TÜRK FEN EĞİTİMİ DERGİSİ
Yıl 15, Özel Sayı, Aralık 2018

Journal of
TURKISH SCIENCE EDUCATION
Volume 15, Special Issue, December 2018

http://www.tused.org

Adopting Educational Robotics to Enhance Undergraduate Students' Self-Efficacy Levels of Computational Thinking

Feri Ardiana ARISTAWATI[1], Cucuk BUDIYANTO [2], Rosihan Ari YUANA[3]

[1] Universitas Sebelas Maret, Surakarta-INDONESIA,
[2] Dr., Universitas Sebelas Maret, Surakarta-INDONESIA , ORCID ID: 0000-0001-7288-3605
[3] Universitas Sebelas Maret, Surakarta-INDONESIA, ORCID ID: 0000-0001-7311-1105

Received: 20.06.2018 **Revised:** 20.09.18 **Accepted:** 15.12.2018

The original language of article is English (v.15, n. Special Issue, December 2018, pp. 42-50. doi: 10.12973/tused.10255a)

ABSTRACT

Computational Thinking (CT) skill, as an essential 21st-century skill, is an important problem-solving and survival abilities in the era of disruption. Universal principles generating a pattern of abstraction develop step-by-step troubleshooting instructions in solving similar problems, perceiving similarities/differences between the patterns, and making a complex problem solvable. The skill could be applied to various engineering fields by emphasising efficiency, accuracy, and capability of problem-solving. Recently, robot enthusiasts in Science, Technology, Engineering and Math (STEM) learning that involves 'assembling, programming and testing' activities, underscore CT skills. This research explores the CT pattern along with participants-developed robotics activities.

Keywords: Computational Thinking Skill, educational robotics, self-efficafy, STEM Learning.

INTRODUCTION

Computational Thinking (CT), which is an important 21st-century skill (Wing, 2006), devotes to the field of computers and almost all related disciplines. These skills, which include the ability to identify universal principles generating abstraction patterns, develop step-by-step troubleshooting instructions in solving the same problem (algorithm thinking), recognising the similarities or differences between patterns, trends, and making data, processes or problems (complex) solvable (decomposition).

Although this topic has been taken a considerable attention recent years, little research and literature have inquired on how to deal with Computational Thinking Skills and how to employ strategies developing those abilities (Brennan and Resnick, 2012). Wings (2011) and Bocconi et al. (2016) only discuss and outline computational thinking, usability, and examples of CT through solving real-world problems. Some have used Scratch for CT, but they have only studied with limited programming and samples (Brennan and Resnick, 2012). A complex CT study with robotics examined how to integrate CT skills into junior and senior

Corresponding author e-mail: cbudiyanto@staff.uns.ac.id

© ISSN:1304-6020

high schools (Atmatzidou and Demetriadis, 2016). The research was an inspiration for robotics learning with constructivism approach at higher education level.

Jean Piaget (1972, cited by Ackermann 2001) states that humans learn not only from the transmission of knowledge but also actively engaging in an activity or experience-based knowledge. Papert (1980) adds that understanding knowledge will be more effective if learners construct their knowledge-related products throughout their own experiences. In Papert's view of constructivism, practical (hands-on) learning uses tangible and intangible objects. Constructivism itself includes learner's active engagement so that (s)he builds an understanding or meaning on what (s)he has done.

Given the 21st-century skills, the researchers proposed robotics learning in designing, constructing, programming, and controlling the tangible objects. Because constructivist learning approach emphasizes a student-centred learning and doing science via guided inquiry activities, some activities, in hand, are intended to provoke computational thinking skills. This research explores the computational thinking pattern along with participants-developed robotics activities. The current study purposes to stimulate the development of computational thinking skills using a constructivist learning approach.

CONCEPTUAL BACKGROUND

Computational Thinking

Computational thinking, which is a problem-solving method commonly used by computer scientists (Eguchi, 2014), advocates the 21st-century skills (Wing, 2006). Computational thinking skill is devoted to mainly the field of computers and almost all related disciplines. This ability pays attention to the utilisation of technology and underlying concept of technology creation.

Peyton Jones (cited in Bocconi et al. 2016) states that computational thinking skill develops patterns of real-world problems as well as designing, developing, refining and explaining how computing technology works. Computational thinking can be developed by integrating it into the compulsory education curriculum. Thus, learners are expected to view the problem from different perspectives, express themselves with various media, and analyse daily problems. Also, developing a CT promotes the future of economic growth, employment in the field of computers and work-preparation (Bocconi et al., 2016).

Educational Robotics

Robotics and automation machines have entered the fields of education and industry. However, the introduction of robotics in education is usually only limited to the technological sophistication of the robot itself (Alimisis, 2012). The robot can be seen as an excellent educational tool to achieve learning. Karim and Mondada (2015) address that modern robotics has been used for such technical and non-technical learning as mathematics, physics, science, language, and music.

Robotics, which is a new learning media to relay knowledge, teaches reality and knowledge to learners (Miglino, Lund, and Cardaci, 1999). Robotics is one of the best learning technologies and media that can integrate knowledge, skills, and attitudes within each other (Miglino et al. 1999). Robotic activities include to design, build, operate, and use robots and computers as control, sensors, and information processing. Robotics is a computational thinking material to meet technological requirements including intelligence and embodiment. Robotics interacts learners with media (Catlin and Woollard, 2014).

Using robotics, which encourage learners to construct their own robots, will introduce new, creative and innovative technologies to learners. Learners will also have a mindset to become active technology/science creators rather than passive technological consumers (Eguchi, 2014).

Constructivist Learning Theory

Constructivist learning theory proposed by Piaget (1972) underpins robotics inspiration as a learning medium. It is argued that learners interpret the theory very well if they apply it to real-life issues or problems through creation and innovation. A constructivist learning model explores student's knowledge by manipulating and constructing tangible objects. The learning activity enables learners to interpret new knowledge with their pre-existing one. In view of Papert's constructivism theory and Vygotsky's socio-cognition approach, robotics activity makes learners to actively build a piece of new knowledge by collaboratively developing critical mental skills with their peers.

Previous Researches

Table 1 summarises the findings from previous researches.

Table 1. *Findings of previous researches*

Title	Author	Main Findings
Advancing students' computational thinking skills through educational robotics: A study on age and gender relevant differences.	Atmatzidou, S., and Demetriadis, S. (2016)	Male and female students had the same achievement, but female students needed more extended training. Also, the activity increased their compulsory thinking skills.
Robotics in education & education in robotics: Shifting focus from technology to pedagogy	Alimisis, D. (2012)	Teachers felt that they were active in utilising technology and designing the rich technological learning. Junior high school students understood the concepts of informatics and kinematics and felt the pleasure while building a robot.
Robotics as an educational tool	Miglino, Lund, and Cardaci (1999)	The related literature revealed two ways to explore computing thinking skills in educational contexts, such as (a) generate system-based interactive knowledge (i.e., internet services, multimedia encyclopedias, etc.) (b) build a simulation laboratory to gain knowledge by following scientific research method(s) (e.g., formulating and testing hypotheses).
Computational thinking and tinkering: Exploration of an early childhood robotics curriculum	Bers, Flannery, Kazakoff, and Sullivan (2014)	The results showed a high achievement level while assembling the robot. However, the achievement level decreased during programming and rose again on conditional programming.
New frameworks for studying and assessing the development of computational thinking.	Brennan, and Resnick (2012, April)	The 8-16 aged children in the Scratch community were accustomed to use such computational thinking skills as sorting code, understanding the patterns (looping) and thinking about the problem-solving steps.
Robotics as a learning tool for educational transformation.	Eguchi (2014a)	Learning robotics effectively involves students in learning STEM concepts, coding, computational thinking and engineering skills that students will need to work in the future.
Educational robots and computational thinking.	Catlin, and Woollard (2014)	The results of robotics have a strong symbiotic relationship with the 'computational thinking' skill. Robotics is also a practical form of computational thinking.

METHODS

Through a qualitative approach, the current study investigated participants' experiences of educational robotics. The participants experienced robotics activities with LEGO Mindstorm EV3 for its sophisticated visual and interactive programming interface. The LEGO Mindstorm also added the advantages of robotics to constructivist learning theory (Kamal, Budiyanto, & Efendi, 2018). Since participants were relatively ancient to modular robotics, the robotics activities began with a brief introduction to LEGO models and modules. Then, they built their robot models using LEGO Mindstorm EV3. During the activities, the researchers simultaneously observed participants' behaviours and conducted semi-structured interviews after all participants completed their sessions.

a) The Sample of the Study

Eight undergraduate students purposefully (Creswell, 2014) drawn from the department of informatics voluntarily participated in the current study. All participants had Informatics Education background and were familiar with programming. As seen in Table 2, they represented four different grades and gender equity. Even though most of them had attended programming courses, they had no previous experience with robotics.

Table 2. *Demographic features of the sample of the study*

Student	Semester	Gender	Programming Lessons	Robotic Experience
Student A	8th semester	Male	Nine times	No
Student B	8th semester	Female	Eight times	Yes
Student C	6th semester	Male	Ten times	No
Student D	6th semester	Female	Six times	No
Student E	4th semester	Male	Five times	No
Student F	4th semester	Female	Five times	No
Student G	2nd semester	Male	2 times	Yes
Student H	2nd semester	Female	2 times	No

b) Setting

This research initially administered a questionnaire to assess the participants' computational thinking proficiencies. The second part included the implementation of the LEGO Mindstorm EV3 as the robotics activity (i.e., robot assembly, programming, and testing) asking them to build a simple model. While participants engaged in the robotics activities, the authors observed the participants' behaviours and took observation notes. In the third part, the participants were interviewed about their experiences of the robotics activity and progress. The interview protocol and codes were analyzed according to the computational thinking skills (see Table 3).

Table 3. *Indicators of the in-depth interview (adopted from Atmatzidou and Demetriadis, 2016, p. 664)*

CT Skills	Description	Students' Skills (Students are able to)
Abstraction	Addresses a creating process from simple to complicated by taking out irrelevant details, finding relevant patterns, and separating ideas from concrete details. The essence of CT is an abstraction (Wing, 2008).	1. Separate important information from unnecessary one. 2. Analyze and specify common behaviours or programming structures between different scripts. 3. Identify abstractions between different programming environments.
Generalization	Transfers a problem-solving process to a wide variety of problems	Extend an existing solution in a problem to yield more possibilities/cases.
Algorithm	Practically writes step-by-step specific and explicit instructions for carrying out a process. Selecting appropriate algorithmic	1. Explicitly state the algorithmic steps. 2. Identify different useful algorithms for a given problem.

	techniques is a crucial part of CT (Kazimoglu et al., 2012).	3. Find the most efficient algorithm.
Modularity	Develops autonomous processes that encapsulate a set of common commands performing a specific function. It might be used in the same or different problems	Develop autonomous code sections for the same or different problems.
Decomposition	Breaks down problems into smaller parts that may be more easily solved. CT uses decomposition while attacking or designing a massively complex task (Wing, 2008).	Break down a problem into smaller/simpler parts that are easier to manage.

c) Data Analysis

After data collection, data reduction was conducted to select relevant and meaningful data. Hence, the authors focused on data leading to problem-solving, discovery, meaning or research questions. That is, the findings answering research questions were handled. Data from in-depth interviews and observations were trialled and analyzed using interactive analysis (Miles and Huberman, 1994). The data were presented to describe recent circumstances purposefully. The early phase of the research progressed as a temporal insight and understanding until a theoretical saturation achieved.

RESULTS

The second part of the research contained to elicit the participants' programming-related barriers. Table 4 summarizes the results of the interview protocols.

Table 4. *The Participants' Programming-related-Barriers*

Students	Algorithm	Syntax	Complex Algorithm
Student A		X	
Student B	X		
Student C		X	
Student D			X
Student E	X		
Student F		X	
Student G	X		
Student H			X

The students' programming barriers consisted of Algorithm, Syntax, and Complex Algorithm. The algorithm is defined as the first step to think how the program should be constructed that requires logical thinking. The syntax, which is perceived as the problem to express a new programming language, compiles an algorithm. Finally, the complex algorithm is the activity with merging loops and conditionals.

The student E concerned a need to think about the appropriate algorithm. The students B and G experienced a similar problem. *"I have difficulty thinking about algorithms. (I) need a long time and have to repeat the questions over and over again. (It is frequent that) I am (ended up) asking friends for help."*

Students A, C, and F faced the difficulty to express the appropriate Syntax. The Student A mentioned about a problem a new programming language: *"I can compile an algorithm, but when applying to the code, it sometimes takes time."* Similarly, Student C referred to the following expression: *"Programming is not a difficult thing (to deal with), (to) write the codes that require mastery (in programming)."*

An increase in complex assignment forced their abilities to solve their problems. For example; the student H expressed the following quotation: *"I can only do either conditional or looping. However, when I have to work on a program that requires conditional and looping combinations, I cannot."*

To follow up their responses to the interview protocols, they were asked to conduct robotics learning to find out how they progressed with their CT skills. The results of each CT area are outlined in the following:

a) Abstraction and Modularity

Table 5 summarizes the results of their responses of the 'abstraction and modularity' areas.

Table 5. *The participants' responses of the 'abstraction and modularity' areas*

Questions	Their responses to the questions
What is the typical behaviour of the robot?	All students demonstrated that the robot moved forward until it detected an obstacle, then it executed the following command
What is the general programming structure?	All students indicated the first and second programs. That is the robot run and stopped till the sensors detected something, only different in the sensor. The third was a combination of the first and second programs.
Which programming did the students do firstly?	All students, but the students B, D, E, and G built their programs in sequence. The student B, D, E, and G tested their robot models after the first program was done to ensure whether the robot runs correctly. They could do the next programs more accessible after testing.

b) Generalization

Table 6 depicts the results of the participants' abilities of constructing a generalization.

Table 6. *The participants' abilities of constructing a generalization*

Questions	Their responses to the questions
Please suggest a more generalized solution to implement sensors covering a wide range of cases. Is the proposed solution more general? Please explain your reason	The student A mentioned line following disability chair, which detected the upcoming obstacles via infrared. The student B mentioned car with traffic light detector and auto brake. The students C and G mentioned line follower for factory vehicle and car with auto brake. The student D mentioned auto cleaner with an infrared sensor (without referring to the colour sensor). The student E mentioned fire extinguisher with a light sensor and a stick for visually impaired people using an infrared sensor. The student F mentioned line follower (without referring to the use of an infrared sensor). The student H did not mention anything.

c) Algorithm

The observation and interviews showed that all participants completed to programming, but the student G failed the assignment of algorithm construction. The algorithm phase started by writing down how the motor had to run. Then, they determined when the motor had to turn off. The student G wrote only how the sensor worked and failed to explain the state of the motor. Subsequently, they implemented their algorithms to the block programming (in LEGO Software Home Edition). Table 7 shows how many times they tried to work on programming until the robot's movements complied with pre-determined objectives.

Table 7. *The number of the participants' programming attempts*

	S A	S B	S C	S D	S E	S F	S G	S H
Program 1	1	2	1	1	1	1	1	1
Program 2	1	2	1	1	1	1	2	1
Program 3	1	3	3	3	3	2	2	1

d) Decomposition

As can be seen from Table 8, the participants were either grouped all items first (GAF) or grouped per slide (GPS). The participants, who were grouped, were crossed in both categories.

Table 8. *The results of the assembling activity*

Steps	S A	S B	S C	S D	S E	S F	S G	S H
GAF	x	x	v	x	v	x	x	v
GPS	x	x	v	x	v	x	v	v
Step 1	v	v	v	v	v	v	v	v
Step 2	x	x	v	v	x	x	v	x
Step 3	v	v	v	v	v	v	v	v
Step 4	x	v	v	v	x	x	v	v
Step 5	v	v	x	v	x	v	x	v
Step 6	v	x	v	v	v	v	v	v
Step 7	x	x	v	v	x	x	x	x
Step 8	v	v	v	v	v	x	v	v
Step 9	v	v	v	v	v	v	v	x
Duration (Minutes)	50	58	24	37	29	40	43	37

When the student G was inquired about his choice of grouping, he selected the components required in each step. He spent 43 minutes for all steps in the assembling activity. The students C, E, and H, in contrast, grouped all components before started to assembling. The rest of the participants directly built their robots without sorting the components out.

DISCUSSION

The research evaluates the effectiveness of educational robotics to endorse undergraduate students' computational thinking. The participants with basic programming knowledge addressed that they had difficulties in algorithm, syntax, and sophisticated algorithm. The participants pointed to their inclination levels towards each area of Computational Thinking in the robotics activities. This research handled the Computational Thinking with five competencies comprising of Abstraction, Modularity, Generalization, Algorithm, and Decomposition (Atmazidou & Demitriadis, 2016). In light of the results, the participants indicated a self-efficacy transformation along the treatment.

a) Abstraction and Modularity

The participants, who demonstrated their CT skills in the area of abstraction and modularity, explained how their robots were structured. That is, LEGO Mindstorm helped them identify and classify the functions of the program and simply ignoring unnecessary details (Weese & Feldhausen, 2017). They further elaborated on how a particular component of the robot responded to the instructions implemented in the program.

b) Generalization

The ability to identify a particular component in the robotics module and associate it in another context seems to well-conveyed in the data analysis. Transferring a solution into a broader context is regarded as a mental construct associated with the Computational Thinking skill (Atmatzidou & Demetriadis, 2014). The participants were asked to mention their devices related to colour and infrared sensors. Hence, they were invited to probe how to associate the devices with the real world.

c) Algorithm

The participants were asked to express their algorithm proficiencies through written responses and projects in LEGO Mindstorm Home Edition software. Although some of the participants repeatedly worked their solutions out due to an increase in the problems' complexities, they enjoyed learning robotics. The block programming facilitated to identify either a single step or multiple steps in a solution. The use of loops underscored the usability of repeated commands (Weese & Feldhausen, 2017) as visually depicted in the interface.

d) Decomposition

Decomposition, which is the ability to break down a complex problem into small parts, makes a complex problem easily solvable. The modular part of LEGO Mindstorm engaged the participants in assembly and disassembly activities. Grouping or ungrouping the components is considered as the simulation breaking complex problems into a solvable smaller situation. The problem-solving strategy, however, would be affected by individual learning behavior (Kamal, Budiyanto, & Efendi, 2018).

CONCLUSION

This research evaluated how the participants developed their computational thinking skills while dealing with robotics learning. It is evident that the participants demonstrated varied computational thinking skills. Each participant tended to craft specialized proficiency levels of computational thinking skills in different areas. The robotics-assisted learning, in this regard, enabled the participants to explore and develop their CT skills. A long-term robotics activity, however, is required since some participants needed a longer time to explore their particular components. Given their interactions with educational robotics in a short-duration, the participants sufficiently comprehended the essence of the robotics learning and managed robotics- and computational thinking-related knowledge. Indeed, because the participants obtained pre-existing knowledge in computational thinking and managed their computational thinking skills, they may have tackled the robotics learning efficiently.

REFERENCES

Ackermann, E. (2001). Piaget's constructivism, Papert's constructionism: What's the difference. *Future of learning group publication*, *5*(3), 1-11.

Alimisis, D., Arlegui, J., Fava, N., Frangou, S., Ionita, S., Menegatti, E., ... Pina, A. (2010). Introducing robotics to teachers and schools: experiences from the TERECoP project. *Proceedings for Constructionism*, *1*, 1-10.

Alimisis, D. (2012). Robotics in education & education in robotics: Shifting focus from technology to pedagogy. In *Proceedings of the 3rd International Conference on Robotics in Education*, 7-14.

Alimisis, D., Moro, M., Arlegui, J., Pina, A., Frangou, S., & Papanikolaou, K. (2007, August). Robotics & constructivism in education: The TERECoP project. In *EuroLogo 40*, 19-24.

Atmatzidou, S., & Demetriadis, S. (2014). How to Support Students' Computational Thinking Skills in Educational Robotics Activities. Paper presented at the Proceedings of 4th International Workshop Teaching Robotics, Teaching with Robotics & 5th International Conference Robotics in Education, Padova, Italy.

Atmatzidou, S., & Demetriadis, S. (2016). Advancing students' computational thinking skills through educational robotics: A study on age and gender relevant differences. *Robotics and Autonomous Systems*, *75*, 661-670.

Bers, M. U., Flannery, L., Kazakoff, E. R., & Sullivan, A. (2014). Computational thinking and tinkering: Exploration of an early childhood robotics curriculum. *Computers & Education*, *72*, 145-157.

Bocconi, S., Chioccariello, A., Dettori, G., Ferrari, A., Engelhardt, K., Kampylis, P., & Punie, Y. (2016). Developing computational thinking in compulsory education. *European Commission, JRC Science for Policy Report*.

Brennan, K., & Resnick, M. (2012, April). New frameworks for studying and assessing the development of computational thinking. In *Proceedings of the 2012 annual meeting of the American Educational Research Association, Vancouver, Canada*, 1-25.

Catlin, D., & Woollard, J. (2014, July). Educational robots and computational thinking. In *Proceedings of 4th International Workshop Teaching Robotics, Teaching with Robotics & 5th International Conference Robotics in Education*, 144-151.

Cresswell, J. W. (2014). Penelitian kualitatif & desain riset. *Yogyakarta: Pustaka Pelajar*.

Eguchi, A. (2014, July). Robotics as a learning tool for educational transformation. In *Proceeding of 4th International Workshop Teaching Robotics, Teaching with Robotics & 5th International Conference Robotics in Education Padova (Italy).*

Kamal, F., Budiyanto, C. W., & Efendi, A. (2018, November). Understanding students behavior during the adoption of modular robotics in learning. *In IOP Conference Series: Materials Science and Engineering, 434*(1), p. 012263. *IOP Publishing.*

Karim, M. E., Lemaignan, S., & Mondada, F. (2015). A review: Can robots reshape K-12 STEM education?. In *Advanced Robotics and its Social Impacts (ARSO), 2015 IEEE International Workshop on*,1-8. IEEE.

Kazimoglu, C., Kiernan, M., Bacon, L., & Mackinnon, L. (2012). A serious game for developing computational thinking and learning introductory computer programming. *Procedia-Social and Behavioural Sciences, 47,* 1991-1999.

Miglino, O., Lund, H. H., & Cardaci, M. (1999). Robotics as an educational tool. *Journal of Interactive Learning Research*, *10*(1), 25.

Miles, M. B., Huberman, A. M., & Saldana, J. (1984). Qualitative Data Analysis: A Methods Sourcebook. *Sage Publications Ltd (CA).*

Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. New York: Basic Books, Inc.

Weese, J. L., & Feldhausen, R. (2017). STEM Outreach: Assessing Computational Thinking and Problem Solving. Paper presented at the 2017 ASEE Annual Conference & Exposition.

Wing, J. M. (2006). Computational thinking. *Communications of the ACM, 49*(3), 33-35.

Wing, J. M. (2008). Computational thinking and thinking about computing. *Philosophical Transactions of the royal society of London A: mathematical, physical and engineering sciences*, *366*(1881), 3717-3725.