ICTTE
FKIP UNS 2015

Robomind Utilization to Improve Student Motivation and Concept in Learning Programming

Rosihan Ari Yuana, Dwi Maryono
Sebelas Maret University, Jl. Ir. Sutami 36A, Surakarta, Indonesia

Corresponding e-mail: rosihanari@staff.uns.ac.id

Abstract: Robomind is one of the PAT (programming assistance tool) software that used as a tool for programming learning and improving the understanding of programming concepts, as well as improving the ability of logical thinking and problem solving. The advantages of Robomind compared to the other PATs are a simple graphical display and attractive, it can be used for students ranging from primary schools, secondary schools and further education. Besides design and features, Robomind has also fulfilled the Robert Gagne's instructional theory concept. The aim of this study is how to utilize robomind as a tool to motivate students in learning computer programming and know the students' responses to its use in understanding the programming concepts. This research is motivated by the lack of student motivation in learning programming that caused by the initial perception that programming is difficult. In addition, some programming elusive concepts also make the lack of motivation to learn programming. This research was conducted at the Department of ICT Education FKIP UNS on Structured Programming course in first semester. Results showed, there were 90% of students feel motivated to learn programming although not previously been familiar with programming at all. In addition, it is also obtained from the survey that 20% of students who previously had studied programming, 85% of whom could have better understanding concepts that were previously less understood.

Keywords: robomind, learning, programing

1. INTRODUCTION

The use of simulation software to support learning in the field of engineering has been widely implemented (Botelho, Marietto, and Ferreira, 2015). With the use of simulation software, it is expected that students can practice as if dealing directly or uses a real tools.

Olsson and Mouzelis (2015) stated that in the field of computational techniques and computer science, programming is a very important thing. However, many students have difficulties in learning programming, especially concerning how to combine the skills and knowledge they have. Most students have difficulties in implementing their knowledge into the coding process. Students are easier to learn and understand the algorithm than express the algorithm into the program. It has been studied for years by researchers and expected a solution how to improve students' understanding in learning programming.

As noted by Korsse, Cilliers, and Calits in 2015, the other problem is the lack of motivation

for students to learn programming because they consider the initial perception that programming is a difficult matter.

According to Koorsse et al (2015), the programming assistant tool (PAT) is a kind simulation software that is specifically used to enhance the students' abilities and understanding of the programming concepts. Several types of PAT have been developed such as ScratchJr (Portelance, Strawhacker, and Bers, 2015), Scratch (Armoni, Meerbaum-Salant, and Ben-Ari, 2015), Robomind, B# (Koorsse et al, 2015), KERMIT and CeeBot (Duenpen, 2015). Not only that, even some kind of PATs can be used to motivate students to learn programming, as proposed by Gomes and Mendes (2015), because in general, the PAT is designed with attractive features, such as Corrupted which is a game (Chau, Robinson Pace, Nash, and Sung, 2014). PAT also has a simple Integrated Development Environment (IDE) and easy to use so that students will have no difficulties in

using this PAT. Thus, students can fully develop the codes based on their algorithms and just focus on coding process.

As a tool to learn programming, generally PAT focuses on the flow control issues, particularly on the issue of sequencing, conditional statements and loops. Because these concepts is a basic material in programming.

2. ROBOMIND

According to Duanpen (2010), Robomind is software developed by Arvid Halma with Java that serves as a tool to learn programming. Robomind is adopted from the Logo programming language. In general, the main objects of Robomind are how to control the movement of the robot through a sequence of instructions written in a particular programming language (see Fig. 1). Simplicity and ease of its use may cause Robomind used by wide range students from elementary to college to learn programming. Furthermore, Robomind can be integrated with Mindstorm NXT robot types that produced by Lego. So Robomind can also simultaneously be used as a learning tool robotics, as well as WeDo robotics kit set (Strawhacker, and Bers, 2014).

The advantages of using Robomind in learning are fulfilling some aspects to the instructional pedagogic theory by Robert Gagne (Duenpen, 2010), namely: gain attention aspect, elicit performance, and provide feedback. In the aspect of gain attention, Robomind has an interface and features that can be both interesting and challenging problem solving cases for users.

Furthermore, Robomind can be used to compare the performance of programs produced by users. For the same case, the performance can be compared between one program to another. So in this case the students can understand the efficiency of algorithms and know how to construct an algorithm in order to obtain more efficient performance of the program. This makes Robomind meet elicit performance aspects.

Related to provide feedback aspect, Robomind can inform the user where the lines of program code that has a error syntax or improper

use of procedure so that the users can fix it immediately.

In this study, Robomind will be implemented in the learning process as a tool for understanding the programming basic concepts as well as to motivate students.

The aim of this study is how to utilize robomind as a tool to motivate students in learning computer programming and know the students' responses to its use in understanding the programming concepts.

This research is conducted at the Department of ICT Education FKIP UNS, and all of its undergraduate students as the research objects, especially the students in the first semester.
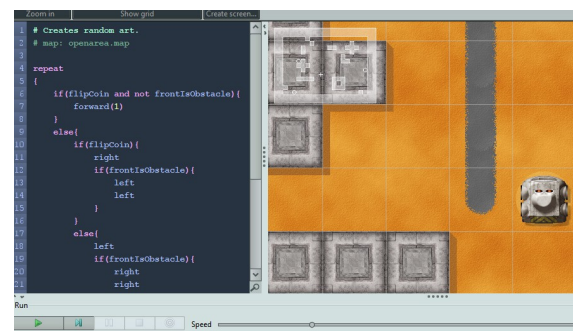

Figure 1. Robomind interface

3. METHOD

Implementation of the Robomind conducted on students who take Structured Programming course in first semesters. This course was chosen due to the basic course of programming courses. Moreover, because of this course is in the first semester so that students will most likely have never practice to programming at any levels of previous education.

In Structured Programming course, students are introduced to Robimind as the first material before Pascal programming language as the main topic for about 4 sessions. The topics given in those sessions include: logic and flow control consisting of sequences, conditional statements and loops as the basic materials in programming. each material will be given 5 problem solving cases with increased levels of difficulty.

PROSIDING ICTTE FKIP UNS 2015
Vol 1, Nomor 1, Januari 2016
Halaman:

ISSN: 2502-4124

ICTTE
FKIP UNS 2015

During each session, students' progress will be evaluated to determine the level of students' understanding of basic concepts of programming. Furthermore, at the end of Robomind material, students' response to the use of Robomind related to their motivation in programming will be obtained.

4. RESULT AND DISCUSSION

At the first session, students are given an explanation about programming in general, background and its benefits. Still in this session, then students were given materials on Robomind, ie how it works and how to use some important features in it.

Once students understand how to use Robomind, then they were given topic on the sequence concept in programming. Related to this concept given, some cases have been solved by the students. An example of given case was the line follower problem. Using the map (default.map) in Figure 2, students were asked to make a some codes so that the robot can move along the white line until a predetermined end point.

Figure 2. Line follower problem

From 5 given cases in the sequence concept material, there were 98% of 60 students can do all cases in time. So from this first session, most students comprehend this concept very well.

In the second session, given the concept of conditional statements. After having given a brief explanation of the concept of conditional statements, students were also given five problems that must be solved by students. One of the given cases was grabbing the beacons (Fig. 2). In this case, students were asked to

make the program code with beacon grabbed by the robot as target. In this session, from 5 cases, there were 87% of students could complete all cases on time.
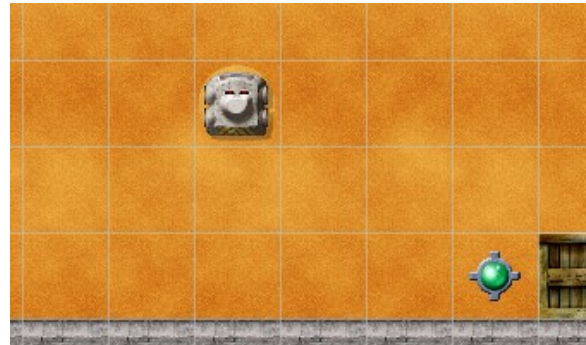

Figure 3. Grabbing the beacon problem

The next basic programming concepts that was provided is about loops. This concept was given at the fourth session. First, we explained to the students about the concept of loops theoretically. In particular, this theoretical explanation focuses on the difference between the concept of loop with the certain numbers and unknown numbers of loop, as well as some examples of implementation on both cases.

Furthermore in this session, 5 cases were given to be solved by the students with robomind. The cases were selected in this phase incorporates the concept of sequence, conditional statements and loops. One of given cases was how to make a line that is parallel to the black line and has the same length, with condition the student may not count the number of robot's steps (see Fig. 4).

By the 5 given cases in this material, there were 76% of students able to complete all cases within the allocated time. While 24% of students could not solve all cases, there are 83% of students were able to complete 3 to 4 of cases, and the remaining 17% could only finish 1 or 2 cases only.

Right after understanding the concept of flow control in programming, students were given a concept concerning to the modular programming using a subprogram or procedure. The introduction of the modular programming is

PROSIDING ICTTE FKIP UNS 2015
Vol 1, Nomor 1, Januari 2016
Halaman:

ISSN: 2502-4124

ICTTE
FKIP UNS 2015

also very important in structured programming. All 5 problem solving cases in this topic could be solved by approximately 82% of students.
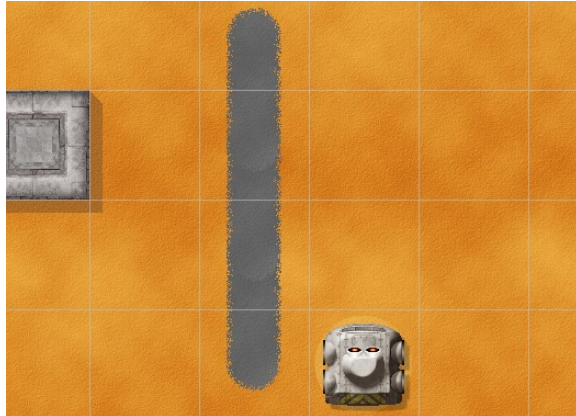

Figure 4. Copyline Problem

Not only can the introduction to the programming concept, Robomind also be shown to the students which program code is more efficient to solve a certain case. This efficiency can be seen clearly through Robomind by running some program code concurrently and see which code that gains the target first. Such of this visualization can provide insight to students about the importance of efficiency in making the program code, even though the concept of algorithm analysis has not been given to students.

Once the students had introduced to the basic concepts of programming, at the end of the session on Robomind material, students were asked to provide feedback on the benefits earned during the use Robomind. Some feedback items asked to students include: ease of Robomind operation, ease in understanding the basic concepts of programming and students' motivation to programming.

The feedback survey results to the 60 students as showed in Table 1. Based on the results, almost all the students regarded Robomind easy to use. The effectiveness of the use of robomind to improve understanding of the basic concepts of programming is known by the high percentage of students (86.75%) who

have grasped the concept of sequences, conditional statements, loops and procedure, which is the average percentage of students who had understood.

Table 1. The student survey results related the use of Robomind in learning programming

Items	Answers Percentage		
	Yes	No	Abstain
Robomind is very easy to operate	98%	2%	0%
Robomind can help your understanding in sequence concept	97%	2%	1%
Robomind can help your understanding in conditional statement concept	88%	10%	2%
Robomind can help your understanding in loops concept	77%	18%	5%
Robomind can help your understanding in modular programming	85%	14%	1%
With the use of Robomind can motivate you to learn programming further	87%	9%	4%

The interesting thing about these results, from 87% of students were motivated to learn programming further, 90% of whom have never known or studied programming at all.

In addition, it is also obtained from the survey that 20% of students who previously had studied programming, 85% of whom could have better understanding concepts that were previously less understood.

5. CONCLUSION

From this study, it can be concluded that the use of Robomind to introduce some basic programming concepts to students in the form of problem solving can improve the motivation of students as well as to improve students' understanding to basic concepts of programming, particularly on the concept of sequencing, conditional statements, loops, and procedures. This is due to the ease of use Robomind features and supported by attractive graphical display

Although in this study Robomind said to be an effective way to improve students' motivation and understanding of the basic concepts of programming, but it is not known the comparative effectiveness against other types of PAT, ie ScratchJr or B #.

6. REFERENCES

Armoni, M., Meerbaum-Salant, O., Ben-Ari, M. (2015). From scratch to "Real" programming. "*ACM Transactions on Computing Education*", 14 (4), 25.

Botelho, W.T., Marietto, M.G.B., Ferreira, J.C.M., Pimentel, E.P. (2015). Kolb's experiential learning theory and Belhot's learning cycle guiding the use of computer simulation in engineering education: A pedagogical proposal to shift toward an experiential pedagogy. "*Journal of Computer Applications in Engineering Education*".

Chau, B., Robinson, A., Pace, J., Nash, R., Sung, K. (2014). Corrupted: A game to teach programming concepts. "*Computer*", 47 (12), 6994080, pp. 100-103.

Duenpen, K. (2010). A Web-based System Design for Enhancing Learning Problem Solving in Artificial Intelligence. "*Proceeding of The Seventh International Conference on eLearning for Knowledge-Based Society*", Thailand, Dec, pp. 42.1 - 42.6.

Gomes, A., Mendes, A. (2015). A teacher's view about introductory programming teaching and learning: Difficulties, strategies and motivations. "*Proceedings - Frontiers in Education Conference, FIE*", Feb, 7044086.

Koorsse, M., Cilliers, C., Calitz, A. (2015). Programming assistance tools to support the learning of IT programming in South African secondary schools. "*Computers and Education*", 82, pp. 162-178.

Olsson, M., Mozelius, P. (2015). Visualization of concepts and algorithms in programming education - A design theoretic multimodal perspective. "*Proceedings of the International Conference on e-Learning, ICEL*", January, pp. 257-264.

Portelance, D.J., Strawhacker, A.L., Bers, M.U. (2015). Constructing the ScratchJr programming language in the early childhood classroom. "*International Journal of Technology and Design Education*".

Strawhacker, A., Bers, M.U. (2014). "I want my robot to look for food": Comparing Kindergartner's programming comprehension using tangible, graphic, and hybrid user interfaces. "*International Journal of Technology and Design Education*".