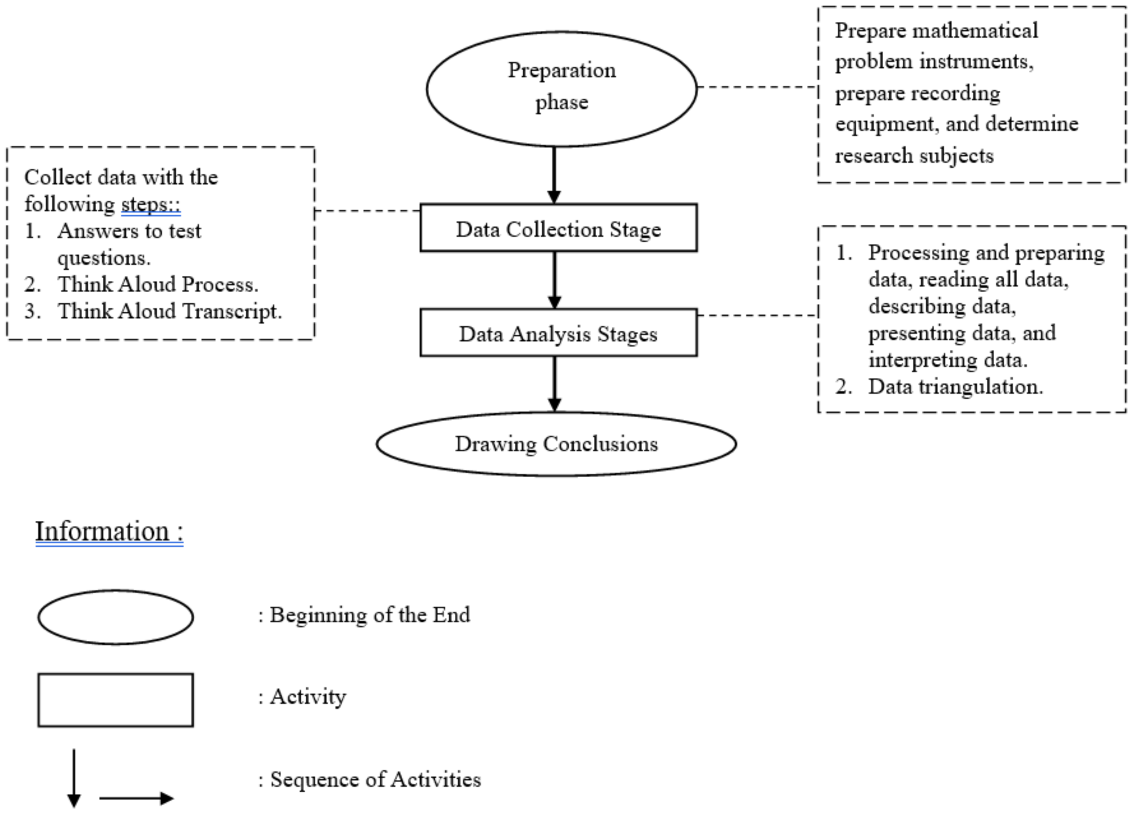


Figure 3. Research Procedure

RESULTS AND DISCUSSIONS

This study describes the computational thinking of students majoring in mathematics tadris FTIK UIN SATU Tulungagung in solving sequence problems based on learning style. This section presents subject data in solving mathematical problems that produce computational thinking processes. These subjects carry out decomposition, abstraction, generalization, algorithmic, and debugging stages. The next one saw the emergence of 4 learning styles: divergent, assimilation, convergent, and accommodation.

The respondents with assimilation and convergent learning styles were 28 students with computational thinking based on assimilation and convergent type 1 learning styles, as many as seven students, students with computational thinking based on assimilation and convergent type 2 learning styles as many as seven students, students with computational thinking based on assimilation learning styles and convergent type 3 as many as five students, computational thinking based on assimilation learning styles and convergent type 4 as many as one students, and computational thinking based on assimilation and convergent learning styles type 5 as many as eight students. Respondents with assimilation and accommodation learning styles were 20 students with computational thinking based on assimilation learning styles and type 1 accommodation, as many as 12 students, and computational thinking based on assimilation learning styles and type 5 accommodation, as many as eight students. Respondents think computationally based on convergent learning styles and accommodation of as many as seven students, and respondents think computationally based on divergent and convergent learning styles of as many as five students.

Computational Thinking of Students in Solving Problems Based on Assimilation and Convergent Learning Styles

The subject of understanding the problem by reading and recognizing known information, namely the Sierpinski triangle stage 1, has a side length of 10 $cm,$ the computational thinking stage of decomposition. In addition, subjects understand the images represented in their mathematical ideas by doodling them with written numbers, an algorithmic stage of computational thinking. In writing, the subject makes a strategy to divide the Sierpinic triangle stage 2 into four parts where each part of the triangle is from the previous triangle, $\frac{1}{4}$ so that the area of each stage of the Sierpinski triangle is the computational thinking stage of generalization (Aho, 2012).

Subjects in planning strategies when digging Sierpinski triangle information up to stage 6 so that they get a line starting from the first term, second term, third term, fourth term, fifth term, and sixth term, which is the computational thinking stage of decomposition. After that, identify the sequence that occurs so that the ratio can be obtained and can be used to answer the number of Sierpinski triangles up to stage 6 with the geometric row formula, which is computational thinking of the stages of decomposition. Furthermore, the subjects in calculating the area of the Sierpinski triangle up to stage 6 are by calculating the area of the Sierpinski triangle stage 1, the area of the Sierpinski triangle stage 2, the area of the Sierpinski triangle stage 3, the area of the Sierpinski triangle stage 4, the area of the Sierpinski triangle stage 5, and the area of the Sierpinski triangle stage 6 which is computational thinking stages of decomposition. This is because decomposition allows cognitively divided complex problems into smaller ones to propose appropriate solutions (Council, 2011).

Subjects are solving strategies with mathematical calculations based on geometric row patterns and ratios obtained to obtain the number of Sierpinski triangles up to stage 6 using geometric row formulas, which are computational thinking stages of decomposition. Subjects are solving strategies with mathematical calculations based on geometric row patterns and ratios obtained to obtain the area of the Sierpinski triangle up to stage 6 using geometric sequence formulas, which are computational thinking stages of decomposition and generalization. In addition, subjects carry out solving strategies with mathematical calculations based on geometric row patterns and ratios obtained to obtain the area of a Sierpinski triangle if it is made continuously using geometric sequence formulas, which are computational thinking abstract stages and generalizations. This is due to abstraction and generalization‖, often used together because abstracts are generalized through parameterization to provide greater utility (Denning, 2016).

Making conclusions and rechecking by mentioning the Sierpinski triangle in stage 6 is the fruit of the Sierpinski triangle, which is the computational thinking stage of decomposition. In addition, the subject also mentions the $S_6 = \frac{3^6-1}{2} = 364$ area of Sierpinski's triangle up to stage 6 is $\frac{186575\sqrt{3}}{2048}\ cm^2$ which is computational thinking decomposition. In addition, if made continuously, the subject makes inferences about the area of the Sierpinski triangle, which is the computational thinking stage of debugging and generalization. $50\sqrt{3}\,cm^2$ If solution implementation, automation operations, modeling, and simulation can be performed, then it can be assessed, tested, debugged, and generalized for various computational problems (Kallia, 2021).

Computational Thinking of Students in Solving Problems Based on Learning Styles of Assimilation and Accommodation

Subjects understand the problem by matching the image with the information and interpreting it as scribbles in the image by counting the number of Sierpinski triangles until stage 6, the computational thinking stage of decomposition. I also calculated the area of the Sierpinski triangle using a triangle in stage 2 divided into four equal parts so that the area is from the Sierpinski triangle in stage 1. In contrast, stage 3 is divided into triangle stage 1, an abstract computational thinking stage. Furthermore, by $\frac{3}{4}\frac{9}{16}$dividing the triangle stage *2* into *four* parts, each *1* part of the triangle $\frac{1}{4}$ is from the previous triangle and up to the Sierpinski triangle, the *ninth* stage, computational thinking, and the algorithmic stages. Because thinking algorithmically produces algorithms that are clear, unambiguous, and replicable (Kalelioğlu et al., 2016).

The subjects plan a strategy by identifying the image of a Sierpinski triangle so that it can be the area of the Sierpinski triangle stage 1, stage 2, stage 3, and so on to produce geometric sequences. It is a computational thinking stage of decomposition. After that, the series can be a ratio based on the series of the number of Sierpinski triangles up to the n-stage with the geometric sequence formula at the time of the decomposition stage. The subjects also found a geometric sequence pattern of the area of the Sierpinski triangle stage 1, stage 2, stage 3, and so on. They obtained the ratio, the computational thinking stage of decomposition, because decomposition is a skill in solving complex problems into simpler problems (NCTM, 2010).

The subjects carried out the strategy by writing the formula of the 6th term geometric sequence in calculating the number of Sierpinski triangles up to stage 6 based on the elements obtained, namely the first term and the ratio to calculate S_6 which is the computational thinking stage of decomposition. Computational thinking stages. The debugging stage occurs when calculating the area of the Sierpinski triangle. The subject appears to have a writing error and is crossed out on the wrong writing. Subjects in carrying out strategies based on geometric row patterns and ratios obtained to obtain the area of the Sierpinski triangle up to stage 6 using geometric row formulas, which are computational thinking stages of decomposition. In addition, subjects in carrying out solving strategies with mathematical calculations based on geometric row patterns and ratios obtained to obtain the area of the Sierpinski triangle if made continuously by not relating to previous terms, only taking the values *of a* and r and using geometric sequence formulas which are computational thinking abstract stages. This shows that the subject mastered the initial row and row material concept. Because abstraction reduces complexity by hiding irrelevant details (Denning, 2016).

The subjects made conclusions and rechecked by writing down the number of Sierpinski triangles to stage 6, as many as 364 triangles, which are computational thinking stages of decomposition. In addition, the subjects also wrote the area of Sierpinski's triangle up to stage 6 is the $\frac{186575\sqrt{3}}{2048}\ cm^2$ which is the computational thinking stage of generalization. Furthermore, the subject also wrote the area of the Sierpinski triangle if made continuously, which is computational thinking with stages of decomposition. $50\sqrt{3}\ cm^2$. This ability is closely linked to decomposition, which allows him cognitively to divide complex problems into smaller ones to generate suitable solutions (Council, 2011).

Computational Thinking of Students in Solving Problems Based on Convergent Learning Styles and Accommodation

Subjects understand problems by interpreting images in the form of scribbles so that they can visualize mathematical problems into mathematical models. There are many triangles at each stage of the Sierpinski triangle, which is the computational thinking stage of abstraction. The subjects in understanding the problem of the area of the triangle at each stage of the Sierpinski triangle, as well as the ratio of each row based on the information that the length of the side of the Sierpinski triangle sta is 10 *cm* and the area is $\frac{25}{2}\sqrt{3}\ cm^2$ what constitutes the computational thinking algorithmic stage. Algorithmic is a problem-solving skill associated with compiling a step-by-step solution to each problem and differs from coding (Selby, 2014).

Subjects in planning strategies can find a geometric sequence pattern based on the number of Sierpinski triangles from stage 1 to stage 4, the computational thinking stage generalization. After that, identify the sequence that occurs so that it can be adjusted based on the series of Sierpinski triangles up to stage 4 with the geometric sequence formula, the computational thinking stage of decomposition. The subjects also planned a strategy for calculating the area of the Sierpinski triangle from stage 1 to stage 6 to obtain geometric row patterns and ratio values, which are computational thinking stages of generalization. After that, identify the sequence that occurs so that it can be a ratio based on the broad sequence of the Sierpinski triangle stages 1 to 6 with the geometric sequence formula, which is the computational thinking stage of decomposition. Because the decomposition and generalization of information structures is the application of computational thinking, information structures are formed as abstract type data by creating new structures, adding information, checking whether there is information, looking for information, changing information, and finally deleting information (Nusantara, 2020).

The subjects in carrying out the strategy by paying attention to the geometric row pattern of the number of Sierpinski triangles from stage 1 to stage 4, the ratio value, and the row formula used are computational thinking stages of decomposition. Subjects in solving strategies with mathematical calculations based on geometric row patterns and ratios obtained to obtain the area of the Sierpinski triangle up to stage 6 using geometric row formulas, which are computational thinking stages of decomposition. In addition, subjects carry out solving strategies with mathematical calculations based on geometric row patterns and ratios obtained to obtain the area of a Sierpinski triangle if it is made continuously using geometric row formulas, which are computational thinking algorithmic and abstract stages. This is because abstraction is a skill that removes characteristics or attributes from an object or entity to reduce it to a set of characteristics (Wing, 2011).

The subjects in making conclusions and checking back by writing down the Sierpinski triangle to stage 6 are $S_6 = \frac{3^6-1}{2} = 364$ Sierpinski's triangle is a computational thinking stage of decomposition. In addition, the subjects also wrote the area of the Sierpinski triangle up to stage 6 is $\frac{186575\sqrt{3}}{2048}\ cm^2$ which is the computational thinking stage of decomposition. Furthermore, the subject in answering the area of the Sierpinski triangle if it is made continuously is $50\sqrt{3}\ cm^2$ which is the computational thinking stage of debugging. This is because debugging is a skill to recognize when actions are not by instructions and a skill to correct errors (Selby, 2014).

Computational Thinking of Students in Solving Problems Based on Divergent and Convergent Learning Styles

The subject of understanding the problem by reading and recognizing information continues to be entered into the logic of thinking so that the Sierpinski triangle stage 1 has a side length and $ten\ cm$ area is $\frac{25}{2}\sqrt{3}\ cm^2$ which is the computational thinking stage of decomposition. The subject understands the problem by representing his mathematical idea by crossing out the picture by writing the first term as 1, the second term as 3, the third term as 9, the fourth term as 27, and so on so that the ratio is obtained which is computational thinking algorithmic stage. Yadav (2013) argues that human creativity can be augmented by computational thinking, specifically by using automation and algorithmic thinking.

Subjects in planning strategies by exploring existing information, what is asked in the problem, and what elements can be used to answer the problem, such as many Sierpinski triangles stage 1, many Sierpinski triangles stage 2, many Sierpinski triangles stage 3, many Sierpinski triangles stage 4, and so on to find a geometric row pattern which is computational thinking stages of decomposition. The subjects in planning the strategy calculate the area of the Sierpinski triangle up to stage 6 by calculating the area of the Sierpinski triangle stage 1, the area of the Sierpinski triangle stage 2, the area of the Sierpinski triangle stage 3, the area of the Sierpinski triangle stage 4, the area of the Sierpinski triangle stage 5, and the area of the Sierpinski triangle stage 6 to produce geometric rows and write the geometric sequence formula in solving the problem but wrong in the process which is computational thinking stages generalization because generalization reduces complexity by replacing multiple entities that perform similar functions with a single construction (Denning, 2016).

The subjects carry out the strategy by identifying the sequence with the first term *being a* and the ratio being r so that many Sierpinski triangles can be applied to the geometric sequence formula, the computational thinking stages of decomposition. The subject, in carrying out the strategy, experienced an error in the work of calculating the number of Sierpinski triangles up to stage 6, where the error was caused by incorrectly determining the ratio of the existing geometric rows because the geometric rows obtained were also inaccurate which is computational thinking of the decomposition stage. The subjects of implementing the strategy also realized that there was an error in answering, especially when writing down the steps when calculating the area of the Sierpinski triangle up to stage 6, which is the computational thinking stage of debugging. As well as applying the geometric sequence formula in calculating the area of Sierpinski's triangle to stage 6, the answer is incorrect: the computational thinking generalization stage. In addition, the subject could not answer the area of a Sierpinski triangle if it was created continuously and without mentioning the use of formulas or mathematical calculations in solving problems that are computational thinking stages of decomposition due to the decomposition of the problem to identify the problem or generate alternative solutions (Tedre, 2016).

The subject, in making conclusions and rechecking by mentioning the number of Sierpinski triangles up to stage 6, is the fruit of Sierpinski's triangle, which is the computational thinking stage of generalization. In addition, the subjects concluded by writing down 364 the area of the Sierpinski triangle until stage 6 is $42{,}6\ cm^2$ which is the computational thinking stage of generalization. The subject did not conclude the area of a Sierpinski triangle if made continuously because it could not answer, which is the computational thinking stage of decomposition. Karen thinks computational can be used to find the best way of solving problems, and then more creative solutions can be found to overcome problems or difficulties (Brennan and Resnick, 2012). This is also supported by a quote from the *International Society for Technology in Education* (ISTE), which explains that computational thinking is a problem-solving process that involves identifying patterns, making abstractions, developing algorithms, and formulating procedures to find solutions (Sukoriyanto, 2020).

CONCLUSION

Based on the results of research and discussion of computational thinking of students in solving problems based on learning styles, there are four characteristics of computational thinking of students in solving problems based on learning styles, namely ***First***, computational thinking of students in solving row problems based on assimilation and convergent learning styles in solving subject problems for The steps to understand problems in the stages of computational thinking include the stages of decomposition, the algorithmic stages and the stages of generalization. The steps of planning strategies in the stages of computational thinking include the stages of decomposition. The steps to implement strategies in the stages of computational thinking include the stages of decomposition, generalization, abstraction, and generalization. The steps to conclude and re-examine the computational thinking stages include the decomposition and debugging stages, where the stages of computational thinking are incoherent, complete, and systematic.

Second, students' computational thinking in solving row problems based on assimilation and accommodation learning styles starts from the step of understanding problems in the stages of computational thinking, including the decomposition, abstraction, and algorithmic stages. The steps of planning strategies in the stages of computational thinking include the stages of decomposition. The steps to implement strategies in the computational thinking stages include the decomposition, debugging, and abstraction stages. The steps to conclude and reexamine the stages of computational thinking include the stages of decomposition, the stages of generalization, and the stages of decomposition, where the stages of computational thinking are coherent, complete, and systematic.

Third, students' computational thinking in solving row problems based on convergent learning styles and accommodation in solving problems based on convergent learning styles and accommodation starts from understanding problems in the stages of computational thinking, including abstraction and algorithmic stages. The steps to plan strategies in the stages of computational thinking include the stages of generalization, decomposition, generalization, and decomposition. The steps to implement strategies in the stages of computational thinking include the stages of decomposition, the algorithmic stages, and the stages of abstraction. The steps to conclude and re-examine the computational thinking stages include the decomposition and debugging stages, where the stages of computational thinking are coherent, complete, and unsystematic.

The steps of planning strategies in the stages of computational thinking include the stage of decomposition and the stage of generalization. The steps to implement strategies in the computational thinking stages include the decomposition, debugging, and generalization stages. The steps to conclude and reexamine the stages of computational thinking include the stage of generalization and the stage of decomposition, where the stages of computational thinking are incoherent, incomplete, and unsystematic.

REFERENCES

Aho, A. (2012). Computation and Computational Thinking. *The Computer Journal*, 56(7): 832–835.

Angeli, C., Voogt, J., Fluck, A., Webb, M., Cox, M., Malyn-Smith, J., & Zagami, J. (2016). "A K-6 Computational Thinking Curriculum". *Framework: Implications for Teacher Knowledge. Educational Technology & Society*, *19*(3), 47–57.

Barra'., M. (2012). Implementasi Gaya Belajar Model Kolb Dalam LC5E Untuk Memaksimalkan Pembelajaran Luas Segitiga Heron Dikelas VIII A SMPN 19 Palu. *Jurnal DIKDAS, 1:1, (September: 2012).*

Benakli, N., Kostadinov, B., Satyanarayana, A., & Singh, S. (2017). Introducing computational thinking through hands-on projects using R with applications to calculus, probability, and data analysis. *International Journal of Mathematical Education in Science and Technology,* 48(3), 393-427.

Boom, K. D., Bower, M., Arguel, A., Siemon, J., & Scholkmann, A. (2018). Relationship between computational thinking and a measure of intelligence as a general problem-solving ability.

Annual Conference on Innovation and Technology in Computer Science Education. *ITiCSE*, 206 211.

Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. *AERA*, 1–25.

Christi, S. R., & Rajiman, W. (2023). Pentingnya Berpikir Komputasional dalam Pembelajaran Matematika. *Journal on Education*, 5(04).

Clements, D. H., Battista, M. T., & Sarama, J. (2001). Logo and Geometry. *Journal for Research in Mathematics Education. Monograph, 10,* 1–177.

Council, N. R. (2011). *Report of a workshop on pedagogical aspects of computational thinking*. The National Academic Press: Newyork.

Creswell, J, W. (2014). *Research Design: Qualitative, Quantitative, and Mixed Methods Approaches, In………… (4th ed., Vol. 59)*. United States of America: SAGE.

Denning, P. J., & Tedre, M., (2016). The long quest for computational thinking. *ACM International Conference Proceeding Series*, 120–129.

DeSchryver, M. D., & Yadav, A. (2015). Creative and Computational Thinking in the Context of New Literacies: Working with Teachers to Scaffold Complex Technology-Mediated Approaches to Teaching and Learning. *Journal Society for Information Technology & Teacher Education, 23*(3).

Gadanidis, G. (2016). Artificial intelligence, computational thinking, and mathematics education. *The International Journal of Information and Learning Technology*, 34(2), 133-139.

Gadanidis, G., Cendros, R., Floyd, L., & Namukasa, I. (2017). Computational Thinking in Mathematics Teacher Education. *Contemporary Issues in Technology and Teacher Education*, 17(4), 458-477.

Grover, S., & Pea, R. (2013). *Computational Thinking in K – 12: A Review of the State of the Field.* 42(1), 38-43. Di akses tanggal 10 Januari 2023, dari https://doi.org/10.3102/0013189X12463051.

Kallia, M., van Borkulo, S. P. Drijvers, P., Barendsen, E., & Tolboom, J. (2021). Characterizing computational thinking in mathematics education: a literature-informed Delphi study. *Research in Mathematics Education*, 23(2), 159 – 187.

Kazakoff, E. R., Sullivan, A., & Bers, M. U. (2013). The Effect of a Classroom-Based Intensive Robotics and Programming Workshop on Sequencing Ability in Early Childhood. *Early Childhood Education Journal, 41*(4), 245–255.

Kolb, A.Y, Kolb, D.A. (2005). Learning styles and learning spaces: Enhancing experiential learning in higher education. *The Academy of Management Learning and Education*, 4(2),193–212.

Marchelin, L.E., Hamidah, D., & Resti, N. C. (2022). Efektifitas Metode Scaffolding dalam Meningkatkan Computational Thinking Siswa SMP pada Materi Perbandingan. *Jurnal Pengembangan Pembelajaran Matematika (JPPM), Vol. 4, Issue 1, February 2022*.

Nagai, Y., Shimogoori, A., Ariga, M., & Georgiev, G. V. (2019). Future learning and design creativity competency. *Proceedings of the International Conference on Engineering Design, ICED, 2019-August*(August), 499–508.

NCTM., (2010). *Principles and Standards for School Mathematics.* Reston, VA: The National Council of Teachers Mathematics, Inc.

Maharani, S., Kholid, M. N., Pradana, L. N., & Nusantara, T. (2019). Problem Solving in the Context of Computational Thinking. *Infinity Journal of Mathematics Education*, *8*(2), 109–116.

Maharani, S., Agustina, Z. F., & Kholid, M. N. (2021). Exploring the Prospective Mathematics Teacher's Computational Thinking in Solving Pattern Geometry Problems. Al-Ishlah: Jurnal Pendidikan, 13(3), 1756–1767.

Maharani, S., Nusantara, T., As'ari, A. R., & Qohar, A. (2019). How is The Student's Computational Thinking Ability On Algebraic? International Journal of Scientific & Technology Research, 8(9), 419–423.

Maharani, S., Nusantara, T., As'ari, A. R., & Qohar, Abd. (2020). Computational Thinking: Media Pembelajaran CSK (CT-Sheet for Kids) dalam Matematika PAUD. Jurnal Obsesi : Jurnal Pendidikan Anak Usia Dini, 5(1), 975–984. https://doi.org/10.31004/obsesi.v5i1.769

Papert, S. (1996). *The Connected Family: Bridging the Digital Generation Gap*. World: 221 Press.

Plucker, J. A. & Beghetto, R. A. (2016). *Revisiting the relationship among schooling, learning, and creativity. In J. C. Kaufman and J. Baer. (Eds.). Creativity and reasoning in cognitive development*. New York: Cambridge University Press.

Putra, M. R. A. L., Mahardika, G. P & Putro, H. P. (2019). Penerapan Kemampuan Problem Solving pada Siswa SMP Menggunakan Pendekatan Computational Thinking (CT) Berbasis Role Playing Game. *Jurnal Ilmiah Teknik Informatika,* 8(2) 158-164.

Safitri, R., Jamal, A., Hermawan, D., & Supriyanto, A. (2022). Pengenalan Pelatihan Computational Thinking untuk Guru dan Siswa SD-SMP di Jabodetabek. *Prosiding Seminar Nasional Pemberdayaan Masyarakat 2022.*

Sagala, R., Naungchalerm, P., Saregar, A., & El Islami, R. A. Z. (2019). Environment-friendly education as a solution to against global warming: A case study at Sekolah Alam Lampung, Indonesia. *Journal for the Education of Gifted Young Scientist,* 7(2), 85 – 97.

Scherer, R., Siddiq, F., & Viveros, B. S. (2019). The cognitive benefits of learning computer programming: A meta-analysis of transfer effects. *Journal of Educational Psychology*, *111*(5), 764–792.

Selby, C. (2014). *How Can the Teaching of Programming Be Used to Enhance Computational Thinking Skills*? Di akses tanggal 3 Maret 2023, dari https:doi.org/10.5694/j.1326-5377.1981.tb113016.x.

Sung, W., Ahn, J., & Black, J. B. (2017). Introducing Computational Thinking to Young Learners: Practicing Computational Perspectives Through Embodiment in Mathematics Education. *Technology, Knowledge and Learning*, 22(3), 1 – 21.

Supiarmo, M. G., Turmudi., & Susanti, E. (2021). Proses Berpikir Komputasional Siswa dalam Menyelesaikan Soal PISA Konten Change And Relationship Berdasarkan Self-Regulated Learning . *Jurnal Numeracy*, 18(1).

Syaeful, M., Prabawa, H.W., & Rusnayati, H. (2018). *Peningkatan Kemampuan Berpikir Komputasi Siswa Melalui Multimedia Interaktif Berbasis Model Quantum Teaching and Learning*. Bandung : Universitas Pendidikan Indonesia.

Ramlah, (2014). Pengaruh Gaya Belajar Dan Keaktifan Siswa Terhadap Prestasi Belajar Matematika. *Jurnal Ilmiah Solusi, Edisi 1 bulan Nopember*.

Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2015). Defining Computational Thinking for Mathematics and Science Classrooms. *Journal of Science Education and Technology*, *25*(1), 127–147.

Wing, J. M. (2017). *Computational Thinking*. Communication of The ACM, 49(3), 33 – 35.

Yadaf, A., Stephenson, C., & Hong, H. (2017). *Computational thinking for teacher education*. Communications of the ACM, 60(4), 55 – 62.