EFFECTS OF MACHINE-LEARNING PROGRAMMING SIMULATOR ON PERFORMANCE, ENGAGEMENT AND PERCEIVED MOTIVATION OF UNIVERSITY STUDENTS IN LEARNING PROGRAMMING

PUTRI TANSA TRISNA ASTONO

UNIVERSITI SAINS MALAYSIA

2024

EFFECTS OF MACHINE-LEARNING PROGRAMMING SIMULATOR ON PERFORMANCE, ENGAGEMENT AND PERCEIVED MOTIVATION OF UNIVERSITY STUDENTS IN LEARNING PROGRAMMING

by

PUTRI TANSA TRISNA ASTONO

Thesis submitted in fulfilment of the requirements
for the degree of
Doctor of Philosophy

August 2024

ACKNOWLEDGEMENT

I would like to express my gratitude to Allah SWT, the Most Gracious, the Most Merciful, who has given me the opportunity to obtain a doctorate degree, and given me the courage, fortitude and determination to undertake this difficult and protracted journey. My deepest gratitude goes to my supervisors, Professor Dr. Wan Ahmad Jaafar Wan Yahaya, Dr. Nur Azlina and Professor Dr. Sriadhi for their unwavering guidance, tolerance, inspiration, and vast knowledge. I was able to finish the study and write this thesis thanks to his insightful advice. In addition to my supervisor, my family, particularly my parents Lili Astono and Ida Royani, my brother and my sister Geniung and Rastra, and also my loyal supporter Hanapi Hasan deserve my sincere appreciation. They have been a pillar of support and a place of refuge as I complete my doctorate. I would also like to thank Universitas Negeri Medan for always supporting me to do this research. Last but not least, I would like to thank everyone who helped me on this journey. I would not have been able to do this without all of you.

TABLE OF CONTENTS

LIST OF TABLES

Page

LIST OF FIGURES

Page

LIST OF APPENDICES

KESAN SIMULATOR PENGATURCARAAN BERASASKAN PEMBELAJARAN MESIN TERHADAP PRESTASI, PENGLIBATAN DAN MOTIVASI PELAJAR UNIVERSITI DALAM PEMBELAJARAN PENGATURCARAAN

ABSTRAK

Memandangkan sistem maklumat digunakan untuk menyokong orang ramai dalam banyak aspek kehidupan mereka, pengaturcaraan atau pemrograman adalah penting. Tetapi berbanding negara lain, Indonesia mempunyai kadar kelulusan yang sangat rendah untuk sains, teknologi, kejuruteraan, dan matematik (STEM). Persekitaran moden menjadikan kebolehan pengaturcaraan semakin diperlukan untuk pelajar. Kemahiran pengaturcaraan sangat dihargai dalam banyak bidang, termasuk pembangunan perisian, perbankan, penjagaan kesihatan, dan juga hiburan. Dengan menggunakan reka bentuk kuasi-eksperimen, kajian ini bertujuan untuk mereka bentuk, membangun, dan mengkaji kesan simulator pengaturcaraan aplikasi pembelajaran multimedia persuasif terhadap prestasi pelajar universiti, penglibatan, dan persepsi motivasi terhadap bahan pembelajaran. Dua teknologi—Simulator Pengaturcaraan Pembelajaran Mesin (Simulator Pengaturcaraan ML) dan Simulator Pengaturcaraan Bukan Mesin (Simulator Pengaturcaraan NoML)—adalah pembolehubah bebas dalam kajian ini. Prestasi pelajar, penglibatan, dan persepsi motivasi berkenaan dengan bahan kursus adalah pembolehubah bersandar. Aliran pengajian pelajar, yang dibahagikan dengan kebimbangan pelajar, berfungsi sebagai pembolehubah moderator. Secara keseluruhan, kajian ini melibatkan seorang pelajar universiti daripada universiti awam di Indonesia. Ujian ANOVA, teknik statistik deskriptif dan inferensi, digunakan untuk menganalisis data kajian. Dapatan kajian

menunjukkan bahawa, dari segi penglibatan, prestasi, dan persepsi motivasi untuk subjek, pelajar yang menggunakan Simulator Pengaturcaraan ML mengatasi mereka yang menggunakan Simulator Pengaturcaraan NoML. Menurut kajian ini, prestasi pelajar universiti, penglibatan, dan persepsi motivasi terhadap bahan kursus semuanya telah meningkat hasil daripada aplikasi simulator pengaturcaraan dalam persekitaran bilik darjah. Di samping itu, penciptaan Simulator Pengaturcaraan ML untuk penyelidikan ini membantu mengembangkan kumpulan aplikasi mudah alih dan web yang meningkatkan keberkesanan bilik darjah kursus pengaturcaraan.

EFFECTS OF MACHINE-LEARNING PROGRAMMING SIMULATOR ON PERFORMANCE, ENGAGEMENT AND PERCEIVED MOTIVATION OF UNIVERSITY STUDENTS IN LEARNING PROGRAMMING

ABSTRACT

Since information systems are used to support people in many aspects of their life, programming is essential. But compared to other nations, Indonesia has a very low graduation rate for science, technology, engineering, and mathematics (STEM). The modern environment is making programming abilities more and more necessary for students. Programming skills are highly valued in many areas, including software development, banking, healthcare, and even entertainment. Using a quasi-experimental design, the study set out to design, develop, and examine the effects of the persuasive multimedia learning application programming simulator on university students' performance, engagement, and perceived motivation towards the learning material. The two technologies—Machine Learning Programming Simulator (ML-Programming Simulator) and Non Machine Learning Programming Simulator (NoML-Programming Simulator)—were the independent variable in this study. Students' performance, engagement, and perceived motivation with regard to the course material were the dependent variables. Students' study streams, which are divided by students anxiety, served as the moderator variable. In all, the study involved one university students from the public university in Indonesia. The ANOVA test, a descriptive and inferential statistical technique, was employed to analyze the study's data. The study's findings indicate that, in terms of engagement, performance, and perceived motivation for the subject matter, students who utilized the ML

Programming Simulator outperformed those who used the NoML Programming Simulator. According to this study, university students' performance, engagement, and perceived motivation toward the course material have all increased as a result of the programming simulator application in the classroom environment. In addition, the creation of the ML Programming Simulator for this research helped to expand the pool of mobile and web applications that enhance the effectiveness of the programming course classroom.

CHAPTER 1

INTRODUCTION

1.1 Overview

These days, programming is crucial as information systems are utilized to support individuals in many facets of their lives. Our lives have been significantly impacted by programming and associated technologies, which include web applications, games, social media, online communication, and cloud storage. In order to create a system, programming must take use of technological advancements like artificial intelligence, machine learning, virtual and augmented reality, mobile programming, the Internet of things, and more. By employing technology, we are able to create any form of technology for any kind of area of life. Education is one area where technology is being used.

Students' need for programming skills is growing in the current climate. In a variety of industries, including software development, banking, healthcare, and even entertainment, the ability to program is highly prized. Students that study programming may find a wide range of employment opportunities. Students need to think critically and creatively in order to solve problems (Baist & Pamungkas, 2017). This ability is helpful in many other facets of life than computer science. As technology evolves and becomes more embedded into our daily lives, programming will become an increasingly important ability. Students may future-proof themselves for the employment market by studying programming now. Learning to code can assist students in developing computational thinking, which is breaking down large problems into smaller, more manageable portions. This ability is transferable to many other areas of education and life. Programming may provide students with a creative

outlet by allowing them to design their own programmes, games, and websites. This can aid in the development of their creativity and problem-solving abilities.

When studying programming, there are still several aspects of programming that need to be enhanced in order to investigate certain groups of learners that have not been previously studied by researchers. Lecturers use technology to teach programming should examine important elements such as performance, engagement, and perceived motivation in order to enhance the effectiveness of instruction.

In subsequent sections of this chapter, the researcher provides a comprehensive analysis of the study's background, statement of the problem, research aims, research questions, and the study's importance. This chapter additionally presents a theoretical framework, constraints, and operational specification for the study carried out by the researcher. Finally, the researcher presents a concise overview of this chapter.

1.2 Background of Study

Computer programming course is important for computer major students. Undergraduate students of computer science can get basic knowledge from the course that is delivered in college. Then they can implement that knowledge in business activity after they graduated. Students who are not proficient in the fundamentals of computer programming may find it challenging to get employment, particularly for developers or software developers. Students majoring in computer sciences are required to take the course programming. Nowadays, reading, writing, and fundamental math skills are required of all students majoring in computers, in addition to having a minimum of basic computational thinking abilities (Angeli & Valanides,

2020). They have to seize those skills to be able to survive in computer science major in university, and to be able to keep pace with lecturers' and their study material.

Computer programmers' employment will fall by 10% between 2021 and 2031, according to the US Bureau of Labour Statistics. Despite this drop, they anticipate 9,600 additional computer programming job opportunities per year owing to individuals who will shift to other industries or retire. As businesses continue to automate basic or repetitive processes, the jobs of people who used to do these tasks become redundant. However, this opens up opportunities for those with diverse skill sets. While simpler jobs may be automated, there will be a greater demand for strategic responsibilities. To remain competitive, programmers must upskill in order to complete these duties.

According to the latest report from the Agency for the Assessment and Application of Technology (2023), Indonesia needs around 600,000 programmers by 2025. However, the number of programmers currently available is only around 100,000 people. This condition shows that Indonesia is still far from the target set and requires concrete action to overcome the programmer HR crisis. Therefore, more guidance is required throughout the lecture session to develop programmers who are capable and skilled. The number of highly qualified programmers that come out of higher education is predicted to rise as a result of employing technology in programming course instruction. Additionally, technology may be utilized to support the teaching of programming courses through gamification, evaluation tools, and other techniques like visualization.

Learning programming is a complicated challenge for beginner college students. It isn't pretty much mastering programming language syntax. It additionally

entails the college students` capacity to increase a set of rules that might resolve a given trouble situation (Umar & Hui, 2012). The problems that students face in phrases of knowledge the primary programming due to the fact they're now no longer but acquainted with a specific programming language (Baist & Pamungkas, 2017). In learning programming course, students often make a mistake in making a correct code for a current problem. They do not understand a line that they are writing. They just write the code without even know or understand the code about, for example concept of variables that might be different from one problem to other problem. Students might only memorized the coding, not the logic. So, when lecturer gave other problem but still the same concept of problem, students could not solve it.

It is reported by study done by Alturki (2016) that only some college students reap complete marks whilst many fail to by skip or drop the course (as much as 65%). Nikula et al. (2011) suggested that greater than 30% computer science undergraduate students around the world dropped out or failed in programming course. Malcolm et al. (2010) associated the failure to release the introductory program with the cancellation of the diploma. These bulging dropout and failure rates must be a major problem for institutions, teachers, and students. The desire of students to eschew classes, which are necessary for all computer majors at universities, will cause problems since it will delay graduation owing to failure. In order to assist college students learn programming better, teachers spend a lot of time and effort on lectures and laboratories. However, considering the high costs of failure and attrition, these efforts may seem useless.

Some students of university in Indonesia feel the same problems in learning programming course. The problems that students face in terms of basic knowledge are

because they are not yet familiar with a particular programming language (Baist & Pamungkas, 2017). In addition, students need to become proficient in three interconnected areas: programming language syntax, design, and programming structure. The same challenges are encountered by Indonesian public university students when taking programming courses. Students pursuing a degree in electronics engineering must complete the programming course. Beginning college students have a challenging task while learning programming. It's not only about learning the syntax of programming languages. It also involves the ability of college students to develop a set of guidelines that might potentially address a particular problem (Umar & Hui, 2012). Some basic concepts that students may encounter in the course such as variables, arrays, and iteration. Students tend to engage in some bad programming behaviors in their attempt to pass the assignments (Preliminary Study).

Student engagement refers to the degree of attentiveness, curiosity, interest, optimism, and enthusiasm students demonstrate when learning or teaching. Uninteresting is not limited to programming, as research shows that students in all subject areas often experience a lack of participation that is marked by boredom and alienation (Mann & Robinson, 2009). Pessimist in the context of programming leads to high education dropouts (Bennedsen & Caspersen, 2007). Very few students have found themselves committed as the teacher gave in particular by introducing new concept of programming (Isiaq & Jamil, 2017). Engagement is often not clearly defined in studies of engagement in computer science education, and it is measured through indirect observation (Edwards et al., 2020). The concept of 'student engagement' has expanded to encompass the level of active participation and interest that students demonstrate in their learning, as well as their level of attachment to their classes, institutions, and peers (Axelson & Flick, 2010).

For this study, a public university in North Sumatra is taken for experiment because it has Information Technology and Computer Education study program in it. Information Technology and Computer Education study program is the part of Electronics Engineering major. That study program offers Bachelor Degree in Informatics and Computer. From the first semester to the seventh semester, students learn about the basic programming until developing a complete information system. To build a complete information system, students also learn about the database, programming structure, and the architecture of the system such as language programming and server. Students need to decide the concept of system, whether it is desktop or web application and also application requirement. So, the basic programming concepts from students need to be strong.

Based on the prelimiary study, they need a tool that can help them to learn programming alone. Because from the researcher’s experience, some students will give up easily when they can not find out the question of their answer regarding to learn programming when they are doing self-studying. They also do not know about the use of array, iteration or looping. So, when they meet an error exception that is thrown by the programming editor, they do not know how to handle it correctly. In the age of advanced technology, it has become common to utilize technology as a means of influencing and altering people's behavior and attitudes. Persuasive technology refers to an interactive computer system specifically designed to modify individuals' attitudes and behaviors (Fogg, 2003). Within the field of persuasive technology, there exists a convergence between computers and the act of influencing called captology, which is an abbreviation for 'computer as persuasive technology.' Captology is a field that specifically deals with the development, study, and evaluation of interactive

computer products that are designed to influence and modify people's attitudes or behaviors (Fogg, 2003).

Simulation tools are developed to help students understand the concept of programming, by showing the structure of coding. Hopefully, they can implement the concept and structure of any problem in any different situations. A study was performed by Medvediev (2019) that determined a tool called E-olymp as a practical teaching aid to prepare students for programming courses. It also can be used in sports as an expert as a computer technology teacher, encouraging independent learning and self-improvement. And also study by Zinovieva et al. (2021) had a look at that the use of simulators in the learning process as an additional tool for the formation of expert competencies provides deeper student involvement in the process of writing code and the practical application of existing knowledge in a more relaxed and more practical higher education environment. A study conducted by Staubitz et al. (2016) found that due to the large number of courses, training groups were unable to manually review, comment on, or vet contributor submissions. So, they resulted a CodeOcean that gives the contributors with right automatic remarks in a well timed way and is capable of examine the given programming duties in an automatic way. Based on study by (Budi et al., 2021) stated that simulation software using Proteus contributes significantly to improving student learning outcomes. Teaching complex subjects involves microcontroller programming, and simulation software should be used to enable students to become more passionate, have more experimentation and develop creativity. Students still experience some difficulties in learning programming courses even though there are some previous studies. In previous studies, there has been no simulator that focuses on the PHP programming language. So, students who study PHP programming in their courses cannot use applications that have been developed by

previous studies. So, this study focuses on helping students learn basic knowledge and programming concepts, by understanding the mistakes they may make during their study time.

This study also implements machine-learning knowledge to complete the simulator. The machine learning skills on programming simulators have long been expected to allow simulators to solve problems based solely on their previous expertise, not just the facts stored in them. According to Wang (2019), machine learning can, for example, evaluate data from students over the past year, learning status and outcomes, and produce corresponding written reports. This important data can be used by teachers and learners to investigate learning difficulties and causes and provide the learning programs and support strategies they need.

This study aims to help students in learning programming course. This study develops a machine-learning simulator that is used as helping tools to teach programming. Since in reality, there are some difficulties in learning programming course for students. According to study by Tan et al. (2009) that it's miles crucial to focus on the reasons that lead undergraduates to carry out poorly in analyzing programming. Solution and opportunity analyzing technique can be applied on the way to help them whilst analyzing programming. Simulation tools may be very beneficial in teaching programming, mainly due to the fact their principal reason is to facilitate students` expertise of code execution through guiding them via a sequence of lively techniques (Hundhausen et al., 2002; Mulholland, 2014). To help students' learning in programming, lecturers always try to provide them with a recent teaching methodology. One of the methodologies is using tools to enhance students in learning programming course. There are some tools that could be used for students, that are

visualization, gamification, pair and collaborative technique, robot programming and assessment tools (Kanika et al., 2020).

From the previous studies that have been described briefly above, simulator is the best strategy that could be implemented to help students in learning programming course. This study introduces a novel simulator technology that is anticipated to outperform existing simulators. This technology is more recent in comparison to past studies that created simulators for the purpose of teaching programming. With machine learning in a programming simulator, it is expected that simulator can solve problems based on previous knowledge not only on data that is stored in that simulator. Machine-learning has been extensively used in lots of realistic programs which includes statistics mining, textual content processing, sample recognition, and clinical picture analysis, which frequently depend upon huge statistics sets (Ji et al., 2020; Kumar et al., 2021). From using label information, characteristic choice algorithms are in particular classified as filters or wrapper approaches (L. Sun et al., 2021; Zhao et al., 2020). The wrapper-primarily based totally techniques are typically used to complete the category task (Liu & Zhao, 2012). The essential step consists of classifiers, assessment standards of features, and locating the ultimate features (Al-Tashi et al., 2019). Implementing machine learning in a programming simulator is the new strategy to create a simulator that can solve more various problems in coding for students.

1.3 Problem Statement

Programming is a crucial subject that majors in computer engineering must complete in order to graduate. In actuality, however, lecturers and universities face problems when students fail their programming courses. Teaching methods,

instructors' subject-matter expertise, students' knowledge and proficiency in cryptography, students' self-discipline, the learning environment, and student market resources are some of the factors contributing to low student performance in programming (Alturki, 2016). It is shown also in preliminary study that undergraduate students felt difficult in some topics of programming, looping topic being the most difficult topic of programming course.

Computer anxiety has been demonstrated to influence computer skills (Owolabi et al., 2014). There is a negative link between mathematics anxiety and academic success (Ashcraft & Kirk, 2001). It found that mathematics anxiety impacts a negative attitude towards computers, which is likely to have a significant impact on computer programming performance (Owolabi et al., 2014). Students who are more comfortable when presented with programming-related activities outperform those who are somewhat anxious about programming in any programming course. As a result, there is every reason to suppose that the degree of worry whether computer anxiety, programming anxiety, or mathematics anxiety should be kept to a minimum in order to accomplish effective programming results.

From the researcher's experience in teaching programming and interview from lecturers, students often make a mistake in making a correct code for a current problem. They do not understand a line that they are writing. They just write the code without even know or understand the code about, for example concept of variables that might be different from one problem to other problem. There had been instances wherein college students could research massive fragments of software code through rote, with none or with infrequently any understanding. There had been additionally college students who wrote massive software code with none syntax and logical

testing, which produced a big wide variety of errors, discouraging the scholars from programming altogether (Radošević et al., 2009). They also do not know about the use of array, iteration or looping. So, when they meet an error exception that is thrown by the programming editor, they do not know how to handle it correctly.

If students are willing to learn, the key issue is whether they are motivated (a side effect for instructors is that highly motivated students are more likely to be helpful in teaching). Students need to be motivated to succeed in their studies. It was reported that more students have a negative and neutral preliminary perception that the programming course is easy to understand and has good grades (Zainal et al., 2012).

Technology may be used as an aid to assist teach programming concepts including gamification, evaluation tools, and visualization, among other strategies. Nonetheless, the use of technology in higher education programming courses is still somewhat restricted. Because programming is a fixed skill that requires a lot of practice, most college students learn it by reading analytical texts or paying attention to their instructors, this leads to less than ideal results (Harimurti et al., 2021). Computer programming needs complicated cognitive competencies. Planning is one of the factors required for studying programming courses; problem solving and logical thinking are other important components of the process of becoming proficient in programming. It's important for instructors and educational institutions to consider these factors and create an environment that fosters high performance, engagement, and perceived motivation in a programming course. This can involve designing interactive and relevant learning experiences, providing support and resources, offering opportunities for collaboration and hands-on practice, and fostering a positive learning community.

In order to help lecturer to increase students' outcomes in programming course, simulation tools might be help. To decide whether students need simulator for programming course or not, also becomes a consideration for the lecturers. Since there are some of students that have a good result even without the simulator. Through a comprehensive analysis of existing literature on the programming course environment, the researcher identified that engagement and perceived motivation has a significant impact on performance in this domain. The determining factor is the level of students' involvement in the programming class environment.

Simulation tools can also be very helpful in teaching programming, especially since their main reason is to facilitate students' experience in executing code by guiding them through a series of animated techniques (Hundhausen et al., 2002; Mulholland, 2014). However, studies on simulation in programming courses are still few. (Radošević et al., 2009) presented an application to simulate a C++ programming for students named Verificator. This study provided a tool to visualize coding to be easier for students by identifying the mistakes that student made in their coding. But again, not every university using C++ as the programming language in its course. In public university of Indonesia, they use PHP also as the programming language. And studies on PHP simulators in programming are still few.

In developing a programming simulator, we can use any technology that can support the simulator to perform better. Machine learning can also be implemented to make a simulator performs dynamically based on its previous knowledge. Previously, machine-learning techniques were used only for marketing, finance, telecommunications, and network analysis. In marketing, machine-learning technology is used for classification and related activities. In finance, machine-

learning technology is used for predictive tasks. Related activities in the field of network analysis use machine-learning techniques. In the field of communications, machine learning technology is used for classification, prediction, and espionage tasks (Wang et al., 2009). According to the literature review, there is currently a paucity of research on the use of machine-learning technologies in simulator programming to aid in training. In order to develop a simulator, the machine learning technology on programming simulators have been expected for some time to allow simulators to solve problems based exclusively on their earlier experience, not only the facts stored in them.

1.3.1 Preliminary Study

To ensure that there is a problem in programming course, researcher has conducted a preliminary study that was helped by 31 students of public university of Indonesia. The study used questionnaire to analyze the difficulties in learning programming course from students' point of view. The questionnaire consists of 4 (four) parts that are profile, performance, technology and anxiety. By answering the questionnaire, we could conclude that some students felt difficult in learning programming course. Besides that, the researcher also collected preliminary research data by doing interview to the lecturers to see their points of view regarding teaching and learning programming course. 3 (three) lecturers from programming expertise were chosen to answer questions from the researcher.

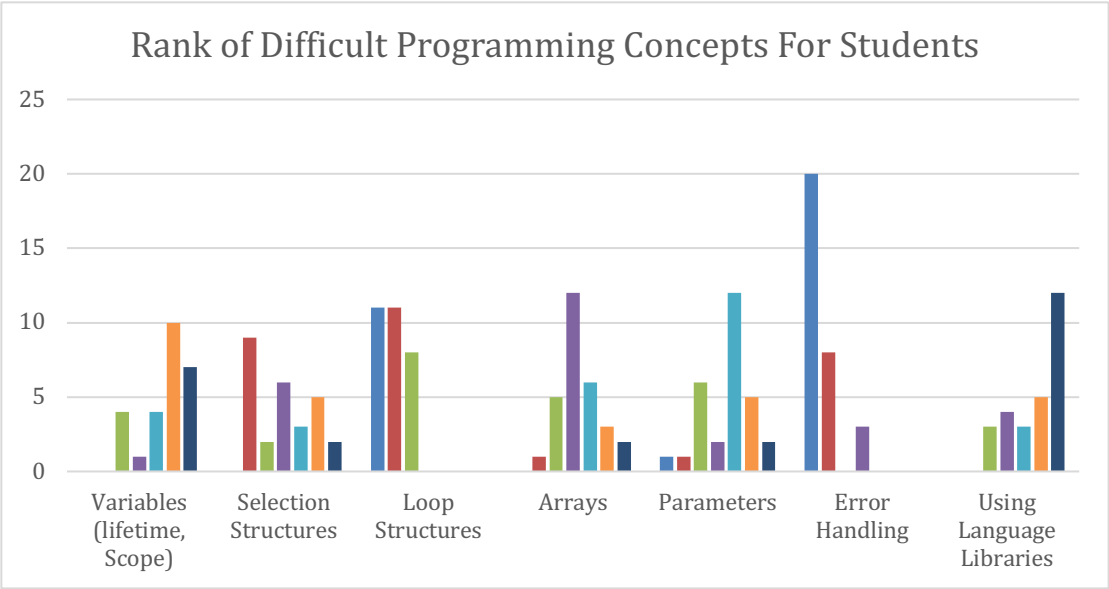

Figure 1. 1 Questionnaire Data - Difficult Concept to Learn for Students

Error management ranks #1 among issues that students find most challenging to master, according to 67% of student correspondents. This is consistent with the information from the student questionnaire on Figure 1.1 about the subjects that they find challenging. Loop structures is the second most challenging topic on the list. After error handling, loop structures were evaluated as the second most difficult topic by 12 out of 30 students taking a programming course. For now, this is rudimentary programming expertise.

In addition, based on the results of the questionnaire Figure 1.2 that has been conducted by this preliminary study, learning programming syntax is the most difficult thing in studying programming courses for students. As many as 67% of student respondents chose it as a problem in studying programming courses. This is related to students' skills in handling errors. If they do not master programming syntax, they will

not be able to handle errors that may appear in their coding. This is in line with the problems that the researcher mentioned earlier.

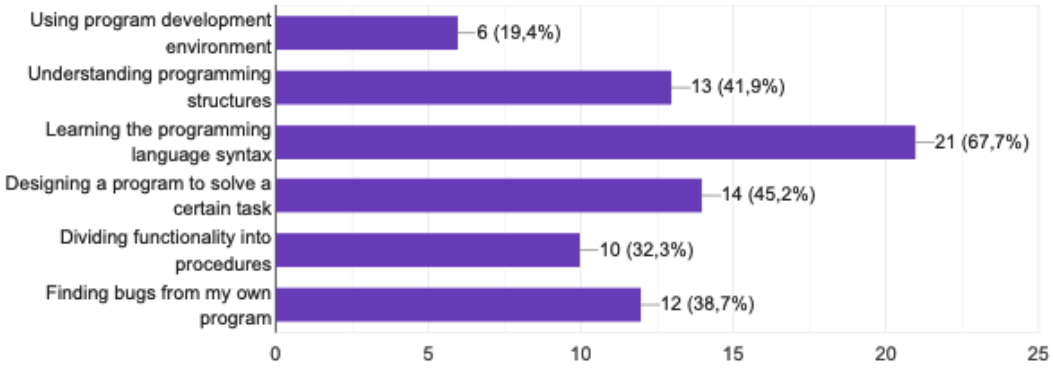

Figure 1. 2 Difficult Issues in Learning Programming Course

Figure 1.3 revealed that 64.5% of student responders find learning programming courses challenging when they study on their own. Similar issues with studying programming courses by college students have also been found in earlier research. Maybe all those students learned was the coding, not the reasoning. Therefore, students were unable to answer the lecturer's other problems that had the same notion. Thus, they require a tool that will enable them to explore programming on their own. Based on the researcher's experience, some students tend to lose up easily when they are unable to locate the answer to a topic pertaining to learning programming during their self-study sessions.

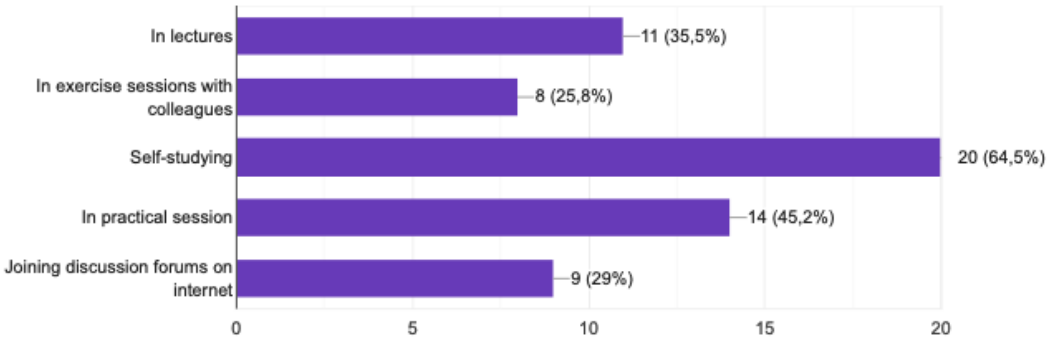

Figure 1. 3 Questionnaire Data - Moment When Student Feel Difficult in Learning Programming

1.4 Research Objective

Objectives of this study are determined to acknowledge research questions. The main purpose of this study is to design and develop a programming simulator that used machine-learning technology that can facilitate learning programming course in order to increase students' outcomes in programming course. Below are the research objectives of this study:

1. To design and develop ML programming simulator to enhance students' performance, engagement and perceived motivation in learning programming.
2. To investigate the effect of ML programming simulator on performance, engagement and perceived motivation with different level of anxiety.
3. To identify the relationship between student engagement and student performance, engagement and perceived motivation, performance and perceived motivation.
4. To propose predictive models of ML programming simulators in the process of learning programming course that is expected to enhance students' performance.

1.5 Research Question

This study is specially designed to answer this question.

1. Are there any difference in (a) performance, (b) engagement and (c) perceived motivation between students that used machine-learning simulator and noML programming simulator?
2. Are there any difference in (a) performance, (b) engagement and (c) perceived motivation between high anxiety students and low anxiety students that used machine-learning simulator and noML programming simulator?
3. Is there any relationship between students' engagement and students' performance in learning programming course that used machine-learning simulator and noML programming simulator?
4. Is there any relationship between students' engagement and students' perceived motivation in learning programming course that used machine-learning simulator and noML programming simulator?
5. Is there any relationship between students' performance and students' perceived motivation in learning programming course?
6. How is the predictive model of using ML programming simulators in the process of learning programming course to classify students that use ML programming simulator and noML programming simulator?

1.6 Hypotheses

Based on research question, there are some hypotheses that can be produced.

$H_0$1 : There is no difference in (a) performance, (b) engagement and (c) perceived motivation between students that used machine-learning simulator and noML programming simulator.

$H_0$2 : There is no difference in (a) performance, (b) engagement and (c) perceived motivation between high anxiety students and low anxiety

students that used machine-learning programming simulator and non-machine learning programming simulator.

$H_0$3 : There is no relationship between students' engagement and students' performance in learning programming course that used machine-learning programming simulator and non-machine learning programming simulator.

$H_0$4 : There is no relationship between students' engagement and students' perceived motivation in learning programming course that used machine-learning programming simulator and non-machine learning programming simulator.

$H_0$5 : There is no relationship between students' performance and students' perceived motivation in learning programming course that used machine-learning programming simulator and non-machine learning programming simulator.

1.7 Significance of The Study

This study develops a machine-learning simulator that can help lecturers teach programming and help students understand more about basic programming concepts. This simulator is an additional tool in teaching and learning process in higher education. The simulator uses machine learning knowledge to make it more dynamic in solving problems in helping students learning programming course.

This study also measures the effectiveness of using simulator. The researcher compares the performance, engagement and perceived motivation of students that used simulator in programming course. Furthermore, students are classified into two

treatment groups that use machine-learning simulator and noML simulator. This study defines the best formula to teach programming concept in class.

The findings of this study contributes in giving the best strategy in teaching and learning programming course in order to produce skilled programmers from higher education to business environment. Lecturers from public and private universities of Indonesia can adapt this strategy to improve performance, engagement and pereived motivation of computer science and information technology students. The simulator that is developed through this study can also be used as a tool in teaching and learning basic concept of programming course.

The results of this study can also be adopted by the Ministry of Higher Education as an appropriate way to teach the concept of computer programming at Indonesian universities, which can deepen their understanding of programming and improve programming performance. In addition, this study may serve as the basis for further research on the impact of methods and educational strategies used to improve college learning.

1.8 Research Framework

Based on Figure 1.4, this study develops a machine-learning simulator and noML simulator as the comparison treatment that are used for students in learning programming course. We check the difference between a group of students that use a machine-learning simulator in programming course and a group of students that use a noML simulator in programming course. The parameters to see the difference are performance, engagement and pereived motivation of students. Besides that, students

are chosen randomly by their cognitive and anxiety. Anxiety is known to contribute to programming ability (Owolabi et al., 2014). Programming anxiety has been described as a worry of doing programming, or fearing the opportunity of the use of coding. Anxiety is maintained through incorrect or dysfunctional appraisal of a situation. Therefore, programming anxiety takes place for college students due to an incorrect evaluation in their capacity to analyze laptop programming (Connolly et al., 2009). Cognitive turned into discovered to be a full-size getting to know function that ought to be considered whilst the usage of virtual video games to study programming (Theodoropoulos et al., 2016).

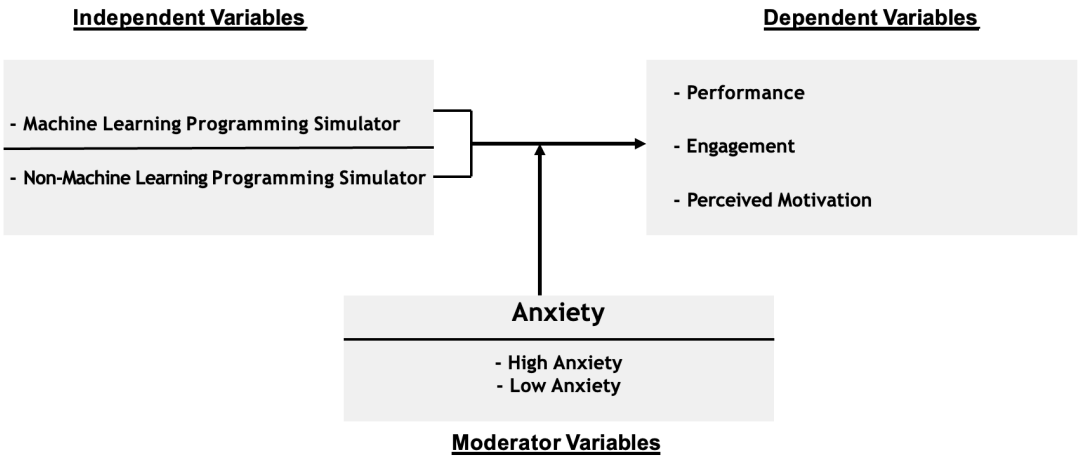

Figure 1. 4 Research Framework

1.9 Theoretical Framework

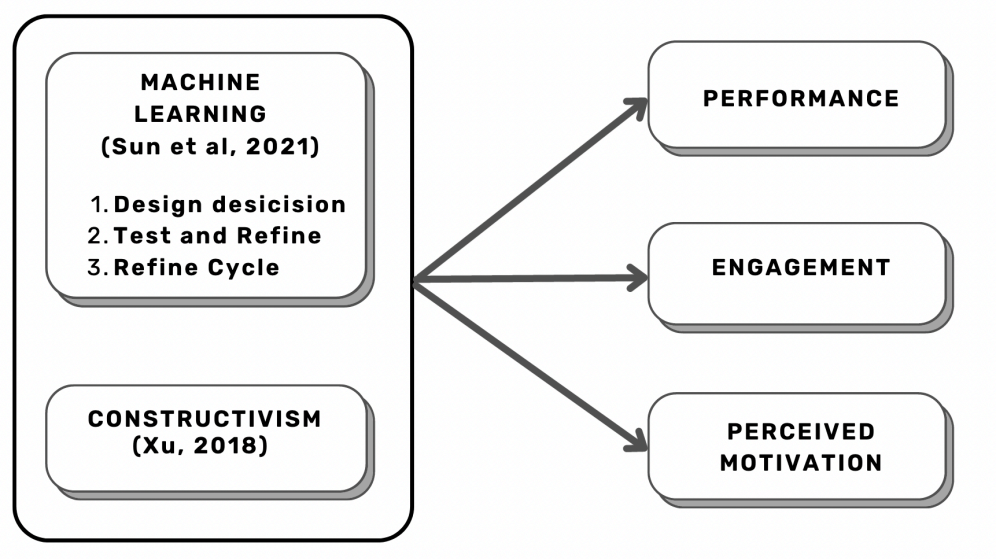

Figure 1. 5 Theoretical Framework

The theoretical framework of this study showed in Figure 1.5. Machine Learning theory by Sun et al. (2021), simulator technology by Isiaq & Jamil (2017), constructivism theory by Xu (2018), students' engagement by Dixson (2015), students' performance by Alturki (2016) and students' perceived motivation by Keller (2018) are the fundamental theories used in this study. The development of ML programming simulator as the strategy of teaching and learning programming course in this study followed the theories of machine learning theory by Sun et al. (2021) that is planted on the simulator by Isiaq & Jamil (2017) as the technology and method in developing a tool in helping students using the constructivism strategy by Xu (2018) based on their engagement using study by Dixson (2015), motivation study by Berg (2005) and performance study by Alturki (2016) in programming course.

Strategy that is used to implement machine-learning technology in learning environment in this study is constructivism. Based on this strategy, knowledge is built

by people through interaction with environment and each other, rather that waiting to be discovered (Xu, 2018). In Xu's (2018) constructivist learning environment, there has been a significant shift in the roles of teachers and students as compared to the traditional teaching style. The constructivist learning paradigm prioritizes the perspective of students, regarding them as the primary agents of cognition and as active creators of information and its significance. Teachers solely facilitate and foster the process of students' production of meaning and are not obligated to directly transmit knowledge to students. The constructivist learning theory promotes a student-centered approach to learning, with teachers providing guidance. Machine learning technology serves as a tool to assist pupils in acquiring new knowledge. The four components, namely teacher, student, teaching content, and learning media, each play distinct roles and exhibit interdependencies when compared to conventional teaching approaches.

The machine learning fits into the study and acted as an intelligent tool for calculating new learning cases or problem similarity values presented to the application in database-stored cases. The problem that is inputted by students is solved using the same knowledge as in the previous case. Machine learning is used in application development because they can intelligently provide solutions for a particular problem based on previous data. Machine learning in this study provides immediate automatic feedback, reducing the burden on crowded science classroom teachers. Based on study by Sun et al. (2021) theory, this study makes the first decision to create an automated feedback prototype based on established learning theory, available research, and previous experience in developing feedback. Next, it tests and refine the feedback system based on an analysis of the data collected in the classroom.

Repeating the cycle of design, testing and improvement can complete an automated feedback system.

In constructivism study by Xu (2018), students tend to be active builders of knowledge rather than passive receivers of external stimuli. Teachers are not knowledge teachers, but facilitators of the educational process. The knowledge given by the textbook is no longer the content of the teacher's teachings, but the purpose of building a positive meaning for the students. The learning media is not a means of helping teachers convey knowledge, but a means of creating a co-learning situation. Conversation is used as one of the student's collaborative and exploratory learning tools. The four elements of teacher, student, teaching material, and learning media have their own roles and interrelationships compared to traditional teaching methods.

In this study, constructivism is implemented as a strategy to help students create co-learning situation. Students use machine-learning simulator to learn programming course by themselves, besides that students also join in-class programming course with a lecturer. We conduct four elements in this study to support constructivism strategy; those are lecturer, students, teaching material and machine-learning simulator as the learning media. All of the elements related to each other to create co-learning situation.

According to study by Alturki (2016), the most reliable way to get a relatively objective measurement is to use student grades before and during the course. Students are assessed and graded using a variety of instruments: midterm exams, quizzes, homework, projects, attendance, lab exams, and final exams. Using these assessment items to assess overall performance is very important to predict student performance and provide appropriate support. Past college grades, high school grades, and standardized tests can also help predict performance.

Student involvement is essential for student learning, particularly in the digital world, where learners usually feel excluded and detached. As a result, instructors and researchers must be capable of evaluating student participation (Dixson, 2015). Student engagement is often defined as the extent to what learners actually connect with course materials, other learners in the course, and the teacher by researching, discussing, and engaging.

An observe discovered that motivation affects the mindset of a students in which the encouraged students extrade to a nice mindset at the same time as the much less encouraged college students may be converted right into a bad mindset (Berg, 2005). Positive attitudes are; observe tough and now no longer surrender even fail. Students are extra encouraged and live encouraged, pushed with the aid of using intrinsic rewards together with optimistic complaint than extrinsic, together with excellent grades (Ryan & Deci, 2000). This is due to the fact the intrinsic rewards provide extra pride than the extrinsic rewards

1.10 Limitations

The scope of this study was limited to students of computer and information technology. Sample data was taken from selected public university in North Sumatra of Indonesia. This study focused on the teaching methods used and was not aimed at examining the outcomes of programming course between genders. Learning content that is used in this study was limited to basic programming concepts of PHP. Therefore, there are two courses – Web Programming and Algorithm and Basic Programming that selected to be studied. The topic selected for this study was about the "error handling" that forms the basic components of the basic web-programming