

APPLICATION OF CONSTRUCT ON SCAFFOLD CONCEPT MAP IN MOBILE PROGRAMMING LEARNING WITH FLUTTER LAYOUT TOPIC

Putra Prima Arhandi¹, Vivin Ayu Lestari², Annisa Taufika Firdausi³, Abdurrasyid
Muhasibi⁴, Dharma Yudistira Eka Putra⁵, Banni Satria Andoko⁶

^{1,2,3,3,4,5} Department of Information Technology, State Polytechnic of Malang

email: putraprima@polinema.ac.id¹, vivin@polinema.ac.id²,
annisa.taufika@polinema.ac.id³, abduurrasyid.muhasibi.20@gmail.com³, 1841720028@student.polinema.ac.id⁵,
ando@polinema.ac.id⁶

Abstract

Flutter is a framework for making mobile applications cross-platform made by Google. From 2019 to 2021 the popularity of flutter is increasing. Flutter use declarative writing style to create layouts. This makes the layout in flutter immutable, and a light blueprint. This research proposes a construct on scaffold concept map method to help students understand the concept of widget arrangement in flutter layout. Construct on scaffold will provide the learner with a framework from an incomplete expert concept map. Some of the nodes and connecting relationships in the framework have been removed, so students must fill in the missing parts with several available answer choices to complete the concept map. To prove the impact of the application of this method, the study was conducted using a pre-post-test group experimental design. Students will do a pre-test, use the EasyFlutter application, and post-test. The results of the pre-test and post-test obtained were tested for normality first, then tested to find out whether there was an average difference between the pre-test and post-test scores. The results of the normality test show that the pre-test data are not normally distributed, and the post-test data are normally distributed, so the next test will use a non-parametric test, namely the Wilcoxon test. The test results show that the post-test mean score is higher than the pre-test mean. Wilcoxon test results also show that the Asymp. Sig. (2-tailed) of 0.01, so it can be concluded that there is a significant difference between the pre-test and post-test scores. The application of the construct on scaffold method has a significant positive impact on the post-test scores of students related to the concept of widget arrangement in flutter.

Keywords : Construct on Scaffold, Concept Map, Flutter

Received: 13-04-2023 | **Revised:** 05-07-2023 | **Accepted:** 10-07-2023

DOI: <https://doi.org/10.23887/janapati.v12i2.60629>

INTRODUCTION

In the computational syllabus (computer science), programming is a common topic to learn and teach [1]. Higher cognitive competencies such as creativity, and critical thinking are needed to learn programming [2]. When learning programming, the failure and dropout rates for novice [3]–[6]. Therefore programming is referred to as a difficult topic to learn [7]–[11]. But on the other hand, Japan actually makes programming one of the important topics in the world of education, even at the elementary school level. This was done not only to gain programming skills, but also to practice logical thinking skills [12].

Many studies have been conducted to find out what difficulties cause novice programmers

to fail and drop out when learning programming [9]. One of the most common difficulties is the lack of problem solving skills [13], [14]. This lack of skills makes novice programmers tend to choose to directly code without clearly defining the problem first [15]. When coding , they tend to do trial and error [15], [16] and rely on feedback from the compiler to resolve errors [14], [15]. Another problem experienced by novice programmers is the difficulty of understanding certain programming concepts [17]. These difficulties may occur due to lack of understanding of basic programming concepts and lack of ability to apply programming skills to real practice [18].

Flutter is a framework for making mobile applications cross-platform (iOS and Android)

made by Google [19], [20]mobile programming learning . From 2019 to 2021 the popularity of flutter is increasing. Flutter use declarative writing style to create layouts. This makes the layout in flutter immutable, and only a light blueprint. In contrast to frameworks that use an imperative writing style, the layout components in the framework can be changed directly after they are created. The difference also lies in the way the program code is written. Frameworks in general will separate the program layout code from the program logic code, but Flutter uses the Dart programming language to create layouts without separating them from the program logic code [19].

In flutter, widgets are the main component in making mobile applications. Everything in flutter can be created using widgets [19], [20]. Starting from text, images, buttons, layouts, orientations, animations, to gestures are widgets [20]. Layout Flutter can be created by arranging widgets [19], [20], as well as putting together a puzzle [19] or Lego [20]. Therefore we need an appropriate learning support method in learning the concept of widget arrangement in flutter layout.

Widget tree is a logical representation of the widget array [19]. This makes the widget tree can be used as a learning medium to understand the concept of widget arrangement in flutter layout. Widget tree is one implementation of concept map. The concept map consists of propositions defined by two nodes and one connecting relation [21]. Concept map is also an implementation of a graphic strategy [22]. The graphic strategy will convert linear text statements into non-linear graphical representations [23]. The text here is the program code of the widget that will be visualized as a node.

Concept-map approaches for computer-based , namely construct-on-scaffold and construct-by-self [21]. Construct on scaffold will provide students with an incomplete expert concept map framework. Some of the nodes and connecting relationships in the framework have been removed, so students must fill in the missing parts with the answer choices provided to complete the concept map. Construct by self will give students the freedom to create their own concept maps without the help of an expert framework. Construct on scaffold provides a better learning effect than the construct by self-approach, this is because the construct on scaffold approach can reduce cognitive load and the possibility of overload on students [24]; the time and energy saved by the learner can be used to complete the concept map and understand the whole material [21].

In this research we propose the use of construct on scaffold concept map approach to increase student understanding on flutter layout topics.

GRAPHICAL STRATEGIES

Graphic strategy is an approach to change the structure of a text into a visual representation, with the aim of providing a clearer and more substantial understanding to the reader [23]. In this study, a graphical strategy is used to convert the source code text of the widget into a visual representation of a node. There are three main categories of graphical learning strategies there are graphic organizers [25] , knowledge maps [26] and concept maps [27]. These three graphic strategies are very identical in terms of their guiding principles and implementation despite the variations in visual presentation. They transform textual statements that are linear into nonlinear graphic displays. Although concept mapping is a graphic strategy similar to knowledge maps and graphic organizers, it has been successfully applied primarily to the learning of scientific subjects [22].

CONCEPT MAP

Concept map consists of propositions defined by two vertices and a connecting relation [21]. Concept map is a form of implementation of graphical strategy. Example of a propositions and relations can be seen in Figure 1 and Figure 2.

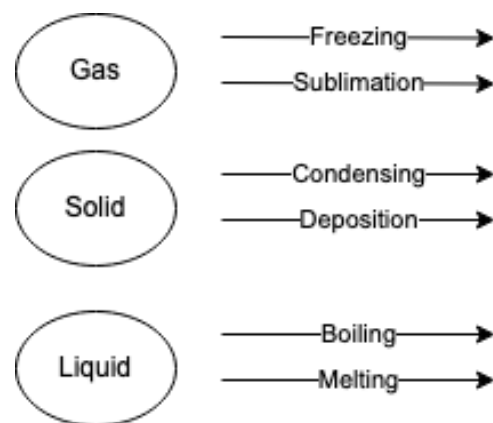


Figure 1. Example of propositions and relations in concept map

Based on the structure concept map can be divided into three main structure namely spoke, chain and net [28], the three structure illustrated on Figure 3,4 and 5 respectively.

Concept map are widely used as technology enhanced learning tools. it can be used in EFL learning tools [29], biology and computer

science [30]. Concept map created by the student can be used to evaluate student understanding about certain concept [29], [31], [32].

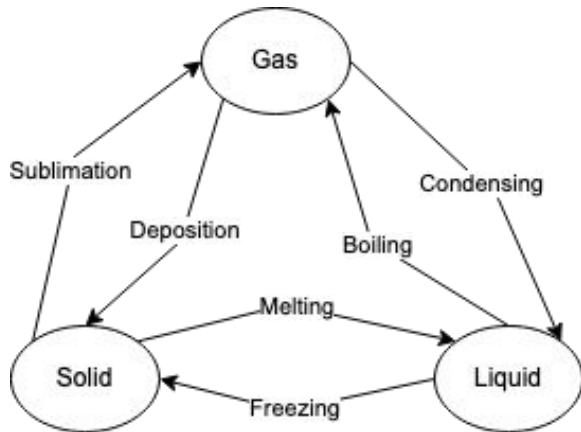


Figure 2 Example Of a Concept Map

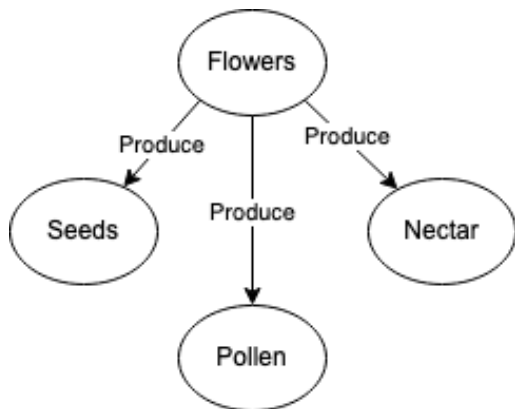


Figure 3 Example of Spoke Concept Map

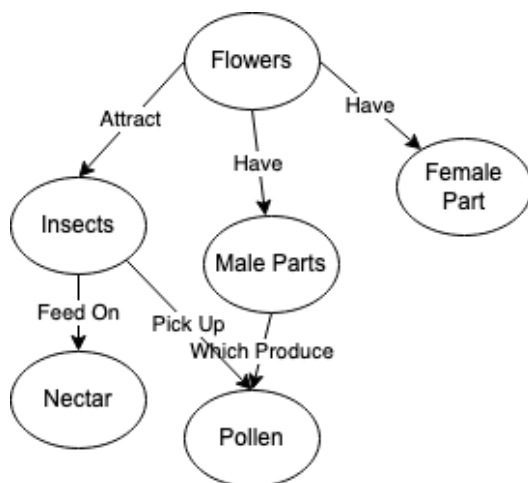


Figure 4 Example of Net Concept Map

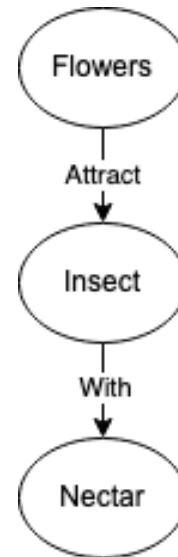


Figure 5 Example of Chain Concept Map

Concept map have several benefits if implemented in Computer Science. Concept map is easy to understand since it shows simple graphical representation of a concept and relation between the concept. Learners can share their concept map and collaborate to learn certain concept. Concept map also can be used to visualized data to lecturer about student misconception and their level of understanding [33]. However using concept map in Computer Science also have some challenging problems such as the difficulty when creating the concept map, cognitive overload, increase workload and scalability [30].

In general there are two types of concept map based on how student create the concept map the open ended concept map [34] and close ended concept map [21], [31]. The close ended is superior to open ended concept map in terms of automatic assessment process [35] because the learner map and expert map can be compared effectively.

CONSTRUCTION OF CONCEPT MAP

To build a concept map learner can use construct by self (open ended) or construct on scaffold (close ended) [21].

Construct on scaffold will provide the learner with a framework from an incomplete expert concept map. Some of the nodes and connecting relationships in the framework have been removed, so students must fill in the missing parts with the available answer choices to complete the concept map framework Figure 6.

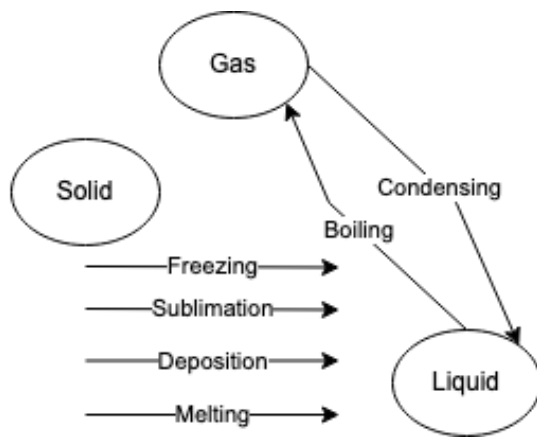


Figure 6 Sample of Construct On Scaffold Concept Map

By using construct on scaffold approach the student can focus on reconstructing the expert concept map thus it will reduce the student cognitive load, and because the student complete the same concept map build by expert the answer from student concept map and expert map can be analyzed programmatically.

FLUTTER WIDGET

Widget tree is an important concept in flutter layout. Widget tree describes a logical representation of widgets. Making the widget tree as minimal as possible, will reduce the time to process views and increase display efficiency [19]. An example of a widget tree in Flutter is shown in Figure 8.

Widgets in the widget tree are represented as nodes. Each widget may have its own state and any changes that occur to the widget will rebuild the widget and the child widgets underneath [19].

In this study the widget tree can be represented as a specialized concept map combination between spoke concept map and chain concept map. Where a node represent a concept and the relation between node can be represented as child or children in flutter layout. The component of concept map for flutter widget illustrated in Figure 7. The Students used the widget tree as practice exercises on the EasyFlutter app to better understand the idea of placing widgets in the Flutter layout, the complete example of a flutter widget concept map shown in Fugre 8.

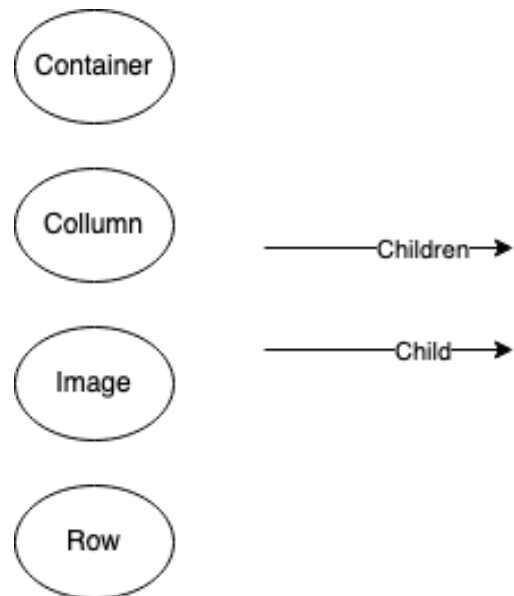


Figure 7 The Building Blocks of a Flutter Concept Map

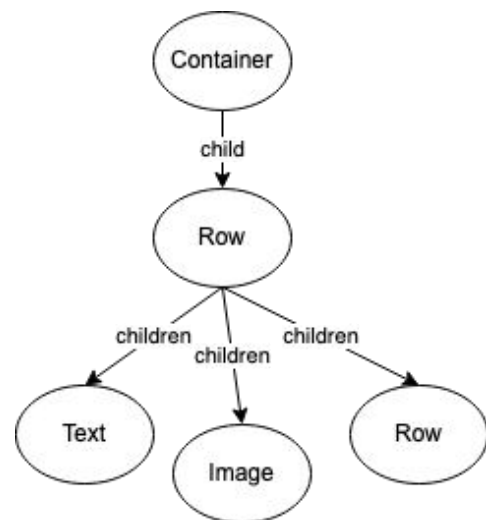
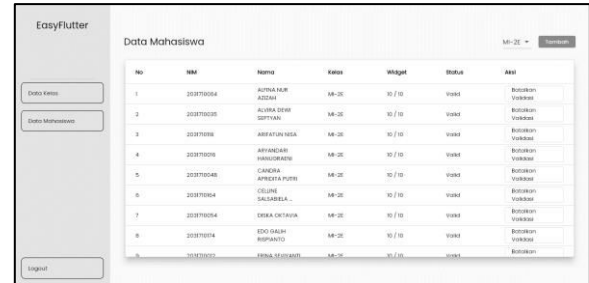


Figure 8. Example of a Flutter Layout Concept Map

EASYFLUTTER

EasyFlutter is a learning support application to help students understand the concept of widget arrangement in flutter layout using the construction-on-scaffold concept-map approach. This application provides practice questions for students using the construct on scaffold concept map approach, which presents concept maps from experts that have been removed from several parts, as well as answer choices that can

be used by students to complete concept maps by drag & drop answer choices to the concept map (empty section) Figure 9. Every time students drag & drop in the process of working on practice questions, student answers will be stored in the answer log Figure 10. This application provides monitoring features for lecturers. Lecturers can manage class data, practice data, log data, and student data Figure 11.



No	NIM	Nama	Kelas	Nilai	Status	Aksi
1	202310004	AUTHA NUR AZIZAH	IM-20	10 / 10	VAJIB	Berkas Videotape
2	202310005	ALYIA SIBER SOPYAN	IM-20	10 / 10	VAJIB	Berkas Videotape
3	202310006	ARIFATUN NISA	IM-20	10 / 10	VAJIB	Berkas Videotape
4	202310008	ARHANI RABINDRANI	IM-20	10 / 10	VAJIB	Berkas Videotape
5	202310040	CANDYA APRIYATI PURI	IM-20	10 / 10	VAJIB	Berkas Videotape
6	202310004	OSLINE SELASELA	IM-20	10 / 10	VAJIB	Berkas Videotape
7	202310004	DEBA CAHYA	IM-20	10 / 10	VAJIB	Berkas Videotape
8	202310014	EDO RAHM HIMPANTO	IM-20	10 / 10	VAJIB	Berkas Videotape
9	202310001	FENIA ALYANTRI	IM-20	10 / 10	VAJIB	Berkas Videotape

Figure 11. Teacher Report Interface

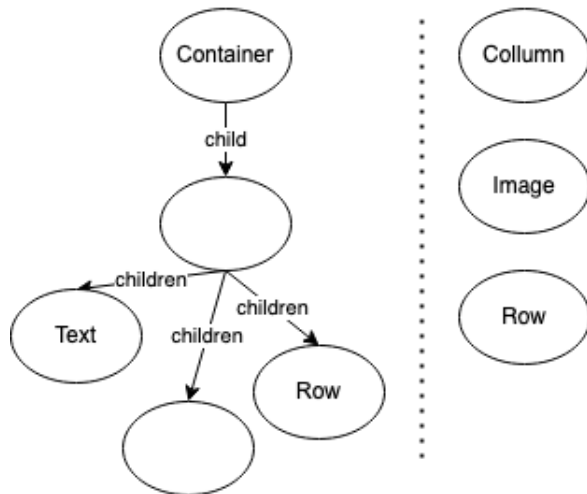


Figure 9. Example Of a Student Concept Map

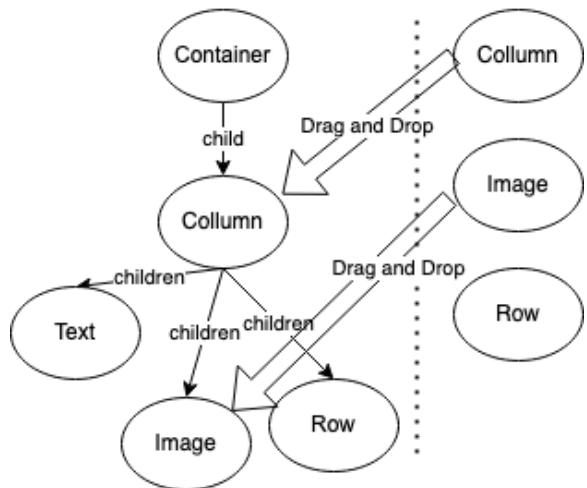


Figure 10. Student Dragging and Dropping Available Widget to Available Concept Map

METHOD

A. Application Development

Easyflutter developed using agile approach, it has two main actor the lecturers and the students. The feature for student and lecturer can be seen on Figure 12 and 13 respectively.

In Easyflutter student can register themselves, login, check available exercise and solve the exercise. While lecturer in Easyflutter have feature to manage class, manage exercise, and evaluate student answers.

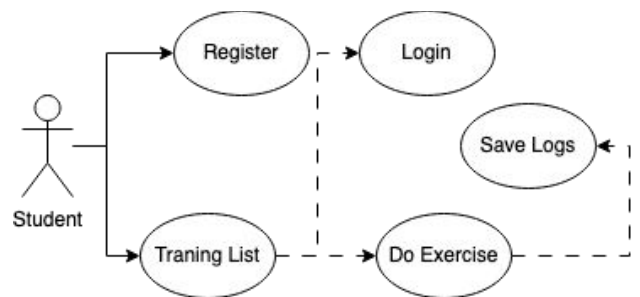


Figure 12. Student Feature in EasyFlutter

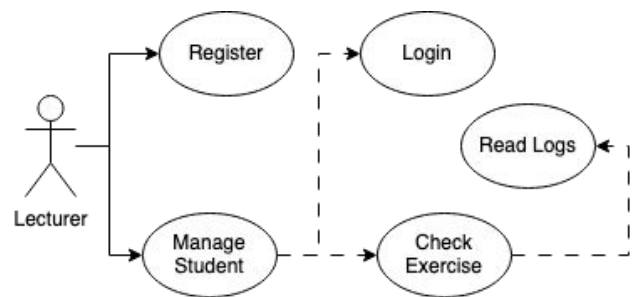


Figure 13. Lecturer Feature in Easy Flutter

Easyflutter build based on flutter for web and using firestore as database. The system architecture for Easyflutter is illustrated in figure 14.

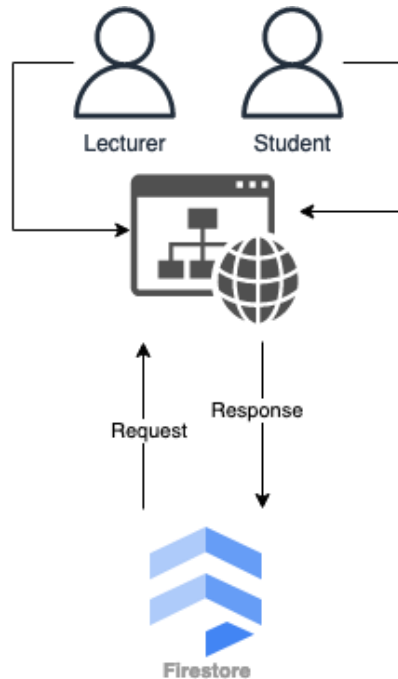


Figure 14. Easy Flutter System Architecture

B. Experiment

An experiment should be carried out to support the hypotheses. To provide a controlled environment, several activities will be built. Each student's activity will be restricted to the steps and time allotted.

The experiment was carried out for two weeks (meetings). 19 sophomore students were involved, majoring in Information Technology from State Polytechnic of Malang.

The first week is used to provide flutter layout material, introduce the application, and make application habituation by doing practice questions, as shown in Figure 15. Complete activities in the first weeks are:

1. Flutter Layout material: 50 Minutes
2. EasyFlutter introduction and explanation: 10 Minutes.
3. EasyFlutter free practice for habituation: 30 Minutes.

Flutter layout materials are provided to students to ensure students have an equivalent knowledge about flutter layout. The material is given using Google Slides for 50 minutes. After the flutter layout material is given, the researcher will introduce the EasyFlutter so that students know an overview of the application they will use. The introduction of the application was carried out by means of a 10-minute application demo. After the introduction of the application, the researcher will direct students to use the EasyFlutter so that students are familiar with using the application. Application

habituation is done by working on sample questions in the EasyFlutter for 30 minutes.

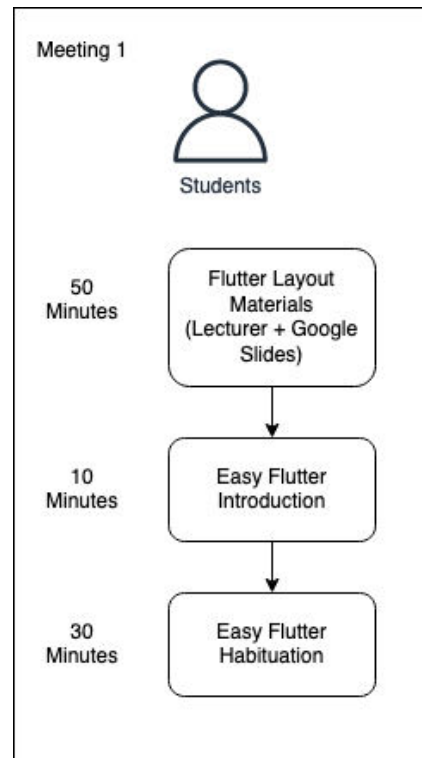


Figure 15. Experimental Setting Week 1

The second week is used to pre-test, app usage, post-test. This experiment illustrated in Figure 16. Complete activities in the second weeks are:

1. Pre-test: 15 Minutes.
2. App usage: 30 Minutes.
3. Post-test: 15 Minutes.

The pre-test is given using a google form with 10 questions multiple choice that can be done in 15 minutes. After the pre-test is given, the researcher will direct the students to use the EasyFlutter application in the hope that it can help them understand the concept of widget arrangement in flutter layout. Students use the EasyFlutter application by doing 10 practice questions in 30 minutes. After using the application, the researcher will give the students a post-test to determine the student's ability regarding the topic of flutter layout after using the EasyFlutter application. The post-test was given using a google form with the same time and questions as the pre-test. The results of pre-test and post-test described in Table 1.

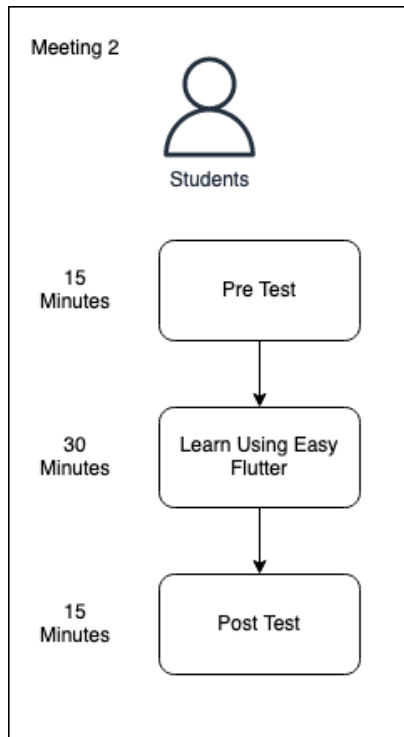


Figure 16. Experimental Setting Week 2

RESULT AND DISCUSSION

Pre-test and post-test results from the experiment will be used to represent students' understanding. The box plot for the pre-test shows the lowest score is 10, the median value is 60, the highest score is 90, mean score is 54.7368. The box plot for the pre-test shows the lowest score is 30, the median value is 60, the highest score is 100, and the mean score is 65.2632. The box plot pre-test and post-test as shown in Figure 17. The line chart for pre-test and post-test showed a significant increase, there were two data that showed a decrease as shown in Figure 18.

C. Analysis

Kolmogorov-Smirnov analysis was used to determine whether the data were normally distributed or not. If the data is normally distributed, it will use parametric analysis. If the data is not normally distributed, non-parametric analysis will be used. Table 2 explain the results of the normality test using the Kolmogorov-Smirnov test.

Table 1. Pre test and Post test Result

Learning Result			
No	Pre-Test	EasyFlutter	Post-Test
1	30	10/10	40
2	80	10/10	90
3	60	10/10	30

4	80	10/10	100
5	60	10/10	60
6	40	10/10	60
7	30	10/10	30
8	10	8/10	30
9	80	10/10	90
10	90	10/10	90
11	30	10/10	60
12	10	10/10	30
13	90	10/10	100
14	40	10/10	80
15	80	10/10	90
16	20	10/10	30
17	80	10/10	100
18	40	5/10	50
19	90	10/10	80

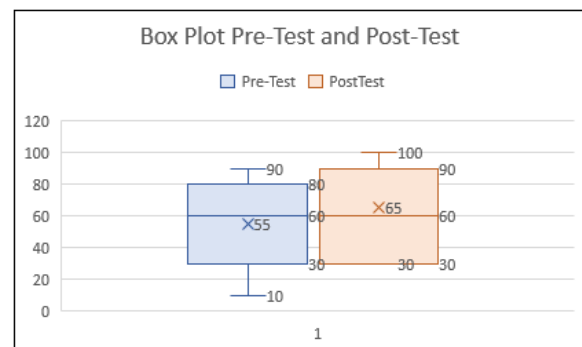


Figure 17. Box Plot for Pre Test and Post Test

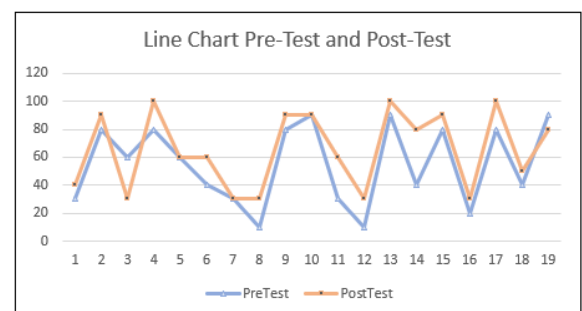


Figure 18, Line Chart for Pre Test and Post Test

Table 2. Pre & Post Test Kolmogorov-smirnov

Test	N	Mean	SD	Asymp. Sig. (2-tailed)
Pre-Test	19	54.7368	28.55179	0.008
Post-Test	19	65.2632	27.56173	0.091

Based on the results of the Kolmogorov Smirnov test in Table 1, the Asymp value Sig. (2-tailed) of the pre-test value is 0.008 (less than 0.05), which means the pre-test data is not normally distributed and the Asymp value. Sig. (2-tailed) of the post-test value is 0.091 (more than 0.05), which means that the post-test data is normally distributed.

Table 3. Pre & Post Test Wilcoxon Ranks

		N	Mean Rank	Sum of Ranks
Post Test – Pre Test	Negative Ranks	2	9.50	19.00
	Positif Ranks	14	8.36	117.00
	Ties	3		
	Total	19		

Based on the result of the Pre-Test and Post-Test Wilcoxon Ranks in in Table 3, it is known that the Negative Ranks or difference is 2 for decreased data, 9.5 for the average decrease, and 19.00 for the number of decreased data. Positive Ranks or the difference (positive) between pre-test and post-test is 14 for increased data, 8.36 for average increase, and 117.00 for increased number of data. There are 3 data that are the same, and the total data is 19.

Table 4. Pre & Post Test Wilcoxon Test Statistics

	Post Test – PreTest
Z	-2.579
Asymp. Sig. (2-tailed)	0.010

Based on the result of pre-test and post-test of the Wilcoxon Test Statistics in Table 4, it is known that the Z value is -2.579 and the Asymp value. Sig. (2-tailed) of 0.01 (less than 0.05), it can be concluded that there is a significant difference between the pre-test and post-test scores.

CONCLUSION

In this experiment, the use of the EasyFlutter application had a significant positive impact on students' post-test scores. The average post-test score is higher (increased) when compared to the pre-test average value, so descriptively it can be concluded that there is an average difference between the pre-test and post-test scores. Next, to prove whether the difference is significant or not, it is necessary to interpret the results of data processing from the Wilcoxon

test. The results showed that there was a significant difference between the pre-test and post-test scores ($p < 0.05$). These results indicate that the EasyFlutter application helps students to increase their post-test scores from their pre-test scores. EasyFlutter application also helps students understand the concept of widget arrangement in flutter layout.

EasyFlutter cannot be utilized in a real class based on the results of this experiment, despite claims to the contrary. It still requires a great deal of supporting data, which necessitates the engagement of a sizable class size, a knowledgeable lecturer, and extended use of the application. However, the data gathered from this experiment might provide a sound basis for doing subsequent study.

REFERENCES

- [1] M. G. Jamil and S. O. Isiaq, "Teaching technology with technology: approaches to bridging learning and teaching gaps in simulation-based programming education," *Int. J. Educ. Technol. High. Educ.*, vol. 16, no. 1, p. 25, Dec. 2019, doi: 10.1186/s41239-019-0159-9.
- [2] S. Bergin, R. Reilly, and D. Traynor, "Examining the role of self-regulated learning on introductory programming performance," in *Proceedings of the 2005 international workshop on Computing education research - ICER '05*, Seattle, WA, USA: ACM Press, 2005, pp. 81–86. doi: 10.1145/1089786.1089794.
- [3] J. Bennedsen and M. E. Caspersen, "Failure rates in introductory programming," *ACM SIGCSE Bull.*, vol. 39, no. 2, pp. 32–36, Jun. 2007, doi: 10.1145/1272848.1272879.
- [4] S. Bergin and R. Reilly, "The influence of motivation and comfort-level on learning to program," 2005.
- [5] M. N. Giannakos, I. O. Pappas, L. Jaccheri, and D. G. Sampson, "Understanding student retention in computer science education: The role of environment, gains, barriers and usefulness," *Educ. Inf. Technol.*, vol. 22, no. 5, pp. 2365–2382, Sep. 2017, doi: 10.1007/s10639-016-9538-1.
- [6] A. Robins, J. Rountree, and N. Rountree, "Learning and Teaching Programming: A Review and Discussion," *Comput. Sci. Educ.*, vol. 13, no. 2, pp. 137–172, Jun. 2003, doi: 10.1076/csed.13.2.137.14200.
- [7] B. Hanks, C. McDowell, D. Draper, and M. Krnjajic, "Program quality with pair programming in CS1," in *Proceedings of*

- the 9th annual SIGCSE conference on Innovation and technology in computer science education, Leeds United Kingdom: ACM, Jun. 2004, pp. 176–180. doi: 10.1145/1007996.1008043.
- [8] T. Jenkins, "ON THE DIFFICULTY OF LEARNING TO PROGRAM".
- [9] M. M. Villamor, "A review on process-oriented approaches for analyzing novice solutions to programming problems," *Res. Pract. Technol. Enhanc. Learn.*, vol. 15, no. 1, p. 8, Dec. 2020, doi: 10.1186/s41039-020-00130-y.
- [10] K. Tsutsui and H. Takada, "A classroom SNS to promote reflective activity in programming learning for children," *Res. Pract. Technol. Enhanc. Learn.*, vol. 13, no. 1, p. 2, Dec. 2018, doi: 10.1186/s41039-018-0069-9.
- [11] A. Carbone, J. Hurst, I. Mitchell, and D. Gunstone, "An Exploration of Internal Factors Influencing Student Learning of Programming".
- [12] N. J. G. Falkner and K. E. Falkner, "A fast measure for identifying at-risk students in computer science," in *Proceedings of the ninth annual international conference on International computing education research*, Auckland New Zealand: ACM, Sep. 2012, pp. 55–62. doi: 10.1145/2361276.2361288.
- [13] M.-H. N. C. Vee, B. Meyer, and K. L. Mannock, "Understanding novice errors and error paths in Object-oriented programming through log analysis".
- [14] M. C. Jadud, "A First Look at Novice Compilation Behaviour Using BlueJ," *Comput. Sci. Educ.*, vol. 15, no. 1, pp. 25–40, Mar. 2005, doi: 10.1080/08993400500056530.
- [15] A. Carbone, J. Hurst, I. Mitchell, and D. Gunstone, "An Exploration of Internal Factors Influencing Student Learning of Programming," p. 11, 2009.
- [16] M.-H. N. C. Vee, B. Meyer, and K. L. Mannock, "Understanding novice errors and error paths in Object-oriented programming through log analysis," p. 8, 2006.
- [17] I. Ouahbi, F. Kaddari, H. Darhmaoui, A. Elachqar, and S. Lahmine, "Learning Basic Programming Concepts by Creating Games with Scratch Programming Environment," *Procedia - Soc. Behav. Sci.*, vol. 191, pp. 1479–1482, Jun. 2015, doi: 10.1016/j.sbspro.2015.04.224.
- [18] C.-Y. Tsai, "Improving students' understanding of basic programming concepts through visual programming language: The role of self-efficacy," *Comput. Hum. Behav.*, vol. 95, pp. 224–232, Jun. 2019, doi: 10.1016/j.chb.2018.11.038.
- [19] A. Biessek, *Flutter for beginners: an introductory guide to building cross-platform mobile applications with Flutter and Dart 2*. 2019.
- [20] S. Giordano and P. Mainkar, *Google Flutter Mobile Development Quick Start Guide*. Place of publication not identified: Packt Publishing, 2019. Accessed: Nov. 24, 2021. [Online]. Available: <https://www.safaribooksonline.com/library/view/title/9781789344967/?ar?orpq&email=^u>
- [21] K. E. Chang, Y. T. Sung, and S. F. Chen, "Learning through computer-based concept mapping with scaffolding aid," *J. Comput. Assist. Learn.*, p. 13, 2001.
- [22] J. D. Novak, "Concept mapping: A useful tool for science education," *J. Res. Sci. Teach.*, vol. 27, no. 10, pp. 937–949, Dec. 1990, doi: 10.1002/tea.3660271003.
- [23] K.-E. Chang, Y.-T. Sung, and I.-D. Chen, "The Effect of Concept Mapping to Enhance Text Comprehension and Summarization," *J. Exp. Educ.*, vol. 71, no. 1, pp. 5–23, Jan. 2002, doi: 10.1080/00220970209602054.
- [24] F. G. W. C. Paas, "Training strategies for attaining transfer of problem-solving skill in statistics: A cognitive-load approach," *J. Educ. Psychol.*, vol. 84, no. 4, pp. 429–434, Dec. 1992, doi: 10.1037/0022-0663.84.4.429.
- [25] A. D. Katayama and D. H. Robinson, "Getting Students 'Partially' Involved in Note-Taking Using Graphic Organizers," *J. Exp. Educ.*, vol. 68, no. 2, pp. 119–133, Jan. 2000, doi: 10.1080/00220970009598498.
- [26] A. M. O'Donnell, D. F. Dansereau, and R. H. Hall, "Knowledge Maps as Scaffolds for Cognitive Processing," *Educ. Psychol. Rev.*, vol. 14, no. 1, 2002.
- [27] J. D. Novak and A. J. Cañas, "Theoretical Origins of Concept Maps, How to Construct Them, and Uses in Education".
- [28] I. M. Kinchin, D. B. Hay, and A. Adams, "How a qualitative approach to concept map analysis can be used to aid learning by illustrating patterns of conceptual development," *Educ. Res.*, vol. 42, no. 1, pp. 43–57, Jan. 2000, doi: 10.1080/001318800363908.
- [29] B. S. Andoko, Y. Hayashi, T. Hirashima, and A. N. Asri, "Improving English reading for EFL readers with reviewing kit-build

- concept map,” *Res. Pract. Technol. Enhanc. Learn.*, vol. 15, no. 1, p. 7, Dec. 2020, doi: 10.1186/s41039-020-00126-8.
- [30] V. dos Santos, É. F. de Souza, K. R. Felizardo, and N. L. Vijaykumar, “Analyzing the Use of Concept Maps in Computer Science: A Systematic Mapping Study,” *Inform. Educ.*, vol. 16, no. 2, pp. 257–288, Oct. 2017, doi: 10.15388/infedu.2017.13.
- [31] T. Hirashima, K. Yamasaki, H. Fukuda, and H. Funaoi, “Framework of kit-build concept map for automatic diagnosis and its preliminary use,” *Res. Pract. Technol. Enhanc. Learn.*, vol. 10, no. 1, p. 17, Dec. 2015, doi: 10.1186/s41039-015-0018-9.
- [32] J. Keppens and D. Hay, “Concept map assessment for teaching computer programming,” *Comput. Sci. Educ.*, vol. 18, no. 1, pp. 31–42, Mar. 2008, doi: 10.1080/08993400701864880.
- [33] V. P. Gurupur, G. Pankaj Jain, and R. Rudraraju, “Evaluating student learning using concept maps and Markov chains,” *Expert Syst. Appl.*, vol. 42, no. 7, pp. 3306–3314, May 2015, doi: 10.1016/j.eswa.2014.12.016.
- [34] K. E. de Ries, H. Schaap, A.-M. M. J. A. P. van Loon, M. M. H. Kral, and P. C. Meijer, “A literature review of open-ended concept maps as a research instrument to study knowledge and learning,” *Qual. Quant.*, vol. 56, no. 1, pp. 73–107, Feb. 2022, doi: 10.1007/s11135-021-01113-x.
- [35] W. Wunnasri, J. Pailai, Y. Hayashi, and T. Hirashima, “Validity of Kit-Build Method for Assessment of Learner-Build Map by Comparing with Manual Methods,” *IEICE Trans. Inf. Syst.*, vol. E101.D, no. 4, pp. 1141–1150, 2018, doi: 10.1587/transinf.2017EDP7177.