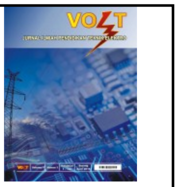
P-ISSN: 2528-5688
E-ISSN: 2528-5696

VOLT

Jurnal Ilmiah Pendidikan Teknik Elektro

Journal homepage: jurnal.untirta.ac.id/index.php/VOLT

Vol. 2, No. 2, October 2017, 81-92



ANALYSIS OF STUDENT DIFFICULTIES IN COMPUTER PROGRAMMING

Abdul Baist[1]✉, Aan Subhan Pamungkas[2]

[1]Department of Mathematics Education, Faculty of Teacher Training and Education
Universitas Muhammadiyah Tangerang, Tangerang 15118, Indonesia
✉Corresponding author e-mail: abdulbaist79@gmail.com

[2]Department of Mathematics Education, Faculty of Teacher Training and Education
Universitas Sultan Ageng Tirtayasa, Serang 42124, Indonesia

Received: 22 August 2017. *Received in revised form*: 02 October 2017. *Accepted*: 02 October 2017

Abstract

Computer programming skills are required in mathematics computing courses. Most students have difficulty making computer programs. This study aims to identify the difficulties faced by students in making computer programs. This research is descriptive quantitative research. The subjects in this study are students of Mathematics Education Departement, Muhammadiyah University of Tangerang. Based on the results of data analysis, the conclusion is: (1) there are significant differences in multidimensional array material between high, medium and low group; (2) there is a significant difference in input / ouput command material between high, medium and low group; (3) there are significant differences about the difficulties experienced by students in understanding the basic concept of programming between high, medium and low groups; (4) there is a significant difference regarding the difficulties experienced by students in finding the fault of their own programs between high, medium and low groups; (5) there is no significant difference in situations that may assist students in programming for lab work in the high, medium and low groups; (6) there is no significant difference in situations that can assist students in programming to do alone tasks between high, medium and low groups; (7) there is no significant difference in the lack of examples shown when practice makes poor performance in programming between high, medium and low groups; (8) there is no significant difference in what makes poor performance in programming a less conducive atmosphere between high, medium and low groups.

Keywords: difficulties, computation, programming

INTRODUCTION

Mostrom (2011) said that programming is the act of understanding a problem, formulating a solution, and writing down the solution in such a way that a computer can use the solution to solve the. Therefore a programmer should understand a problem first. When the problem had been understood, programmer would use problem solving technic to solve the problem. At last step a programmer need to communicate that solution so that computer could follow instruction that been given.

Computer programming skill is needed in computational mathematics course. In that course, problem in mathematics is translated into a computer program. The program is used to get an answer from a given problem.

Making a computer program is not an easy thing. This skill need another skills, like designing algorithm, writing program in certain program language, and understanding syntax from the program (Rahmat, Shahrani, Latih, Yatim, Zainal, & Rahman, 2012).

For most students, programming is a new thing. It needs basic skill to reach advanced level in computer programming. Students need to know basic syntax, structure, and style of a program language gradually. These make students feel difficult often.

Difficulties in programming are common for novice students. As Evan and Simkin (1989) said that computer programming is very complicated for many novice students at university level. Meanwhile Ala-Mutka (2004) said that difficulties faced by students are not in syntax or understanding of concept, but rather basic program planning (Mhashi & Alakeel, 2013, pp. 15).

Based on score of computational mathematics courses there are about 70% of students achieved below grade of B. It indicated that there are a problem happened. Therefore an action is needed to find out the cause.

The aimi of this study is to find out difficulties faced by students in computer programming. The result of this study is expected to be input for computational mathematics course. What kind of action is needed so that the outcomes of the lectures get better.

METHOD

This study used survey method. Groves, Fowler, Couper, Lepkowski, Singer, Tourangean (2009), said that survey is a systematic method for gathering information from (a sample of) entities for the purposes of constructing quantitative descriptors of the attributes of the larger population of which the entities are. Systematic is deliberate and meaningfully distinguishes surveys from other ways of gathering information. The quantitative descriptors are called statistics.

This study used questionnaire adapted from instrument used by Milne and Rowe (2002), Tan, Ting, Ling, (2009), and Derus and Ali (2012). To obtain the necessary information, the questionnaire consists of two major sections, the background information of respondents in general, and the experience of respondents when learning programming.

The number of respondents in this study about 132 students. Respondents are sixth semester students of Mathematics Education Department, Muhammadiyah University of Tangerang

RESULTS AND DISCUSSION

Of the 132 respondents who answered the questionnaire, 15% were male or about 20 people. While for female respondents 85% or about 112 people. This can be seen in the following diagram

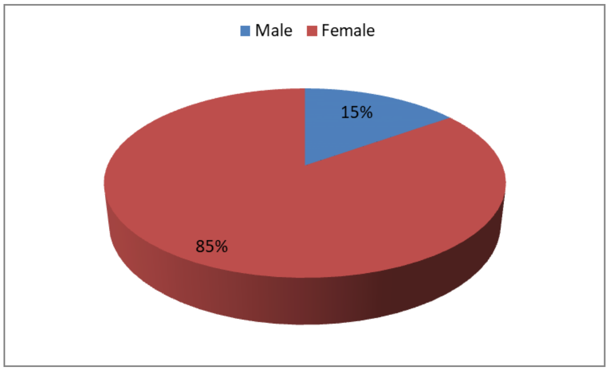


Figure 1. Percentage of respondents based on gender

Regarding the question have they ever studied computer programming, most of them have never studied computer programming or about 98% that is as many as 129 people. This means that for the first time they are familiar with computer programming. This is illustrated in the following diagram

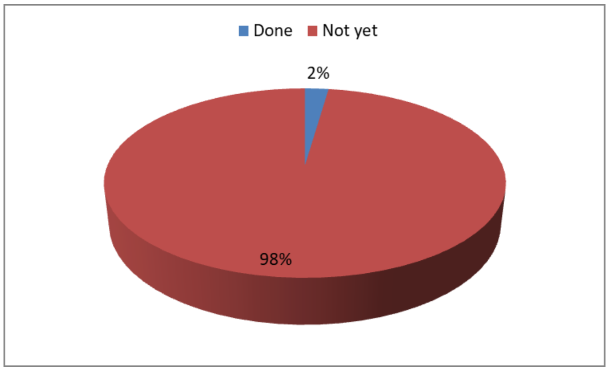


Figure 2. Percentage of respondents based on experience learning computer programming

Student performance is determined based on the score of mathematics computational exam. the values are grouped into high, medium, and low. This grouping will be used to determine if there is a significant difference between each problem analyzed and the student's performance. The grouping can be seen in the following table.

Table 1. Students performance based on groups

Performance	Frequency	Percentage (%)
High	10	8
Medium	30	42
Low	92	50

The data used in this study comes from a questionnaire distributed to students, the questionnaire is ordinal scale. Questionnaires are given to students to know their opinions on understanding the basic material of programming. The level of understanding of students on the basic materials of programming can be seen in Table 2. Scale used in each item is Likert scale. The material is sorted from the smallest average score. To see if there is a difference between the problems analyzed and the student's performance, the Kruskal-Wallis test is used

Table 2. Level of students' understanding on basic topic on programming course

Topics	Mean	Stdev
Multidimensional Array	2,34	0,91
Looping Statements (e.g: *while, for*)	2,45	0,98
Function	2,73	0,95
Branch Statements (e.g: *if-else, switch*)	2,92	0,97
Array Data Structure	3,11	0,96
Variables	3,45	0,99
Input/Output Statements (e.g: *input, fprintf*)	3,72	0,93

From the table above can be seen also that the average student does not understand the material Multidimensional Array. This can be seen further in the following diagram

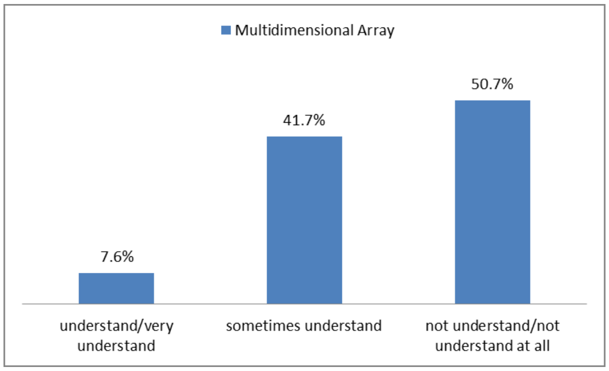


Figure 3. Level of students' understanding on Multidimensional Array

From Figure 3 it is seen that most students are not familiar with Multidimensional Array material. Here are the results of testing the difference rates by group

Table 3. Ranks of students' understanding on Multidimensional Array

Ranks

Performance		N	Mean Rank
Answer	High	10	90.85
	Medium	30	77.58
	Low	92	60.24
	Total	132	

Table 4. Test Statistics of students' understanding on Multidimensional Array

Test Statistics^{a,b}

	Answer
Chi-Square	10.114
df	2
Asymp. Sig.	0.006

From Table 3 and Table 4 it can be seen that there are significant differences in Multidimensional Array material between high, medium, and low group. The H value of the test is 10.114 with degrees of freedom 2 and the value p = 0.006. Meanwhile the mean rank of the high group was 90.85, the moderate group was 77.58, and the low group was 60.24. This means that the level of understanding on Multidimensional Array material between high, medium, and low groups differ significantly. These results indicate that the group is well versed with Multidimensional Array material.

As for the material Input / Output Input the average student approaches the category of understanding with the material. This can be seen further in the following diagram

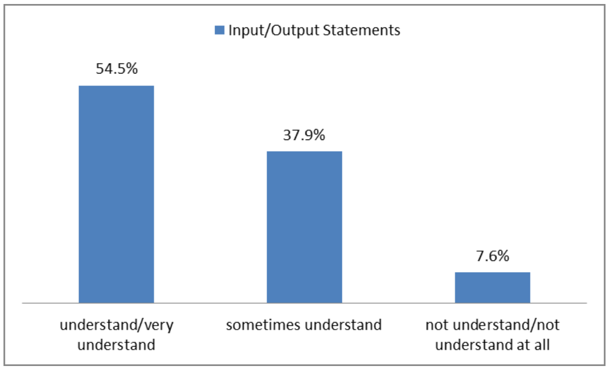


Figure 4. Level of students' understanding on Input/Output Statements

From Figure 4 it can be seen that most of the students are familiar with the Input / Output Input materials. As many as 54.5% of students understand the material Input / Output Commands. The possibility of this happening because the material is quite easy. The syntax they learned was not too complicated.

To see further differentiation rates, further testing is required. Here are the results of testing the difference rates by group

Table 5. Ranks of students' understanding on Input/Output statements

Ranks

Performance		N	Mean Rank
Answer	High	10	96.30
	Medium	30	79.53
	Low	92	59.01
	Total	132	

Table 6. Test Statistics of students' understanding on Input/Output statements

Test Statistics^{a,b}

	Answer
Chi-Square	14.472
df	2
Asymp. Sig.	0.001

From Table 5 and Table 6 it can be seen that there is a significant difference in Input / Output command material between high, medium, and low group. The H value of the test is 14.472 with degrees of freedom 2 and the value p = 0.001. Meanwhile the mean rank of the high group was 96.30, the mean group was 79.53, and the low group was 59.01. This means that the level of understanding of the Input / Output Input materials between high, medium, and low groups differs significantly. These results indicate that the high group understands the material Input / Output Input.

Overall the level of students' understanding of basic programming materials is at a moderate level. From Table 2 it can be seen that students have difficulty in understanding the basic material of programming. According to Milne and Rowe (2002), students' difficulties in learning programming is caused by less familiar to students with the rules contained in computer programs. This is reinforced by interviews with some students who stated that they have never studied computer program.

The difficulty level when studying the programming can be seen in Table 7. The scale used in each item in Table 7 is the Likert scale (1, strongly disagree; 2, disagree; 3, neutral; 4, agree; 5, strongly agree). Difficulty sorted from the largest average value.

Table 7. Difficulty while learning programming

Difficulties	Mean	Stdev
Understanding basic concepts of programming structure	4,09	0,95
Designing a program	3,87	0,98
Learning the programming language syntax	3,73	0,97
Using program development environment	3,48	1,04
Finding bugs from my own program	3,24	1,03

From Table 7 it can be seen that the average student has difficulty in understanding the basic concept of programming structure. This can be seen further in the following diagram.

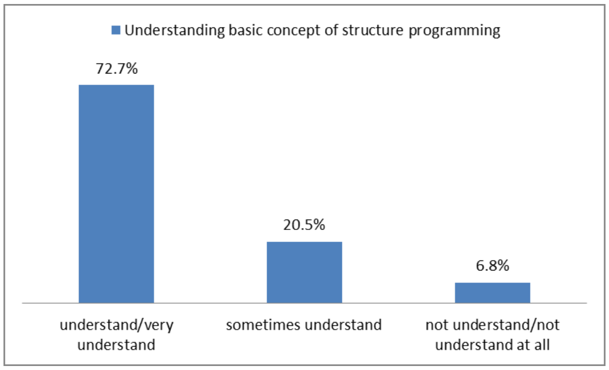


Figure 5. Difficulty while understanding basic concept of structure programming

From Figure 5 shows that most students have difficulty when understanding the basic concept of programming structure is about 72.7%. Here are the results of testing the difference rates by group.

Table 8. Ranks of difficulty while understanding basic concept of structure programming

Ranks

Performance		N	Mean Rank
Answer	High	10	40.40
	Medium	30	51.10
	Low	92	74.36
	Total	132	

Tabel 9. Test Statistics of difficulty while understanding basic concept of structure programming

Test Statistics[a,b]

	Answer
Chi-Square	15.147
df	2
Asymp. Sig.	0.001

From Table 8 and Table 9 it can be seen that there are significant differences regarding the difficulties experienced by students in understanding the basic concept of programming between high, medium, and low groups. The H value of the test is 15.147 with degrees of freedom 2 and the value p = 0.001. Meanwhile the mean rank of the high group was 40.40, the moderate group 51.10, and the low group 74.36. This means that the degree of difficulty in understanding the basic concepts of programming between high, medium, and low groups differ significantly. These results indicate that the low group has difficulty in understanding the basic concepts of programming.

From Table 7 it can be seen that the average students tend to be neutral in the difficulties they encounter when finding fault with their own programs. This can be seen further in the following diagram

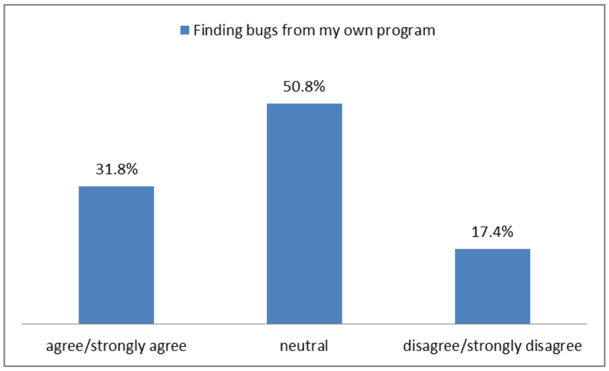


Figure 6. Difficulty while finding bugs from my own program

From Figure 6 it appears that most students feel neutral meaning they assume that they sometimes find it difficult and sometimes do not find it difficult to find fault with their own program. Here are the results of testing the difference rates by group.

Table 10. Ranks of difficulty while finding bugs from my own program

Ranks

Performance		N	Mean Rank
Answer	High	10	37.55
	Medium	30	56.95
	Low	92	72.76
	Total	132	

Table 11. Test Statistics of difficulty while finding bugs from my own program

Test Statistics[a,b]

	Answer
Chi-Square	11.712
df	2
Asymp. Sig.	0.003

From Table 10 and Table 11 it can be seen that there are significant differences regarding the difficulties students experience in finding out the errors of their own programs between high, medium, and low groups. The H value of the test is 11.712 with the degrees of freedom 2 and the value p = 0.003. Meanwhile the mean rank of the high group was 37.55, the moderate group 56.95, and the low group 72.76. This means that the degree of difficulty in finding the fault of the program itself between high, medium, and low groups differed significantly. These results indicate that the low group has difficulty finding the fault of their own program.

From Table 7 it can be seen that there are three types of interrelated difficulties when studying programming: i) difficulty understanding basic concepts of programming structure, ii) designing a program, and iii) studying programming language syntax. These three things are common for beginners when just learning programming. This is because programming capabilities involve the ability to create algorithms, write in certain program languages, and understand the syntax of the program's language (Rahmat, Shahrani, Latih, Yatim, Zainal, & Rahman, 2012).

Table 12. Situations that would help to learn programming

Situations	**Mean**	**Stdev**
Practical in lab	4,06	0,92
Discussion with lecturers or friends	3,91	0,96
In small group exercise sessions	3,68	0,98
In lectures	3,39	1,00
While working alone on programming coursework	3,06	1,00

Table 12 is the student's opinion of situations that they think may help study programming. The scale used in each item in Table 12 is the Likert scale (1, strongly disagree; 2, disagree; 3, neutral; 4, agree; 5, strongly agree). From Table 12 it can be seen that the average student agreed to a situation that can help them in learning programming is practice in the laboratory. This can be seen further in the following diagram

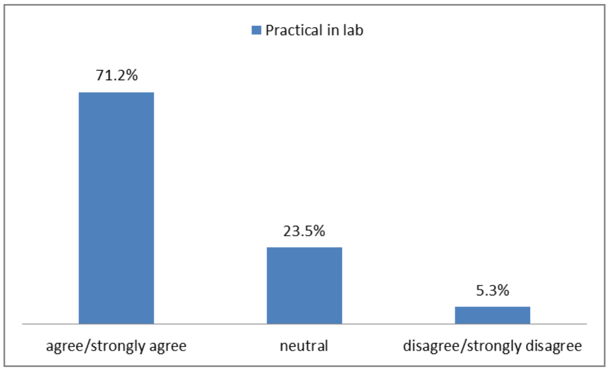


Figure 7. Practical in lab would help to learn programming

From Figure 7 it appears that most students agree that practice in the laboratory can help them in learning the programming. Here are the results of testing the difference rates by group.

Table 13. Ranks of practical in lab

Ranks

Performance		N	Mean Rank
Answer	High	10	62.30
	Medium	30	61.77
	Low	92	68.50
	Total	132	

Table 14. Test Statistics of practical in lab

Test Statistics[a,b]

	Answer
Chi-Square	0.932
df	2
Asymp. Sig.	0.628

From Table 13 and Table 14 it can be seen that there is no significant difference in situations that may help them in learning programming is laboratory practice between high, medium, and low groups. The H value of the test is 0.932 with degrees of freedom 2 and p value = 0.628. Meanwhile mean rank of high group was 62,30, medium group 61,77, and low group 68,50. This means that practice in the laboratory as a situation that can help learn the programming between high, medium, and low groups does not differ significantly. From Table 12 it can be seen that the average student's neutral opinion for a situation that can help them in learning programming is doing their own programming tasks. This can be seen further in the following diagram.

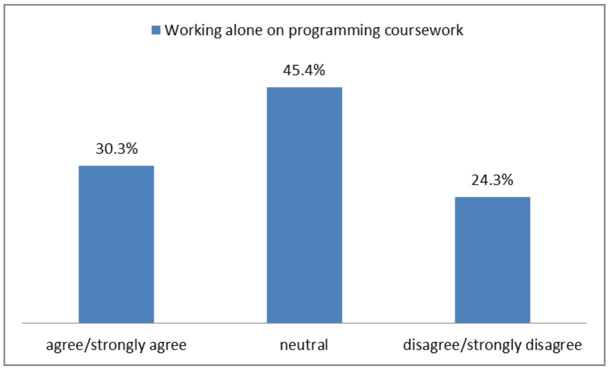


Figure 7. Working alone on programming coursework would help to learn programming

From Figure 7 it appears that most students are neutral in their view that doing their own programming tasks can help them in learning the programming. Here are the results of the differentiation levels based on the group.

Table 15. Ranks of working alone on programming coursework

Ranks

Performance		N	Mean Rank
Answer	Tinggi	10	68.20
	Sedang	30	64.13
	Rendah	92	67.09
	Total	132	

Table 16. Test Statistics of working alone on programming coursework

Test Statistics[a,b]

	Answer
Chi-Square	0.176
df	2
Asymp. Sig.	0.916

From Table 15 and Table 16 it can be seen that there is no significant difference in situations that can help them in learning programming is doing their own programming tasks between high, medium, and low groups. The H value of the test is 0.176 with degrees of freedom 2 and the value p = 0.916. Meanwhile the mean rank of the high group was 68.20, the group was 64.13, and the low group 67.09. This means that working on your own programming task as a situation that can help learn the programming between high, medium, and low groups does not differ significantly.

Most students agree that practical activities in the laboratory can help them learn the basics of programming effectively and discuss with lecturers or friends. This is similar to what was revealed by Gomes and Mendes (2007) that learning programming requires intensive practice and practice. Activities in the laboratory will certainly help students understand the difficulties they experience and arouse students' interest in programming (Parham, 2003).

Table 17. Factors that lead to poor performance in programming

Factors	**Mean**	**Stdev**
Less examples of practical use are shown	3,48	0,98
Computers provided in labs are not functioning well	3,36	0,97
Teaching methodology is less effective	3,34	0,95
Students' lack of interest to learn	3,27	0,93

Factors	Mean	Stdev
Syllabus focuses too much on theory	3,20	0,99
Syllabus coverage per semester is too wid	3,14	0,99
Learning environment that is not conducive	3,03	0,99

Table 17 is the student's opinion of the factors that make them get poor performance when learning programming. The scale used in each item in Table 4 is the Likert scale (1, strongly disagree; 2, disagree; 3, neutral; 4, agree; 5, strongly agree).

From Table 17 it can be seen that the average student believes that the neutral factor that makes poor performance in programming is the lack of examples that are displayed when practicing. This factor is at the top of the list. This can be seen further in the following diagram

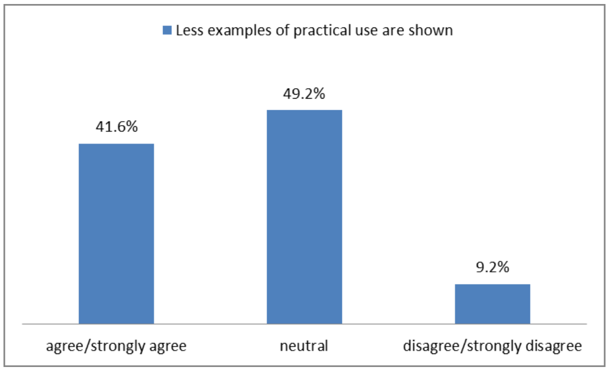


Figure 8. Less examples of practical use are shown lead to poor performance in programming

From Figure 8 it is seen that 49.2% of students think neutrally against the lack of examples shown when practicing. But as many as 41.6% of students agree to the lack of examples shown when practicing. It may be said that this indicates that the actual example given by the lecturer when the practice is still lacking. Here are the results of testing the difference rates by group

Table 18. Rank of less examples of practical use are shown

Ranks

	Performance	N	Mean Rank
Answer	Tinggi	10	60.10
	Sedang	30	59.40
	Rendah	92	69.51
	Total	132	

Table 19. Test Statistics of less examples of practical use are shown

Test Statistics[a,b]

	Answer
Chi-Square	2.186
df	2
Asymp. Sig.	0.335

From Table 18 and Table 19 it can be seen that there is no significant difference in the lack of examples shown when the practice makes their performance poorly in programming between high, medium, and low groups. The H value of the test is 2.186 with the degrees of freedom 2 and the value p = 0.335. Meanwhile the mean rank of the high group was 60.10, the moderate group was 59.40, and the low group was 69.51. This means that the lack of examples shown when the practice of making poor performance in programming between high, medium, and low groups does not differ significantly.

From Table 17 it can be seen that the average student assumes neutral that the factors that make poor performance in programming is less conducive learning atmosphere. This factor is at the bottom of the list. This can be seen further in the following diagram

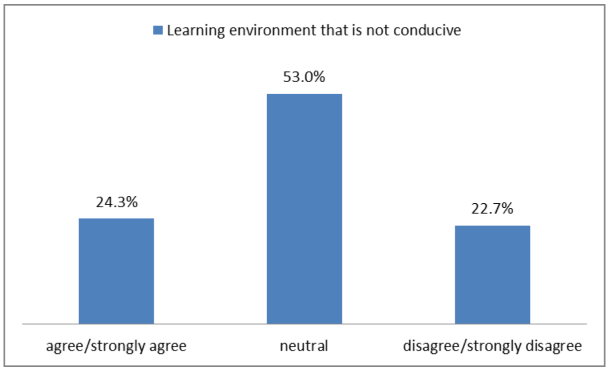


Figure 9. Learning environment that is not conducive lead to poor performance in programming

From Figure 9 seen as many as 53.0% of students thought neutral to the less conducive learning atmosphere. It may be said that they are in doubt with the learning atmosphere they are experiencing whether making performance in programming is bad or not. Here are the results of testing the difference rates by group.

Table 20. Ranks of learning environment that is not conducive

Ranks

Performance		N	Mean Rank
Answer	Tinggi	10	29.50
	Sedang	30	68.50
	Rendah	92	69.87
	Total	132	

Table 21. Test Statistics of learning environment that is not conducive

Test Statistics[a,b]

	Answer
Chi-Square	12.048
df	2
Asymp. Sig.	0.002

From Table 20 and Table 21 it can be seen that there is a significant difference regarding what makes their performance poor in programming is a less conducive learning environment between high, medium, and low groups. The H value of the test is 12.048 with degrees of freedom 2 and the value p = 0.002. Meanwhile the mean rank of the high group was 29.50, the group was 68.50, and the low group was 69.87. This means that the less conducive learning atmosphere as a situation that makes performance in programming poorly between high, medium, and low group significantly different. The interesting thing to be found from Table 20 can be seen that significant differences occur in high clusters. The average high group considers that the less conducive learning atmosphere makes bad kinreja in programming.

From the discussion of factors that make poor performance in programming can be said that students have difficulty programming because lecturers do not provide adequate examples for students or lack of examples provided, and computer equipment in the laboratory does not work well, and teaching methods used lecturers less effective. Teaching strategies and techniques are important in order to convey information to students. In order for students to master basic problem-solving skills, the teaching methods used must be applicable and make students engage in practical activities (Ismail, Ngah, Umar, 2010).

CONCLUSION

For beginners, computer programming is not an easy thing. It takes some important skills to master programming such as creating a programming design, and writing it into a particular programming language. The difficulties that students face in terms of understanding the basic programming because they are not yet familiar with a particular programming language. Plus three interrelated things that are programming structure, make design, and programming language syntax they have to master.

In addition, lecturers do not provide sufficient examples for students, as well as computer equipment in the laboratory is not working properly, and teaching methods used less effective lecturers make their performance poor in learning programming.

Based on the results of data analysis, the conclusion is: (1) there are significant differences in multidimensional array material between high, average and low group; (2) there is a significant difference in input / ouput command material between high, average and low group; (3) there are significant differences about the difficulties experienced by students in understanding the basic concept of programming between high, average and low groups; (4) there is a significant difference regarding the difficulties experienced by students in finding the fault of their own programs between high, average and low groups; (5) there is no significant difference in situations that may assist students in programming for lab work in the high, average and low groups; (6) there is no significant difference in situations that can assist students in programming to do alone tasks between high, average and low groups; (7) there is no significant difference in the lack of examples shown when practice makes poor performance in programming between high, average and low groups; (8) there is no significant difference in what makes poor performance in programming a less conducive atmosphere between high, average and low groups;

To overcome the difficulties that students encounter in learning programming, it takes a situation that they think can overcome it. Situations that can help them in mastering programming, according to them are practical activities in the laboratory and discussions with lecturers or friends.

REFERENCES

Derus, S. R. M., & Ali, A. Z. M. (2012). Difficulties in learning programming: Views of students. In *1st International Conference on Current Issues in Education (ICCIE 2012)* (pp. 74-79).

Gomes, A., & Mendes, A. J. (2007). Learning to program-difficulties and solutions. In *International Conference on Engineering Education–ICEE* (Vol. 2007).

Groves, R. M., Fowler, Jr., F. J., Couper, M. P., Lepkowski, J. M., Singer, E., dan Tourangean, R. (2009). *Survey Methodology (second edition)*. Wiley & Sons: New Jersey.

Ismail, M. N., Ngah, N. A., & Umar, I. N. (2010). Instructional strategy in the teaching of computer programming: a need assessment analyses. *TOJET: The Turkish Online Journal of Educational Technology*, *9*(2).

Mhashi, M. M., & Alakeel, A. L. I. M. (2013). Difficulties facing students in learning computer programming skills at Tabuk University. In *Proceedings of the 12th International Conference on Education and Educational Technology (EDU'13), Iwate, Japan* (pp. 15-24).

Milne, I., dan Rowe, G. (2002). Difficulties in Learning and Teaching Programming-Views of Students and Tutors. *Education and Information Technologies, 7:1, 55–66.*

Mostrom, J. E. (2011). *A Study of Students Problems in Learning to Program*. Umea: Department of Computing Science, Umea University, Sweden.

Parham, J. R. (2003). An Assessment and evaluation of computer science education. *Journal of Computing Sciences in College, vol. 19, no. 2, 115 – 127*.

Rahmat, M., Shahrani, S., Latih, R., Yatim, N. F. M., Zainal, N. F. A., & Ab Rahman, R.

(2012). Major problems in basic programming that influence student performance. *Procedia-Social and Behavioral Sciences*, *59*, 287-296.

Tan, P. H., Ting, C. Y., & Ling, S. W. (2009). Learning difficulties in programming courses: undergraduates' perspective and perception. In *Computer Technology and Development, 2009. ICCTD'09. International Conference on* (Vol. 1, pp. 42-46). IEEE.