A Proposal of Code Typing Problem for Basic Java Programming Learning

1st Mustika Mentari
Department of Information Technology
State Polytechnic of Malang
Malang, Indonesia
must.mentari@polinema.ac.id

2nd Nobuo Funabiki
Department of Information and Communication Systems
Okayama University
Okayama, Japan
funabiki@okayama-u.ac.jp

3rd Huiyu Qi
Department of Information and Communication Systems
Okayama University
Okayama, Japan
pbr17iqa@s.okayama-u.ac.jp

4th Yan Watequlis Syaifudin
Department of Information Technology
State Polytechnic of Malang
Malang, Indonesia
qulis@polinema.ac.id

5th Imam Fahrur Rozi
Department of Information Technology
State Polytechnic of Malang
Malang, Indonesia
imam.rozi@polinema.ac.id

6th Komang Candra Brata
Department of Computer Science
Brawijaya University
Malang, Indonesia
k.candra.brata@ub.ac.id

2023 International Electronics Symposium (IES) | 979-8-3503-1473-1/23/$31.00 ©2023 IEEE | DOI: 10.1109/IES59143.2023.10242575

***Abstract*—Java is an object-oriented programming language that helps programmers to run programs easily. A web-based Java Programming Learning Assistant system (JPLAS) has been developed to help improve skills through self-learning. Learning Java programming for beginners requires quick practice routines in code typing problems, but JPLAS didn't have that problem type yet. In this paper, we propose Code Typing Problems (CTP) as a new type in JPLAS. CTP is a new instance that displays code references that must be rewritten that cannot be copied and pasted. CTP evaluation was given to 21 second-semester students taking a Java programming course. The problem-solving did on 8 Java programming topics, where each case consisted of at least two Java files with 624 lines of program code. The results show that the highest average level of truth is 99.1%, and the trend of the ratio of the average number of submissions to the number of question lines is decreasing. A decreasing ratio indicates an increase in students' understanding while using CTP.**

***Index Terms*—java programming, learning assistant system, code typing problems**

I. Introduction

Java language is a good choice for beginners in learning programming. The reason is that Java language is a programming language that is safe to use, easy to understand in terms of memory management, and supports application development or game creation [1] [2] [3]. Java Programming is often difficult for students to learn and spends time understanding the material [4]. Self-learning of Java programming is a solution for student problems compared to traditional methods. Students determine their own strategies when learning to program and can test their understanding anytime, anywhere, quickly, and responsively [4]. The need for self-learning Java programming training has been carried out by the Java Programming Learning Assistant System (JPLAS) [5].

JPLAS is a form of self-regulated programming learning consisting of element fill-in-blank problems (EFP), value trace problems (VTP), statement fill-in-blank problems (SFP), and code writing (CWP) problems [5], grammar-concept understanding problem (GUP) [6], code correction problem (CRP), code completion problem (CCP) [7], code amendment problem (CAP) [8]. The fill-in-blank problem simultaneously trains the ability to read and write the Java programming language [9]. VTP in JPLAS presents a collection of questions regarding variable values in the program code. VTP focuses on mapping the output messages generated by the provided program code [10]. The fill-in-blank problem statement in JPLAS presents the question as program code. Students who use this facility must fill in the blank program code in certain parts of the available complete program code [11]. CWP at JPLAS requires students to implement the complete program code starting from the beginning according to the problems given in the questions. The program code written as an answer will be automatically checked behind the scenes using JUnit [12]. The grammar-concept understanding problem presents the source code questions about grammar concepts [6]. The code correction problem provides code of questions that are deliberately wrong so that they can be corrected by the user or student so that the answers can pass the JUnit unit test [13]. The code completion problem presents an incomplete source as a problem that is solved by answering the blank part of the code. [7]. The code amendment problem presents generating errors in certain parts. The user must examine which part is wrong and correct that part. [8]. The sub-problems provided by JPLAS can improve Java programming skills with independent learning.

This study [14] explains that programming skills are higher with program modifications. The initial stage before this modification is carried out requires an understanding of syntax or grammar in the Java language. This stage is done by rewriting the existing program code. This method of retyping program code is more targeted according to the expected goals if the program's source code referred to is in the form of a code that

Authorized licensed use limited to: BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE. Downloaded on October 02,2023 at 14:19:35 UTC from IEEE Xplore. Restrictions apply.

cannot be modified to minimize students' forgetfulness due to just copying and pasting program code. The learning process in the data structure practicum course for students majoring in the information technology department at the State Polytechnic of Malang requires weekly training in code typing problems from the examples provided, then observing the source code model. The JPLAS type to overcome this problem in the form of a code typing problem is still not available.

The purpose of this paper is to apply Code Typing Problems (CTP) at JPLAS to students majoring in Information Technology in semester one as novice programming students in the Data Structures course and discuss the results of the job test. CTP is a new instance that displays the main program code for code references that need to be rewritten in a code format that cannot be copied and pasted, and the code entry program response form that is generated from the main program code. Java Programming is the main object that is tested on the web that is created. There are two types of CTP to test student's basic programming skills, namely, code typing problems basic (CTP-Basic) and code typing problem plus (CTP-Plus). These two things make students not only focus on how to write Java code but also on understanding the logic flow of each topic being taught.

Proof of increasing programming skills using CTP in JPLAS is examined and described. The evidence came from reviews of the results of student trials on a total of eight topics tested. Moreover, the causes of students' difficulties in learning basic programming were also examined by students who still often made mistakes even though they had done the challenge questions provided by CTP many times. Proving will also review students' logical abilities in understanding case studies that must be completed in the form of Java program code.

The evaluation step was carried out by testing CTP on JPLAS on 21 Indonesian students to complete a total of around 624 lines of program code. The code is divided into 17 Java files, which consist of 8 learning topics in the Data Structures course. The learning topics in question are objects, arrays of objects, searching, sorting part 1, sorting part 2, Stack, Queue, and Linked List. The average number of correct answers from all students for all questions in CTP reached $98,85\%$. The results of the evaluation of CTP work for lecturer students can get to know more about each student's programming abilities. Lecturers can also vary the procedures in CTP-JPLAS to add to the challenges in understanding Java Programming for their students. The results show that the highest average level of truth is 99.1%. The ratio of the average number of submissions to the number of problem lines is downward. A decrease in the ratio indicates that students' understanding of the use of CTP is improving.

This paper consists of five sections as follows: Section I discusses the Introduction. Section II describes the Related Works of CTP. Section III discusses the implementation of CTP in JPLAS. Section IV presents an evaluation of the implementation results for students, and Section V concludes the results of working on this paper and future research.

II. Related Works

In this section, we discuss the related works about some types of JPLAS.

A. JPLAS Overview

JPLAS has several types of problems with various levels of difficulty. Based on the MVC model, Java for the model (M), HTML/CSS/JavaScript for the View (V), and JSP for the controller (C). JPLAS adopts the MySQL database server. JPLAS uses a database to store information on users, problems, and answers. [15].

JPLAS has several problem types that continue to be developed, especially in Java Programming. Variations in the types of JPLAS [16] learning provides opportunities for students learning programming to increase their ability to learn independently and briefly find out or confirm whether their answers are wrong or right. This variant also has something to do with the level of understanding already owned by novice students learning Java Programming. A beginner studying Java Programming needs independent practice outside lectures to strengthen skills and develop abilities [17].

- GUP requires the system to answer questions related to the grammar concepts of the Java language. Answers can consist of numbers, words, or short sentences. The answer from the user will be validated by matching the string between the answer key and the answer from the student [6].
- VTP trains students to read program code from top to bottom to determine the logical flow of the presented problems. The results of trials by students who try to show primarily satisfactory results, only a few do not receive good results and require more handling or attention related to learning programming [10].
- CRP requires the user to correct the source code presented, which has errors. Corrections by students must be thorough until the code passes the test code given to JUnit [13].
- EFP prompts you to fill in blanks or missing elements in the specified source code with the correct word. The position of empty elements is explicitly shown in the source code [].
- CCP requires students to locate missing elements from the source code and complete them with the correct code. Finding the location of the fault is the initial challenge that must be sought [7].
- CAP changed several elements. Change in the form of intentional errors so the user examines and corrects them [8].
- CWP on JPLAS used at a higher level Java learning can be done. Students must write source code to solve the problems presented successfully [18]. The results of the source code created by students will be automatically verified using JUnit to check whether it is true or false. CWP shows that verified results have good validity and effectiveness [19].

Authorized licensed use limited to: BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE. Downloaded on October 02,2023 at 14:19:35 UTC from IEEE Xplore. Restrictions apply.

B. JPLAS Limitation

Recent JPLAS problems have been discussed in this section, especially for early learning beginner sessions, although some references discuss Java Programming learning technology at intermediate and upper levels. Most of what has been discussed requires students to type source code, regardless of the difficulty level of Java Programming learning technology. The correctness of the source code written requires practice so that students are accustomed to using syntax in the Java language to support an understanding of program logic that needs continuous training.

JPLAS did not yet have the type of problem learning basic Java programming for retyping program code. The increase in students' ability to learn Java programming without them realizing it by doing the routine retyping of the program code questions provided, and there is also no type of problem JPLAS is doing retyping program code while observing errors. Existing from the question of the program code provided.

III. Proposal of Code Typing Problems (CTP)

In this section, we present the implementation of CTP on JPLAS. CTP consists of two types, namely CTP-Basic to train the memory of students and also the initial program flow and also CTP-Plus to train deeper into the program's logic flow.

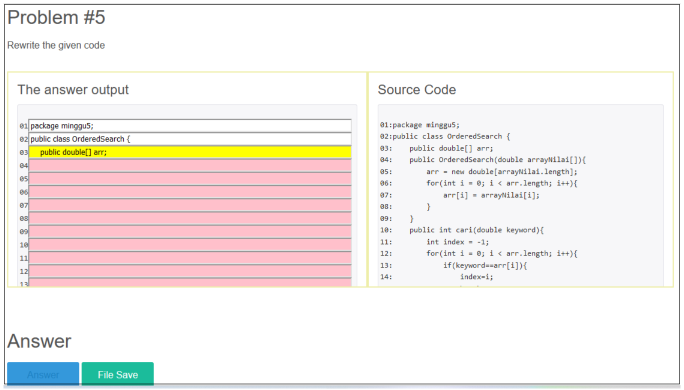
CTP was made using JPLAS's User Interface generator, which uses HTML, JavaScript, and CSS. Adjustments were then made to divide the right and left sections of the page into areas of the program code questions that could not be edited and copied and pasted and space for retyping the program code answers.

Students can directly access the CTP website address that has been hosted. They work on every problem available at CTP. After finishing work, they can send the results of their work to the Learning Management System (LMS) used during the data structure course.

A. CTP-Basic

The CTP-Basic page divides the core halves on the right and left. The right side contains source code regarding Objects, Array of objects, Sorting (Bubble Sort), and Searching (Ordered Search). The source code displayed does not have a hint and cannot be copied and pasted by students. They see only the left of this. On the left is the student answer area. As shown in "Fig. 1", this answer area has three types of colour clues as a marker of the correct answer from code typing or rewriting the program code they are doing. The white mark indicates that the program line is accurate and neat. The yellow mark means that the Java language syntax is correct; it's just that the writing of the program code is still untidy. The pink indicates still errors when retyping the program code from the source code questions provided.

After all the source code questions have been rewritten in the answers section, the student presses the submit button to verify whether the answers that have been written are correct. If the answers are not entirely accurate, then code-typing answers that are still colored pink or yellow can be corrected by the student.



```
package minggu5;
public class OrderedSearch {
    public double[] arr;
```

```
package minggu5;
public class OrderedSearch {
    public double[] arr;
    public OrderedSearch(double arrayNilai[]){
        arr = new double[arrayNilai.length];
        for(int i = 0; i < arr.length; i++){
            arr[i] = arrayNilai[i];
        }
    }
    public int cari(double keyword){
        int index = -1;
        for(int i = 0; i < arr.length; i++){
            if(keyword==arr[i]){
                index=i;
```

Fig. 1. CTP-Basic Page View

Behind the layer, the verification button will record information such as the number of question program lines, how many correct answers program lines, the number of attempts to answer questions in each problem number, and the percentage of correct answers. In addition, there is also a button to save the code typing problem's solutions into the .txt file format, the contents of which are program blocks in the form of java classes that have been answered. This file can be used further by students.

An example of a source code snippet for a searching topic that discusses the Ordered Search method is shown in "Fig. 2". All program lines on this topic are discussed, starting from the code for making packages, creating classes, and the variables used for logic-ordered search. Students studying data structures have gone through the topics taught in introductory programming courses. CTP will make students automatically repeat topics in basic programming, starting from variables, operators, selection, looping, arrays, or functions.

```
package minggu5;
public class OrderedSearch {
    public double[] arr;
    public OrderedSearch(double arrayNilai[]){
        arr = new double[arrayNilai.length];
        for(int i = 0; i < arr.length; i++){
            arr[i] = arrayNilai[i];
        }
    }
```

Fig. 2. CTP-Basic Code Snippet Example

B. CTP-Plus

CTP-Plus has the same appearance as CTP-Basic. The difference is that CTP-Plus has a question program code line that is still wrong. If the student is not careful and tricked into retyping the code, the answer is automatically considered inappropriate. CTP questions are given assuming the student working on them already understands the theory of the previously provided topics. That is, students should

Authorized licensed use limited to: BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE. Downloaded on October 02,2023 at 14:19:35 UTC from IEEE Xplore. Restrictions apply.

be careful and be able to detect if there are lines in the program that are not correct. The topics used in CTP-Plus are Stack, Queue, and Linked List. These three materials have their variations of operations, and they are diverse. If at the time they understand the material, in theory, they already understand, then they continue to work on this CTP-Plus. It is possible that students still make mistakes when solving existing problems. The mistake happened because the types of operations of the three topics discussed are quite a lot. The main effect of CTP-Plus requires students to type one by one line of code, learning from mistakes, and retyping while thinking the logic flow according to the material previously studied.

```
1  package minggu9;
2  public class Stack {
3      int size;
4      int top;
5      int data[];
6      public Stack(int size) {
7          this.size = size;
8          data = new int[size];
9          top = -1;
10     }
11     public boolean IsFull() {
12         if (top == size - 4) {
13         return true;
14         } else {
15         return true;
16         }
17     }
18     public boolean IsEmpty() {
19         if (top == -10) {
20         return true;
21         } else {
22         return false;
23         }
24     }
```

Fig. 3. CTP-Plus Code Snippet Example

One of the program code snippets used in CTP Plus is shown in “Fig. 3”. There was an error in the source code of the questions in lines 12 and 19. There are constants in the form of numbers that are deliberately blamed on the problem. Student understanding will be tested at this point. Before working on this problem, students were taught the logic flow and the correct constant value in the problem question.

IV. Evaluation

In this section, we present an evaluation of CTP in JPLAS. The system was tested on 21 Indonesian students taking data structure courses. Students work on a total of 8 topics. This topic is divided into two Java classes or at least two question numbers. Each topic consists of two Java classes because the data structure lecture material obtained by these students has started introducing object-oriented programming, even though it is only about what objects are and how to apply them to create object classes from existing classes. Separate from the class that has the main function. Students can start CTP by solving questions that consist of rewriting program code on CTP-Basic, as well as rewriting and modifying program code that is still greeting on CTP-Plus. Students can submit several times until their answer is correct. CTP evaluation is divided into two, namely evaluation for CTP-Basic and CTP-Plus.

A. CTP-Basic

The results of CTP-Basic work by a total of 21 students with a total of 10 problem numbers can be seen in “Fig. 4”.The figure shows the average number of submissions for the ten problem numbers and the average percentage of correct answers for all questions.

On average, most submissions were made by student number 10, with 109 submissions for all program lines, of which 203 had to be rewritten. The lowest average number of submissions was made by student number one, with 3 submissions for the same number of question lines.

The average correct answer shown in CTP-Basic is $99,1\%$. The observation results show that Student number one, who has the lowest average number of submissions, is the student who has the best level of programming ability among other students. Moreover, student number 10 has completed the retyping code in CTP-Basic well. The student’s history explained that during the third submission on problem number 4, the code typing answer was correct but not perfect. There are still several lines of program code that have not been rewritten with neat spacing. Therefore, student number 10 tried to perfect the program code and made him repeatedly submit it. Most trials were carried out when rewriting the looping section, as shown in “Fig. 5”. There are many whitespace areas that students often forget to write about.

Most of the correctness of all students’ answers to all problems in CTP-Basic has been carried out properly. Eight problems can be solved perfectly until 100% is correct, there are only two problems that do not reach maximum percentages, but they can still be said to be very good because the correctness of the answers is more than 94%.

TABLE I
CTP-Basic Result
(Average Number Of Submissions, Number Of Lines, and Comparison Ratio)

Problem	Average Number of Submissions	Number of Lines	Comparison Ratio
1	177	20	8,85
2	7	15	0,47
3	2	5	0,40
4	21	22	0,95
5	16	30	0,53
6	15	16	0,94
7	36	42	0,86
8	7	12	0,58
9	15	29	0,52
10	7	12	0,58

The relationship between the number of lines for each question and the average number of submissions has been further evaluated, as shown in “Table. I”. The relationship between the two items is calculated using a comparison ratio.

Authorized licensed use limited to: BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE. Downloaded on October 02,2023 at 14:19:35 UTC from IEEE Xplore. Restrictions apply.

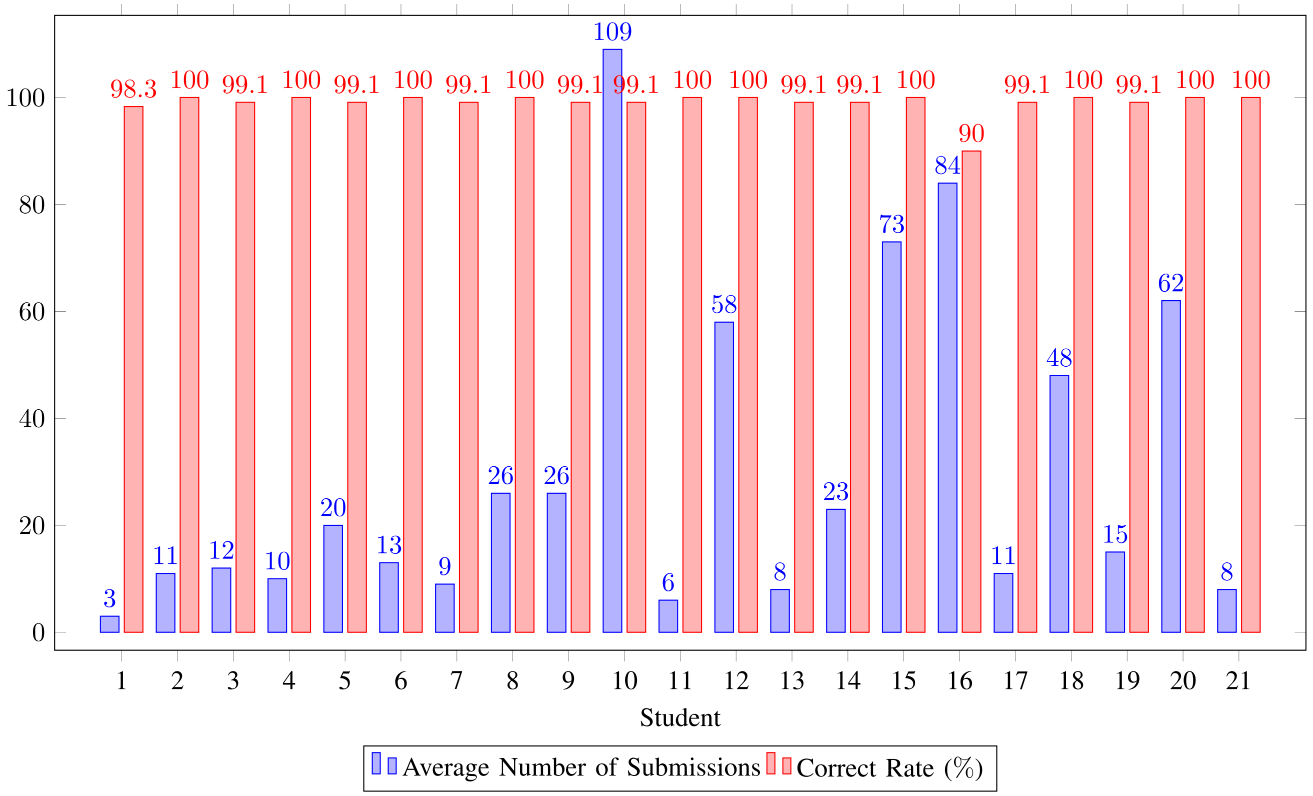
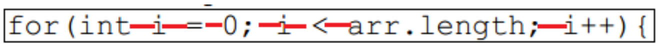


Fig. 4. CTP-Basic Result (Number of Submissions and Correct Rate)



```
for(int i = 0; i < arr.length; i++){
```

Fig. 5. Opportunity of imperfections in CTP-Basic

If the value is closer to zero, it shows that student learning outcomes are improving. Question number one has the highest percentage. The first time students do CTP, students face many problems with question number one. Students need adjustments when they first start using CTP. Over time the comparison ratio is far below the percentage of the first problem. Students quickly adapt to work on problem number two. Question number two's initial comparison ratio was 8.85 to 0.47, indicating a considerable distance between the ratio values. The comparison ratio of most of the tables shows that the six question numbers are worth less than 0.6. Apart from question number 1, questions 4, 6, and 7 have ratio values of 0.95, 0.94, and 0.86. Although it is still not too far from the value of 1, this ratio shows extraordinary results. The observation results show that the number of submissions for questions 4, 6, and 7 for most students is not too high. The highest number of submission contributions was student number 10, for the reasons previously explained.

B. CTP-Plus

The CTP-Plus evaluation is the same as CTP-Basic by presenting the same problems to 21 students. There is a slight difference with CTP-Basic; CTP-Plus has fewer problems but a vaster number of lines of program code. "Fig. 6" shows the average number of submissions and the average percentage of correct answers for all CTP-Plus questions. On average, the highest number of submissions was made by student number 10, with 47 attempts for all program lines that had to be rewritten, as many as 421 lines. The lowest average number of submissions was made by student number six of 4 for the same number of lines of questions.

The average correct answer displayed on CTP-Plus is 98.61%. The observation results showed that Student number six, who had the lowest average number of CTP-Plus submissions, was one of the students who achieved 100% correctness for the seven CTP-Plus questions. Facts show that only three students got 100% truth in working for CTP-Plus. Student number 10 has successfully retyped the code on CTP-Plus, with a correct proportion of 98% of all questions. The history of the student explained that there were questions that caused the student to try to answer repeatedly, namely questions 11, 13, and 16. These three questions had the most program lines in CTP-Plus, namely 62, 96, and 163 program lines. The history of student number 10 when solving the third problem shows that the student did not finish retyping the program code but had completed it. Maybe out of curiosity, the program lines written are correct or not. So, type a few lines, then click submit, even though the third question is a long program line. That's what causes the number of applications high; the student has not completed the program code to the

Authorized licensed use limited to: BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE. Downloaded on October 02,2023 at 14:19:35 UTC from IEEE Xplore. Restrictions apply.

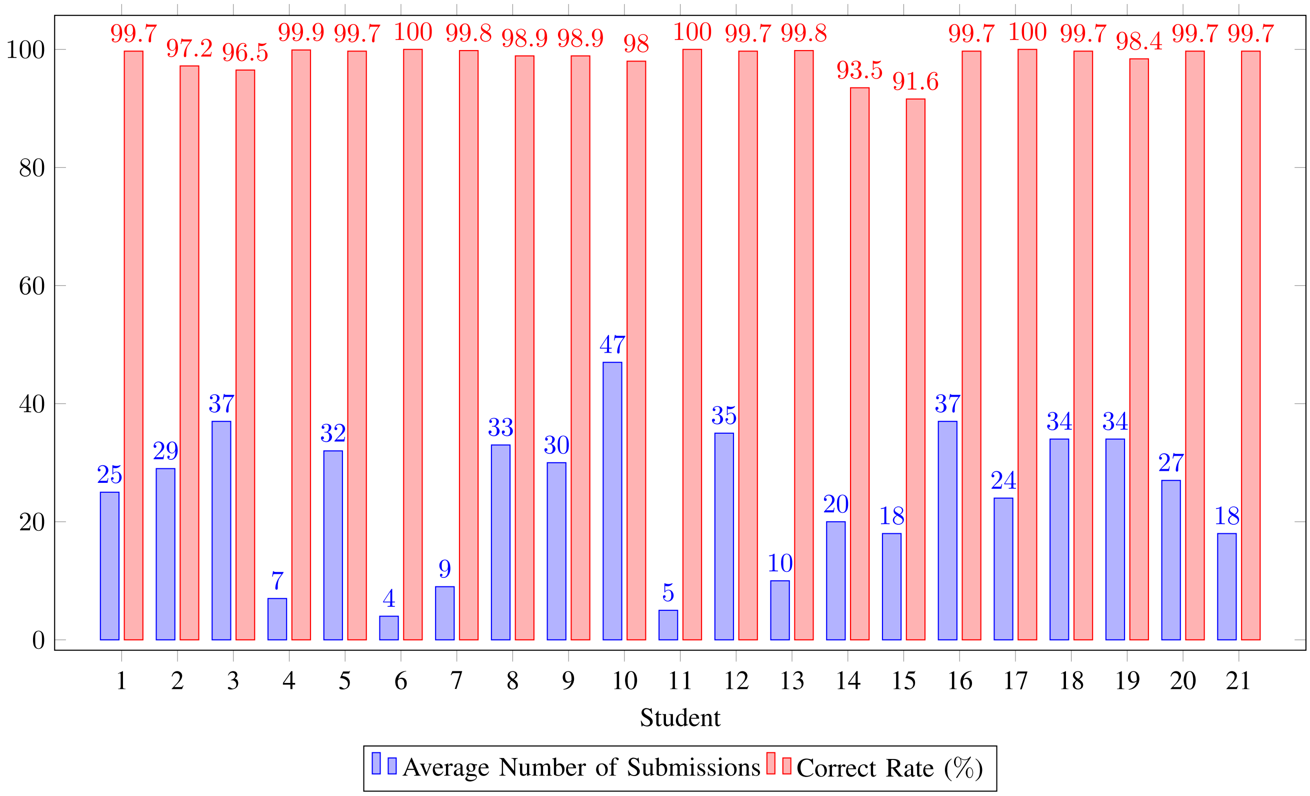


Fig. 6. CTP-Plus Result (Number of Submissions and Correct Rate)

end of the program line but has submitted it.

Most of the correctness of all students' answers to all questions on CTP-Plus has been appropriately implemented. Two questions can be solved perfectly up to 100% correct, and five questions do not reach the maximum percentage but can still be said to be very good because the correct answers are more than equal to 97%.

TABLE II
CTP-PLUS RESULT
(AVERAGE NUMBER OF SUBMISSIONS, NUMBER OF LINES, AND COMPARISON RATIO)

Problem	Average Number of Submissions	Number of Lines	Comparison Ratio
11	26	62	0,41
12	7	18	0,39
13	64	96	0,67
14	14	46	0,30
15	3	9	0,33
16	53	163	0,32
17	4	27	0,15

CTP-Plus has gone through an evaluation process and examined further the relationship between the number of lines for each question and the average number of submissions, which has been carried out further, as shown in "Table. II". The relationship between the two items is calculated using a comparison ratio. If the value is closer to zero, it shows that student learning outcomes are improving. A pretty good trend can be seen from the "Table. II". The longer students use CTP-Plus, their understanding of Java Programming increases. That fact can be seen from the comparison ratio of questions 11 to 17, which has a decreasing trend. Indeed, there are still comparison ratios that do not have a downward trend but instead increase. Still, the increase is also not too high, as in the comparison ratio of question 13. The number of submissions of most students in this number is not that high on average. It's just that it is affected by the number of submissions of one student, number 10, who submits many times, affecting the overall average. The number of program code lines for question no 16 is quite long, namely 163 program lines, but the ratio of the average number of submissions to the total number of lines is only 0.3. This ratio shows that students increasingly understand how to write Java program lines correctly and how the correct logic flows for the topics of discussion that are explained that week. The subject studied that week was Linked List material which had two face-to-face learning sessions.

There is some CTP impact on the student. The fewer lines of the program, the number of submissions will be; The number of submissions decreases as the number of meetings increases (the comparison of the number of program lines remains to be considered); The ability of students is increasing in every meeting.

CTP-Basic and CTP-Plus show the ratio value for comparing the number of submissions and the number of lines of program code questions that look different. CTP-Basic still

Authorized licensed use limited to: BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE. Downloaded on October 02,2023 at 14:19:35 UTC from IEEE Xplore. Restrictions apply.

has a ratio value far above one. In contrast, CTP-Plus, even though the questions provided are not only reiterating the program code but also look at program code mistakes, has a matching ratio value below one and has a trend of decreasing scores. The results show that the longer students use CTP, their understanding of Java Programming increases.

V. Conclusion

This paper presents the instance of CTP in JPLAS for basic programming learning. There are two types of CTP, namely a) CTP-Basic, which presents problems in the form of program code available in the data structure starting from the chapter on variations of the searching to variations of the sorting method. The complete program code of the questions on CTP provided to improve memory about the syntax of Java Language and understanding of the program's logic flow; b) CTP-Plus, which presents problems at a higher level, namely not writing the full program code of the questions, but also examining some errors in the program code images presented. CTP-Plus does not only focus on understanding program logic flow but also modifying incorrect program code. CTP, which was applied to 21 students who followed the structured data course, achieved a correct answer level of $99,1\%$ for CTP-Basic and $98,61\%$ for CTP-Plus. The results of the analysis of answers from the comparison ratio between the average number of submissions and a number of lines show that the average number of submissions decreases the longer it takes them to solve problems with CTP. This shows that CTP has achieved its goal, which is to improve the ability to understand and also the flow of logical thinking using the Java language.

In future work, the development of CTP can be added not only regarding retyping program code but adding variations of questions that are more challenging. In addition, it is better to add the limitation that you cannot submit before the program line has been completed or most of it has been completed so that it is not considered that the student has tried to type many times but failed.

Acknowledgment

We would like to express our deepest appreciation to 21 State Polytechnic of Malang students who worked on CTP in the data structure course.

References

[1] M. S. Khoirom, M. Sonia, B. Laikhuram, J. Laishram, and D. Singh, "Comparative Analysis of Python and Java for Beginners," International Research Journal of Engineering and Technology, 2020, [Online]. Available: www.irjet.net

[2] C.-Y. Tsai, "Improving students' understanding of basic programming concepts through visual programming language: The role of self-efficacy," Comput Human Behav, vol. 95, pp. 224–232, Jun. 2019, doi: 10.1016/j.chb.2018.11.038.

[3] R. Scherer, F. Siddiq, and B. Sánchez Viveros, "The cognitive benefits of learning computer programming: A meta-analysis of transfer effects.," J Educ Psychol, vol. 111, no. 5, pp. 764–792, Jul. 2019, doi: 10.1037/edu0000314.

[4] G. Al Farsi, R. M. Tawafak, S. I. Malik, and B. H. Khudayer, "Facilitation for Undergraduate College Students to Learn Java Language Using E-Learning Model," International Journal of Interactive Mobile Technologies, vol. 16, no. 8, pp. 4–17, 2022, doi: 10.3991/ijim.v16i08.28689.

[5] N. FUNABIKI, K. K. ZAW, N. ISHIHARA, and W.-C. KAO, "JAVA PROGRAMMING LEARNING ASSISTANT SYSTEM: JPLAS," in IAENG Transactions on Engineering Sciences, WORLD SCIENTIFIC, Jan. 2018, pp. 517–530. doi: 10.1142/9789813230774_0037.

[6] S. T. Aung et al., "A Proposal of Grammar-Concept Understanding Problem in Java Programming Learning Assistant System," Journal of Advances in Information Technology, vol. 12, no. 4, 2021, doi: 10.12720/jait.12.4.342-350.

[7] H. H. S. Kyaw, N. Funabiki, N. Ishihara, M. Kuribayashi, and W.-C. Kao, "Web-Server Implementation of Code Completion Problem for Java Programming Learning Assistant System," in 2019 IEEE International Conference on Consumer Electronics - Taiwan (ICCE-TW), IEEE, May 2019, pp. 1–2. doi: 10.1109/ICCE-TW46550.2019.8991871.

[8] H. H. S. Kyaw, N. Funabiki, and W.-C. Kao, "A Proposal of Code Amendment Problem in Java Programming Learning Assistant System," International Journal of Information and Education Technology, vol. 10, no. 10, pp. 751–756, 2020, doi: 10.18178/ijiet.2020.10.10.1453.

[9] N. Ishihara, N. Funabiki, Tana, and W.-C. Kao, "An extension of statement fill-in-blank problem in Java programming learning assistant system," in 2015 IEEE 4th Global Conference on Consumer Electronics (GCCE), IEEE, Oct. 2015, pp. 354–358.

[10] K. T. Mon, K. H. Wai, N. Funabiki, and H. H. S. Kyaw, "Value Trace Problems for Code Reading Study of JavaScript Programming," International Journal of Information and Education Technology, vol. 12, no. 5, pp. 430–435, 2022, doi: 10.18178/ijiet.2022.12.5.1637.

[11] N. Ishihara and N. Funabiki, "A Proposal of Statement Fill-in-Blank Problem in Java Programming Learning Assistant System," in 2015 IIAI 4th International Congress on Advanced Applied Informatics, IEEE, Jul. 2015, pp. 247–252. doi: 10.1109/IIAI-AAI.2015.192.

[12] X. Lu et al., "Test Code Improvements for Code Writing Problem in Java Programming Assistant System," in 2021 IEEE 10th Global Conference on Consumer Electronics (GCCE), IEEE, Oct. 2021, pp. 599–603. doi: 10.1109/GCCE53005.2021.9621949.

[13] N. Funabiki et al., "An Extension of Code Correction Problem for Java Programming Learning Assistant System," in Proceedings of the 2020 8th International Conference on Information and Education Technology, New York, NY, USA: ACM, Mar. 2020, pp. 110–115. doi: 10.1145/3395245.3396439.

[14] Kuo-En Chang, Bea-Chu Chiao, Sei-Wang Chen, and Rong-Shue Hsiao, "A programming learning system for beginners-a completion strategy approach," IEEE Transactions on Education, vol. 43, no. 2, pp. 211–220, May 2000, doi: 10.1109/13.848075.

[15] N. Ishihara, N. Funabiki, M. Kuribayashi, and W.-C. Kao, "A Software Architecture for Java Programming Learning Assistant System," International Journal of Computer and Software Engineering, vol. 2, no. 2, Jul. 2017, doi: 10.15344/2456-4451/2017/116.

[16] X. Lu, N. Funabiki, S. T. Aung, Y. Jing, and S. Yamaguchi, "An Implementation of Java Programming Learning Assistant System in University Course," in 2023 11th International Conference on Information and Education Technology (ICIET), IEEE, Mar. 2023, pp. 215–220. doi: 10.1109/ICIET56899.2023.10111422.

[17] S. L. Aung, N. K. Dim, S. M. M. Aye, N. Funabiki, and H. H. S. Kyaw, "Investigation of Value Trace Problem for C++ Programming Self-study of Novice Students," International Journal of Information and Education Technology, vol. 12, no. 7, pp. 631–636, 2022, doi: 10.18178/ijiet.2022.12.7.1663.

[18] N. Funabiki, Y. Wang, N. Ishihara, and W.-C. Kao, "An offline answering function for code writing problem in Java Programming Learning Assistant System," in 2017 IEEE International Conference on Consumer Electronics - Taiwan (ICCE-TW), IEEE, Jun. 2017, pp. 241–242. doi: 10.1109/ICCE-China.2017.7991085.

[19] K. H. Wai, N. Funabiki, S. T. Aung, K. T. Mon, H. H. S. Kyaw, and W.-C. Kao, "An Implementation of Answer Code Validation Program for Code Writing Problem in Java Programming Learning Assistant System," in 2023 11th International Conference on Information and Education Technology (ICIET), IEEE, Mar. 2023, pp. 193–198. doi: 10.1109/ICIET56899.2023.10111392.

Authorized licensed use limited to: BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE. Downloaded on October 02,2023 at 14:19:35 UTC from IEEE Xplore. Restrictions apply.