# Identifying the Factor of Mathematical Reasoning That Affects the Ability to Programming Algorithm

Sulis Janu Hartati[1], Anik Vega Vitianingsih[2], Neny Kurniati[3], Sulistyowati[4] & Muhajir[1]

[1] Faculty of Education, Universitas Dr. Soetomo, Surabaya, Indonesia

[2] Informatics Departments, Universitas Dr. Soetomo, Surabaya, Indonesia

[3] Department of Informatics Engineering, Universitas Nahdatul Ulama, Sidoarjo, Indonesia

[4] Information System Study Program, Universitas Dinamika, Surabaya, Indonesia

Correspondence: Anik Vega Vitianingsih, Informatics Departments, Universitas Dr. Soetomo, Surabaya, Indonesia. E-mail: vega@unitomo.ac.id

## Abstract

This paper examines the limited proficiency to engage in programming algorithms among university students in information technology and information system in several universities across Surabaya, Indonesia. The purpose of this research is to find the most influential factor in learning programming algorithm using a quantitative approach. The research subjects were second-semester information technology students in several private universities in Surabaya, Indonesia. The research instruments were mathematical reasoning and basic algorithm programming test. Mathematical reasoning tests incorporated linear algebraic, basic calculus, and mathematical logic. The data analysis used was variant-based Structural Equation Modelling, also known as Partial Least Squares - Structural Equation Modelling based on Smart-PLS 3. With $\alpha = 5\%$, the research results conclude that mathematical reasoning positively influences algorithm programming ability with an R score of 0.999, and that the most influential variable among mathematical reasoning abilities was algebra with an R score of 0.732.

**Keywords:** analysis of factors, quantitative approach, smart-pls, programming algorithm, mathematical reasoning

## 1. Introduction

### 1.1 Introduce the Problem

This study was motivated by the findings of Hartati (2014). A review of the literature shows that the algorithm programming is a course that is not easy for the students majoring in Information System and Informatics in several private universities in Indonesia (Marisa, 2008; Sarah-Bibi, 2015; Arifin, 2017). These results indicate that since 2008, the algorithm programming has been a challenge for the lecturers in several private universities in Indonesia. The learning objectives of programming algorithms include making computational algorithms, which is a skill that must be taught in schools (Weintrop et al., 2016). Mohaghegh (2016) recommends computational thinking as a core topic in basic and secondary education. Meanwhile, according to Hsu, Chang, & Hung, (2018), computational thinking has been an interesting discussion in various countries, including in Australia and North Korea, since 2006. Computational thinking is the everyday living skills needed by everyone, not just programming skills used commonly by computer scientists (Wing, 2006, 2008). According to (Haseski, Ilic, & Tugtekin, 2018), computational thinking is an important part of learning in the 21st century. (Weintrop et al., 2016) assigned 10 initial skills in computational thinking that must be taught in school. One of them is making an algorithm. Based on these considerations, the difficulty in understanding algorithms for students in several private universities in Indonesia is an interesting problem that requires a solution. To this end, we seek to identify the factors of mathematical logic that affect the ability to programming algorithms (Pendidikan, Kebudayaan, & Indonesia, 2018). However, the fact is that the mathematical reasoning ability of Indonesian students is still far behind that of other countries. In 2018 Indonesia ranked 72 out of 78 countries with a score of 379 (OECD Better Policies For Better Lives, 2019). While the average value of PISA participants from 78 countries is 489 (OECD Better Policies For Better Lives, 2019). Based on the results of the national examination evaluation conducted by the Indonesian Ministry of Education and Culture, the national exam scores for high school mathematics subjects decreased, in 2017, 2018, and 2019 namely 37.61, 33.86, and 34.94, respectively. This value is far below the standard set by the

Indonesian Ministry of Education and Culture, which is 70-85 (Utomo & Narulita, 2018). This is a challenge for private university lecturers organizing Informatics Engineering Study Programs to continue to provide greater opportunities for high school graduates to study computer science.

*1.2 Explore the Importance of the Problem*

Hartati (2014) mentioned that the characteristics of the knowledge learned in programming algorithms are conceptual and metacognitive. Therefore, prospective new students must have above-average mathematical thinking skills. This explains why the study programming algorithm is included in the category of natural sciences in the national education system in Indonesia. However, these requirements are not a major concern in several private universities in Indonesia. With the consideration that the college should give the opportunity to every citizen to continue to higher education in accordance with his choice. As mandated by the Constitution Article 31, every citizen is entitled to education (Mohammad, 2010). Consequently, the lecturer or educator of algorithm programming must think hard to solve the challenge. The general purpose of learning to programme algorithm is to provide a computational rationale, specifically in the stages of turning certain inputs into specific outputs (Kachroo, Krishen, & Agarwal, 2017; Eglen, 2009; Ou, 2005) using computers. Computational thinking is presented in the form of a flowchart and/or pseudo code. Hartati (2017) concluded that making programming algorithms requires creative mathematical logic to compile a set of reasoning by deduction, which is a systematic collection of several propositions. According to Johar, Yusniarti, and Saminan (2018), Indonesian children are not accustomed to the training of logical thinking, particularly proportional reasoning. Tourniaire and Pulos in 1985 argued that proportional reasoning is reasoning about the equality of two ratios (Johar, Yusniarti, & Saminan, 2018). However, according to Septia, Prahmana, Pebrianto, and Wahyu (2018), logical ability can be enhanced with continuous training. Based on these facts, mathematical reasoning training is needed for Informatics Engineering study students whose majority of students has low mathematical abilities. For this reason, mathematical reasoning factor analysis is needed that can improve the ability to make basic level computational algorithms.

*1.3 Describe Relevant Scholarship*

This paper aims to identify the most influential mathematical logic that increases capabilities to programming algorithms. This discussion is important, as part of an effort to improve the ability of the programming algorithm because it cannot stand alone. Especially for subject participants who have mathematical abilities below 70. It includes the logical ability to solve problems in linear algebra, calculus, and mathematical logic. There are two thoughts underlying the selection of those three abilities. First, the three materials are the basic mathematics material taught in the first semester, as well as in programming algorithms. Second, based on Plato's opinion contained in Attridge (Attridge & Inglis, 2013), people who have the competency to calculate in general understand other knowledge faster. If they are trained in arithmetic, then the ability to understand knowledge is expedited. Calculation competencies are studied in basic calculus and algebra. In basic calculus, there are calculations on number theory. In algebra, there is a discussion on matrix operations.

Several studies relating to the problem of improving the ability of programming algorithms are still being discussed today in Indonesia and elsewhere. Maulana (2017) believes that logic and algorithms are abstract material, making it difficult for students to learn. He overcame this problem by creating learning media called EL-algorithm, such as modules, Edmodo, and the like. However, it does not address the fundamental difficulty of students which is increasing the ability of abstraction. In addition, there is no information concerning the research that was conducted and the research subject. Likewise, Nuraini (2015) argues that making the matrix multiplication operation algorithm is considered difficult for some people to understand. The difficulties include arranging the steps, the logic sequence, the decision making, and the arithmetic process. She accomplished this by making a flowchart and pseudocode for matrix multiplication. The solution from the perspective of learning theory does not solve the problem. Whereas Combéfis et al. (2017) argue that computer scientists must have the skills to design efficient algorithms. This means that according to them, computer scientists are not only able to make algorithms, but also must produce efficient algorithms. The efficiency is measured using the analysis called big O, or running time (McConnell, 2001; Kozen, 1991) in the form of the mathematical functions found by the algorithm generator. Thus, high-level mathematics logic capability is necessary for programming algorithms.

Hwang, Kuo, Yin, and Chuang (2010) developed a learning system that can detect the behaviour of the students in the context of learning in the real world with the help of technology (sensors) supported by Heuristic Algorithms. The students are guided to operate the objects of the real world through the digital world so that a learning pathway model is found according to the students' needs. However, what was developed was elementary ecological learning and was carried out outside Indonesia. Thus, there are no factors that affect the ability to increase the

capacity to programming algorithms. Inglis and Attridge (2016) argue that studying advanced mathematics will develop conditional logic, although not for all students who are forced to study advanced mathematics. Thus, to improve the ability of basic level programming algorithms for students with mathematical abilities below 70 years requires the analysis of basic reasoning mathematical factors that affect the ability to compile a basic level programming algorithm. This is important because it is very difficult to find publications that discuss the topic.

Some studies use SEM (Stands for Structural Equation Modelling) to perform factor analysis. According to Khosrow-Pour (Schmalbach & Maza-Avila, 2018), the aim of this technique is to explain the relationship between latent variables and observed variables. SEM is a standard for analyzing cause-effect relationships between latent constructs (Hair, Ringle, & Sarstedt, 2011). According to Gefen, Straub, and Boudreau (Achjari, 2004), the use of SEM assumes that the relationship between items and their constructs and relationships between constructs is linear. While for Sarstedt, Hair, Cheah, Becker, & Ringle (2019), SEM is a high-level construction, which facilitates modelling of the construction at a more abstract high-level dimension and a more concrete low-level subdimension. There are two methods, namely covariant-based called CB-SEM and variant-based called VB-SEM, which were introduced by Wold in 1966 (Hamdollah & Baghaei, 2016), with the name PLS-SEM (Partial Least Square - SEM). One of the differences between the two is that the CB-SEM requires a large sample size, a minimum of 200 (Schmalbach & Maza-Avila, 2018), while PLS-SEM can be used for small samples. PLS-SEM is widely used in research in the fields of business, management, accounting studies, computer science, social science, engineering, and social science. As for education, such as mathematics has not been as much as mentioned above. Hamdollah and Baghaei (2016) use PLS-SEM to investigate factors that contribute to the apathy dimension in adulthood. These factors include cognitive, emotional, and behaviour. They found that various apathetic factors related to cognition to varying degrees. Shahijan, Rezaei, and Amin (2016) investigated the impact of perceived brand orientation, intercultural friendship, and university reputation in international classes on the satisfaction and intentions of continuing student studies in higher education in Malaysia. Their findings indicate that brand orientation and university reputation positively influence satisfaction and intentions to continue studies, intercultural friendships affect intentions to continue studies but not with satisfaction. Hamdollah and Baghaei (2016) studied the factors that contribute to the psychological and sociocultural adaptation and welfare of students in the international class of the host country (Malaysia). Their model shows that the length of stay in the host (Malaysia) negatively affects the attachment to the host. Furthermore, positive attitude adjustment influences psychological and sociocultural adaptations. Hauser, Paul, & Bradley (2012) studied the relationship between changes in computer self-efficacy and computer anxiety and their effects on performance on online computer-related and face-to-face education tasks. They concluded that, in face-to-face education, ongoing dialogue is needed with students to enhance learning, logical organizing is essential in online media. Al-Azawei and Lundqvist (2015) investigated factors related to student satisfaction in online Arabic courses. Their findings indicate that learning styles and gender do not have a significant effect on perceived usefulness, perceived ease of use, and of course satisfaction.

Given the small sample in this study, PLS-SEM was chosen, whereas for data processing, Smart-PLS was chosen. According to Garson (2016) and Monecke and Leisch (2012), Smart-PLS is free software commonly used for path models. The software was developed by Hair et al. (2011), Monecke and Leisch (2012), Wong (2013), Al-Azawei and Lundqvist (2015), Hamdollah and Baghaei (2016), Garson (2016) and Sarstedt et al. (2019) on the Java Eclipse platform. In other words, Smart-PLS is the leading software application for PLS-SEM.

*1.4 State Hypotheses and Their Correspondence to Research Design*

Based on the explanation above, this paper seeks to determine the level of influence of mathematical reasoning abilities on the ability to make programming algorithms. It analyses the most influential mathematical logic on the ability to create a programming algorithm. In this research, mathematical logic refers to algebra, calculus and the logic of mathematics. The hypothesis tests the extent mathematical logic influences the ability to create a computing algorithm, and the results show that it has a very significant effect on the ability to create a computational algorithm.

This results showed: (1) there is a correlation between algebraic and programming algorithm ability with a correlation score of 0.935, (2) there is a correlation between mathematical logic and programming algorithm ability with a correlation score of 0.796, (3) there is a correlation between basic calculus and programming algorithm ability with a correlation score of 0.664, (4) there is a correlation between mathematical reasoning and algebraic ability with a correlation score of 0.624, (5) there is no correlation between mathematical reasoning and basic calculus ability, where the correlation score is only 0.125, (6) there is a correlation between basic calculus and algebraic ability with a correlation score of 0.659.

## 2. Method

### 2.1 Procedures in the Study

Flow in Figure 1 is a research process that begins with the selection of research samples. Then, proceed with tests of mathematical logic abilities, algebra, basic calculus, and programming algorithms. The ability test results are then processed and analyzed using smart-pls software. The results of testing are obtained (i)the Average Capability of Mathematical Logic, Algebra, Basic Calculus, and Programming Algorithms, (ii) the value of R Square for the ability to Programming Algorithms, algebra, and calculus, (iii)correlations between the variables programming algorithm, algebra, calculus, and logic of mathematics, and (iv)the Path Coefficient of model.

### 2.2 The Conceptual Model of Research

There are six hypotheses that are tested: (1) $H_1$: Proficiency in algebra influences algorithm programming proficiency, (2) $H_2$: Proficiency in the logic of mathematics influences algorithm programming proficiency, (3) $H_3$: Basic arithmetic capabilities influence algorithm programming proficiency, (4) $H_4$: Proficiency in the logic of mathematics influences algebra proficiency, (5) $H_5$: Proficiency in the logic of mathematics influences basic arithmetic proficiency, (6) $H_6$: Proficiency in basic arithmetic capabilities influences algebra proficiency. The conceptual model of this hypothesis is presented in Figure 2.
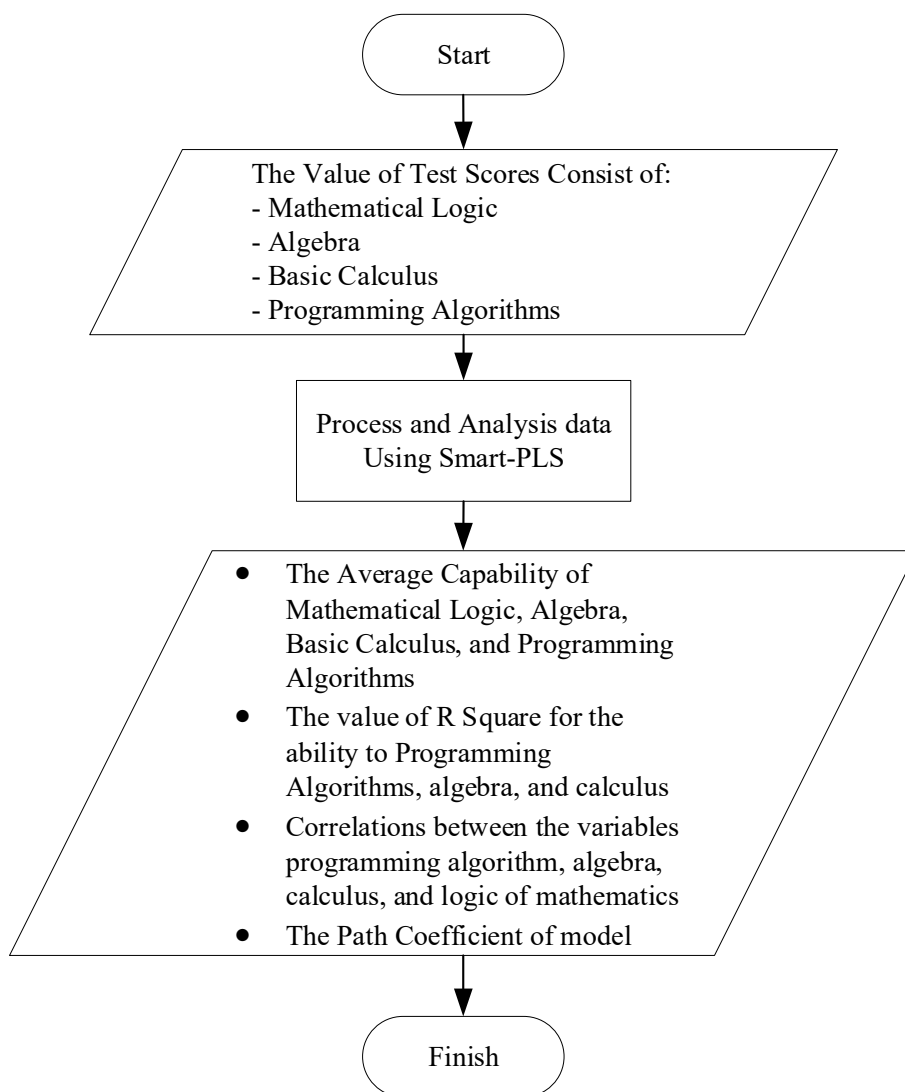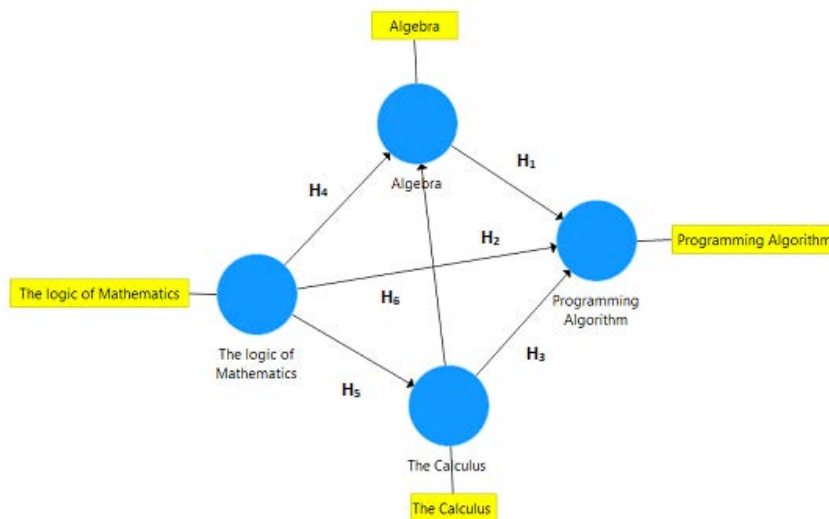


Figure 1. Flow of the research

Figure 2. The conceptual model of research

## 3. Results

### 3.1 General Description of Data

The sample comprises of 54 students studying linear algebra and programming algorithms. At the beginning of semester 2, they were given a test of basic calculus and logic of mathematics, which is a semester 1 course. At the end of semester 2, they were given an algebra test and a programming algorithm. The test results are presented in the form of a graph of the average ability of algebra, logic of mathematics, calculus, and programming algorithms shown in Figure 3.
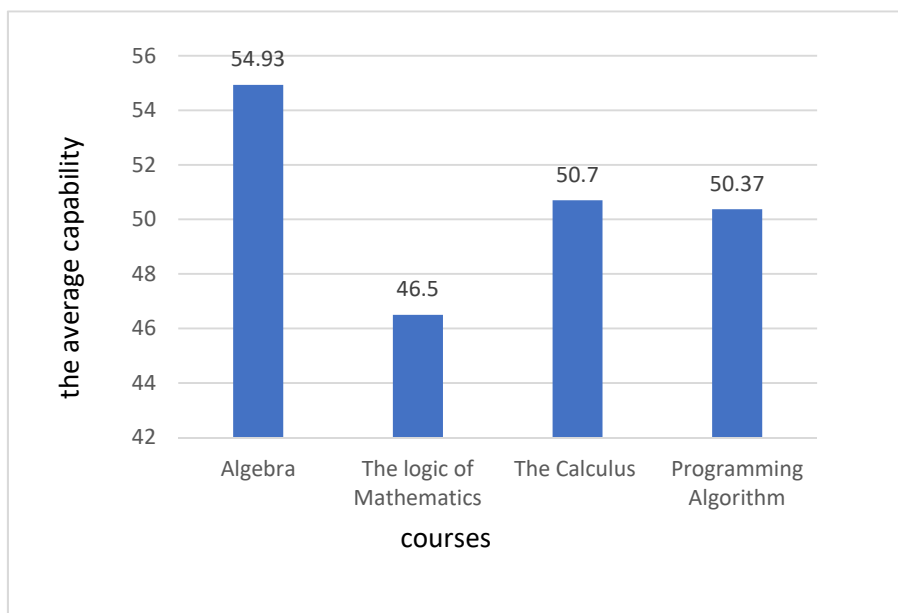


Figure 3. Graph of the value of the average algebra, logic mathematics, calculus, and programming algorithm

The result of the processed data showed that the average score for mathematical reasoning was below 55. The highest average score was in algebraic ability, and the average score is 54.93, with a standard deviation of 10.9. It is followed by a programming algorithm which scored an average score of 50.37 with a standard deviation of

10.13. The third is basic calculus with an average score of 50.7 and standard deviation of 9.54. The lowest average score is in mathematical logic ability, where the average score is 46.5, with a standard deviation of 16.36.

The algebra test material includes addition, subtraction, and matrix multiplication operations. Matrix operations provide the basic ability to understand operations on variable arrays, both one-dimensional (1-D) and two-dimensional (2-D). The test results show that students cannot perform arithmetic operations for array variables. The error occurred because students did not understand the terms of the multiplication operation on two matrices.

The calculus test questions consisted of basic arithmetic operation, equality, and inequality. The equation is divided into two parts, explicit and implicit. Most of the mistakes in basic arithmetic operations were in the operations with a rational number less than one. It resulted in an incorrect equality solution. For inequality problems, most of the students were unable to give a set of solutions. Understanding equality and inequality theory are basics for differentiating "command" or "instruction" and "condition" to programming algorithms. Students were inexperienced to change implicit equality into explicit equality. Creating "command" and "instruction" in the program's algorithm is related to the ability to explicitly make algebraic equations. Meanwhile "condition" is related to the ability to determine the implication and relation of two or more prepositions.

The logic of mathematics tests consisted of true proposition value problems. In relation to calculus, it leads to the creation of value example that includes solution to inequality. In relation to programming algorithm, it is understanding of condition concept. The algorithm programming test seeks is to determine the students' ability to understand the basics of algorithms that were presented in flowcharts and pseudocode.

*3.2 Correlation Analysis*

Based on the results of the data processing by utilizing Smart-PLS, the R Square model is obtained and presented in Table 1. R Square model's function is to elaborate on the independent variable's contribution to dependent variables (Hamdollah & Baghaei, 2016). In this research, the dependent variable is the programming algorithm. Independent variables are algebra, calculus, and logic of mathematics. The results of the calculation are shown in Table 1.

Table 1. The value of R Square for the ability to programming algorithm, algebra, and calculus

| Elective Courses | The value of R Square |
|---|---|
| Programming Algorithm | 0.999 |
| Algebra | 0.732 |
| Calculus | 0.016 |

Table 1 showed that, altogether, algebraic proficiency, mathematical reasoning proficiency, and basic calculus proficiency contributed to programming algorithm proficiency by 99.9%. Mathematical logic and basic calculus collectively contributed to algebraic proficiency by 73.2%. Whereas mathematical logic has a very low contribution to calculus proficiency at only 1.6%. Correlations between the variables (capability) programming algorithm, algebra, calculus, and mathematical logic are presented in Table 2.

Table 2. Correlations between the variables programming algorithm, algebra, calculus, and logic of mathematics

| The variables | Programming Algorithm | Algebra | The Calculus | The logic of mathematics |
|---|---|---|---|---|
| Programming Algorithm | 1.000 | 0.935 | 0.664 | 0.796 |
| Algebra | 0.935 | 1.000 | 0.659 | 0.624 |
| Calculus | 0.664 | 0.659 | 1.000 | 0.125 |
| The Logic of Mathematics | 0.796 | 0.624 | 0.125 | 1.000 |

According to (Sarwono, 2006), the criteria to explore the correlation of two variables are as follows: (1) if the value of correlation coefficient is $r = 0$, it means there is no correlation between the two variables, (2) if the value of correlation coefficient is $0 < r \leq 0.25$, it means they have very weak correlation, (3) if the value of correlation coefficient is $0.25 < r \leq 0.5$, it means they have sufficient correlation, (4) if the value of correlation coefficient is $0.5 < r \leq 0.75$, it means they have strong correlation, (5) if the value of correlation coefficient is $0.75 < r \leq 0.99$, it means there is very strong correlation, (6) if the value of correlation coefficient is $r = 1$, it means they have the perfect correlation.

Table 2 showed that there is a correlation between the ability of algebra with the capability to programming algorithm by 93.5%. There is a correlation between the ability of the calculus with the capability to programme algorithms by 66.4%. A correlation between the ability of the logic of mathematics with the capability to programme algorithms by 66.4%. There is a correlation between the ability of algebra with the capability of calculus by 65.9%. There is a correlation between the ability of algebra with the capability of mathematical logic by 62.4%. Whereas the correlation between the ability of logic of mathematics and calculus is very small with 12.5% (which is a very weak correlation). This is in accordance with the value of R Square, in which the contribution of the variables that affect the ability of calculus is very small at 1.6%. It means that the correlation between the ability of mathematical logic and the ability of calculus is very weak.

Table 3. The path coefficient

| The variables | Original Sample (O) | Sample Mean (M) | Standard Deviation (STDEV) | T Statistics (\|O/STDEV\|) | P Values | Description |
|---|---|---|---|---|---|---|
| Algebra→ Programming Algorithm | 0.393 | 0.333 | 0.118 | 3.339 | 0.001 | Influential |
| The Calculus → Programming Algorithm | 0.342 | 0.391 | 0.148 | 2.316 | 0.021 | Influential |
| The Calculus → Algebra | 0.591 | 0.492 | 0.286 | 2.068 | 0.039 | Influential |
| The Logic of Mathematics → Programming Algorithm | 0.508 | 0.626 | 0.256 | 1.982 | 0.048 | Influential |
| The Logic Of Mathematics → Algebra | 0.550 | 0.574 | 0.223 | 2.468 | 0.014 | Influential |
| The Logic of Mathematics → Calculus | 0.125 | 0.044 | 0.369 | 0.338 | 0.736 | No effect |

According to Nugroho (2014), Table 3 can be arranged into a mathematical equation as follows equation (1), (2), and (3).

$$Programming\ algorithm = 0.33\ Algebra + 0.39\ Calculus + 0.63\ Logic\ of\ Mathematics \tag{1}$$

$$Algebra = 0.49\ Calculus + 0.578\ Logic\ of\ Mathematics \tag{2}$$

$$Calculus = 0.04\ Logic\ of\ Mathematics \tag{3}$$

## 4. Discussion

With $\alpha$ of 5%, if P values are less than 5%, it can be concluded that a variable may affect other variables. Based on Table 3, the result is that the ability of the algebra, the ability of calculus, and the ability of logic of mathematics have a positive effect on the ability to programming algorithm, which means that every improvement on the proficiency of algebra, calculus, and logic of mathematics will improve the ability to programme algorithms. The ability of calculus and logic of mathematics also has a positive effect on the ability of algebra, which means that any increase in calculus proficiency and logic of mathematics will improve algebra proficiency. Whereas mathematical logic proficiency does not affect calculus proficiency. The graph of coefficient path distribution is supported by Figure 4.

a. Algebra proficiency and programming algorithms



b. Calculus proficiency and programming algorithms



c. The logic of mathematics proficiency and
programming algorithm



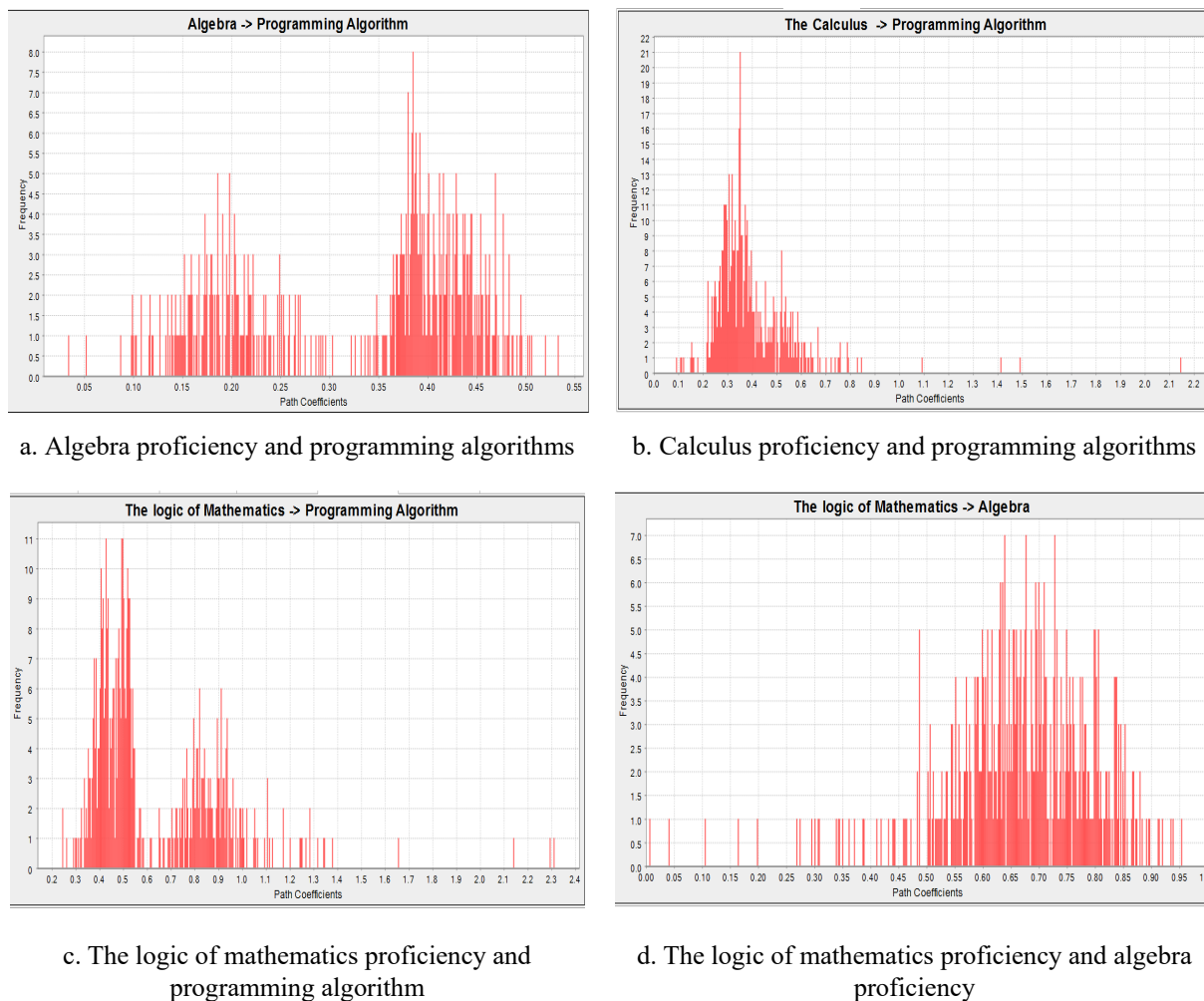d. The logic of mathematics proficiency and algebra
proficiency

Figure 4. The graph of coefficient path distribution

Figure 4.a. shows the highest frequency of coefficient path distribution of algebra proficiency and its influence on programming algorithms is around 0.4, meaning that the ability of algebra affects the ability to programming algorithm. The graph of the coefficient path distribution of the basic arithmetic ability and its influence on programming algorithms in Figure 4.b. shows the highest frequency is around 0.35, meaning that calculus proficiency affects the ability to programming algorithms. The graph of coefficient path distribution of the ability of logic of mathematics compared to the ability of programming algorithm in Figure 4.c. shows the highest frequency is around 0.5, meaning that the logic of mathematics proficiency affects the ability to programming algorithms. The graph of coefficient path distribution of the ability of the logic of mathematics to the ability to algebra in Figure 4.d. shows the highest frequency is around 0.55, meaning that programming logic proficiency affects algebra proficiency.

The graph of coefficient path distribution of the logic of mathematics proficiency and its influence on basic calculus capabilities in Figure 5 shows the highest frequency is located around the 0.12, meaning that the logic of mathematics proficiency does not affect basic calculus proficiency.
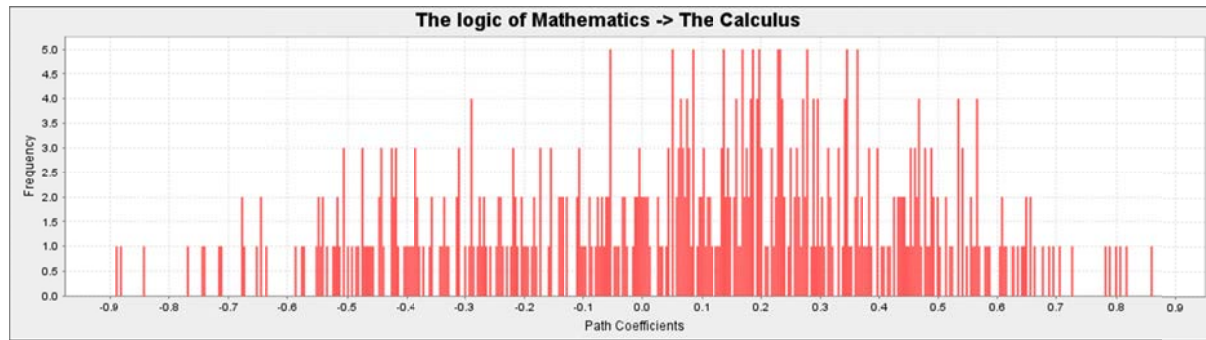
Figure 5. Graph of coefficient path distribution of logic of mathematics proficiency and its influence on basic calculus capabilities

The indirect relationship between variables is presented in Table 4. An indirect relationship is when there is a third variable that mediates the relationship between the two variables (Rigdon, Becker, & Sarstedt, 2019).

Table 4. Indirect effects between variables

| The Variables | Original Sample (O) | Sample Mean (M) | Standard Deviation (STDEV) | T Statistics (|O/STDEV|) | P Values | Description |
|---|---|---|---|---|---|---|
| The Logic of Mathematics → Calculus →Algebra → Programming Algorithm | 0.029 | 0.042 | 0.080 | 0.362 | 0.717 | No Effect |
| The Logic of Mathematics → Algebra → Programming Algorithm | 0.216 | 0.179 | 0.087 | 2.47% | 0.014 | Influential |
| The Logic of Mathematics → Calculus → Programming Algorithm | 0.043 | 0.000 | 0.185 | 0.230 | 0.818 | No Effect |
| The Logic of Mathematics → Calculus →Algebra | 0.074 | 0.091 | 0.196 | 0.376 | 0.707 | No Effect |

Based on Table 4, if the value of the p Values = 0.014 is less than α =0.05, it may be said that the logic of mathematics proficiency has an indirect influence on the ability to programming algorithms through the mediating variable of algebra proficiency. This means that algebra proficiency mediates an increase in the logic of mathematics and programming algorithms proficiency. Therefore, it is necessary to improve algebra proficiency after the logic of mathematics to improve the ability of programming algorithms. This explanation is reinforced with the graph of data distribution of the indirect effect of the ability of the logic of mathematics on the ability to programme algorithms with the highest frequency at five points between 0.3 and 0.4. It means that the logic of mathematics proficiency indirectly affects the ability to programme algorithms through the mediating variable of algebra proficiency, as shown in Figure 6.
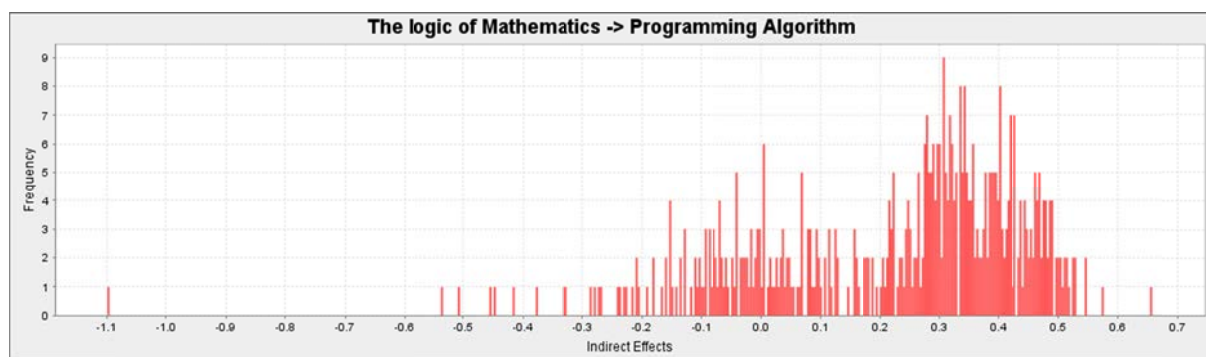


Figure 6. Graph of the spread of the influence of the data is not directly between the logic of mathematics proficiency and the ability to programming algorithm

The logic of mathematics proficiency and calculus proficiency does not affect the ability to programme algorithms, because the value of the P-value is less than α = 0.05. It means that calculus proficiency as a mediating variable is not able to increase the influence of the logic of mathematics on the ability to programming

algorithms. This is because the logic of mathematics proficiency does not affect calculus proficiency. Hence, calculus proficiency as a mediating variable is also unable to improve the proficiency of the logic of mathematics and algebra. The explanation above is supported by the data distribution of the indirect effect of the ability of calculus on the ability to programming algorithms, as shown in Figure 7 and Figure 8.

The data distribution in Figure 9 shows the highest frequency at five points between the numbers 0.2 and 0.27, which means that calculus proficiency does not indirectly affect the ability to programming algorithms. The data distribution in Figure 8 shows the highest frequency at the point between 0.0 and 0.1, which means that the logic of mathematics proficiency (through the mediating variable the ability of calculus) does not affect algebra proficiency. The model of correlations between the ability to programming algorithm, algebra, calculus, and logic of mathematics is shown in Figure 9. Based on Figure 9 and Table 2, algebra proficiency has the most influence on the ability to programming algorithm by 93.5%.
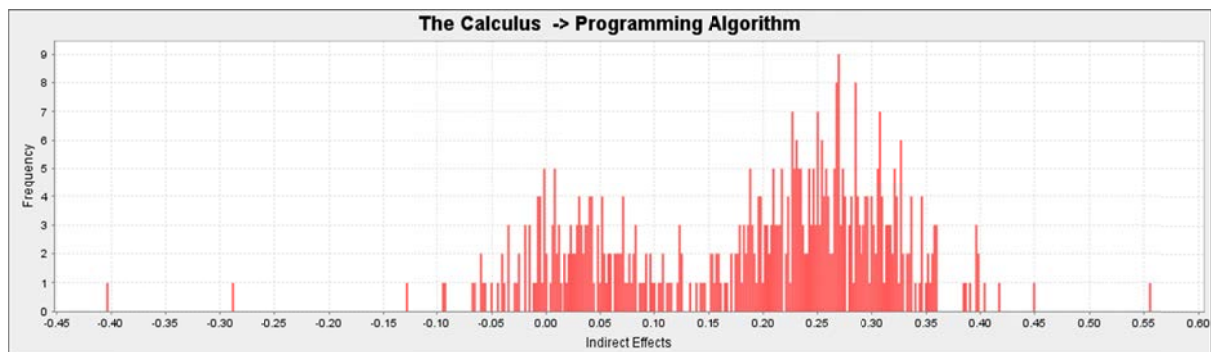


Figure 7. Graph of data distribution of the indirect effect of calculus proficiency on the ability to programming algorithm
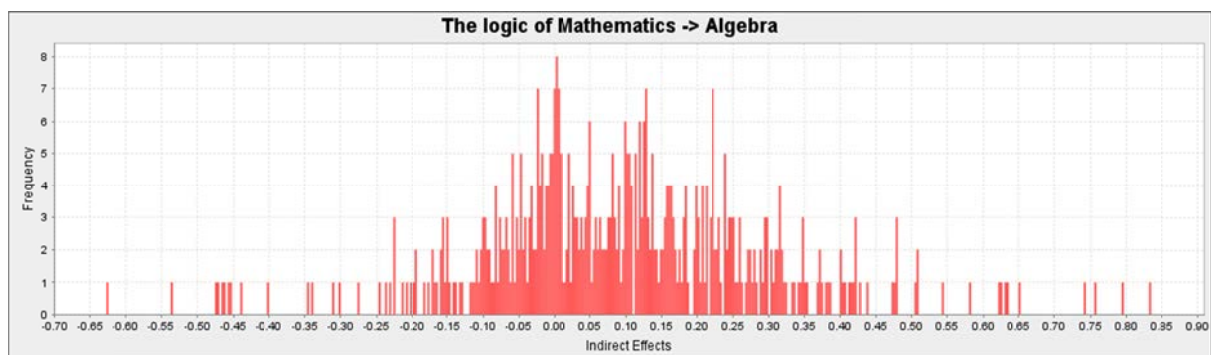


Figure 8. Graph of data distribution of the indirect effect of the logic of mathematics proficiency on algebra proficiency
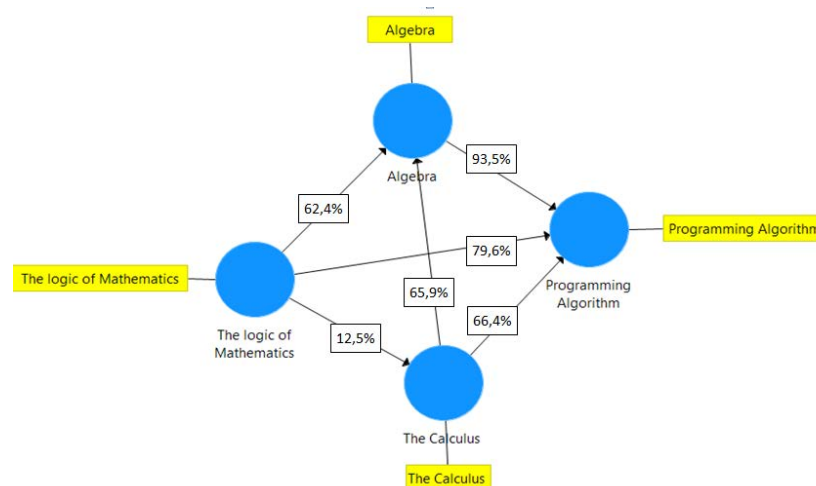
Figure 9. Correlations between the ability to programming algorithm, algebra, calculus, and logic of mathematics

## 5. Conclusion

Proficiency in algebra, calculus, and logic of mathematics has a positive effect on the ability of programming algorithms. The mathematical logic with the highest influence over the ability to programme algorithms in algebra, with a percentage of 93.5%. It is followed by the logic of mathematics by 79.6% and calculus by 66.64%. Calculus proficiency and the logic of mathematics proficiency have a positive effect on algebra proficiency. It means an increase in calculus and logic of mathematics proficiency will improve algebra proficiency. The influence of calculus proficiency on algebra proficiency is 65.9%, and the influence of the logic of mathematics proficiency on algebra proficiency is 62.4%.

Proficiency in the logic of mathematics does not affect calculus proficiency. As a result: (1) Calculus proficiency as a mediating variable is not able to increase the influence of the logic of mathematics proficiency on algebra proficiency. (2) Calculus proficiency as a mediating variable is unable to increase the influence of the logic of mathematics proficiency on the ability to programming algorithms. Calculus proficiency (through algebra) does not affect proficiency in the logic of mathematics indirectly. It means algebra proficiency as a moderating variable is unable to improve the influence of calculus proficiency on proficiency in the logic of mathematics.

Algebra proficiency as a mediating variable increases the effect of the logic of mathematics proficiency on the ability to programming algorithms. Therefore, it is necessary to increase algebra proficiency after mathematical logic in order to improve the ability to programme algorithms.

Based on the findings, it is recommended that the programming algorithm lecturers collaborate with algebra, basic calculus, and mathematical logic to improve learning material and methods. The findings may become a stepping stone for future research such as media development for programming algorithm study that prioritizes improvement in algebra, basic calculus, and mathematical logic.

## Acknowledgments

## References

Achjari, D. (2004). PARTIAL LEAST SQUARES : ANOTHER METHOD OF STRUCTURAL EQUATION MODELING ANALYSIS. *Jurnal Ekonomi Dan Bisnis Indonesia*, *19*(3), 238-248.

Al-Azawei, A., & Lundqvist, K. (2015). Learner differences in perceived satisfaction of an online learning: An extension to the technology acceptance model in an arabic sample. *Electronic Journal of E-Learning*, *13*(5), 408-426.

Arifin, R. W. (2017). Media Pembelajaran Berbasis Video Animasi Pada Mata Kuliah Logika Dan Algoritma 1.

*Bina Insani ICT Journal*, *4*(1), 83-94. https://doi.org/10.1111/j.1751-1097.2010.00740.x

Attridge, N., & Inglis, M. (2013). Advanced Mathematical Study and the Development of Conditional Reasoning Skills. *PLoS ONE*, *8*(7). https://doi.org/10.1371/journal.pone.0069399

Combéfis, S., Barry, S. A., Crappe, M., David, M., De Moffarts, G., Hachez, H., & Kessels, J. (2017). Learning and teaching algorithm design and optimisation using contests tasks. *Olympiads in Informatics*, *11*, 19-28. https://doi.org/10.15388/ioi.2017.02

Eglen, S. J. (2009). A quick guide to teaching R programming to computational biology students. *PLoS Computational Biology*, *5*(8). https://doi.org/10.1371/journal.pcbi.1000482

Garson, G. D. (2016). *Partial Least Squares: Regression & Structural Equation Models*. Statistical Publishing Associates.

Hair, J. F., Ringle, C. M., & Sarstedt, M. (2011). PLS-SEM: Indeed a silver bullet. *Journal of Marketing Theory and Practice*, *19*(2), 139-151. https://doi.org/10.2753/MTP1069-6679190202

Hamdollah, R., & Baghaei, P. (2016). Partial least squares structural equation modeling with R. *Practical Assessment, Research and Evaluation*, *21*(1), 1-16.

Hartati, S. J. (2014). Design of Learning Model of Logic and Algorithms Based on APOS Theory. *International Journal of Evaluation and Research in Education (IJERE)*, *3*(2), 109-118. https://doi.org/10.11591/ijere.v3i2.5743

Hartati, S. J. (2017). Kerangka Teori untuk Peranan Penalaran Matematika pada Pembelajaran Algoritma Komputasi. *SENAMAS 2017*, (1982), 1-11.

Haseski, H. I., Ilic, U., & Tugtekin, U. (2018). Defining a New 21st Century Skill-Computational Thinking: Concepts and Trends. *International Education Studies*, *11*(4), 29. https://doi.org/10.5539/ies.v11n4p29

Hauser, R., Paul, R., & Bradley, J. (2012). Computer self-efficacy, anxiety, and learning in online versus face to face medium. *Journal of Information Technology Education: Research*, *11*(1), 141-154. https://doi.org/10.28945/1633

Hsu, T. C., Chang, S. C., & Hung, Y. T. (2018). How to learn and how to teach computational thinking: Suggestions based on a review of the literature. *Computers and Education*, *126*(June), 296-310. https://doi.org/10.1016/j.compedu.2018.07.004

Hwang, G. J., Kuo, F. R., Yin, P. Y., & Chuang, K. H. (2010). A Heuristic Algorithm for planning personalized learning paths for context-aware ubiquitous learning. *Computers and Education*, *54*(2), 404-415. https://doi.org/10.1016/j.compedu.2009.08.024

Inglis, M., & Attridge, N. (2016). Does mathematical study develop logical thinking?: Testing the theory of formal discipline. In *World Scientific* (Vol. 3). https://doi.org/10.1142/q0020

Johar, R., Yusniarti, S., & Saminan. (2018). The analysis of proportional reasoning problem in the Indonesian mathematics textbook for the junior high school. *Journal on Mathematics Education*, *9*(1), 55-68. https://doi.org/10.22342/jme.9.1.4145.55-68

Kachroo, P., Krishen, A., & Agarwal, S. (2017). Fuzzy logic programming based knowledge analysis for qualitative comparative analysis. *Quality and Quantity*, *51*(5), 2101-2113. https://doi.org/10.1007/s11135-016-0453-8

Kozen, D. C. (1991). *The Design and Analysis of Algorithms for Asynchronous Multiprocessors*. Springer-Verlag, Inc.

Marisa, F. (2008). Pengembangan Web Based Learning Dalam Matakuliah Algoritma Pemrograman 1 Di STMIK Pradnya Paramita Malang. *Jurnal Ilmiah Teknologi Informasi Asia*, *3*(1), 14-18.

Maulana, G. G. (2017). Pembelajaran Dasar Algoritma Dan Pemrograman Menggunakan El-Goritma Berbasis Web. *Jurnal Teknik Mesin*, *6*(2), 8. https://doi.org/10.22441/jtm.v6i2.1183

McConnell, J. J. (2001). *Analysis of Algorithms–An Active Learning Approach*. Canada: Jones and Bartlett Publisher, Inc.

Mohaghegh, M. (2016). Computational thinking : the skill set of the 21st century. *(IJCSIT) International Journal of Computer Science and Information Technologies*, *7*(3), 1524-1530.

Mohammad, M. (2010). *D. Naskah Komprehensif Perubahan Undang-Undang Dasar Negara Republik*

*Indonesia Tahun 1945 Latar Belakang, Proses, dan Hasil Pembahasan 1999-200*2. Sekretariat Jenderal dan Kepaniteraan Mahkamah Konstitusi .

Monecke, A., & Leisch, F. (2012). SemPLS: Structural equation modeling using partial least squares. *Journal of Statistical Software*, *48*(3). https://doi.org/10.18637/jss.v048.i03

Nuraini, R. (2015). Desain Algorithma Operasi Perkalian Matriks Menggunakan Metode Flowchart. *Jurnal Teknik Komputer Amik Bsi*, *1*(1), 146. Retrieved from https://ejournal.bsi.ac.id/ejurnal/index.php/jtk/article/view/245

OECD Better Policies For Better Lives. (2019). *PISA 2018 Assessment and Analytical Framework*.

Ou, X. (2005). *A logic-programming approach to network security analysis* (Ph.D dissertation).

Pendidikan, M., Kebudayaan, D. A. N., & Indonesia, R. (2018). *PERATURAN MENTERI PENDIDIKAN DAN KEBUDAYAAN REPUBLIK INDONESIA*.

Rigdon, E. E., Becker, J. M., & Sarstedt, M. (2019). Factor Indeterminacy as Metrological Uncertainty: Implications for Advancing Psychological Measurement. *Multivariate Behavioral Research*, *54*(3), 429-443. https://doi.org/10.1080/00273171.2018.1535420

Sarah-Bibi, H. J. (2015). Efektivitas Model Blended Learning Terhadap Motivasi dan Tingkat Pemahaman Mahasiswa Mata Kuliah Algoritma dan Pemrograman. *Jurnal Pendidikan Vokasi*, *5*(1), 74-87. https://doi.org/http://dx.doi.org/10.26858/jpkk.v2i2.2250

Sarstedt, M., Hair Jr, J. F., Cheah, J. H., Becker, J. M., & Ringle, C. M. (2019). How to specify, estimate, and validate higher-order constructs in PLS-SEM. *Australasian Marketing Journal (AMJ)*, *27*(3), 197-211. https://doi.org/10.1016/j.ausmj.2019.05.003

Sarwono, J. (2006). *Metode Penelitian Kuantitatif & Kualitatif*. In Graha Ilmu.

Schmalbach, J. C. V., & Maza-Avila, F. J. (2018). Structural equation models applied for evaluating service quality and satisfaction in the healthcare system of Cartagena de Indias D. T. Y C. (Colombia). *Economics and Sociology*, *11*(2), 200-215. https://doi.org/10.14254/2071-789X.2018/11-2/14

Septia, T., Prahmana, R. C. I., Pebrianto, & Wahyu, R. (2018). Improving students spatial reasoning with course lab. *Journal on Mathematics Education*, *9*(2), 327-336. https://doi.org/10.22342/jme.9.2.3462.327-336

Shahijan, M. K., Rezaei, S., & Amin, M. (2016). International students' course satisfaction and continuance behavioral intention in higher education setting: an empirical assessment in Malaysia. *Asia Pacific Education Review*, *17*(1), 41-62. https://doi.org/10.1007/s12564-015-9410-9

Utomo, A. P., & Narulita, K. S. E. (2018). Diversification of Reasoning Science Test Items of Timss Grade 8 Based on Higher Order Thinking Skills : A Case Study of Indonesian Students. *Journal of Baltic Science Education*, *17*(1), 152-161.

Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2016). Defining Computational Thinking for Mathematics and Science Classrooms. *Journal of Science Education and Technology*, *25*(1), 127-147. https://doi.org/10.1007/s10956-015-9581-5

Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, *49*(3), 33-35. https://doi.org/10.1145/1118178.1118215

Wing, J. M. (2008). Computational thinking and thinking about computing. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, *366*(1881), 3717-3725. https://doi.org/10.1098/rsta.2008.0118

Wong, K. K. K.-K. (2013). Partial Least Squares Structural Equation Modeling (PLS-SEM) Techniques Using SmartPLS. *Marketing Bulletin*, *24*(1), 1-32. Retrieved from http://www.researchgate.net/profile/Ken_Wong10/publication/268449353_Partial_Least_Squares_Structural_Equation_Modeling_(PLS-SEM)_Techniques_Using_SmartPLS/links/54773b1b0cf293e2da25e3f3.pdf

**Copyrights**