Jurnal Pendidikan Matematika
Volume 17, No. 2, May 2023, pp. 167-188
P-ISSN: 1978-0044, E-ISSN: 2549-1040, DOI: https://doi.org/10.22342/jpm.17.2.20042.167-188
Website: https://ejournal.unsri.ac.id/index.php/jpm
Accredited by SINTA 2: http://sinta2.ristekdikti.go.id/journals/detail?id=1811

Computational Thinking Skills Indicators in Number Patterns

Yullys Helsa[1], Dadang Juandi[2], Turmudi[3]

[1]Department of Primary School Education, Universitas Negeri Padang, Padang, Indonesia
[1, 2, 3]Department of Mathematics Education, Universitas Pendidikan Indonesia, Bandung, Indonesia
Email: yullys@fip.unp.ac.id

Abstract

This research aims to examine a) what computational thinking indicators have been developed by researchers, b) what computational thinking indicators can be used in learning mathematics appropriately, and c) how to describe the development of student computational thinking indicators from the answers of computational thinking tests. This research is a qualitative descriptive study through a process of collecting data from literature reviews, integrated computational thinking math tests, and interviews. Data collection instruments used research notes, interview sheets, and CT question sheets. The results showed that a) 20 computational thinking indicators had been studied by researchers, b) computational thinking indicators that could be used in learning mathematics include problem decomposition, abstraction, pattern recognition, procedural algorithms, and generalizations, and c) From the student answers, five proposed computational thinking indicators can be developed even though they were not perfect. The general implication of this research is that there are five indicators of computational thinking skills that can be used in mathematics learning, specifically in number patterns, which include problem decomposition, abstraction, pattern recognition, pattern recognition, procedural algorithm, and generalization. The researchers developed all five computational thinking skills indicators in the instructional designs of not only the number pattern concept but also combination, geometry, combinatorics, etc.

Keywords: Indicators, Computational Thinking Skills, Learning Mathematics, Number Pattern

Abstrak

Tujuan penelitian ini adalah untuk mengetahui a) apa saja indikator *computational thinking* yang sudah dikembangkan para ahli, b) apa saja indikator *computational thinking* yang bisa digunakan dalam pembelajaran matematika dengan tepat, dan c) bagaimana gambaran perkembangan indikator *computational thinking* mahasiswa dari jawaban soal *computational thinking* yang diberikan. Penelitian ini merupakan penelitian deskriptif kualitatif dengan mengumpulkan data dari hasil literatur review, tes matematika terintegrasi *computational thinking*, dan wawancara. Pemilihan sampel untuk interview dan tes dilakukan melalui teknik *convenient sampling.* Instrumen pengambilan data menggunalan catatan penelitian, lembar wawancara, dan lembar soal *computational thinking*. Teknik analisis triangulasi data digunakan pada penelitian ini. Hasil penelitian menunjukkan a) ada 20 indikator *computational thinking* yang telah diteliti para ahli, b) Indikator *computational thinking* yang dapat digunakan dalam pembelajaran matematika meliputi dekomposisi masalah, abstraksi, pengenalan pola, algoritma prosedur, dan generalisasi, dan c) Dari jawaban mahasiswa diperoleh gambaran bahwa kelima indikator *computational thinking* yang diajukan dapat berkembang meskipun belum sempurna. Implikasi penelitian ini secara global adalah ditemukannya lima indikator *computational thinking skills* yang dapat digunakan dalam pembelajaran matematika, khususnya bilangan, adalah dekomposisi masalah, abstraksi, pengenalan pola, algoritma prosedural, dan generalisasi. Peneliti bisa mengembangkan kelima indikator computational thinking skills dalam pendesainan pembelajaran matematika terutama konsep bilangan. Namun, tidak hanya bilangan tetapi juga kombinasi, geometri, kombinatorik, dan lainnya

Kata kunci: Indikator, Kemampuan Berfikir Komputasi, Pembelajaran Matematika, Pola Bilangan

How to Cite: Helsa, Y., Juandi, D., & Turmudi. (2023). Computational thinking skills indicators in number patterns. *Jurnal Pendidikan Matematika, 17*(2), 1-22.

©2023 Jurnal Pendidikan Matematika – Universitas Sriwijaya. This is an open access article under the CC-BY-NC-SA license (https://creativecommons.org/licenses/by-nc-sa/4.0/).

Received January 4, 2023; Revised March 13, 2023; Accepted May 29, 2023

INTRODUCTION

Computational thinking (CT) is a competency that plays an important role in the 21st century and should be integrated into learning in schools (Dede et al., 2013; Guggemos, 2021). Yadav et al. (2017) mentioned that Computational thinking (CT) is an essential skill for all youth to succeed in our technology and information-rich world. While CT has a growing presence within K-12 classrooms, libraries play an essential role in introducing these critical skills to all. Computational thinking assists in keeping pace with the rapid developments of the 21st century through problem-solving techniques with its application in a very broad area (Korkmaz et al., 2017; Rey et al., 2020). Computational thinking skill is believed to be the basic skill a person must have, much like reading, writing, and arithmetic (Wing, 2006). It is also viewed as a potential field that supports both individual and global development in a fast-paced world bearing a significant economic benefit (Cansu & Cansu, 2019). As such, computational thinking skills are necessary to have in the 21st century. In Indonesia, CT has been integrated into Kurikulum Merdeka (Emancipated Curriculum) through Permendikbudristek Regulation No. 008 on the learning outcomes of early childhood, elementary education, and secondary education in Kurikulum Merdeka (Kemendikbudristek, 2022). The regulation signifies that the integration of CT in mathematics learning is a priority now.

In brief, computational thinking is a thinking process using computational steps to formulate problems and solutions so that solutions can be generated effectively in information processing (Goodson et al., 2020; Wing, 2011). The main essence of CT is to think like a computer scientist when faced with a problem (Grover & Pea, 2013). Computational thinking allows one to solve problems like a computer through decomposition processes, pattern recognition, abstraction, and algorithm design (Kidd et al., 2017) In this process, CT involves cognitive and metacognitive activities (Cutumisu et al., 2019; Kang et al., 2022).

Computational thinking is one of the abilities that need to be developed through exercises and is one of the basic types of knowledge for high-level problem-solving abilities (Tim Penyusun Materi ITB, 2020). Therefore, for CT to develop properly, CT should be nurtured from an early age and become a part that needs to be developed in education. However, there are still many teachers who have little to no knowledge of CT and students still have difficulty solving higher-order thinking questions (Putra et al., 2022). An in-depth analysis indicates that CT questions are classified as problem-solving which are completed in nature so that their level is above higher-order thinking skills. Students already have difficulty in solving questions that are classified as higher-order thinking skills, even more with CT questions. This is also supported by the results of the meta-analysis conducted by Helsa et al. (2023) which concludes the student's CT skills as still low.

Historically, the development of CT in education was introduced in 1980 by Papert but with a different name (Rodríguez-Martínez et al., 2020). Later, (Wing, 2006) brought this term to light through his published discussion of computational thinking. Since then, there have been more in-depth discussions regarding CT. Indeed, currently, many countries have integrated CT into their school curriculum (Babazadeh & Negrini, 2022). CT integration in learning is carried out using various methods such as using Dynamic Mathematics Software (Van Borkulo et al., 2021) and Scratch (Fagerlund et al., 2021). Some try to develop students' CT using scaffolding in learning (Nurwita et al., 2022).

The integration of CT into the curriculum is also based on the consideration that the scope of CT in problem-solving is not only related to computers but also various scientific disciplines, including humanities, mathematics, arts, and natural sciences (Khosrow-Pour, 2018; Morze et al., 2022). There has been a trend in publications showing the relationship between STEM and CT (Anwar & Herman, 2022). Wu and Yang (2022) claimed that CT has a very close relationship with mathematical thinking. There are several components in learning mathematics that intersect with Computational Thinking Skills such as problem-solving, modeling, data analysis, interpretation, statistics, and probability, making it necessary to develop mathematics learning tools that integrate CT skills to develop mathematical thinking skills. Thus, CT helps students not only develop and apply mathematical concepts and competencies using software or programming but also solve problems in CT (Wu & Yang, 2022).

The idea of introducing computational thinking concepts into the curriculum, especially mathematics creates an important foundation for the development of future computational skills (Seiter & Foreman, 2013). This is because the application of CT in the curriculum (learning process) will enable students to see the relationship between subjects and life inside and outside the classroom. Learning to use CT will enable students to learn to think abstractly, algorithmically, and logically, and be prepared to solve complex and open-ended problems (Yadav et al., 2018). Therefore, it is only natural that CT is highly recommended to be the main lesson for students in higher education so they can prepare qualified future graduates (Kang et al., 2022; Sondakh et al., 2020). Unfortunately, the ability of students to computational thinking is still relatively low, especially after the pandemic (Rosali & Suryadi, 2021). Therefore, it is necessary to make efforts to improve students' CT.

To develop students' CT in learning mathematics, clear indicators are needed to support the assessment and help create materials necessary for making improvements. Assessment is carried out to increase student performance and confidence in their abilities and improve the learning process (Fagerlund et al., 2020; Martínez et al., 2020; Yambi, 2020). Even though many countries have integrated CT into their school curriculum, there are still difficulties in assessing this competency because conceptual boundaries are still vague, such as indicators or constituent components (Relkin et al., 2020; Weintrop et al., 2021). It is important to determine these indicators so that they can assist in compiling an effective, practical, and directed assessment form (Veerasamy et al., 2022). In other words, the formulated indicators will help in designing the test questions that will be given to students.

Therefore, further review is needed regarding CT indicators that can be integrated into learning mathematics so that it facilitates evaluation. This review can be done through a systematic literature review (SLR). There has been much research related to CT using the SLR method such as research from Suharto (2022) regarding the trend of publications related to CT from 2012-2021. However, this research only discusses the research trend of computational thinking skills in the learning process in schools in general. Another SLR study was performed by Acevedo-Borrega et al., (2022) regarding what pedagogical tools are used in integrating CT into learning. This research was conducted by examining research trends related to CT based on three perspectives, namely conceptual, documentary, and pedagogical. There has also been a trend of publications related to CT assessment, such as research from (Tang et al., 2020). However, Tang's research focuses on learning in schools in general that are not related to CT indicators that can be used in learning mathematics. There is yet an SLR study that discusses CT indicators that can be used in learning mathematics, especially number patterns. Thus, the novelty of this research is to eliminate this gap through an analysis of existing research trends to find indicators that can be used.

Generally, there is no problem with the CT indicators used by researchers. However, from the many studies, it appears that many indicators emerge. Several indicators have similar purposes but with different terms. Therefore, an analysis was needed to narrow them down so that they are more effective in assessing students' abilities. Therefore, this study could pose as a convenient tool to decide which components can be used as a standard for evaluating CT skills, especially in mathematics learning. Therefore, the purpose of this research was 1) to see the development of CT components used by previous researchers; 2) to find what are the operational definitions of each indicator, and 3) to look at the intersections of the various components based on operational definitions. In the end, this research attempts to narrow down all the indicators into components that could be assessed and integrated into teaching mathematics at schools.

METHODS

This research is qualitative descriptive research by collecting data from literature reviews then conducting CT-integrated mathematics tests and interviewing CT test participants. The research method used to answer the first and second problems related to CT indicators, which is developed by experts, is to use the literature review method. For the third objective, the method used is qualitative methods through tests and interviews.

The main thing in any research is to find, select, weigh, and read the literature contained in a literature review (Creswell & Clark, 2018). The literature review is not only done by reading the literature but evaluating it comprehensively and critically based on a collection of studies that have been done before. The selection of this article was also carried out with the inclusion and exclusion criteria. The inclusion and exclusion criteria can be seen in Table 1.

Table 1. The inclusion and exclusion criteria for the selected articles

Criteria	Inclusion	Exclusion
Type	Indexed journal or conference	Unindexed journal
Title	"Computational thinking" mentioned	Not mentioned
Abstract	"Component/ Indicators" mentioned	Not mentioned
Language	English	Not in English
Proceeding articles	Not proceeding articles	Proceeding articles
Time	2011-2022	Before 2011
Context	Through lessons in school	Not through lessons in school

This research method begins by collecting literature with the keyword "component computational thinking". A total of 64,300 articles were found and then screened. The screening process is presented in Figure 1. Figure 1displays the process of elimination and screening that was carried out and 25 articles were used for analysis. Furthermore, articles that are used as literature are processed using coding with criteria. The coding process was carried out by three people. Coding is based on CT component criteria. Each article analyzes what components appear and are then made in a certain code. If a new component appears, then a new code and node are created. After analyzing the 25 articles, an analysis was carried out to find out which indicators could be used in learning mathematics.

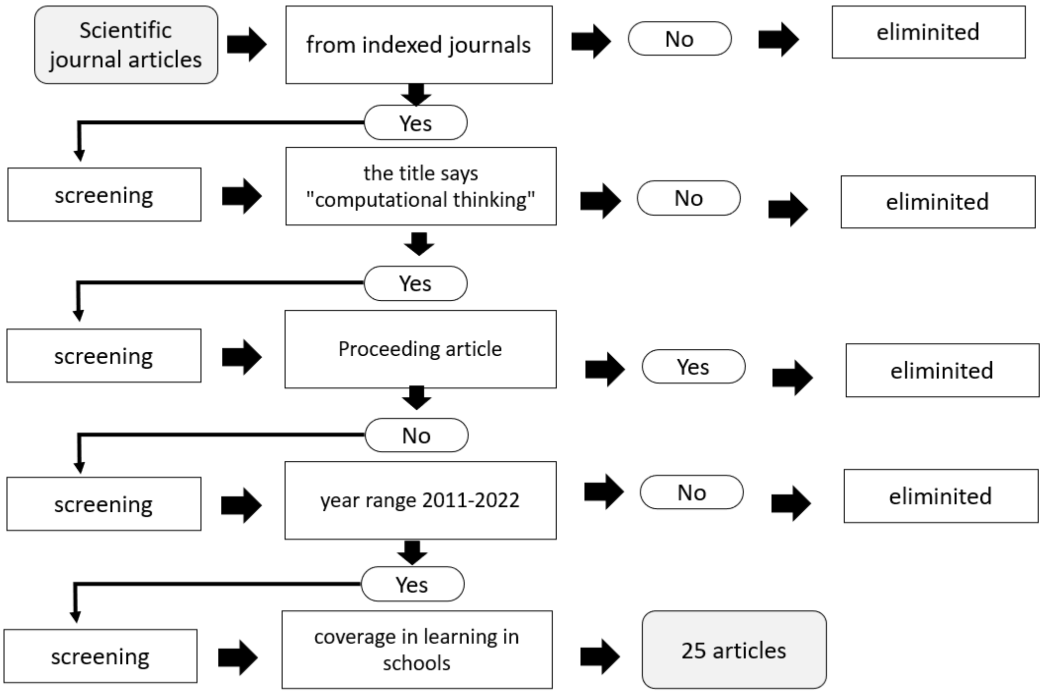


Figure 1. Article screening process diagram

This research was followed by qualitative methods using math tests and interviews. The population in this study were students at the elementary school teacher education program, while the sample was first-year students of the elementary school teacher education. A total of 468 students were used as samples. The sample selection technique for answering test questions and interviews was the convenient sampling method. The instruments used were CT test question sheets and interview sheets. This test consists of three questions that contain indicators that have been obtained before. The test questions were adopted from the Bebras competition questions (Blokhuis et al., 2015, 2016, 2017). Out of all student answers, several students were selected to represent the answers to be analyzed to see which CT indicators appeared. Then, the CT test answers were triangulated with the results of interviews with students to find out the reasons for choosing the steps in answering the questions.

RESULTS AND DISCUSSION

Research related to this CT indicator begins with identifying its components. The analysis and coding process was assisted by two encoders for reliability purposes. The encoders are lecturers from Padang State University and graduates of mathematics education master's degree, namely Vita Nova Anwar and Desmaiyanti. The checking process according to the set code was performed manually. The encoders read the entire selected article. Then, they find the indicators listed in the article. The indicators that appear are written down in notes and formulated in tabulations. Indicators that are similar and different are marked. Figure 1 depicts the coding process for the chosen 25 articles chosen.

Figure 2. Coding process by the encoder

Based on the results of the analysis, it was found that there were 20 types of CT components that researchers proposed. The 20 types CT components found included 1) data collection; 2) data analysis; 3) data representation; 4) problem decomposition; 5) abstraction; 6) algorithms and procedures; 7) automation; 8) pattern recognition; 9) parallelization; 10) debugging and evaluation; 11) modularization; 12) creativity; 13) collaboration; 14) troubleshooting; 15) critical; 16) simulation; 17) logical thinking; 18) assessing different solutions; 19) cognitive planning; and 20) reasoning.

The following describes the operational definition of each component based on the 25 articles used. Data collection gathers data using multiple computing tools, generates data for large and complex systems, and rearranges them in meaningful ways. Data analysis uses computational tools to analyze data, draw valid conclusions, and assess the strengths/weaknesses of data representation/representational systems. Data representation represents ideas in a computationally meaningful way and communicates and presents data in a variety of ways. Problem Decomposition breaks down a complex problem into smaller parts and reframes the problem into a recognizable problem. Abstraction focuses only on important information and ignores irrelevant information. Algorithms and procedures develop step-by-step problem solutions or rules to be followed in solving problems. Automation involves using digital and simulation tools to mechanize problem solutions. Pattern recognition and generalization identify emerging patterns, looking for commonalities among several problems. Parallelization means being able to process information together at one time. Debugging and evaluation find errors yourself and fix, identifies, and solves problems, develop strategies to make things work, and include troubleshooting. Modularization means breaking large tasks into smaller parts. Creativity means being able to solve problems that might occur when faced with new situations. Collaboration means learning to work together effectively to solve unstructured problems using computing. Problem-Solving uses specific skills and strategies to solve problems most logically and effectively and uses the basic concepts of computer science. Critical means using critical thinking and problem-solving skills/strategies to solve problems in the best way. The simulation uses experiments (simulations) to find solutions. Logical Thinking means being able to think logically. Assessing different solutions means being able to provide different solutions. Cognitive planning means being able to make problem-solving plans. Reasoning means being able to explain the choice of solutions used. In Table 2, the proposed CT components are presented.

Table 2. The analysis result of CT components

No	Researcher	Indicator Number																			
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1.	Barr & Stephenson (2011)	√	√	√	√	√	√	√		√							√				
2.	Lee et al., (2011)		√			√		√													
3.	Grover & Pea (2013)	√		√		√			√	√	√	√									
4.	Yadav et al., (2014)						√								√	√					
5.	Selby & Wollard (2014)				√	√	√		√		√										
6.	ISTE (2016)	√	√	√	√	√	√	√	√												
7.	Angeli et al (2016)				√	√	√		√		√										
8.	Wing (2016)				√	√	√	√	√												
9.	Basu et al., (2016)			√	√	√					√	√									
10.	Weintrop et al., (2016)	√	√	√	√	√	√														
11.	Bocconi et al., (2016)				√	√	√	√	√		√										
12.	Yadav et al., (2016)	√	√	√	√	√	√	√		√							√				
13.	Fronza et al., (2017)					√					√	√									
14.	Shute et al., (2017)	√	√	√	√	√	√		√		√										
15.	Guggemos et al., (2019)				√	√	√	√	√		√										

No	Researcher	Indicator Number																			
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
16.	Csizmadia et al., (2019)				√	√	√		√		√										
17.	Fagerlund et al., (2020)				√	√	√		√				√								
18.	Hadad et al., (2020)				√	√	√		√		√										
19.	Relkin et al., (2020)			√			√				√	√									
20.	Polat et al., (2021)						√						√	√	√	√					
21.	Kong & Wang (2021)					√	√			√	√	√			√						
22.	Asbell-Clarke et al., (2021)				√	√	√		√												
23.	Degiene & Dolgopolvas (2022)				√	√	√		√		√										
24.	Babazadeh & Negrini (2022)	√	√	√	√	√	√		√				√	√	√	√		√	√	√	√
25.	Refvik & Bjerke (2022)					√				√											

The table above reveals that the highest number of research performed was in 2016, with as many as 7 articles. There is a trend for many researchers on several indicators. If we look closely, indicators number 17, 18, 19, and 20 are rarely used by researchers, while the most indicators used are numbers 4, 5, 6, 8, and 10. This trend is shown clearly in Figure 3.

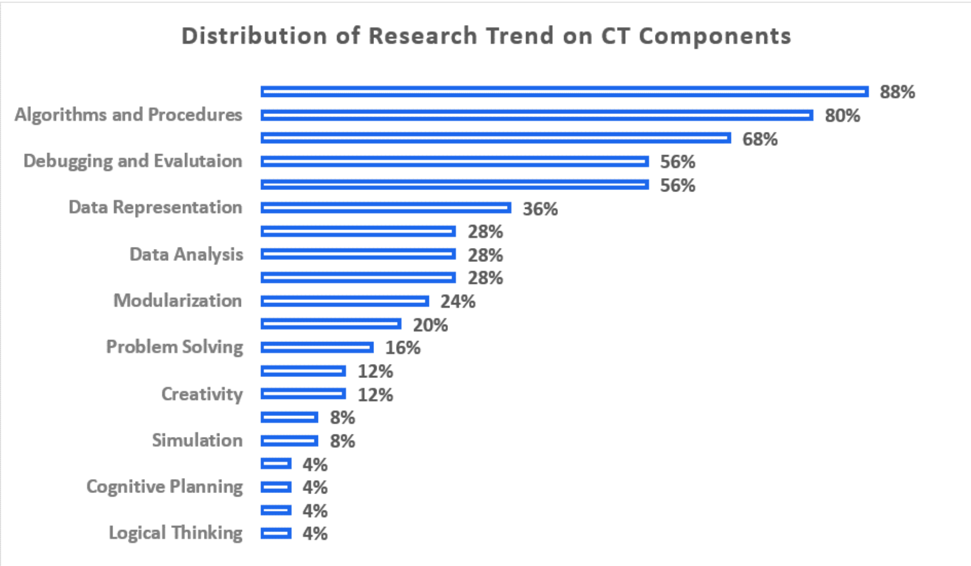


Figure 3. Researcher trend data on CT component

There is a trend for many researchers on several components of CT. If we look closely, reasoning, cognitive planning, assessing different solution and logical thinking components are rarely used by researchers, while the most indicators used are abstraction, algorithms and procedures, problem decomposition, debugging and evaluation, pattern recognition and data representation. This trend is shown clearly in Figure 3.

Based on Figure 3, several components used by most experts, amounting to more than 50% include problem decomposition, abstraction, procedural algorithms, pattern recognition, and debugging and evaluation. This tendency occurs since these five components already include other components.

These components can be used to represent CT components in general. Researchers' belief in having one component over the other is closely related to the operational definition of each proposed component. Therefore, this research also conducted a more in-depth analysis related to the operational definition of each component proposed by researchers.

Figure 3 displays that the most agreed component by all researchers is abstraction, which amounted to 88%. Abstraction implies ignoring unnecessary details (Selby et al., 2015). In other words, abstraction separates important information from useless information (Shute et al., 2017). Abstraction is considered a constructive thinking tool that represents CT (Acevedo-Borrega et al., 2022). Therefore, it is natural for all researchers to include abstraction as a component of CT. Abstraction is not a novel idea in mathematics, for every time you solve a math problem, it is necessary to convert information into something that can be used in solving problems. In mathematics, abstraction includes modularization, critical thinking, and cognitive planning. The second most common component is algorithms and procedures, with a percentage of 80%. Algorithms and procedures simply mean finding solutions (Bocconi et al., 2016). This component defines reusable procedures that solve a series of problems (Asbell-Clarke et al., 2021). In addition, this component requires problem-solving skills related to designing a step-by-step solution to a problem and, therefore, is different from coding (Selby et al., 2015). This component is very compatible with mathematics because solving mathematical problems requires planning problem-solving procedures before stepping into solving the problem.

The third most common component is problem decomposition, with a percentage of 68%. Problem decomposition refers to the skill of breaking down complex problems into simpler ones (Angeli et al., 2016). Csizmadia et al., (2019) defined problem decomposition as ways of thinking about components or parts. The parts can then be understood, completed, developed, and evaluated separately. This makes complex problems easier to solve, new situations better understood, and large systems easier to design. In other words, reducing the complexity of the problem makes it easier to understand the problem (Asbell-Clarke et al., 2021). This component is also always used in solving mathematical problems, especially questions with higher-order thinking skills. When decomposing the problem, the data are changed into simpler forms, so they are less complex. Thus, it takes the ability to present data, represent data, and read data properly.

The fourth component most frequently found in articles is pattern recognition, which has the same percentage as problem decomposition. In pattern recognition, there is an analysis of trends and groupings of a collection of objects, tasks, or information (Asbell-Clarke et al., 2021). Pattern recognition is also closely related to data representation ability (Waterman et al., 2020). When the data has been presented at its maximum, it will be easier to see patterns. An in-depth analysis of learning is needed regarding the patterns that emerge among students in comparison with the patterns that are expected to show (Basu et al., 2016). Patterns are a common representation of mathematics. Mathematics is sometimes defined as the science of patterns. The fifth component with a percentage above 50% is generalization. Generalization is related to concluding so this component is both the key to the completion and the failure of a problem. Generalization can only take place properly if other processes are carried out correctly.

The description of the indicators above is still general and can be used in many subjects. If we look at the operational definitions proposed for each indicator, we can see the intersection of several indicators. These intersections can be conical and can be used in learning mathematics. If conical, the 20 components above are then divided into main components and companion components. The description of the two types of components is presented in Table 3.

Table 3. Table of indicator intersections on computational thinking skills components

No.	Main Component	Complementer Components
1.	Decomposition	Data collection, data analysis, data representation
2.	Pattern recognition	Simulation, logical thinking, creativity, assessing different solution
3.	Abstraction	Modularization, cognitive planning, critical
4.	Algorithms and procedures	Parallelization, automation, problem solving, reasoning, collaboration
5.	Generalization	Debugging and evaluation

These findings have implications for the translation of CT indicators that will be used in learning mathematics. The description of indicators that can be used in learning mathematics is displayed in Table 4.

Table 4. CT indicators in learning mathematics

	Component		Indicator
1.	Decomposition	-	collect data using several computing tools
		-	rearrange data in a meaningful and recognizable way
		-	use computing tools to analyze data and draw valid conclusions
		-	break down complex questions into smaller parts
		-	communicating and presenting data in a variety of ways
2.	Pattern recognition	-	identify the patterns that emerge with the experiment and find commonalities between the questions
		-	logically provide creative ideas for problem solutions in a variety of ways
3.	Abstraction	-	find important information relevant to the problem raised critically
		-	present a problem-solving plan
4.	Algorithms and procedures	-	develop solution steps or rules for solving problems
		-	explain the reason for choosing the step
		-	involves the use of tools in problem-solving
5.	Generalization	-	identify and solve problems
		-	conclude the solution to a problem
		-	find mistakes yourself and fix them

To assess CT, an assessment tool is needed. Assessment tools can be used as tests or non-test. As for the assessment in the form of a test, it is done by designing questions that contain CT indicators.

Researchers agree that the CT component can be assessed in stages so that all indicators do not have to be contained in just one question.

These five main indicators can be used in learning mathematics (see Table 4). To strengthen the analysis from the researchers above, testing was carried out on students by giving Computational Thinking questions adopted from Bebras' questions (Blokhuis et al., 2015, 2016, 2017). The Bebras questions are questions used in the computational thinking competition. This test is to see what indicators appear in problem-solving by students. The following is a description of the CT test questions, examples of student answers, as well as an analysis of the indicators that appear.

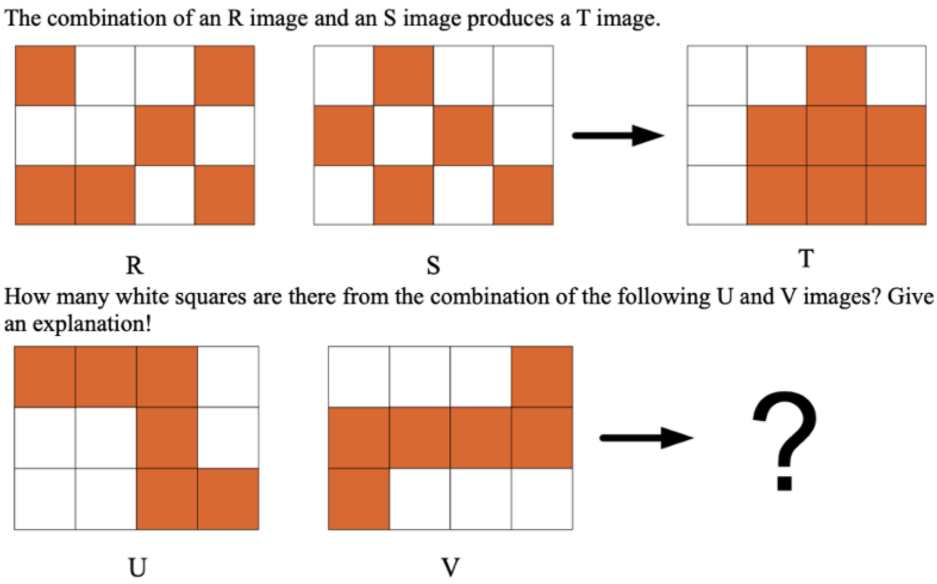
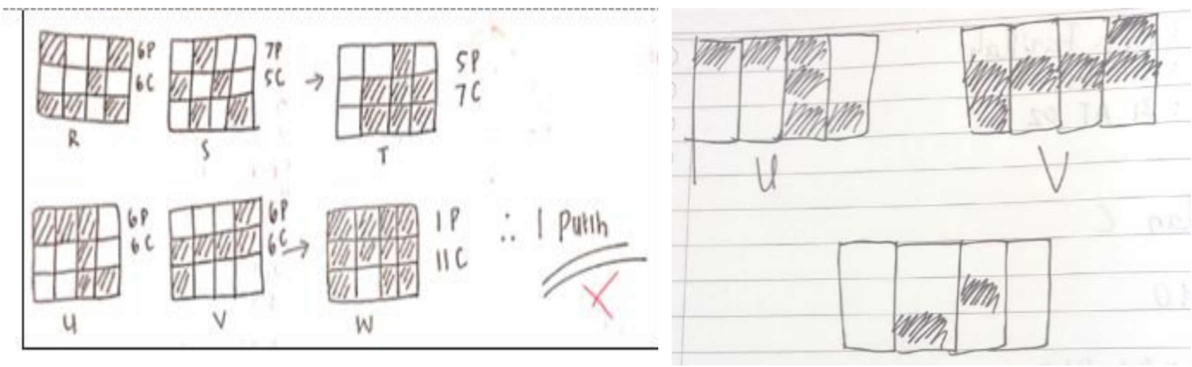


Figure 4. The 1st CT's question

Figure 4 is the 1st computational thinking question given to students. Students are given instructions in the form of a combination of R images and S images to produce T images. Next, students are asked to provide an explanation of the number of white boxes from the combined U and V images.



(a) (b)

Translated into English:
P = Putih = White
C = Coklat = Brown

Figure 5. Student's answer for the 1st CT's question

Figure 5 is the student's answer for the 1st CT's question. Based on the student's answers Figure 5 (a) the students put the U and V pictures close together so that the shaded box covers the empty box. As a result, only one square in W was empty after the merging process was done. Student A slightly made a mistake in analyzing the data presented in pictures R, S, and T. Student A's error is related to the lack of ability to see hidden patterns in the questions given. It is the teacher's responsibility to teach students how to detect patterns so that the patterns that appear are following the desired patterns in the questions (Basu et al., 2016). The correct answer was given by student's answer (b). Student (b) managed to find the pattern in the pictures R, S to become T. When two boxes of different shapes were combined, the box that formed was white. Meanwhile, if two boxes of the same shape were combined, a white box will form. As a result, student (b) gets the correct answer, that is, there are 10 white squares.

From this problem, the visible CT indicators were problem decomposition, abstraction, and pattern recognition. From the two answers given, students were able to present data in different forms that were easy to understand (problem decomposition), retrieve important information needed (abstraction), and provide creative solutions in different ways (pattern recognition). Although the presentation of the data provided was almost the same, there was a slightly different process for finding answers.

This is supported by student recognition during interviews. The following are the results of interviews with students regarding the answers to question number 1.

Student A	:	*"I combined the boxes in images U and V. In my opinion, the empty box will be covered by the shaded box so if one of the boxes is shaded, the box in image T will also be shaded. Therefore, in image W, there is only one empty box in the third row, the second column, because in that box there are no shaded boxes in both U and V images."*
Student B	:	*"In my opinion, the solution to the problem is each different color will produce a white color in the box. So, this box is brown because of the meeting of chocolate with chocolate. The key word is that each different color will produce white and each of the same colors will produce brown."*

From the results of the interviews, it is concluded that the two students have tried to carry out computational thinking processes including decomposition of data from images R, S, and T, abstraction, namely searching for important information from what is known, as well as pattern recognition in images R and S to form images T. From both students we can see that these three indicators have started to appear even though there were some mistakes.

Question 2:

"A couple of villages are separated by forests causing villagers to have difficulty traveling to other villages. For this reason, access will be made in the form of a road from one village to another provided that the number of households in the two villages to be connected is greater than a secret number. Out of the six existing villages, only four connecting bridges can be built as shown in the picture (the number shown in the picture is the number of households in the village)."

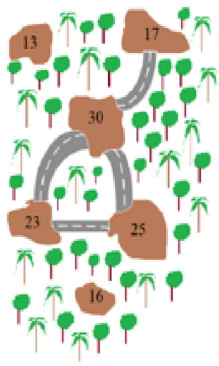
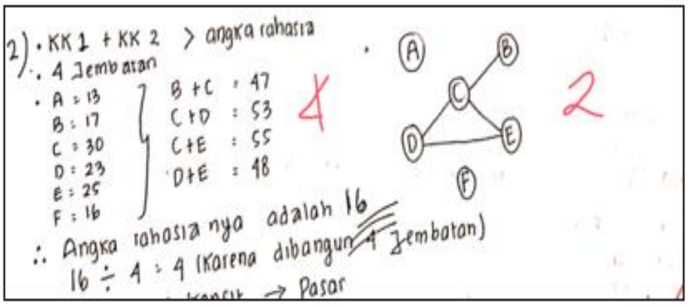


Figure 6. The 2nd CT's question

Figure 6 is the 2nd computational thinking question. This question is about the sum of family numbers and secret numbers. The secret number is greater than the sum of family number. The following are some student responses to the questions given.

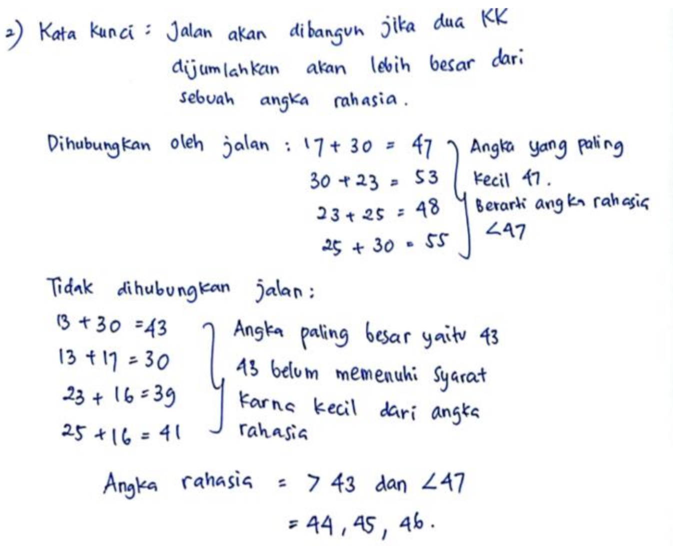


(a)

Translated into English:
2) The result of the sum of the family numbers (KK) 1 and 2 is greater than the secret number. The are four bridges.

A = 13	B + C = 47
B = 17	C + D = 53
C = 30	C + E = 55
D = 23	D + E = 48
E = 25	
F = 16	

The secret number is 16.
16 : 4 = 4, because of four bridges were built.



(b)

2) Keywords: the road will be built if the sum of the two family numbers are greater than the secret number.

Connected by road:
17 + 30 = 47
30 + 23 = 53
23 + 25 = 48
25 + 30 = 55
The smallest number is 47 means the secret number < 47

Unconnected by road
13 + 30 = 43
13 + 17 = 30
23 + 16 = 39
25 + 16 = 41
The largest number is 43, 43 does not meet the criteria because its number is smaller than the secret number

Secret number = > 43 and < 47
= 44, 45, 46

Figure 7. Student's answers to the 2nd CT's question

Figure 7 is student's answers to question number 2. Based on the answer of student Figure 7 (a) still could not find the right solution for the given problem. Student (a) also made a mistake in fulfilling the generalization indicator. In the process used, students (a) show good skills in representing problem data into a pattern representation but fail to generalize or make conclusions. Generalizing means formulating a solution in generic terms so that it can be applied to different problems (Angeli et al., 2016). On the other hand, student (b) almost found the right solution with a description of logical reasons. Student (b) was able to generalize from pattern recognition that was previously done. This is supported by students' accounts during interviews. The following is an excerpt from the interview results related to question number 2.

From the answers above, the CT indicators that emerged were problem decomposition (presenting the problem in an easy-to-understand form), abstraction (choosing important information), pattern recognition (providing creative ways to solve problems), and generalization (making conclusions about solutions to problems). The secret number asked for in this question was 46. Student (a) tried to visualize road access in each village according to the question.

Student A	:	*"I first made an example of the six villages with the initials A, B, C, D, E, and F. Then, I calculated the possible number of family number from the two villages. I found that 16 was less than the number of family number between the two available villages. Because there are four bridges, the secret number is 16:4, which is 4."*
Student B	:	*"I counted the number of family number from the two villages connected by a bridge. There are four bridges, so the smallest number of family number is 47. The key word in the question is that the number of family number must be greater than the secret number, so the secret number must be less than 47. Then, I also added up the family number from the two villages that were not connected by bridges. I found that the largest number was 43. So, I concluded that the secret number must be greater than 43 but less than 47, so the possible secret numbers are 44, 45, or 46."*

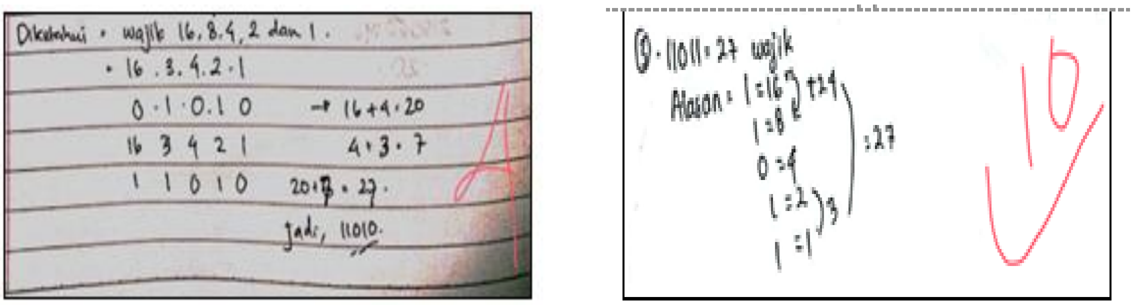
Based on the results of the interviews, the indicators of CT that were well-developed in students were problem decomposition and abstraction. However, the students were still not skilled in the process of pattern recognition and generalization.

Question 3:

"There are five cards that are placed in a row. Each card contains a certain number of diamond symbols, from left to right, respectively, 16, 8, 4, 2, and 1 diamond. At the bottom of the card is written the number 0 or 1. The number 0 is written if the card is closed (no diamonds are visible). The number 1 is written if the card on it is open (you can see the diamond image). Cards can be used to generate codes for numbers. For example, there are 10 diamonds shown in the following image, so the code for the number 10 is 01010."

Figure 8. The 3rd CT's question

Figure 8 depicts the third CT question given to students. This question required students to develop decomposition and abstraction. The following are some student responses to the questions given.



Translated into English:

Given:
Diamond 16, 18, 8, 4, 2, and 1

16	3	4	2	1		
0	1	0	1	0	=>	16 + 4 = 20
16	3	4	2	1	=>	4 + 3 = 7
1	1	0	1	0	=>	20 + 7 = 27

11011 = 27 diamonds
Reason

$$\left.\begin{array}{l}\left.\begin{array}{l}1 = 16\\ 1 = 8\end{array}\right\} 24\\ 0 = 4\\ \left.\begin{array}{l}1 = 2\\ 1 = 1\end{array}\right\} 3\end{array}\right\} 27$$

(a) (b)

Figure 9. Student's answer for the 3rd CT's question

Figure 9 is the student's answer for the 3rd CT's question. Students (a) made mistakes in pattern recognition, so the results obtained were wrong. The students were confused about finding patterns even though they had arranged a way to get the number 27. Meanwhile, student (b) managed to find the pattern correctly. Based on the student's answers above, the students were able to provide reasons for choosing the settlement step (Zoud & Namukasa, 2023). This problem is contained in the procedure algorithm indicator. The two answers also showed creative ideas for solving problems contained in the pattern recognition indicator. The following are the results of interviews with students regarding the completion of question number 3.

Student A	:	*"I opened the 16, 8, and 2 poker cards to make 27."*
Student B	:	*"I added cards 16 and 8 to get 24. To get 27, I needed 3 more diamonds. So, I opened cards 3 and 1 so that if the poker cards are added together it makes up 27 diamonds."*

Based on the results of the interviews, the two students were close to what the question wanted, but student (a) made mistakes in adding up so the results obtained were wrong. From the results of the interviews, the CT indicator that appears and develops well was the selection of the completion step (procedure algorithm). The least developed indicator is pattern recognition. The mistakes made indicate that students (a) do not understand the patterns provided by the questions. This is related to the low reading literacy skills of students. Based on interviews with students, most students admit that they are less inclined to read CT questions because they are generally in the form of quite long sentences. They also have difficulty understanding word-for-word questions because they are not used to them.

Based on the three questions above, the indicators that appeared in the third student's answers to the Computational Thinking questions were 1) problem decomposition, 2) abstraction, 3) pattern recognition, 4) procedure algorithm, and 5) generalization. In problem-solving, in general, these five indicators have developed but the development was still not optimal and there are still a few mistakes when they solve a given problem.

The findings reveal that 20 indicators have been developed by researchers during the 2011-2022 timeframe. These 20 indicators are narrowed down into five main indicators based on the definitions of each component that can be used in learning mathematics. Furthermore, the results of tests and interviews found that almost all five indicators appeared but were not yet fully acquired.

CONCLUSION

CT helps the newer generation of the 21st century to develop high-level thinking patterns like computers in solving problems. The scope of Computational thinking is very broad and has the potential to maximize the ability development of future generations. To realize this, it is necessary to integrate CT into the curriculum of schools and universities. An in-depth understanding of the CT constituent components is needed to help in the assessment process so that we can integrate CT seamlessly. The CT component can be broken down into a CT indicator so that the assessment tool can be adjusted in such a way as to be valid for assessing CT.

The results of the research showed that there were at least 20 CT indicators that have been studied by researchers. Out of the 20 indicators, CT indicators that can be used in learning mathematics include problem decomposition, abstraction, pattern recognition, procedural algorithms, and generalizations. Then, from the results of the analysis of student answers and interviews, we found five proposed CT indicators that could be developed even though they were not perfect. The results of this study are targeted specifically for teachers or other researchers who will carry out assessments to get conclusions regarding students' CT in mathematics lessons, especially number patterns. This is a novelty because previously it was quite difficult to measure students' CT abilities as there was no definite indicator in conducting the assessment.

The limitation of this research is that it only analyzes the components and indicators of CT skills in general and then relates them to mathematics. A more in-depth analysis is needed regarding research that focuses on discussing CT in mathematics.

ACKNOWLEDGMENTS

We would like to thank a group of lecturers from Padang State University and graduates of mathematics education master's degree, namely Vita Nova Anwar and Desmaiyanti, who became the encoders of this study, as well as undergraduate students of Padang State University, who were willing to be subjects in this research.

REFERENCES

Acevedo-Borrega, J., Valverde-Berrocoso, J., & Garrido-Arroyo, M. D. C. (2022). Computational thinking and educational technology: A scoping review of literature. *Education Sciences*, *12*(1). https://doi.org/10.3390/educsci12010039.

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 and Society*, *19*(3), 47–57. Retrieved from https://www.jstor.org/stable/jeductechsoci.19.3.47 .

Anwar, V. N., & Herman, T. (2022). Bibliometric Analysis of publication trends stem approach based on computational thinking in learning mathematics [in Bahasa]. *JPMI*, *5*(5), 1387–1396. https://doi.org/10.22460/jpmi.v5i5.1387-1396.

Asbell-Clarke, J., Rowe, E., Almeda, V., Edwards, T., Bardar, E., Gasca, S., Baker, R. S., & Scruggs, R. (2021). The development of students' computational thinking practices in elementary and middle-school classes using the learning game, Zoombinis. *Computers in Human Behavior*, *115*. https://doi.org/10.1016/j.chb.2020.106587.

Babazadeh, M., & Negrini, L. (2022). How is computational thinking assessed in European K-12 education? A systematic review. *International Journal of Computer Science Education in Schools*, *5*(4), 3–19. https://doi.org/10.21585/ijcses.v5i4.138.

Basu, S., Biswas, G., Sengupta, P., Dickes, A., Kinnebrew, J. S., & Clark, D. (2016). Identifying middle school students' challenges in computational thinking-based science learning. *Research and Practice in Technology Enhanced Learning*, *11*(1). https://doi.org/10.1186/s41039-016-0036-2.

Blokhuis, D., Csizmadia, A., Millican, P., Roffey, C., Schijvers, E., & Sentence, S. (2017). *UK Bebras Computational Thinking Challenge. UK Bebras*. http://www.bebras.uk/answer-booklets.%0Ahtml.

Blokhuis, D., Millican, P., Roffey, C., Schrijvers, E., & Sentance, S. (2015). *UK Bebras Computational Thinking Challenge. UK Bebras*. http://www.bebras.uk/answer-booklets.html.

Bocconi, S., Chioccariello, A., Dettori, G., Ferrari, A., Engelhardt, K., Kampylis, P., & Punie, Y. (2016). Developing Computational Thinking in Compulsory Education - Implications for policy and practice. In *Joint Research Centre (JRC)* (Issue June). https://doi.org/10.2791/792158.

Cansu, F. K., & Cansu, S. K. (2019). An overview of computational thinking. *International Journal of Computer Science Education in Schools*, *3*(1), 17–30. https://doi.org/10.21585/ijcses.v3i1.53.

Creswell, J. W., & Clark, V. L. P. (2018). *Designing and Conducting Mixed Methods Research*. SAGE Publications. Retrieved from https://us.sagepub.com/en-us/nam/designing-and-conducting-mixed-methods-research/book241842.

Csizmadia, A., Standl, B., & Waite, J. (2019). Integrating the constructionist learning theory with computational thinking classroom activities. *Informatics in Education*, *18*(1), 41–67. https://doi.org/10.15388/infedu.2019.03.

Cutumisu, M., Adams, C., & Lu, C. (2019). A scoping review of empirical research on recent computational thinking assessments. *Journal of Science Education and Technology*, *28*(6), 651–676. https://doi.org/10.1007/s10956-019-09799-3.

Dede, C., Mishra, P., & Voogt, J. (2013). Working Group 6: Advancing computational thinking in 21. *International Summit on ICT in Education*, 1–6. https://ris.utwente.nl/ws/files/6168377/Advancing_computational_thinking_in_21st_century_learning.pdf.

Fagerlund, J., Hakkinen, P., Vesisenaho, M., & Viiri, J. (2020). assessing 4th grade students' computational thinking through scratch programming projects. *Informatics in Education*, *19*(4), 611–640. https://doi.org/10.15388/INFEDU.2020.27.

Fagerlund, J., Häkkinen, P., Vesisenaho, M., & Viiri, J. (2021). Computational thinking in programming with Scratch in primary schools: A systematic review. *Computer Applications in Engineering Education*, *29*(1), 12–28. https://doi.org/10.1002/cae.22255.

Fronza, I., El Ioini, N., & Corral, L. (2017). Teaching computational thinking using agile software engineering methods: A framework for middle schools. *ACM Transactions on Computing Education*, *17*(4), 1–27. https://doi.org/10.1145/3055258.

Goodson, B., Sarna, M., & Associates, A. (2020). *Measuring Computational Thinking and Computer Science Outcomes*. Education Innovation and Research (EIR) Project Directors and Evaluators Technical Assistance Meeting. https://oese.ed.gov/files/2021/03/CSCTOutcomes_508.pdf.

Grover, S., & Pea, R. (2013). Computational thinking in K-12: A review of the state of the field. *Educational Researcher*, *42*(1), 38–43. https://doi.org/10.3102/0013189X12463051.

Guggemos, J. (2021). On the predictors of computational thinking and its growth at the high-school level. *Computers and Education*, *161*, 104060. https://doi.org/10.1016/j.compedu.2020.104060.

Guggemos, J., Seufert, S., & Román-González, M. (2019). Measuring computational thinking - Adapting a performance test and a self-assessment instrument for German-speaking countries. *16th International Conference on Cognition and Exploratory Learning in Digital Age, CELDA 2019*, *Celda*, 183–191. https://doi.org/10.33965/celda2019_201911l023.

Hadad, R., Thomas, K., Kachovska, M., & Yin, Y. (2020). Practicing formative assessment for computational thinking in making environments. *Journal of Science Education and Technology*, *29*. https://doi.org/10.1007/s10956-019-09796-6.

Helsa, Y., Suparman, Juandi, D., & Turmudi. (2023). A meta-analysis of the utilization of computer technology in enhancing computational thinking skills: Direction for mathematics learning. *International Journal of Instruction*. https://doi.org/10.29333/iji.2023.16239a.

Kang, C., Liu, N., Zhu, Y., Li, F., & Zeng, P. (2022). Developing college students' computational thinking multidimensional test based on life story situations. *Education and Information Technologies*. https://doi.org/10.1007/s10639-022-11189-z.

Kemendikbudristek. (2022). *Learning outcomes in early childhood, elementary education, and secondary education in the independent curriculum [in Bahasa]*. https://kurikulum.kemdikbud.go.id/wp-content/unduhan/CP_2022.pdf.

Khosrow-Pour, M. (2018). *Encyclopedia of Information Science and Technology, Fourth Edition* (Fourth). IGI Global. https://doi.org/10.4018/978-1-5225-2255-3.ch417.

Kidd, T., R, L., & Morris, J. (2017). *Handbook of Research on Instructional Systems and Educational*

Technology. IGI Global. http://dx.doi.org/10.4018/978-1-5225-2399-4.

Kong, S. C., & Wang, Y. Q. (2021). Item response analysis of computational thinking practices: Test characteristics and students' learning abilities in visual programming contexts. *Computers in Human Behavior*, *122*. https://doi.org/10.1016/j.chb.2021.106836.

Korkmaz, Z., Çakir, R., & Özden, M. Y. (2017). A validity and reliability study of the computational thinking scales (CTS). *Computers in Human Behavior*, *72*, 558–569. https://doi.org/10.1016/j.chb.2017.01.005.

Lee, I., Martin, F., Denner, J., Coulter, B., Allan, W., Erickson, J., Malyn-Smith, J., & Werner, L. (2011). Computational thinking for youth in practice. *ACM Inroads*, *2*(1), 32–37. https://doi.org/10.1145/1929887.1929902.

Martínez, V., Mon, M. A., Álvarez, M., Fueyo, E., & Dobarro, A. (2020). E -Self-Assessment as a Strategy to Improve the Learning Process at University. *Education Research International*, *2020*. https://doi.org/10.1155/2020/3454783.

Morze, N., Barna, O., & Boiko, M. (2022). The relevance of training primary school teachers computational thinking. *CEUR Workshop Proceedings*, *3104*(Icteri 2021), 141–153. Retrieved from https://ceur-ws.org/ .

Nurwita, F., Kusumah, Y. S., & Priatna, N. (2022). *Exploring students' mathematical computational thinking ability in solving pythagorean theorem problems*. *13*(2), 273–287. https://doi.org/10.24042/ajpm.v13i2.12496.

Polat, E., Hopcan, S., Kucuk, S., & Sisman, B. (2021). A comprehensive assessment of secondary school students' computational thinking skills. *British Journal of Educational Technology*, *52*(5), 1965–1980. https://doi.org/10.1111/bjet.13092.

Putra, Z. H., Ramiati, R., Zufriady, Z., Hidayat, R., Jismulatif, J., Hermita, N., & Sulistiyo, U. (2022). *Development of computational thinking tasks based on Riau Malay culture for primary school*. 1–23. https://doi.org/10.1080/03004279.2022.2150063.

Relkin, E., de Ruiter, L., & Bers, M. U. (2020). TechCheck: Development and validation of an unplugged assessment of computational thinking in early childhood education. *Journal of Science Education and Technology*, *29*(4), 482–498. https://doi.org/10.1007/s10956-020-09831-x.

Rey, Y. A. R. del, Cambinda, I. N. C., Deco, C., Bender, C., Avello-Martínez, R., & Villalba-Condori, K. O. (2020). Developing computational thinking with a module of solved problems. *Computer Application in Engineering Education*, *29*(3), 506–516. https://doi.org/10.1002/cae.22214.

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.

Rosali, D. F., & Suryadi, D. (2021). An Analysis of Students' Computational Thinking Skills on The Number Patterns Lesson during The Covid-19 Pandemic. *Formatif: Jurnal Ilmiah Pendidikan MIPA*, *11*(2), 217–232. https://doi.org/10.30998/formatif.v11i2.9905.

Seiter, L., & Foreman, B. (2013). Modeling the learning progressions of computational thinking of primary grade students. *ICER 2013 - Proceedings of the 2013 ACM Conference on International Computing Education Research*, 59–66. https://doi.org/10.1145/2493394.2493403.

Selby, C., Dorling, M., & Woollard, J. (2015). Evidence of assessing computational thinking. *IFIP TC3*

Working Conference: A New Culture of Learning: Computing and Next Generations, 232–242. http://www.ifip2015.mii.vu.lt/file/repository/IFIP_Proceedings.pdf.

Selby, C., Dorling, M., & Woollard, J. (2014). Evidence of assessing computational thinking. *IFIP TC3 Working Conference: A New Culture of Learning: Computing and Next Generations*, 232–242. http://www.ifip2015.mii.vu.lt/file/repository/IFIP_Proceedings.pdf.

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.

Sondakh, D. E., Osman, K., & Zainudin, S. (2020). A proposal for holistic assessment of computational thinking for undergraduate: Content validity. *European Journal of Educational Research*, *9*(1), 33–50. https://doi.org/10.12973/eu-jer.9.1.33.

Suharto, H. (2022). Systematic literature review (SLR) computational thinking learning science in the period 2012 - 2021. *International Journal of Educational Technology and Instruction*, *1*(1), 1–13. https://ijeti-edu.org/index.php/ijeti/article/view/1.

Tang, X., Yin, Y., Lin, Q., Hadad, R., & Zhai, X. (2020). Assessing computational thinking: A systematic review of empirical studies. *Computers and Education*, *148*(Mc 147). https://doi.org/10.1016/j.compedu.2019.103798.

Tim Penyusun Materi ITB. (2020). *Computational thinking in primary and secondary education* [in Bahasa]. Lembaga Penelitian dan Pengabdian Kepada Masyarakat. Retrieved from https://www.researchgate.net/profile/Ginar-Niwanputri/publication/350383897_Computational_Thinking_Learning_and_Teaching_Guide_for_Primary_and_Secondary_Schools_in_Indonesia/links/605cc073458515e8346fdb11/Computational-Thinking-Learning-and-Teaching-Guide-for-Primary-and-Secondary-Schools-in-Indonesia.pdf.

Van Borkulo, S., Chytas, C., Drijvers, P., Barendsen, E., & Tolboom, J. (2021). Computational thinking in the mathematics classroom: Fostering algorithmic thinking and generalization skills using dynamic mathematics software. *ACM International Conference Proceeding Series*. https://doi.org/10.1145/3481312.3481319.

Veerasamy, A. K., Laakso, M. J., & D'Souza, D. (2022). Formative assessment tasks as indicators of student engagement for predicting at-risk students in programming courses. *Informatics in Education*, *21*(2), 375–393. https://doi.org/10.15388/infedu.2022.15,

Waterman, K. P., Goldsmith, L., & Pasquale, M. (2020). Integrating computational thinking into elementary science curriculum: An examination of activities that support students' computational thinking in the service of disciplinary learning. *Journal of Science Education and Technology*, *29*(1), 53–64. https://doi.org/10.1007/s10956-019-09801-y.

Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2016). Defining computational thinking for mathematics and science classrooms. *Journal of Science Education and Technology*, *25*(1), 127–147. https://doi.org/10.1007/s10956-015-9581-5.

Weintrop, D., Rutstein, D. W., Bienkowski, M., & McGee, S. (2021). Assessing computational thinking: An overview of the field. *Computer Science Education*, *31*(2), 113–116. https://doi.org/10.1080/08993408.2021.1918380.

Wing, J. M. (2006). *Computational thinking*. *Researchgate.Net*, 3–3. https://doi.org/10.1109/vlhcc.2011.6070404.

Wing, J. M. (2011). Research Notebook: Computational thinking -what and why? *The Link Magazine*. Retrieved from https://www.cs.cmu.edu/link/research-notebook-computational-thinking-what-and-why.

Wing, J. M., & Stanzione, D. (2016). Progress in computational thinking, and expanding the HPC community. *Communications of the ACM*, *59*(7), 10–11. https://doi.org/10.1145/2933410.

Wu, W. R., & Yang, K. L. (2022). The relationships between computational and mathematical thinking: A review study on tasks. *Cogent Education*, *9*(1). https://doi.org/10.1080/2331186X.2022.2098929.

Yadav, A, Krist, C., Good, J., & Caeli, E. N. (2018). Computational thinking in elementary classrooms: measuring teacher understanding of computational ideas for teaching science. *Computer Science Education*, 1–30. https://doi.org/10.1080/08993408.2018.1560550.

Yadav, A, Stephenson, C., & Hong, H. (2017). Computational thinking for teacher education. *Communications of the ACM*, *60*(4), 55–62. Retrieved from https://edtechbooks.org/-TN/.

Yadav, Aman, Hong, H., & Stephenson, C. (2016). Computational thinking for All: Pedagogical approaches to embedding 21st century problem solving in K-12 classrooms. *TechTrends*, *60*(6), 565–568. https://doi.org/10.1007/s11528-016-0087-7.

Yadav, Aman, Mayfield, C., Zhou, N., Hambrusch, S., & Korb, J. T. (2014). Computational thinking in elementary and secondary teacher education. *ACM Transactions on Computing Education*, *14*(1), 1–16. https://doi.org/10.1145/2576872.

Yambi, T. de A. C. (2020). Assessment and Evaluation in Education EDUC 540 Spring 2020. *Researchgate*, Retrieved from https://www.researchgate.net/publication/342918149_ASSESSMENT_AND_EVALUATION_IN_EDUCATION.

Zoud, R., & Namukasa, I. (2023). Computational thinking workshop: A new way to learn and teach mathematics. *journal of research in science*, *Mathematics and Technology Education*, 6(2), 99-119. https://doi.org/10.31756/jrsmte.624.