PRIMARY SCHOOL PRE-SERVICE TEACHERS' PERCEPTIONS OF *SCRATCH* AND ITS ROLE IN FACILITATING STUDENTS TO LEARN CODING

Theresia Yunia Setyawan
Primary School Teacher Education Program, Sanata Dharma University
theresiayunia@usd.ac.id

Abstract

This study was a qualitative survey study aiming at describing the perceptions of primary school pre-service teachers of *Scratch* programming software. It also intended to unfold their viewpoints of the software's roles in supporting their own learning process as well as in facilitating primary school students to learn coding. The data gathered in this study was derived from a questionnaire distributed to 50 primary school pre-service teachers of Sanata Dharma University taking the *Media Pembelajaran Berbasis* ICT course. The questionnaire was in the form of open-ended and semi open-ended questions and was distributed to the respondents after they had attended *Scratch* programming sessions during the course. The data gathered through the questionnaire was coded and classified into categories to form patterns of primary school pre-service teachers' perceptions of *Scratch* programming and their viewpoints of the software's roles in supporting their own learning process as well as in assisting their future role as primary school teachers facilitating their students to learn the skills of coding. The results of the data analysis showed that the primary school pre-service teachers had positive perceptions of *Scratch*. They considered the program as good and attractive for young learners. It was also found out that the program was beneficial not only in improving their digital literacy skills by facilitating them learn simple coding or programming but also in helping them advance their thinking as well as problem-solving skills.

Keywords: primary school, pre-service teachers, perceptions, *scratch*, coding

1. Introduction

Digitalization has become inseparable part of everyday life and it makes it inevitable for people to become more and more reliant on the use of technology to perform their day-to-day works. As the world becomes increasingly digitalized and automated, children of today needs to learn to be creators and not just consumers of digital technology [2]. In order to embrace their role as digital creators, these millennial children need to familiarize themselves with and learn the skills of coding.

In its broad sense, coding is programming. It means writing the step-by-step instructions that tell a piece of technology – usually a computer – what to do [3, 4]. Coding also means arranging the instructions so that the program works as smoothly and quickly as it can, and doing all this in a way that other coders or programmers can follow in case they need to look at or modify them [13].

Coding has become increasingly important because it teaches children to problem-solve by performing computational thinking. It requires them to think like a computer by breaking down tasks into a logical sequence of smaller steps, discarding unnecessary elements, diagnosing errors, and inventing new approaches when the conventional ones do not work. In short, coding teaches them to learn to think [6].

Coding can also power creativity and innovation [12]. It encourages learners to work things out for themselves and be problem-solvers as well as empowers them to be digital makers for jobs in the future. The skills of coding also equip them with the ability to keep up with rapid changes in the digital world by providing them with the ability to cope with uncertainties of the programming results while at the same time keep their logical thinking in scripting the programming instructions [6].

One of the best software that can help student start learning coding is *Scratch*. It is a visual programming software developed specifically for children by the Lifelong Kindergarten Group at the Massachusetts Institute of Technology (MIT) Media Lab. As it is intended for young learners, *Scratch* is without doubt easy to use. It consists of attractive colorful blocks of texts with simple written instructions on them. What makes it even easier is that these blocks snap together like Lego or puzzle pieces to create basic programs. By using this software, learners can easily adapt already-made codes to make their work faster and better or even make their own games, quizzes, or animations [1, 8, 11].

The importance of coding for working and living in the digital era has urged teachers to start thinking about teaching its skills to their students as early as possible [6, 12]. As millennial teachers, they should prepare their students to face life and work in the digitalized future. However, as coding is still a relatively new subject to primary schools in Indonesia, teachers at this level of education need to get themselves familiar with the subject. Providing them with the opportunities to learn coding using unsophisticated software such as *Scratch* is just an initial effort to evoke their awareness about the benefits of coding for the future generation.

This paper is trying to identify the perceptions of primary school pre-service teachers learning coding for the first time using *Scratch*. It intends to reveal how these pre-service teachers view their learning experience with the program. These perceptions are central not only to reveal their genuine feelings towards the process of learning *Scratch* itself but also to predict the probability of their adopting and making use of the program to facilitate their future millennial students to learn coding.

2. Method

This study was a one-shot survey study taking qualitative approach as its framework. Unlike quantitative surveys aiming at describing statistical parameters of the data gathered from their respondents, this study aimed to identify the variation of the respondents' perceptions of the topic under study [7]. As a one-shot survey research, this study followed only one empirical cycle of (1) identifying research problems, (2) determining research respondents, (3) gathering data from the respondents, (3) analyzing the data gathered, and (4) generating hypotheses based on the data analysis process.

The respondents of this study were fifty sophomores from Primary School Teacher Education Program of Sanata Dharma University who learned coding using *Scratch* as part of their ICT course. At the end of their five-week learning experience using the program, the students were given a written questionnaire containing five questions about *Scratch*. Four questions were open-ended questions while one was semi open-ended where they were asked to choose among the options or add their own response when they thought they had other applicable responses for the question.

The data gathered from the questionnaire were classified into two main categories. Those categories were respondents' general thoughts about *Scratch* and their follow-up actions regarding the use of the program in their later professional life as primary school teachers. The two categories were then elaborated into a few sub-categories detailing the respondents' thoughts about *Scratch* (i.e. its advantages, benefits, and constraints) and the kind of follow-up actions they would take after learning the program.

3. Results

The results of the data analysis phase showed that the majority of the respondents (96.15%) thought that *Scratch* was not easy to learn. Their responses revealed that they had difficulty in assembling the *Scratch* command blocks to generate their own scripts (84.32%). They also stated that they strived to use the program because of their unfamiliarity with its user interface (7.84%) and because of the fact that the program used English as its language interface (7.84%). It was also found out that some respondents had difficulties in using and creating their own sprites (3.92%), and in using sound features of the program (3.92%).

Apart from the difficulties, however, the respondents testified that *Scratch* was good because of its ability to help them create simple games and animations (80.39%). The rest of the respondents noted down that the program was attractive because it contained colorful command blocks that made it easy for users to choose among the scripts. They also agreed that the fact that it had cute cartoon sprites with varieties of costumes or movements made the program even more attractive for beginners.

When asked about the benefits of learning *Scratch* for them, 33.33% of the respondents mentioned problem-solving, critical thinking, and digital literacy skills as the skills that they learned while learning the program. The rest of the respondents stated that they were compelled to use two, or at least one, of the skills while learning coding with *Scratch*. However, these respondents noted that they were also required to be more creative and think logically in order to generate their own scripts and make their own games or animations. As follow-up actions, the questionnaire responses showed that the majority of the respondents (98.04%) said they would continue learning the program because they wanted to use it to create educational games as well as animations for their future students. In addition, they also wanted to teach *Scratch* to their students because they believed that the program could help their students improve their digital literacy skills, learn to think creatively and logically as well as to problem-solve.

4. Discussion

The data analysis process showed that the majority of the primary school pre-service teachers considered *Scratch* as a not-so-easy program to learn because they experienced difficulty in assembling the command blocks while trying to make characters (sprites) do something. Despite their five-week learning experience with *Scratch*, this struggle could be rooted from the fact that it was their first encounter with the program as well as with coding in general. As indicated by their responses, they found it challenging to think of and pick up specific command blocks for specific instructions because they were not yet familiar with the script tab and its contents. Simply put, they still strived to decide which blocks to use and under what categories they belonged to.

Scratch was developed with constructivist learning theory in mind [10]. Accordingly, it is acknowledged that learners will learn to use the program at their best when they are personally involved and engaged in the processes of designing their own projects. While *Scratch* only required its users to drag and snap command blocks together to move sprites or characters, learners still need to think of correct sequences of blocks in order for their sprites to move or do specific things. Experimenting with different categories of block commands and trouble-shooting the generated scripts from time to time will help novice users to familiarize themselves with the block palettes and the commands under each category [8]. It is through these processes of experimenting and trouble-shooting that learners attempt to use their logical thinking to work things out for themselves and find solutions to solve problems concerning their own generated instructions in the program.

The difficulty of which the primary school pre-service teachers had to deal with in assembling command blocks to create instruction scripts, creating their own sprites as well as adding sounds in *Scratch* could result from the lack of intensive practices and experiences in using the program. The amount of time provided for them to experience and learn to use the program was presumed to have effect on their familiarity with the program. Considering their experience with *Scratch* and coding in general, it was quite likely that they did not have sufficient time to learn and use the program autonomously. Provided with more time and learning experience, it is expected that these student teachers can familiarize themselves with the program and, therefore, dealing with their difficulty in producing novel scripts of their own by continuing advancing their digital literacy skills as well as their thinking and problem-solving skills as they learn to create something using the program.

The data analysis process also showed that 80.39% of the primary school pre-service teachers thought that *Scratch* was good because it could help them create simple games and animations. This simplicity could due to the fact that the program was actually designed for young learners with little or no programming skills. As revealed in the data analysis process, the student teachers felt more confident with the program because they did not need to memorize any bits of code to program and because they could minimize the risk of syntax errors while using the program [10].

The primary school pre-service teachers also reported that *Scratch* was attractive because of its colorful interface. This colorfulness was perceived from the different colors of its ten categories of commands (i.e. *motion*, *looks*, *sound*, *pen*, *data*, *events*, *control*, *sensing*, *operators*, and *more blocks*) in the scripts tab. In practice, as they noted, the color difference between categories were helpful in choosing and arranging block commands in the block palette. Because a series of instructions for a sprite consisted of blocks from multiple categories, the difference in colors made it easier for them to switch between categories and locate appropriate block commands they needed [9].

The built-in costumes of *Scratch* sprites or characters were also part of the program the primary school pre-service teachers considered as interesting. The costume sets were not only attractive but also helpful in easily moving a sprite. As a costume functions as an alternate appearance of a sprite, they did not need any specific block commands and were only required to use multiple costumes of the sprite to give it the impression of movement, or put it simply, to move it [5].

The responses given by the primary school pre-service teachers also showed that the majority of them (98.04%) would continue to learn *Scratch* because they wanted to create their own games and animations using the program. They noted that *Scratch* was advantageous not only in visualizing learning materials but also in advancing their digital literacy skills and making them learn to think logically as well as problem-solve creatively. Considering its benefits, the pre-service teachers said they would later make attempts to teach their students to use *Scratch* and hence help the future millennial generation to develop their own digital literacy, thinking and problem-solving skills by creating something novel using the program.

The benefits of *Scratch* stated by the primary school pre-service teachers underline the important characteristics of the program, i.e. tinkerable and meaningful [11]. It is considered *tinkerable* because it lets learners experiment with commands and code blocks in a way they might tinker with mechanical or electronic components. This tinkerability characteristic encourages hands-on learning and supports a bottom-up approach to creating scripts where blocks of code are assembled, tested and put together into larger units. In other words, tinkerability helps learners discover the functionality of various blocks of codes autonomously [8].

Scratch is also considered as providing meaningful learning experiences for its novice users because it allows them to choose among different types of projects (such as such as stories, games, animations, or simulations), so they can work only on projects they are really interested in. The program also makes it easy for learners to personalize their chosen projects by importing photos and sound clips, as well as creating graphics of their own. As the learners work on their own personally meaningful projects, it is more likely that they use their thinking and problem-solving skills to learn by using essential mathematical and computational concepts needed to make their projects work properly [11].

5. Conclusions

Based on their five-week learning coding experience with *Scratch*, it was found out that the primary school pre-service teachers of Sanata Dharma University had positive perceptions of *Scratch* and of its role in supporting the process of learning to code. Even though the pre-service teachers thought that the program was not easy to learn, they noted that it was good because they could create their own animated projects such as games and interactive stories without any specific knowledge of programming skills. In addition, it could also help them visualize their learning materials and so provide more engaging learning experiences for their future students.

The primary school pre-service teachers also indicated that *Scratch* was attractive and advantageous. The simple colorful interface of the program helped them move around different categories of scripts and find ones appropriate for their projects. Most importantly, the pre-service teachers were motivated to learn more about *Scratch* and teach their future students to use it because they were aware that learning *Scratch* was not merely learning about coding or programming but also learning to continuously develop and use digital literacy, thinking and problem-solving skills necessary to live and survive in millennial era.

ACKNOWLEDGMENT

Many people have contributed to this research. The researcher would like to thank sophomore students of Primary School Teacher Education Program of Sanata Dharma University, who had been willing to take part in the *Media Pembelajaran Berbasis* ICT course offered in the department and participate as respondents of the research. The researcher would also like to thank fellow ICT teachers of Sanata Dharma University, who had helped in validating and verifying the instrument used in this research.

REFERENCES

[1] A. Wilson and D. C. Moffat, Evaluating Scratch to Introduce Younger Schoolchildren to Programming, 10 October 2010, http://scratched.gse.harvard.edu/resources/. Accessed 11 April 2017.

[2] B. Quinn, Computer Coding more in Demand than Languages, Survey Shows, 3 March 2014, www.theguardian.com/education/. Accessed 11 April 2017.

[3] C. McCue, Coding for Kids. Hoboken, NJ: John Wiley & Sons, Inc., 2015.

[4] C. Vorderman, J. Woodcock, S. McManus, C. Steele, C. Quigley, and D. McCafferty, Help Your Kids with Computer Coding. New York: DK Publishing, 2014.

[5] E. A. Vlieg, Scratch by Example: Programming for All Ages. New York: Springer Science and Business Media, 2016.

[6] G. Hinsliff, Should Kids Learn to Code?, 3 December 2015, www.theguardian.com/news/. Accessed 11 April 2017.

[7] H. Jansen, "The logic of qualitative survey research and its position in the field of social research methods," Forum Qualitative Sozialforschung/ Forum Qualitative Social

Research, 11(2), Art. 11, May 2010, http://www.qualitative-research.net/. Accessed 13 April 2017.

[8] J. Maloney, M. Resnick, N. Rusk, B. Silverman, and E. Eastmond, “The Scratch programming language and environment,” ACM Trans. Comput. Educ. 10, 4, Article 16, 15 pages, November 2010.

[9] J. Woodcock, Coding Projects in Scratch. New York: DK Publishing, 2016.

[10] K. A. Peppler and Y. B. Kafai, “Creative Coding: Programming for Personal Expression,” The 8th International Conference on Computer Supported Collaborative Learning (CSCL), Rhodes, Greece, Vol. 2, pages 76-78, June 2009, www.researchgate.net/publication/. Accessed 13 April 2017.

[11] M. Resnick, et al., “Scratch: Programming for all,” Communication of the ACM, vol. 52 (11), pp. 60-67, November 2009.

[12] N. Morrison, Teach Kids how to Code and You Give Them a Skill for Life, 27 December 2013, www.forbes.com/sites/nickmorrison/. Accessed 11 April 2017.

[13] S. Andrews and A. Dixon, Coding for Kids: Scratch. London: Dennis Publishing Ltd., 2014.

Primary School Pre-service Teachers Perceptions of Scratch and Its Role in Facilitating Students to Learn Coding

ORIGINALITY REPORT

100%	100%	2%	3%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	eprints.uny.ac.id Internet Source	100%

Exclude quotes	On	Exclude matches	Off
Exclude bibliography	On		