JURNAL PEMBELAJARAN FISIKA

http://jurnal.fkip.unila.ac.id/index.php/JPF

Vol 9 (1), 2021, 101-112 ISSN: 2302-0105 (p); 2684-9828 (e)

How to Bring Computational Thinking Approach to The Non-Computer Science Student's Class???

Hervin Maulina[1]*, Abdurrahman Abdurrahman [1], Ismu Sukamto[2]

[1] Physics Education, Faculty of Teacher Training and Edication, Universitas Lampung, Indonesia

[2] PGSD, Faculty of Teacher Training and Edication, Universitas Lampung, Indonesia

* *e-mail: hervin.maulina@fkip.unila.ac.id*

Received: November 17, 2020 *Accepted: July 17, 2021* *Published: July 19, 2021*

Abstract: Computational Thinking (CT) skill is the ability to solve problems with computer thinking. In addition, CT can be seen as a structured and systematic approach that can be implemented in learning. This study aims to bring the computational thinking approach to the non-computer science student's class and involved 35 undergraduate students of physics education in the computational physics course. The research method used was the mixed method sequential explanatory design (Creswell & Plano Clark, 2011), with the following design. Broadly speaking, the flow of the mixed method research method with an explanatory sequential design in this study includes the collection of quantitative data obtained from student self-evaluation instruments related to the understanding of the CT approach stage. The results showed that the Computational Thinking (CT) approach can be applied to non-computer science students in online learning which includes 6 stages of implementation and 6 stages of implementation. Other results indicate that this method can be used in improving student CT skills.

Keywords: Computational thinking, physics, problem solving

DOI: http://dx.doi.org/10.23960/jpf.v9.n1.202109

INTRODUCTION

Disruption of technology in the last few years has become a central issue that is being discussed in various aspects of life, including education. This era requires humans to have high-level thinking, analytical, outside of routine, and all systems involved in it are not manual or automatic. In addition, this disruption of technology affects the world of education in several main aspects, namely curriculum, learning, and assessment. If education is not able to produce a competent next generation in accordance with the demands of this era, it will be replaced by machines or robots. Even in the future, some jobs will disappear and replace with types of jobs that involve automation in the process.

Educators around the world face various challenges in the use of digital technology in learning. So this is very important for educators to integrate or add new skills in the world of education. These skills are computer programming or coding known as computational thinking (CT). These skills can be implemented in non-computer science classes. CT itself is a new literacy in the 21st century (Wing, 2011) and is a very important skill for students (Yadav et al., 2016; Monday and Nasri, 2019). However, this concept is actually not new. Over the past few years, CT was expressed as a skill in programming that students of all fields need to master (Grover and Pea, 2013), which is presented as algorithmic thinking. This is evidenced by the presence of automation processes such as computer systems using regular and sequential steps (Yadav et al., 2017). However, there has been a change in definition, from thinking like computer scientists when facing problems (Román-González et al. 2017) now extending to other disciplines (Anderson, 2016; Shute et al., 2017).

So far, research on CT has only focused on concepts, practices, and perspectives (Brennan and Resnick, 2012; Lye and Koh, 2014; Román-Gonzálezetal, 2017), and in-service and pre-service teacher opinions regarding CT contributions (Gunbatar, 2019). However, if observed further, there has been no research on strategies for implementing CT as an approach to learning. This is very important to do in equipping students in facing the era of technological disruption where everything moves very fast. In addition, this CT approach is also predicted to be able to improve problem solving skills when viewed from the learning stage. Therefore it si important to know how to implement a CT approach for non-computer science students.

METHOD

This study involved 35 undergraduate students of physics education in the computational physics course with the research method used, namely the mixed methods explanatory sequential design (Creswell & Plano Clark, 2011) (Figure 1).

Figure 1. Research methodolgy

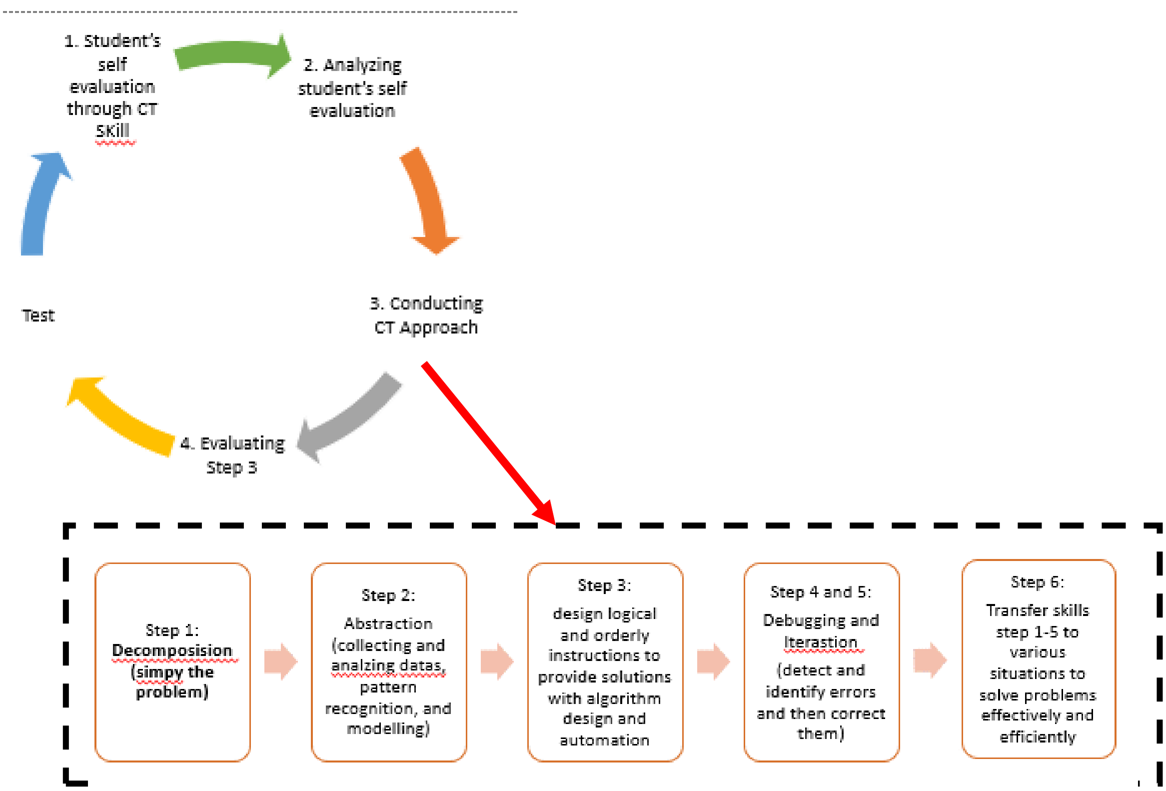


Figure 2. Step by step CT Approach implementattion

Broadly speaking, the flow of the mixed methods research method with the explanatory sequential design in this study includes quantitative data obtained from the student self-evaluation instrument understanding the CT approach stage. Next, collect qualitative data that is useful to help explain or elaborate on the quantitative results obtained from the Focus Group Discussion and In-depth Interviews. The final step of this method is the interpretation of data that has been obtained from the elaboration of quantitative and qualitative data (modified from Creswel, 2009).

Population and Sample

The population in this study was preservice physics teacher of faculty of teacher training and education. This study also involved 35 preservice physics teacher in Computational Physics subject.

Data Collection and Instrument

The instruments used in this study are questionaires, tests, and deep interview. Questionaires were distrbuted to the subject by online. Deep interview done to the subject by texting in whatsapp, and the last test were given to the subject before and after doing implementation of C Approach.

Data Analysis

The datas were analyzed quantitatively and qualitatively from questionaires, deep interview, and test.

RESULT AND DISCUSSION

The implementation of the CT approach to non-computer science education students is very interesting to be carried out with tretera steps in this research method. In detail the results of the research at each stage of its implementation are presented as follows:

Stage 1: Student self-evaluation of CT skills

At the first meeting, the researcher gave a physics problem that had to be solved using numerical methods. After that, students conduct self-evaluation through a questionnaire given via google form.

Stage 2: Analyze the results of self-evaluation

Based on the data in Figure 3, it can be concluded that in the components of decomposition and abstraction, more than 60% of students already really understand and understand. However, at a more advanced stage, namely algorithm design, debugging and iteration, more than 50% of students were categorized as confused and did not understand. So that these results will serve as a basic guide in implementing the CT approach. In this study, the iteration and debugging processes were combined, because according to the researcher this process could be run simultaneously and inseparably. In addition, in the sixth stage the CT approach, namely generalization, is carried out by giving students other similar problems (Figure 4).

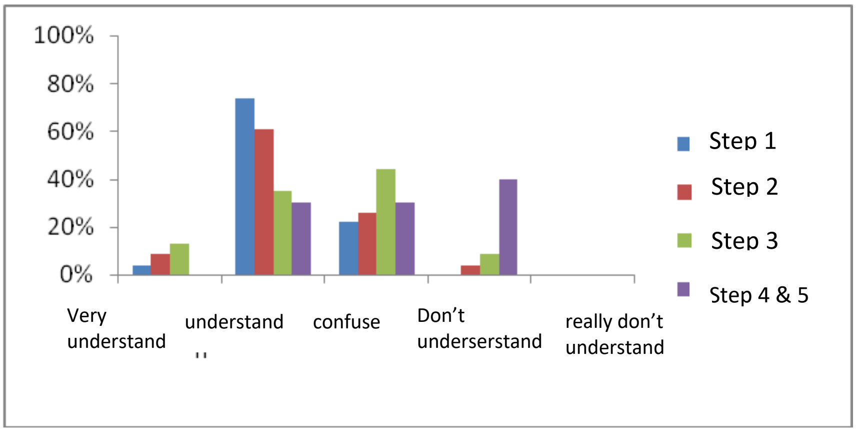


Figure 3. Student's mastery concept in pra implementation

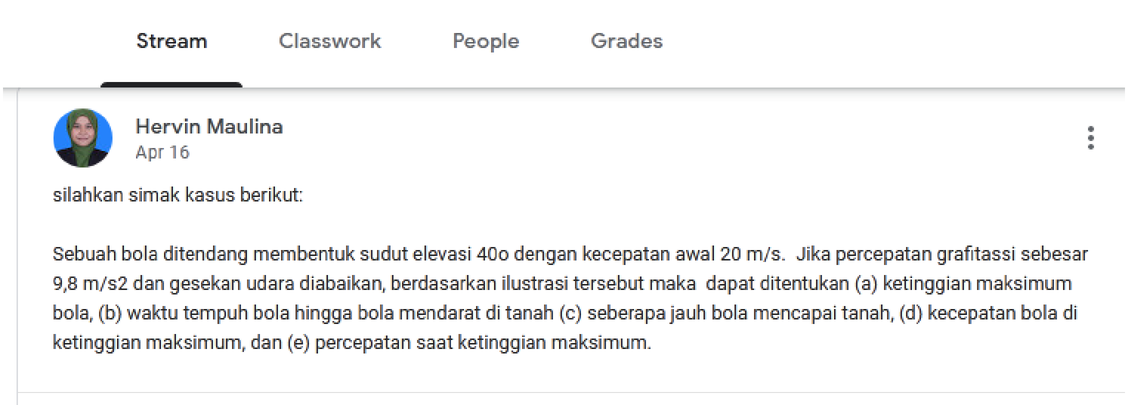


Figure 4. Lecturer gave another similar problem

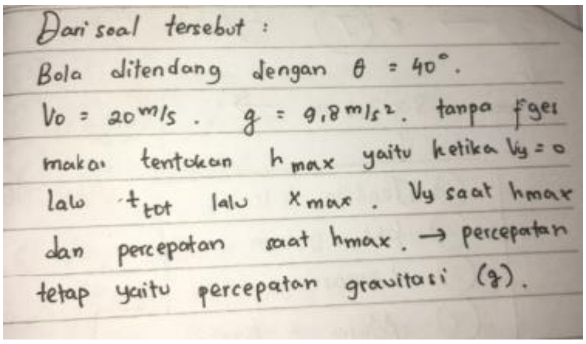


Figure 5. Student's decomposition result

Stage 3: Application of the CT approach

At this stage the lecturer provides 1 case of physics which must be completed with the CT approach steps and uploads the results of their work through Google Classroom. The results of student work have shown quite a significant development. The results can be described as follows:

1) Decomposition

At this stage students are asked to explain again and simplify the questions using language that they can easily understand (Figure 5). Based on the students' answers in Figure 5, it shows that students are able to explain and simplify the problems given by using their own language.

2) Abstraction

At this stage students try to collect important data that is known in the questions. Next they tried to analyze what physics concepts were applied to these problems, so that they could determine the equations used to solve the problem. The final stage in abstraction is visualizing the problems given using a particular model. The complete student work data is shown in Figure 6.

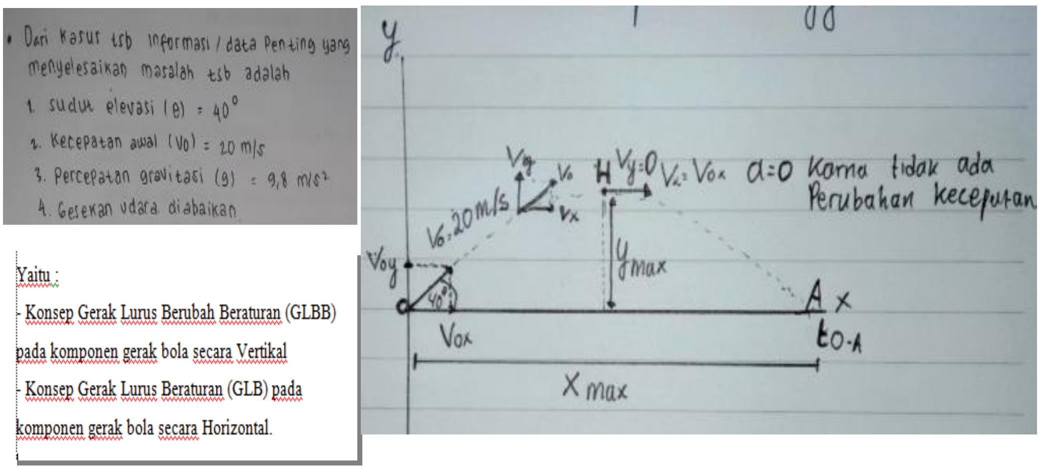


Figure 6. Step for data collecting, patten form, and modelling

3) Algorithm design

At this stage, students make solving algorithms, problem solving and coding in the Matlab software. The algorithm made by students (Figure 7) is in the form of a series of steps to solve the problem in the form of a flowchart. Each symbol on the flowchart describes the execution carried out on the system. The next step for students to make automated problem solving in the form of coding on the matlab as shown in Figure 8.

4) Debugging and iteration

At this stage students detect and identify errors, and then correct errors, when the solution does not work properly. This process is repeated until a suitable solution is found. The debugging process will appear on the right side of the editor screen in Figure 9. Coding that is predicted to be wrong will appear on the right side of the screen and will be colored orange. So that students can immediately improve.

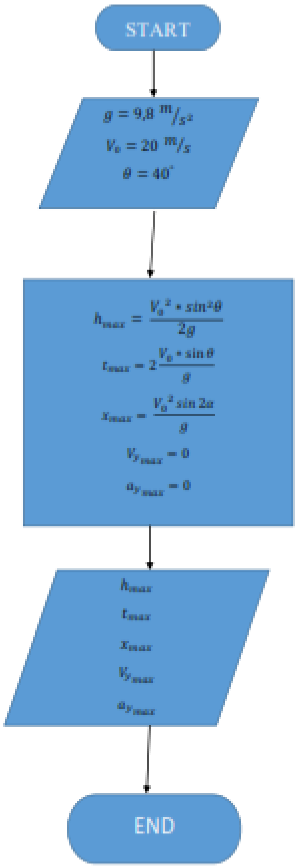


Figure 7. Student’s algorithm

Nama : Indah Ayu Wirastiti
NPM : 1813022011

```
clc; close;clear;
teta=input('Sudut elevasi= ')
v0=input('kecepatan awal= ')
g=9.8;
%konsep parabola
vx=v0*cos(teta)
vy=v0*sin(teta)
disp('a. ketinggian maksimum: ')
hmax=vy.^2./(2*g)
disp('b. waktu tempuh bola di udara: ')
txmax=2*v0*sin(teta)./g
disp('c. jarak tempuh bola dalam sumbu x: ')
x=vx*txmax
disp('d. kecepatan bola saat ketinggian
maksimum: ada 2 jenis kecepatan berdasarkan sumbunya,
yaitu v1 dan v2')
disp('pertama, kecepatan pada sumbu vertikal: ')
v1=vy-g*(0.5*txmax)
disp('kedua, kecepatan pada sumbu horizontal: ')
v2=vx
disp('e. percepatan saat ketinggian maksimum')
ghmax=g
```

Figure 8. Otomation using matlab software

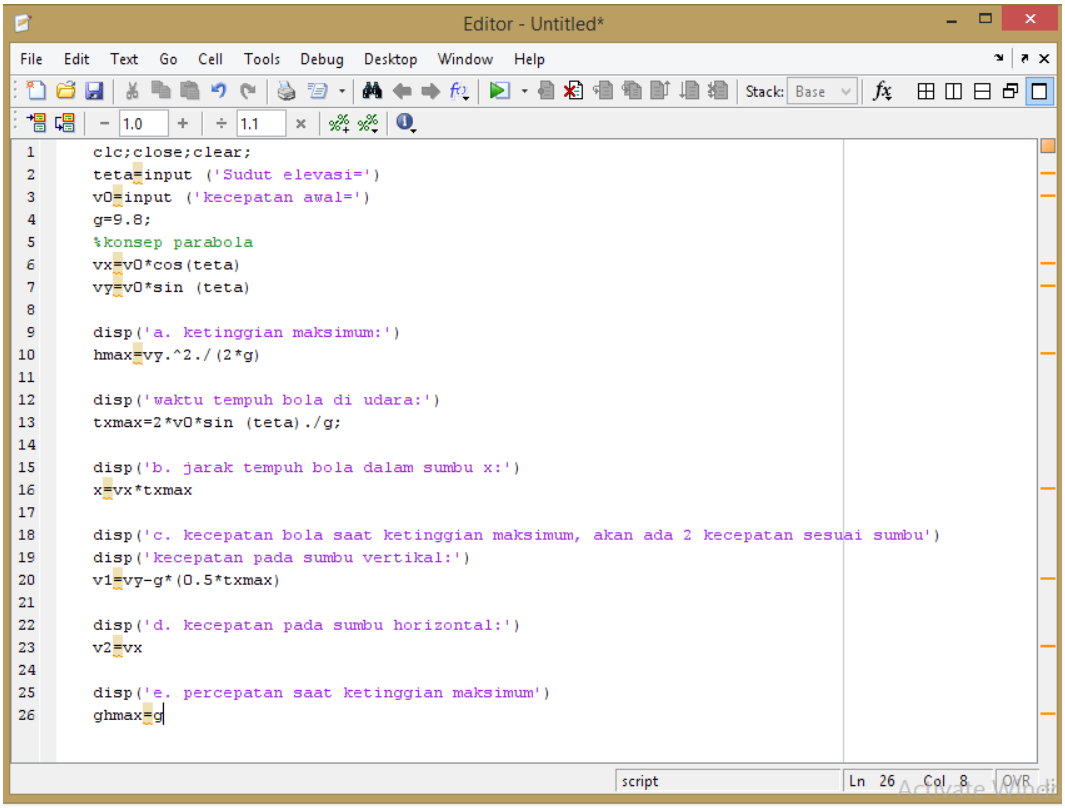
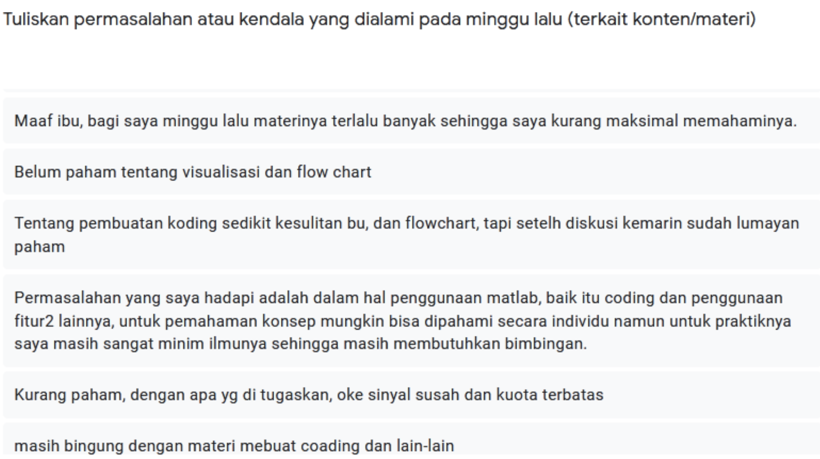


Figure 9. Debugging process

Stage 4: evaluation of the application of the CT approach

This fourth stage is carried out to evaluate the learning process and receive input and suggestions as well as obstacles experienced during learning. The resume of evaluation results conducted by students is presented in Figure 10.


Tuliskan permasalahan atau kendala yang dialami pada minggu lalu (terkait konten/materi)

Maaf ibu, bagi saya minggu lalu materinya terlalu banyak sehingga saya kurang maksimal memahaminya.

Belum paham tentang visualisasi dan flow chart

Tentang pembuatan koding sedikit kesulitan bu, dan flowchart, tapi setelh diskusi kemarin sudah lumayan paham

Permasalahan yang saya hadapi adalah dalam hal penggunaan matlab, baik itu coding dan penggunaan fitur2 lainnya, untuk pemahaman konsep mungkin bisa dipahami secara individu namun untuk praktiknya saya masih sangat minim ilmunya sehingga masih membutuhkan bimbingan.

Kurang paham, dengan apa yg di tugaskan, oke sinyal susah dan kuota terbatas

masih bingung dengan materi mebuat coading dan lain-lain

Figure 10. Learning evaluation by students

Stage 5: Test

In order to find out the effectiveness of this CT approach, students are then given a written test to independently create and solve physics problems (Examples of student work results are attached). The results of the evaluation of the test show that all students have been able to generalize the concept into other simple problems. The results of self-evaluation before and after the implementation of CT are presented in Table 1. Based on the results of data analysis in Table 1, it can be concluded that there has been an increase in the level of understanding of students pre and post implementation.

Table 1. Comparartion of pre and post test results

Activity/step	Very understand		understand		confuse		Don’t understand		realy don’t understand	
	pra	pasca	pra	pasca	pra	pasca	Pra	pasca	pra	pasca
Decomposition	4%	57%	74%	43%	22%					
Abstraction	9%	49%	61%	51%	26%		4%			
Algorithm design	13%		35%	29%	44%	63%	9%	8%		
Iteration and debugging			30%	14%	30%	72%	40 %	14%		

As explained in the literature review section, 4 steps in solving problems (problem solving) consisting of understanding the problem, devising a plan, carrying out the plan, and looking back (Polya, 1957) are contained in the implementation of the CT approach adopted from Shute. et al., 2017. May be it can be analyzed for further stdudy about the CT approach is able to fosters student problem solving skills.

CONCLUSION

Based on the results and discussion, it can be concluded that the Computational Thinking (CT) approach can be applied to non-computer science students in online learning. In addition, in this study there are findings that show that the CT approach can foster problem solving skills in non-computer science students.

ACKNOWLEDGEMENTS (OPTIONAL)

The authors would like say thanks to Lampung University who have given the fund to implement this study by using fund of BLU Universitas Lampung.

REFERENCES

Adjie, Nahrowi dan Maulana. (2006). Pemecahan Masalah Matematika. UPI PRESS. Bandung.

Anderson, N. D. (2016). A call for computational thinking in undergraduate psychology. Psychology Learning and Teaching, 15(3), 226–234.

Aunurrahman. (2011). Belajar dan Pembelajaran. Bandung: Alfabeta.

Bond, T. G., & Fox, C.M. 2007. *Applying The Rasch Model: Fundamental Measurement in the Human Sciences, 2nd Edition*. Lawrence Erlbaum Associates, Publishers. Mahwah, New Jersey. London.

Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. *American Educational Research Association Meeting*, 13–17 April 2012. Vancouver, British Columbia, Canada.

Cohen, J. (1988). Statistical power analysis for the behavioral sciences. New York: Academic.

Conklin, J. (2005). A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives Complete Edition.

Doleck, T., Bazelais, P., Lemay, D. J., Saxena, A., & Basnet, R. B. (2017). Algorithmic thinking, cooperativity, creativity, critical thinking, and problem solving: Exploring the relationship between computational thinking skills and academic performance. *Journal of Computers in Education*, 4(4), 355–369.

Fraenkel, J. R., Wallen, N. E., & Hyun, H. H. (2011). How to Design and Evaluate Research in Education. New York: McGraw-Hill Humanities/Social Sciences/Languages.

Grover, S., & Pea, R. (2013). Computational thinking in K–12: A review of the state of the field. Educational Researcher, 42(1), 38–43.

Gunbatar, M. S. (2019). Computational thinking within the context of professional life: Change in CT skill from the viewpoint of teachers. *Educational and Information Technologies*, doi.org/10.1007/s10639-019-09919-x.

Israel, M., Pearson, J. N., Tapia, T., Wherfel, Q. M., & Reese, G. (2015). Supporting all learners in schoolwide computational thinking: A cross-case qualitative analysis. Computers & Education, 82, 263–279.

Korkmaz, Ö., Çakir, R., & Özden, M. Y. (2017). Avalidity and reliability study of the computational thinking scales (CTS). *Computers in Human Behavior*, 72,558–569.

Linacre, J. M. (2010). Predicting responses from Rasch measures. *Journal of Applied Measurement*, *11*(1), 1-10.

Lye, S. Y., & Koh, J. H. L. (2014). Review on teaching and learning of computational thinking through programming: What is next for K-12? *Computers in Human Behavior*, 41,51–61.

Mohaghegh, D. M., & McCauley, M. (2016). Computational thinking: The skill set of the 21st century. International Journal of Computer Science and Information Technologies., 7(3), 1524–1530.

Moreno-León, J., Román-González, M. & Robles, G. (2018). On computational thinking as a universal skill: a review of the latest research on this ability. *IEEE Global Engineering Education Conference* (EDUCON 2018).18–20.

Polya, G. 1957. How To Solve It. Peinceton University Press.

Robertson, S. I. (2017). Problem Solving: Perspectives form Cognition and Neuroscience. London and New York: Taylor & Francis Group.

Román-González, M., Pérez-González, J. C., Moreno-León, J., & Robles, G. (2018b). Can computational talent be detected? Predictive validity of the computational thinking test. *International Journal of Child Computer Interaction*, 18,47–58.

Román-González, M., Pérez-González, J. C., Moreno-León, J., & Robles, G. (2018). Extending the nomological network of computational thinking with non-cognitive factors. Computers in Human Behavior, 80, 441–459.

Senin, S., & Nasri, N. M. (2019). Teachers' concern towards applying computational thinking skills in teaching and learning. International Journal of Academic Research in Business And Social Sciences, 9(1), 297–310.

Shute, V. J., Sun, C., & Asbell-Clarke, J. (2017). Demystifying computational thinking. Educational Research Review, 22,142–158.

Sumintono, B. dan Widhiarso, W. (2013). Aplikasi Model Rasch untuk Penelitian Ilmu-Ilmu Sosial. Jakarta: TrimKom.

Voogt, J., Fisser, P., Good, J., Mishra, P., & Yadav, A. (2015). Computational thinking in compulsory education: Towards an agenda for research and practice. *Education and Information Technologies*, 20(4), 715–728

Voogt, J., Fisser, P., Good, J., Mishra, P., & Yadav, A. (2015). Computational thinking in compulsory education: Towards an agenda for research and practice. Education and Information Technologies, 20(4), 715–728.

Weinberg,A.E.(2013).Computational thinking: An investigation of the existing scholarship and research (Doctoral dissertation, Colorado State University).

Widoyoko, E. P. 2011. *Teknik Penyusunan Instrumen Penelitian*. Yogyakarta: Pustaka Pelajar.

Wing, J. M. (2006). Computational thinking. Communications of the ACM, 49,33–35.

Wing,J.(2011).Research notebook: Computational thinking—What and why? The magazine of the Carnegie Mellon University School of Computer Science. The Link Magazine, Spring. Carnegie Mellon University, Pittsburgh. Retrieved from http://people.cs.vt.edu/~kafura/CS6604/Papers/CT-What-And-Why.pdf. Accessed 14 April 2019.

Yadav, A., 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.

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