Computational Story: Learning Media for Algorithm and Programming Based on Computational Thinking

Lala Septem Riza, Beni Handoko, Yaya Wihardi, Herbert

Abstract**: *The subject of algorithm and programming is one of the important knowledge that need to be mastered by students in computer science. To learn it, students are required to have several skills, such as logic, critical thinking, problem solving, etc. Therefore, the purpose of this research is to design and build multimedia applications to support critical thinking, called computational stories, on studying basic programming with a problem-posing model approach by involving cases in everyday life. In this application, students are given some cases in the form of code with the C programming language. Then, students are asked to arrange a correct story line from the choices available. From this research experiment, the results show that this multimedia can improve critical thinking based on comparison of pre-test and post-test results with an average of 32.9 to 72.47. With an average gain value of 0.58 included in the medium category.***

Keywords**: *Algorithm, Programming, Computational Thinking, Critical Thinking, Learning Media.***

I. INTRODUCTION

In the development of the computer science curriculum today there is one fundamental scientific discipline that must be mastered, namely algorithms and programming. An algorithm is a sequence of logical steps in problem solving that are arranged systematically. For some students studying Basic Programming subjects is not something easy. This is because these subjects require students to think critically, logically, structurally, and systematically. To help students understand the concepts of algorithms and programming, several studies have been conducted with various teaching strategies. For example, the use of cyberblog based intelligent tutoring systems to improve capabilities in the field of algorithms [1]. Blended learning for the teaching of algorithms for undergraduate students is done by Papadakis [2]. Animation can also be used in teaching algorithms so students easily understand [3]. Other strategies on teaching algorithm and programming can be found in [4, 5, 6, 7], which is by developing many algorithms that can be applied in general cases.

Algorithm learning and programming are also closely related to concepts in Computational Thinking. This concept is the basis of computer science with intellectual abilities consisting of algorithmic thinking, pattern recognition, abstraction, and decomposition [8]. The teaching model with the concept of computational thinking is aligned with the needs of students' 21st century digital skill development. The application of a combination of Computational Thinking and media & information literacy can be found in [9], where this study offers 7 ideas, namely creativity, abstraction, data and information, algorithm, programming, internet, and global impact. Moreover, this study also illustrates one tool that has the ability of 7 aspects, namely scratch. This tool is made for students who are just learning programming through videos, stories, music, simulations, interactive art, or games [10]. Some other examples of research to improve students' abilities in algorithms and programming specifically by involving the concept of computational thinking can be found in [11, 12, 13].

This research aims to develop learning media by considering the concept of Computational Thinking to teach algorithm and programming. In this learning media, problem-based learning [14] is applied through coding translation strategies and story problems. In principle, students are given a problem which is a piece of coding in the C Programming Language and also given a choice of answers which are pieces of the story line. So students are asked to arrange story problems from the available choices so that the storyline accurately describes the case and the solution interpreted in the coding in the problem. With this strategy, students are required to think critically what cases or stories that fit the solution that is the code in the given problem.

Problem posing, which is the strategy used in this study, is a way for students to formulate a problem together when looking for a solution [15]. This posing problem is widely implemented in teaching in the fields of mathematics [16-20], biology [21] and computer science [22]. In the use of problem posing in programming, problem posing is illustrated as a programming paradigm that combines all types of programming. In this concept, it not only separates the proposed solution from the code implementation, but also stores the problem in code explicitly, so. A program consists of a collection of problems, not just a collection of solutions without problems.

Revised Manuscript Received on October 15, 2019.

Lala Septem Riza, Department of Computer Science Education, Universitas Pendidikan Indonesia, Bandung, Indonesia. Email: lala.s.riza@upi.edu.

Beni Handoko, Department of Computer Science Education, Universitas Pendidikan Indonesia, Bandung, Indonesia. Email: benihandoko@student.upi.ed

Yaya Wihardi, Department of Computer Science Education, Universitas Pendidikan Indonesia, Bandung, Indonesia. Email: yayawihardi@upi.edu

Herbert, Department of Computer Science Education, Universitas Pendidikan Indonesia, Bandung, Indonesia. Email: herbert@upi.edu

Retrieval Number: A9738109119/2019©BEIESP
DOI: 10.35940/ijeat.A9738.109119

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

II. THE MODEL OF LEARNING MEDIA BASED ON COMPUTATIONAL STORY

In designing the learning media model, we use the translational method of the problem in the form of coding which is a solution into a story line that is the answer to the question. In other words, in this translation model each answer consists of pieces of the story and the problem is a program code that has been provided in the media. In this model students are required to be able to understand programming problems by not trying to do coding in a computer then compiling the code. Students are required to think like a computer. The coding translation model for this story is based on components that we have determined beforehand, according to the topic of the case, material in programming, and the type of questions as shown in Figure 1. It can be seen that there are two aspects considered in building a question as follows: themes on programming (e.g., data types, array, variables, operators, logic, branch, filtering, and looping) and cases (e.g., calculating min, max, mean, sorting, etc.). On the answer side, we consider two types of QA (i.e., filling in the blank and choosing and arranging the story) and several levels of correctness, such fully correct answer, a bug answer, and wrong answer.

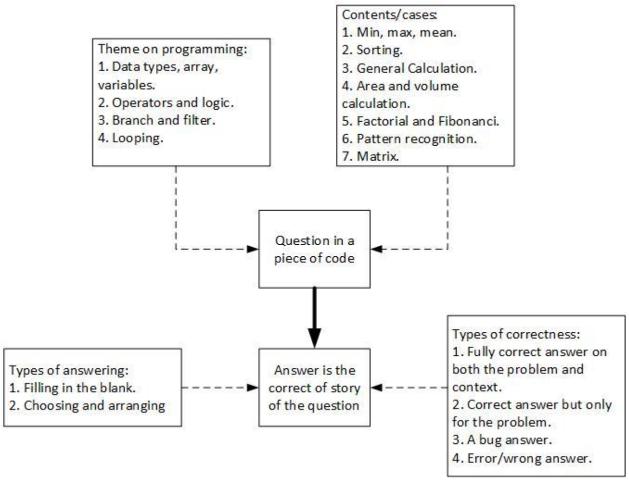

Theme on programming:
1. Data types, array, variables.
2. Operators and logic.
3. Branch and filter.
4. Looping.
Contents/cases:
1. Min, max, mean.
2. Sorting.
3. General Calculation.
4. Area and volume calculation.
5. Factorial and Fibonanci.
6. Pattern recognition.
7. Matrix.
Question in a piece of code
Types of answering:
1. Filling in the blank.
2. Choosing and arranging
Answer is the correct of story of the question
Types of correctness:
1. Fully correct answer on both the problem and context.
2. Correct answer but only for the problem.
3. A bug answer.
4. Error/wrong answer.

Fig 1. The model to construct question and answer with computational story

All aspects of Fig 1 are arranged in the user interface as in Fig 2. It can be seen that the question contains a short command and a piece of code in C while the answer part consisting of options and boxes of answers.

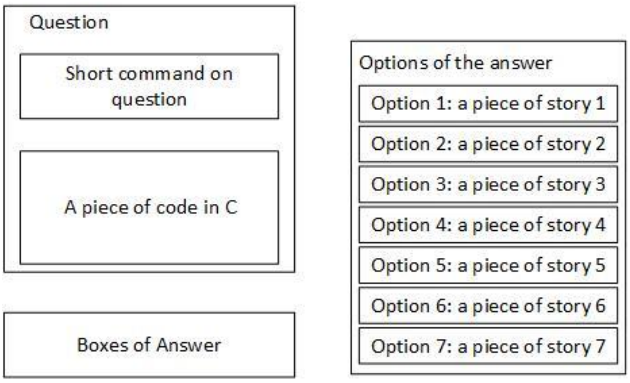
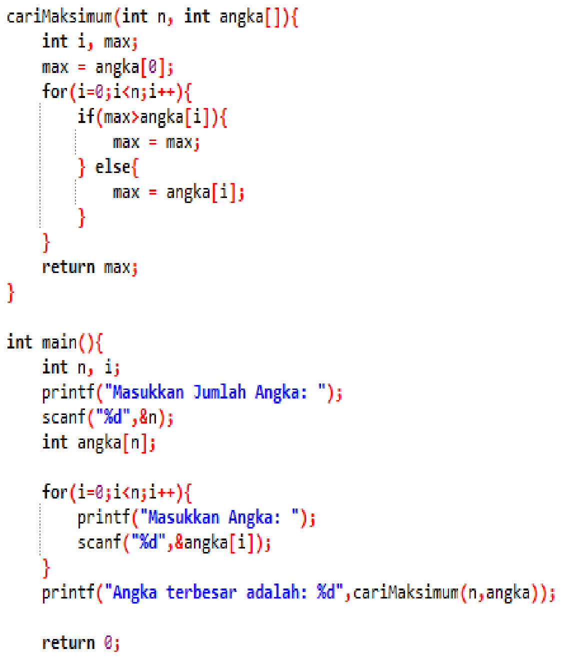

Question
Short command on question
A piece of code in C
Boxes of Answer
Options of the answer
Option 1: a piece of story 1
Option 2: a piece of story 2
Option 3: a piece of story 3
Option 4: a piece of story 4
Option 5: a piece of story 5
Option 6: a piece of story 6
Option 7: a piece of story 7

Fig 2. Mockup of graphical user interface

Then, in order to response a question, students need to have the following capabilities of critical thinking as follows: (i) recognize the problem; (ii) find ways that can be used to deal with problems; (iii) gather and compile information needed to solve problems; (iv) get to know assumptions and values that are not stated; (v) use appropriate, clear, and specific language in discussing a problem or a matter that is received; (vi) evaluate and assess facts, and statements; and (vii) look at the logical relationship between problems with the answers given.

Moreover, some aspects in computational thinking of students also are evaluated, such as capability to decompose the problem, recognize the pattern, and think algorithmic. To arrange this learning media content, several stages need to be carried out as follows: firstly, we need to construct a piece of code to be a question. The code should comply with the theme of programming and cases. Then, we need to arrange a complete story to be the correct answer. After that, we need to modify the story to be options of answers that provides several levels of correctness. In this case, accuracy of story should be taken into account. The following is an example of question and answer on the computational story:

Question: “Mr. Jono is a mathematics teacher. He wants to know who the highest grade in Class 2A is. This problem can be solved by the following code in Fig 3. Tell us how Mr. Jono solve his problem as indicated by the piece of code.



```
cariMaksimum(int n, int angka[]){
    int i, max;
    max = angka[0];
    for(i=0;i<n;i++){
        if(max>angka[i]){
            max = max;
        } else{
            max = angka[i];
        }
    }
    return max;
}

int main(){
    int n, i;
    printf("Masukkan Jumlah Angka: ");
    scanf("%d",&n);
    int angka[n];

    for(i=0;i<n;i++){
        printf("Masukkan Angka: ");
        scanf("%d",&angka[i]);
    }
    printf("Angka terbesar adalah: %d",cariMaksimum(n,angka));

    return 0;
```

Fig 3. The piece of code to solve the problem.

Options of answer:

a. Mr. Jono looked at his absence book to see the number of students in class X.

b. After that, Mr. Jono began to see the value of students in his class. Mr. Jono made value 1 as the reference value to become the first biggest value.

c. After that, Mr. Jono began to see the value of students in his class. The first value is made as the biggest value first by Mr. Jono.

d. After that, Mr. Jono began to see the value of students in his class. Mr. Jono took the first value as the smallest value

e. After that, Mr. Jono compared the first grade with the next grade on the list of scores repeatedly as many students as there were in the class. If the first value is greater than the next value, the largest value is still the first value and if the first value is smaller than the next value, the largest value is the next value.

f. After that Mr. Jono compared the value of 1 with the next value in the list of scores repeatedly as many as the number of students in the class. If the first value is greater than the next value, the largest value is still the first value and if the first value is smaller than the next value, the largest value is the next value.

g. After that, Mr. Jono compared the first grade with the next grade on the list of scores repeatedly as many students as there were in the class. If the first value is greater than the next value, the largest value is still the first value and if the first value is smaller than the next value, the smallest value is the next value.

The answer:

- Fully correct answer: a, c, e
- Correct but it is not suitable to the context: a, c, g
- A bug answer: a, b, f

III. EXPERIMENTAL DESIGN

There are several steps in the experimental design. The first stage starts with making questions that will be used in the experiment. The questions made include pretest questions, posttest questions, and media problems. Then after the stages of making questions are finished, the next step is to validate the questions by experts. Validated questions will be checked whether the questions are in accordance with the indicators made. Then after the problem is validated, the next step is to determine the population and sample that will be used. The population and sample used were students majoring in Software Engineering at The school of BPPI Baleendah, Bandung. The next step is to start the experiment by introducing research to be done on the selected sample. After students understand the purpose of this study, then a pretest is conducted to find out the student's initial knowledge. After the pretest, the next step is to experiment with learning media.

IV. RESULTS AND DISCUSSION

Retrieval of students' understanding data is by doing pre-test and post-test consisting of several multiple choice questions. Students will be grouped into upper, middle, and lower groups. The results of this class grouping can be based on pre-test scores. As for the explanation of the grouping as follows. Based on the pre-test scores it was found that:

- Average: 32.9
- Standard deviation: 7.29

From these data it can be concluded the group boundaries are:

- Upper group limit: 32.9 + 7.29 = 40.19
- Lower group limit: 32.9 - 7.29 = 25.61

Then the class division of 31 participants was as follows:

- The upper group, all students who had a pre-test score greater than 40.19 were 2 people.
- Middle group, all students who have pre-test scores with a range of 25.61 - 40.19 are 26 people.
- The lower group, all students who have a pre-test score below 25.61 with 3 people.

Moreover, we perform some analysis to obtain information regarding students' critical thinking. The following is some analysis on some students:

- Student 1: This student is able to understand the problem well. This is indicated by the user directly choosing the correct answer and in the context given. Based on existing indicators, students are able to recognize problems, find ways that can be used to deal with problems, gather and compile information needed to solve problems, recognize assumptions and values that are not stated, use the right story arrangement in discussing a problem or thing that is received, evaluates and evaluates facts, observes a logical relationship between the problems with the answers given, and draws conclusions or opinions about the issue or problem being discussed.
- Student 2: The student does not understand the question well, the answer chosen is still wrong. Based on existing indicators, students are able to recognize problems, find ways that can be used to deal with problems, gather and compile information needed to solve problems, recognize assumptions and values that are not stated, evaluate and assess facts, and attract conclusions or opinions about the issue or issue being discussed. However, students do not arrange their stories correctly so they answer incorrectly because students are not careful and cannot arrange their stories correctly.
- Student 3: The student is able to understand the problem well. This is indicated by the user directly choosing the correct answer and in the context given. Based on existing indicators, students are able to recognize problems, find ways that can be used to deal with problems, gather and compile information needed to solve problems, recognize assumptions and values that are not stated, use the right story arrangement in discussing a problem or thing that is received, evaluates and evaluates facts, observes a logical relationship between the problems with the answers given, and draws conclusions or opinions about the issue or problem being discussed.

Student 3: Students does not understand the problem well, the answer chosen is still wrong. Based on existing indicators, students are able to recognize problems, find ways that can be used to deal with problems, gather and compile information needed to solve problems, recognize assumptions and values that are not stated, evaluate and assess facts, and attract conclusions or opinions about the issue or issue being discussed. However, students do not arrange their stories correctly so they answer incorrectly because students are not careful and cannot arrange their stories correctly.

V. CONCLUSIONS

Based on the results of experimental discussions about computational stories in the learning media of algorithms and programming to support critical thinking, some conclusions can be drawn, including the following:

1. The learning media with the Computational Story model has been completed using the five stages of the waterfall software development method.
2. The development of learning multimedia is a follow-up study that has been done before. In the previous research, the model used in developing multimedia was a computational story model from story to coding, while this research was the opposite of previous research, namely the translational model from coding to story.

This computational Story learning media can also improve students' abilities that can be measured by increasing pre-test and post-test scores. The average pre-test score obtained by students was 32.9, and the average post-test score was 72.47. The increase has an average gain of 0.58 and falls into the medium category.

REFERENCES

1. L. S. Riza & B. L. Putro, "Development of cyberblog-based intelligent tutorial system to improve students learning ability algorithm," In Journal of Physics: Conference Series, Vol. 1013, No. 1, 2018, pp. 012157. IOP Publishing.
2. S. Papadakis, K. Paparrizos & E. Rossiou, "Using Blended Learning in Traditional Face-to-Face Instruction: A case study teaching Algorithms to undergraduate students," In E-Learn: World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education, 2006, pp. 839-845
3. M. D. Byrne, R. Catrambone & J. T. Stasko, "Evaluating animations as student aids in learning computer algorithms," Computers & education, 33(4), 1999, pp. 253-278.
4. L. S. Riza, D. Handian, R. Megasari, A. G. Abdullah, A. B. D. Nandiyanto, & S. Nazir, "Development of R package and experimental analysis on prediction of the CO2 compressibility factor using gradient descent," Journal of Engineering Science and Technology, 13(8), 2342-2351, 2018.
5. L. S. Riza, Iip, E. P. Nugroho, & Munir, "MetaheuristicOpt: An R Package for Optimisation Based on Meta-Heuristics Algorithms," Pertanika Journal of Science & Technology, 26(3), pp. 1401-1412, 2018.
6. L. S. Riza, D. S. Firdaus, E. Junaeti, T. Hidayat, A. G. Abdullah, A. B. D. Nandiyanto, & C. U. Abdullah, "A Concept And Implementation Of Instructional Interactive Multimedia For Deaf Students Based On Inquiry-Based Learning Model," Journal of Engineering Science and Technology, 13(7), 2016-2035, 2018.
7. L. S. Riza, I. F. Nasrulloh, E. Junaeti, R. Zain, & A. B. D. Nandiyanto, "gradDescentR: An R package implementing gradient descent and its variants for regression tasks," In 2016 1st International Conference on Information Technology, Information Systems and Electrical Engineering (ICITISEE) (pp. 125-129). IEEE. 2016.
8. S. Grover & R. Pea, "Computational thinking in K–12: A review of the state of the field," Educational researcher, 42(1), 2013, pp. 38-43.
9. S. Gretter & A. Yadav, "Computational thinking and media & information literacy: An integrated approach to teaching twenty-first century skills," TechTrends, 60(5), 2016, pp. 510-516.
10. M. Resnick, J. Maloney, A. Monroy-Hernández, N. Rusk, E. Eastmond, K. Brennan, … & Y. B. Kafai, "Scratch: Programming for all," Commun. Acm, 52(11), 2009, pp. 60-67.
11. W. Zhao & V. J. Shute, "Can playing a video game foster computational thinking skills?" Computers & Education, 141, 2019, pp. 103633.
12. P. João, D. Nuno, S. F. Fábio & P. Ana, "A Cross-analysis of Block-based and Visual Programming Apps with Computer Science Student-Teachers," Education Sciences, 9(3), 2019, pp. 181.
13. C. S. González-González, F. Herrera-González, L. Moreno-Ruiz, N. Reyes-Alonso, S. Hernández-Morales, M. D. Guzmán-Franco & A. Infante-Moro, "Computational Thinking and Down Syndrome: An Exploratory Study Using the KIBO Robot," In Informatics, Vol. 6, No. 2, 2019, p. 25
14. S. L. Brown & M. I. Walter, "The art of problem posing," Psychology Press, 2005.
15. E. A. Silver, J. Mamona-Downs, S. S. Leung & P. A. Kenney, "Posing Mathematical Problems: An Exploratory Study," Journal for Research in Mathematics Education, 27 (3), 293-309, 1996.
16. I. Lavy & A. Shriki, "Problem posing as a means for developing mathematical knowledge of prospective teachers," In Proceedings of the 31st Conference of the International Group for the Psychology of Mathematics Education (Vol. 3, pp. 129-136), 2007.
17. K. E. Chang, L. J. Wu, S. E. Weng & Y. T. Sung, "Embedding game-based problem-solving phase into problem-posing system for mathematics learning," Computers & Education, 58(2), 775-786, 2012.
18. T. Hirashima, S. Yamamoto & Y. Hayashi, "Triplet structure model of arithmetical word problems for learning by problem-posing," In International Conference on Human Interface and the Management of Information (pp. 42-50). 2014, Springer, Cham.
19. T. Hirashima & M. Kurayama, "Learning by problem-posing for reverse-thinking problems," In International Conference on Artificial Intelligence in Education (pp. 123-130). Springer, Berlin, Heidelberg, 2011.
20. T. Hirashima, T. Yokoyama, M. Okamoto & A. Takeuchi, "Learning by problem-posing as sentence-integration and experimental use," In AIED (Vol. 2007, pp. 254-261), 2007.
21. N. S. Peterson & J. R. Jungck, "Problem-posing, problem-solving and persuasion in biology education," Academic Computing, 2(6), 1988.
22. C. Landauer & K. L. Bellman, "Problem posing interpretation of programming languages," In Proceedings of the 32nd Annual Hawaii International Conference on Systems Sciences. 1999. HICSS-32. Abstracts and CD-ROM of Full Papers (pp. 10-pp). IEEE. 1999.

AUTHORS PROFILE

Lala Septem Riza He obtained PhD from Universidad de Granada in 2015. Now he is a lecturer in the department of computer science education and teaches the following subjects: machine learning, big data analysis, parallel and distributed computing. His research interest is on learning media for studying algorithm and programming, machine learning for educational evaluations, and data analysis in bioinformatics and astrophysics.

Beni Handoko He was graduated from Department of Computer Science Education. His research is about learning media for students on higher school.

Yaya Wihardi He obtained the master degree at Universitas Indonesia. Now, he is a lecturer at Department of Computer Science Education. His research interest is on education, computer vision, and artificial intelligence.

Herbert He obtained the master degree at Institut Teknologi Bandung. Now, he is a lecturer at Department of Computer Science Education and teaches some subjects related to algorithm and programming, such as web programming, object oriented programming, etc. His research interest is on education, information system, and game programming.