Jurnal Pendidikan Teknologi dan Kejuruan, Vol. 25, No. 1, May 2019, 43-51
ISSN: 0854-4735, accredited by KEMENRISTEKDIKTI, Decree No: 51/E/KPT/2019
DOI: 10.21831/jptk.v25i1.22638

ANALYSIS OF PROGRAMMING SKILLS CONCEPT IN DEVELOPING PROBLEM SOLVING SKILLS

Rina Harimurti[1], Ekohariadi[2], Munoto[3], I Gusti Putu Asto B[4], Elizabeth Titiek Winanti[5]
[124]Department of Informatics Engineering, Universitas Negeri Surabaya, Indonesia,
[3]Department of Electrical Engineering, Universitas Negeri Surabaya, Indonesia,
[5]Department of Civil Engineering, Universitas Negeri Surabaya, Indonesia
E-mail: rinaharimurti@unesa.ac.id

ABSTRACT

Programming refers to a set of processes of writing, testing and correcting the program code aimed at creating a program that can do certain works in accordance with the programmer's instructions, it requires the mastery of algorithms, logic and programming languages. This study discusses the concept of programming skills to develop problem solving skills, which focuses on secondary school students. Various activities allow students to collaborate and develop problem solutions. Some previous studies suggest that students can obtain knowledge and programming skills through formal and non-formal education even though formal education shows better results of students' attainments. Given a basic programming skills, students are expected to have a high level of reasoning for their better problem solving skills to solve complex problems using computers through an integrated approach between thinking skills and computational thinking (CT). Computer programming has the effect of reasoning abilities for students, problem solving and self-efficacy. Currently, programming-based computation has been introduced in school curriculum at different levels. Consequently, teachers are required to teach relevant materials relevantly to the current curriculum.

Keywords: computational thinking, problem solving skills, programming skills

INTRODUCTION

Programming is a set of processes of writing, testing, repairing (debugging), and maintaining the code that constructs a particular computer program. This code is written in different programming languages. Moreover, programming is more than coding because it exposes someone to computational thinking that involves problem solving skills using computer science concepts, i.e. abstraction and decomposition [1].

Programming aims to create a program that can do calculations or other works in accordance with the programmer's instructions. To successfully perform it, several skills need to be mastered by a programmer including algorithms, logic, programming languages, and some Mathematics knowledge. Therefore, a programmer should appropriate the whole required skills and cannot negotiate one of them to be left.

In addition, programming is an art of using interconnected algorithms using a particular programming language, which further results on a certain computer program. Indeed, different programming languages support different programming styles, which is commonly called programming paradigm. Hence, given the basic programming skills, a programmer is expected to have higher reasoning level as well as his better problem solving skills.

This study discusses a concept of programming skills to develop problem solving skills. This study consists of four sections, namely the overviews relating to the basic principles of programming skills,

Received December 27, 2018; Revised March 15, 2019; Accepted April 24, 2019

the implementation of the skills to develop problem solving skills, the methods, and the conclusion, respectively.

Programming is not only about writing codes but also the capability to analyze a situation, identify key components, conduct modelling data and processes, and create or improve certain programs through a design approach. Because of its complexity, it is often conducted as a team-based task in professional settings [2]. It will teach someone how to express a mindset into a software form. By this far, children tend to be taught only about how to use a software. They do not learn about the concept of how a software is made and performed. By an early introduction of the programming concepts will lead to children's better reasoning and problem solving skills.

Programming is an aspect of computer science taught at schools considered as the most challenging. Various activities can be initiated to allow students collaborate and construct problem solutions [3]. For instance, some important programming procedures can be drawn as follows: (1) using algorithms, (2) inserting paired comments into program code, (3) developing program code from the algorithm in pairs, and (4) finding bugs in the program code. Reading and tracking program codes are also significant to support programming learning so that students perceive problem solving skills that are very important in programming. Pedagogical strategies in learning such skills are widely developed including but not limited to how teachers can support students in learning to read and write a program.

McDougall [4] conveys that there is a value that adopts the pedagogical approach in teaching informatics. She explains that students get knowledge and programming skills through either formal or informal ways. She, then, concludes that a group of fifth grade students are skilful in running the MicroWorlds multimedia programming and acquire both programming skills and techniques from various sources outside classroom activities. She also states that the formalization of the student model starts from searching various human resources and obtaining programming skills from formal and informal ways.

The implementation of computer-based services requires problem solving, designing skills, logical thinking skills, understanding how computers and networks operate, and programming competencies. Those abilities are simply considered as a set of programming skills. A study conducted in Finland finds and classifies three approaches and methods of learning programming skills such as a formal education by school curriculum, informal education by online practices and off-school courses, and informal forums (*hackathon*) [5].

Sumartini [6] states that problem solving skills is a process to overcome difficulties faced to achieve the expected goals. In the field of computer science, students must master problem solving skills to solve programming problems such as algorithms, logic and programming. There are five strategic procedures related to performing problem solving skills involving (1) presenting a problem clearly so that it does not confront an ambiguous meaning, (2) asserting the problem clearly so that it does not remain unclear, (3) compiling alternative hypotheses and procedures that are expected to be used to solve the problem, (4) testing hypotheses and performing actions to obtain solutions (e.g. data collection and data processing), and (5) if one solution is obtained, the next step is to re-check and to choose the best solution.

Problem solving can be interpreted as a goal, a process, and a set of skill. Problem solving as a process means as an activity that prioritizes the importance of procedures, steps of the undertaken strategy to troubleshoot problems. Creative competence and problem solving are part of the 21st century skills. The use of creative digital technology to solve problems is related to computational thinking as a series of cognitive and metacognitive strategies where students are involved in active designing, creation, and mobilization processes of computational methods [2].

Zubaidah [7] states that problem solving skills include some other skills such as identification and the ability to search, select, evaluate, organize, consider various alternatives, and interpret information. One must be able to find various solutions from different perspectives in solving multilayered problems. Problem solving requires effective and creative collaboration between teachers and students to be able to use technology, handle massive information, define and understand the elements contained in the subject matters, identify sources of information and strategies compulsory to solve problems. In addition, it cannot be separated from critical thinking skills that are fundamental skills in generating solvencies Students should also be able to apply the right tools and techniques effectively and efficiently in solving problems.

The important thing related to problem solving is that, to solve multifaceted problems using computers, integrating cognitive and critical thinking skills remains necessary. The problem solving skills cover four main principles: (1) how to solve problems and to design systems referring to fundamental concepts in computer science, (2) creating and using various levels of abstraction to understand and solve problems more effectively, (3) thinking algorithmically along with the ability to apply mathematical concepts efficiently, and (4) understanding the consequences of scale, which is not only for sake of efficiency but also economic and social considerations [8]. Moreover, computational thinking consists of several elements such as computing, communication, coordination, recollection, automation, evaluation, and designing. Figure 1 shows an overview of the principle of computational thinking.

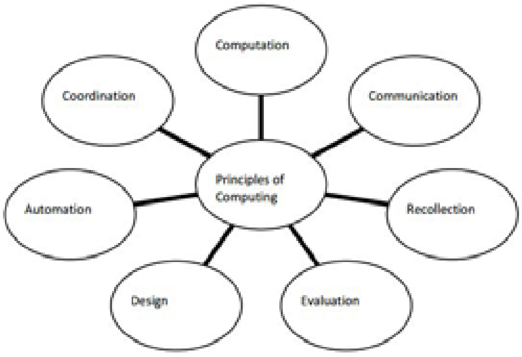


Figure 1. Seven Principles of Computational Thinking

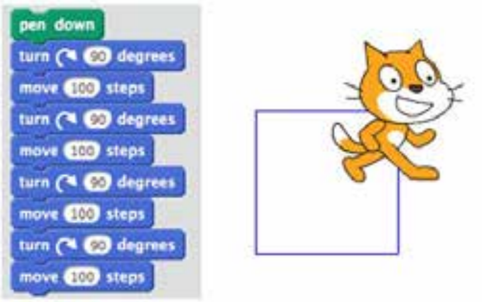
Sarantos & Kallia [9] investigate whether computer programming has influenced high school students' reasoning abilities, problem solving skills, and self-efficacy in Mathematics. They suggest that there are significant differences in students' reasoning skills who participate in the programming class compared to those who are not involved in the class. Their findings then imply that formal education of computer programming has a significant influence on generating students' problem solving skills.

Computation has been introduced in school curricula in many countries. Teachers might generate challenges and strategies used to teach computation. A study conducted in England in 2014 portrayed that teachers were preparing to include computation into the compulsory curriculum [3]. That study concluded that there were five approaches that could be undertaken during teaching and learning process, namely programming activities, task contextualization, collaborative tasks, developing computational thinking, and programming skills. Due to curriculum changes, teachers are required to teach relevantly to the development of the curriculum by coping with the introduction of the new contents. They need to learn any appropriate pedagogic technique to carry out the new materials, especially for the aspects of computer science related to programming algorithms and the development of computational thinking skills. The latest literature relating to computer science education in schools highlights a number of ways to make computer science concepts accessible, interesting, fun, and, more importantly, giving in-depth conceptual understanding.

Programming education can be used as a medium to build knowledge and modelling tools to involve students in creative problem solving activities [2]. When students are involved in such activities, they can improve their problem solving skills. While some utilizations of technology might involve students in passive situations where there is almost no room to produce knowledge. On the other hand, the use of technology can involve students in building creative knowledge, of which the process indicates that technology can be aimed to improve co-creative learning process. There are five levels of students' co-creative learning process in computer programming namely (1) passive exposure to teacher-guided explanations, videos, or tutorials about programming; (2) step-by-step procedural programming activities where there is no potential space for students' creativity; (3) original content creation through individual programming; (4) original content creation through team-based programming, and (5) shared knowledge creation through programming.

Another study investigated the effects of Scratch and Alice tools on the teaching and learning process of programming comparatively by considering student involvement, reflective thinking and problem solving skills, and computational thinking skills. The results showed that the teaching of Scratch programming more influenced students' involvement and reflective thinking skills for generating problem solving than Alice. It had been stated that teaching with Alice influences students' computational thinking skills [10]. The research was conducted based on the observation in which the students had various difficulties in programming learning such as a complex process to require numbers of skills. Students tent to have troubles in understanding program code of the programming process. The phenomenon implies attention to the importance of programming tools in programming languages [11]. Considering these problems, it is clear that using tools to help students understand the logic of programming and the ability to think relevantly to the twenty-first century becomes an easier way to contribute to draw solutions of confronted problems experienced in indispensable programming.

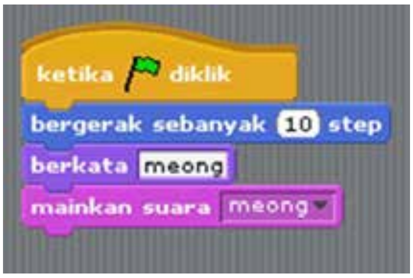
In the discussion above, Scratch programming has a positive impact because it is a free and user-friendly programming language. Up to now, programming languages are known as something that is difficult to learn even many who believe only smart people can operate it. However, this assumption is no longer appropriate because a relatively easy-to-learn programming language has been developed named Scratch programming [12]. Scratch is a new programming language released in May 2007. It supports the development of computer games, interactive stories, graphics, computer animation, and other multimedia projects. Figure 2 depicts the example of a picture using Scratch programming language.



```
pen down
turn ↻ 90 degrees
move 100 steps
turn ↻ 90 degrees
move 100 steps
turn ↻ 90 degrees
move 100 steps
turn ↻ 90 degrees
move 100 steps
```

Figure 2. Drawing a Rectangle Using Scratch

Scratch programming does not use text form but in graphic to put the used codes. Graphic forms are made in such a way as to form pieces of puzzles known as code blocks. Making a Scratch program is like compiling a puzzle. Code blocks can only be arranged if they are interconnected. Thus, the program is distant from syntax errors. Moreover, in Scratch, there are three types of code blocks in which each can generate a relation. There are more than 100 Scratch block codes that are included in eight code block categories.



```
ketika 🏁 diklik
bergerak sebanyak 10 step
berkata meong
mainkan suara meong ▾
```

Figure 3. Block Script Used as the Basis for Writing Scripts

A Scratch application consists of many sprites, of which each of them has several scripts, costumes, and types of sounds. Scripts are programs for creating an application. Costume is an image that will be displayed, while the types of sounds include sounds that will be played in the application. The program results will be shown in an area called stage.

Storing Scratch data can use code block variables. Similar to other programming forms, variables in Scratch can be created as either local or global variables. The code block variables can be used to control the application performance. The value of the code block variables can be displayed on the stage. If no longer needed, code block variables can be deleted. Whereas, for the sake of digit manipulation, Scratch has a number of code block that consists of several blocks of code for mathematical operations, i.e. arithmetic and logic.

To support the running process of the application, Scratch provides control blocks used for looping and conditional processes. In addition, it can also be used to control and coordinate script activities by sending and receiving messages between sprites. Learning using Scratch programming language will make it easier for students to understand the right programming concepts and ensure that novice programmers learn the right way to compile and formulate programming logic. Since Scratch is not a text-based programming language, it does not need to follow a complex set of syntax rules. This is because writing errors that do not comply with the programming language syntax rules will make the application unable to run. Scratch simplifies application development by using the same logic and basic programming concepts as other programming languages.

Another study conducted by Xiao-Ming [13] tested a new method regarding the importance of a computer programming concept to the students that underlies the development of problem solving skills. Computer programming is a subject that requires a strategy to teach students the ability to solve problems and involves a large number of programming logic activities that become a challenge for students. Therefore, students should be provided supports and tutorials for computer programming.

Akcaoglu [14] conveys that computer programming can be used to teach students about problem solving. In her study, students were required to perceive designing and problem solving skills, so that, the assessment was performed under the aspects of system analysis and design, decisision making, and problem solving using case-study approach on game design. The results showed that there was a significant influence of game designing on students' learning and problem solving skills. The computer learning innovation involveda collaborative problem-solving learning model to reveal students' problem solving skills, especially in some schools in Shanghai [15]. The implementation was confronted by an obstacle on how to develop the students' skills. Thus, a collaborative learning was used to deal with the problem, of which there were two models permeated namely social interaction and problem solving model using cognitive and metacogntive approach. The study was conducted in two months with 59 students who came from two different classes. The results depicted that the experiment class using collaborative group model had higher problem solving skills compared to the control class.

The new undertaken approach is collaborative learning which is believed to be an effective approach to increase student's interaction and introduce a conducive learning environment that provides new experiences. Interactions in collaborative learning do not occur automatically. Without strategies, proper guidance, and appropriate supporting tools, the results of this collaborative learning approach will not show satisfactory results. There are two classes with small samples used as research participants. One class consists 25 students randomly selected for the experimental class to implement collaborative learning using posing-based problem strategies. Another class consists 28 students as the control group, who learn with conventional collaborative learning activities.

The results show that the use of posing-based problems shows a significant impact on students' learning achievement, especially on programming and problem solving skills. In addition, the results also show that student collective efficacy in learning increases and the cognitive load of students decreases compared to those who learn with conventional collaborative learning approaches. This approach is done by giving instructions to the control group with programming tasks to solve given problems. After that, the results of the program are corrected by students themselves and feedbacks given by the teacher. Whereas, students in the experimental group are given treatment with examples of tasks at the beginning of learning activities. The tasks are given with different problems among several groups of students to provide opportunities for them solve problems with the other groups. Such activities strengthen students' active participation and responsible attitudes for learning process, knowledge construction, problem creation, problem solving, and reflection. This is the reason why the experimental group is better than the control group in terms of learning achievement and lower cognitive load. However, the limitation of this study is that the sample size is relatively too small, so that the results do not allow generalizations to represent cases with large sample sizes and the approach only focuses on programming content. Consequently, the strategic impact cannot necessarily be generalized to other learning content related to computer science. Further explanations portray that positive results will be achieved if students are given assignments related to student learning content and guide them to be actively involved in construction, creation, and reflection activities to improve problem solving skills [16].

Students should master problem solving skills to compete and accelerate their self-competence in digital world in which so many technology is controlled by the use of various softwares [17]. That is, problem solving skills should always be interconnected with computational thinking skills especially on innovative education.

METHOD

Literature studies were used to compose the present study by collecting relevant theoretical references to the problems proposed. The literature studies were mostly in a form of surveys and discussive literatures in certain fields depicting a brief concept, of which each part of the concept is used as a related source to other studies in showing the chronological or thematic linkages. In preparing this study, a thematic structure approach was used in which the obtained references had been clustered according to the theme or topic. This method was shorter in term of the organization.

References used in the present study were in the form of books, articles, and journals that contained theory, analysis, and research that had been conducted related to programming and problem solving skills. The journals, articles, and books used as sources were published from 2009 to 2018.

The references used can be categorized as follows: basic programming skills concepts, the importance of programming skills given to students, related curriculum to programming and problem solving skills, and the importance of programming skills in developing problem solving skills. The keywords used in the search to search the references were programming and problem solving skills.

CONCLUSION

Programming has been introduced in the school curriculum in several developed countries. Different programming languages support different programming styles, which is called as programming paradigm. An early introduction of programming concepts to the students will impact on their reasoning level, of which the result will lead to perceiving high problem solving skills. By giving the basis of programming skills, students are expected to have a high reasoning level so that the problem solving skills are better. Four important things related to programming include (1) using algorithms in programming, (2) inserting paired comments into program code, (3) developing program code from the algorithm in pairs, and (4) finding bugs in the program code. The implementation of computer-based services requires problem solving skills, designing skills, logical thinking skills, and understanding of how computers and networks operate. Those skills can be considered as a set of programming skills. Studies related to programming skills have been done in many countries, one of which is in Finland. The results of the study suggest three approaches for learning programming skills namely formal education, informal education, and informal forums.

Problem solving can be interpreted as a goal, a process, and skills. Creative competence and problem solving skills are part of the 21st century skills using digital technology as a tool to solve problems related to CT. In many countries, computational thinking has been permeated within school curriculum to make a challenge for teachers and demand teaching strategies. Teachers are required to conceive competency of lesson planning, benefitting learning sources, using learning media, and facilitating learning feedbacks for better curriculum implementation and development [18].

One strategy that can be implemented is the use of tools to help students understand the programming logic and the ability to think relevantly to the twenty-first century in an easier way and to contribute to the solution of problems experienced in programming. The use of computer might be helpful as learning media or a computer programming that can enhance problem solving skills. Results show that the increasing students' learning attainment is influenced by conducive learning environment and atmosphere that can also attract students' learning eagerness [19]. The results showed that the teaching of Scratch programming more influences students' involvement and reflective thinking ability for problem solving skills than Alice programming.

Another result of research supports the concept of programming skills using posing-based problem strategies. This strategy shows significant results to improve student's learning achievement, especially in programming skills, and to enhance students' problem solving skills. By the application of this strategy, students can perceive collective efficacy in conducting collaborative learning that increases cognitive skills compared to those who learn with conventional collaborative learning approaches. Moreover, students can be assisted by the use of the strategy and introduced to a new learning strategy that can lead them to master problem solving skills.

REFERENCES

[1] S. Y. Lye and J. H. L. Koh, "Review on Teaching and Learning of Computational Thinking through Programming: What is next for K-12?," *Comput. Human Behav.*, vol. 41, no. C, pp. 51–61, Dec. 2014.

[2] M. Romero, A. Lepage, and B. Lille, "Computational Thinking Development through Creative Programming in Higher Education," *Int. J. Educ. Technol. High. Educ.*, vol. 14, no. 1, p. 42, Dec. 2017.

[3] S. Sentance and A. Csizmadia, "Computing in the Curriculum: Challenges and Strategies from A Teacher's Perspective," *Educ. Inf. Technol.*, vol. 22, no. 2, pp. 469–495, Mar. 2017.

[4] A. McDougall and M. Boyle, "Student Strategies for Learning Computer Programming: Implications for Pedagogy in Informatics," *Educ. Inf. Technol.*, vol. 9, no. 2, pp. 109–116, Jun. 2004.

[5] P. Tuomi, J. Multisilta, P. Saarikoski, and J. Suominen, “Coding Skills as a Success Factor for a Society,” *Educ. Inf. Technol.*, vol. 23, no. 1, pp. 419–434, Jan. 2018.

[6] T. S. Sumartini, “Peningkatan Kemampuan Pemecahan Masalah Matematis Siswa melalui Pembelajaran Berbasis Masalah,” *Mosharafa J. Pendidik. Mat.*, vol. 5, no. 2, pp. 148–158, Aug. 2018.

[7] S. Zubaidah, “Keterampilan yang Diajarkan melalui Pembelajaran,” in *Seminar Nasional Pendidikan dengan tema su-isu Strategis Pembelajaran MIPA Abad 21*, 2016.

[8] J. J. Lu and G. H. L. Fletcher, “Thinking About Computational Thinking,” *ACM Sigcse Bull.*, vol. 41, no. 260–264, 2009.

[9] S. Psycharis and M. Kallia, “No Title,” *Instr. Sci.*, vol. 45, pp. 583–602, 2017.

[10] H. Yildiz Durak, “The Effects of Using Different Tools in Programming Teaching of Secondary School Students on Engagement, Computational Thinking and Reflective Thinking Skills for Problem Solving,” *Technol. Knowl. Learn.*, Aug. 2018.

[11] O. Berge, R. E. Borge, A. Fjuk, J. Kaasbøll, and T. Samuelsen, “Learning Object-Oriented Programming,” in *Norsk informatik konferanse*, 2016.

[12] Ekohariadi, I. F. Kurniawan, and R. E. Putra, *Pemrograman Visual Menggunakan Scratch*. Surabaya: Unesa University Press, 2015.

[13] X.-M. Wang and G.-J. Hwang, “A Problem Posing-Based Practicing Strategy for Facilitating Students’ Computer Programming Skills in the Team-Based Learning Mode,” *Educ. Technol. Res. Dev.*, vol. 65, no. 6, pp. 1655–1671, Dec. 2017.

[14] M. Akcaoglu, “Learning Problem-Solving through Making Games at the Game Design and Learning Summer Program,” *Educ. Technol. Res. Dev.*, vol. 62, no. 5, pp. 583–600, Oct. 2014.

[15] X. Gu, S. Chen, W. Zhu, and L. Lin, “An Intervention Framework Designed to Develop the Collaborative Problem-Solving Skills of Primary School Students,” *Educ. Technol. Res. Dev.*, vol. 63, no. 1, pp. 143–159, 2015.

[16] C.-L. Lai and G.-J. Hwang, “An Interactive Peer-Assessment Criteria Development Approach to Improving Students’ Art Design Performance Using Handheld Devices,” *Comput. Educ.*, vol. 85, no. C, pp. 149–159, Jul. 2015.

[17] M. Roman-Gonzalez, J.-C. Perez-Gonzalez, and C. Jimenez-Fernandez, “Which Cognitive Abilities Underlie Computational Thinking? Criterion validity of the Computational Thinking Test,” *Comput. Human Behav.*, vol. XXX, pp. 1–14, 2016.

[18] S. Waluyanti, D. Santoso, M. Munir, B. Wulandari, and A. Raduan, “Professional Development For Vocational High School Teachers Through Incremental Teacher Competence,” *J. Pendidik. Teknol. dan Kejuru.*, vol. 24, no. 2, pp. 278–285, 2018.

[19] U. Haryanto, “Peningkatan Kemampuan Memecahkan Masalah melalui Media Komputer dalam Pembelajaran Matematika pada Siswa SMKN 1 Ngawen,” *J. Pendidik. Teknol. dan Kejuru.*, vol. 22, no. 4, p. 432, Jan. 2016.