Advances in Engineering Research, volume 209
International Joint Conference on Science and Engineering 2021 (IJCSE 2021)

Scratch Programming Learning Model Through Team Based Project to Train Pre-Service Science Teachers' Computational Thinking Skill

Novi R. Dewi[1,*] Prasetyo Listiaji[1] Arka Yanitama[1] Risa D.Hardianti[1]

Isa Akhlis[2] Ismail O. Kurniawan[3]

[1]*Departement of Integrated Science, Universitas Negeri Semarang, Indonesia*
[2]*Departement of Physics, Universitas Negeri Semarang, Indonesia*
[3]*ICT Unit, Universitas Negeri Semarang, Indonesia*
[*]*Corresponding author. Email: noviratnadewi@mail.unnes.ac.id*

ABSTRACT
Computational thinking skills (CTS) are needed for pre-service science teachers as a provision in organizing 21st-century learning. This research focused on developing a Scratch programming learning model by means of a team-based project to train science teacher candidate CTS. Development was carried out using the ADDIE model. The resulting model products were lesson plans, textbooks, and CTS assessment tools. Experts' validation result of the entire product obtained valid criteria. The products were implemented in the course on Development of Media and Teaching Aids for Science Learning. At the implementation stage, the CTS of pre-service science teachers obtained results of 15.56 low, 64.44 moderate, and 20.00% high. Based on these results, the scratch programming learning model by means of a team-based project can train CTS for the pre-service science teachers. Scratch programming through the team-based project can be an offer to be applied to the education for pre-service teachers because of the ease of implementation of a visual block-based programming language.

Keywords: *Scratch programming, Learning model, Team-based project, Computational thinking skill, Pre-service science teacher.*

1. INTRODUCTION

Computational thinking skills (CTS) are currently needed by everyone in the 21st-century era because they are considered to make it easier to solve problems in today's world [1, 2]. CTS is a set of problem-solving skills based on computer techniques that are needed for almost all careers [3]. Studies on CTS have been conducted previously by many researchers, but more focused on students [4-6]. Research on CTS in pre-service, especially science teachers, has not been done much. Most of the pre-service science teachers have never acquired computational thinking skills during college [7]. Even though it is very useful to pre-service science teachers to improve the teaching and learning process in the era of the industrial revolution 4.0 [7].

One way to train CTS is to introduce a visual-based programming language because text and logo-based programming takes a long time and is not very successful [8]. Scratch is considered a powerful tool for embedding computational thinking in education because it is a visual and media-rich programming that enables users to create games and stories [8, 9]. Scratch puts users into the world of computing without having much experience in programming languages. On the other hand, visual programming learning must be combined with the right model. One of the appropriate models to use is the team-based project because it has been widely recognized as a useful strategy for improving all levels of education, including higher education [10]. With this model, prospective science teacher students will be able to create computational projects based on the visual scratch programming in groups.

Referring to the importance of training computational thinking skills for pre-service science teachers, and the advantages of learning the visual scratch programming combined with the team-base project learning model, it is necessary to conduct the research on the development of visual scratch programming learning models by means of team-based projects to train computational thinking skills (CTS) of the pre-service science teachers.

2. METHODS

This research is a Research and Development (R & D) model using 5 stages, namely analysis, design, development, implementation, and implementation (ADDIE) [11]. The schematic stages can be seen in Figure 1. The product developed is in the form of a scratch programming learning model that uses a team-based project. The product development carried out consists of semester lesson plans, textbooks, and computational thinking assessment tools.

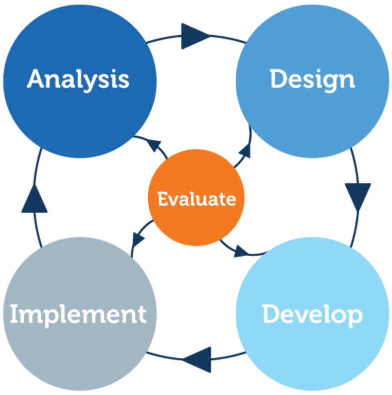


Figure 1 ADDIE model.

2.1. *Analysis*

The activities carried out at this stage include an analysis of the appropriate courses in science education, students' initial knowledge of scratch, the availability of student hardware and software, the availability of student internet networks, analysis of the number of meetings, and analysis of programming material to be provided.

2.2. *Design*

The activity carried out at this design stage is the design of the scratch programming learning model by means of a team-based project whose output is in the form of lesson plans and textbooks. In addition, a draft assessment tool was also designed to measure the computational thinking skills of the pre-service science teachers.

2.3. *Development*

After going through the design stage, the next stage is product development. The finished product is validated by 5 experts so that a product quality score is obtained. The experts give a rating of 1 to 4 on each item quality indicator of the product. The experts also provide suggestions if product revisions are needed. Product quality is determined by score according to Aiken's V formula (equation 1) and the criteria in Table 1 [12].

$$V = \frac{\Sigma s}{[n(c-1)]}$$

V = validation score,
s = $r - lo$,
r = score form expert,
lo = minimum score,
n = amount of expert,
c = maximum score

Table 1. Criteria of validation by expert validation [12]

V score	Criteria
< 0.87	Not valid
> 0.87	Valid

2.4. *Implementation*

The learning model product that has been validated is then tested for use. The usage test was carried out in the Development of Media and Teaching Aids for Science Learning in Science Learning course by involving 45 students of the Science Education Study Program, Universitas Negeri Semarang as the pre-service teachers. After using a scratch programming learning model using a team-based project, then measuring the computational thinking skills (CTS) students of the pre-service science teachers. CTS pre-service science teachers are categorized into low, medium, and high categories according to Table 2.

Table 2. CTS categorizes of pre-service science teacher [13]

Categorize	Formula
Low	$S < M - SD$
Medium	$M - SD < S < M + SD$
High	$M + SD \leq S$

S is the score of CTS, M is the mean score, and SD is the standard deviation.

2.5. *Evaluation*

Evaluation is carried out at all stages so that suggestions are obtained for product improvements that have been developed.

3. RESULTS AND DISCUSSION

3.1. *Products*

The product of the *Scratch* programming learning model that uses team-based projects is contained in the semester lesson plan for the Development of Media and Teaching Aids for Science Learning course. Scratch programming is taught to pre-service science teachers in 4 meetings containing project-based learning syntax. The syntax result of the learning model developed is shown in Figure 2.

In addition to the lesson plan, a textbook on the Development of Media and Teaching Aids for Science Learning course was also developed which contains material on *Scratch* programming and computational thinking skills (CTS). *Scratch* programming material is written in 2 chapters consisting of materials: (1) introduction, (3) *Scratch* installation, (4) *Scratch* work, (5) *Scratch* tutorial, (6) digital storytelling, (7) games, (8) simulation, and (9) project exercises. The CTS material is presented in 1 chapter with details of materials: (1) introduction. (2) CTS elements, (3), *Scratch* for CTS development, and (4) Exercise. The cover of the book text and examples of the scratch project can be seen in Figures 3 and 4.

The CTS assessment tool was developed according to aspects of the CTS including decomposition, pattern recognition, abstraction, algorithmic thinking, and generalization. An example of a CTS evaluation tool is shown in Figure 5.

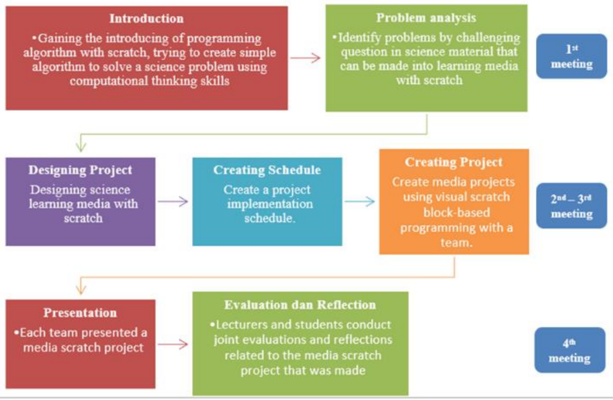
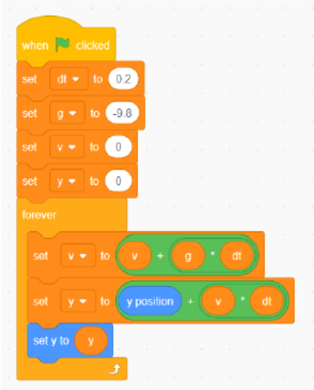
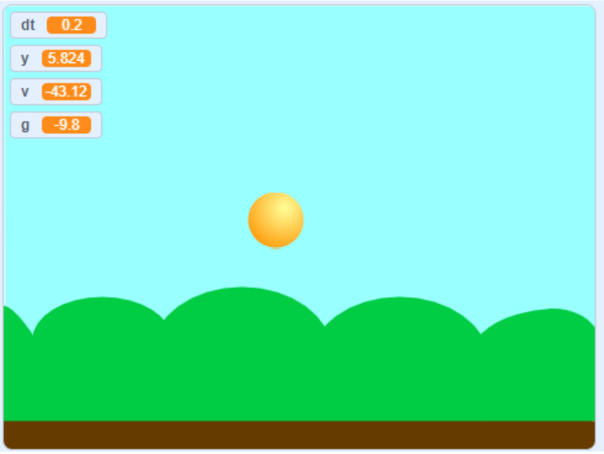


Figure 2 Syntax of Scratch programming learning model that uses mean of team-based project.

Figure 3 Textbook cover.



(a) visual block code



(b) interface

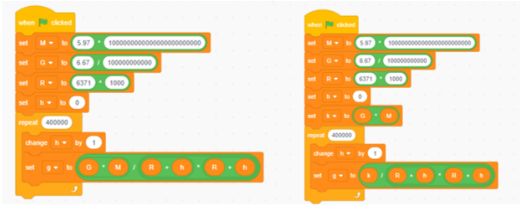
Figure 4 Example of Scratch project (free fall motion), (a) visual block code, (b) interface.

A science teacher wants to make a simulation of changes in the value of the earth's gravitational acceleration depending on the height of the point from the earth's surface by using a block scratch code. The equation used is...

$$g = G\frac{M}{(R+h)^2}$$

where G = general constant of gravity, M = mass of the earth, R = radius of the earth, h = distance of the point from the earth's surface, g = earth's gravitational acceleration

There are two proposed block code flows,



If analyzed in more detail, which code block is more efficient? Why?

Figure 5 Question example of CTS assessment for algorithm thinking aspect.

3.2. *Products validation*

The results of the validation of the learning model in the form of lesson plan products, textbooks, and CTS assessment tools are shown in Figure 6. All products have a V score or Aiken index of more than 0.87 so that it can be said to be valid according to experts. The obtained

Aiken index can be used as a basis for measuring the content validity of a product [14].

If we analyse in more detail each item in all items in all products obtains valid criteria. Based on the results of this validation, all products that are part of the Scratch programming learning model that uses team-based project are valid to be used at the implementation stage. However, from the experts there were several improvements, namely typing improvements in all products, adding references to lesson plans, improving image sizes in textbooks, and improving sentences in the CTS assessment tool.

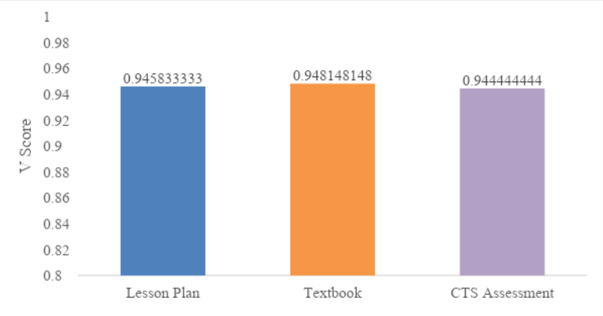


Figure 6 Validation results from experts.

3.3. Computational thinking skill (CTS) of pre-service science teacher

After the model is judged valid by the expert, then all products are implemented in lectures. Then measure the CTS of pre-service science teachers using an assessment tool that has been developed. The results of the CTS of pre-service science teachers after receiving Scratch programming lessons are shown in Figure 7. The median score obtained was 43.28 with a standard deviation of 14.41.

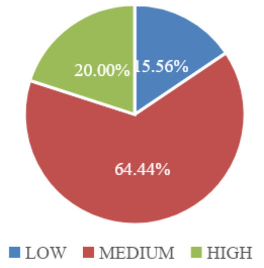


Figure 7 CTS result of pre-service teacher.

Based on these results, the scratch programming learning model by mean of a team-based project can train CTS for pre-service science teacher. This is supported by data that only 15.56 % of pre-service science teacher sample get CTS in the low category. Scratch programming can train CTS well because visual block-based programming makes it easier for students to understand the flow of problem solving [8]. On the other hand, team-based projects can also train CTS which has proven to be widely used in the field of computer science [15].

4. CONCLUSION

Scratch programming learning model by mean of team-based project to train CTS science teacher candidates has been developed. The resulting model products were lesson plans, textbooks, and CTS assessment tools. Experts' validation result of the entire product obtained valid criteria. The products were implemented in the course on Development of Media and Teaching Aids for Science Learning. At the implementation stage, the CTS of pre-service science teachers obtained results of 15.56 % low, 64.44% moderate, and 20.00% high. This research can be developed to test the effectiveness of learning Scratch programming through a team-based project to improve the CTS of pre-service science teachers.

AUTHORS' CONTRIBUTIONS

All authors conceived and designed this study. All authors contributed to the process of revising the manuscript, and at the end all authors have approved the final version of this manuscript.

REFERENCES

[1] Y. Tabesh, Computational thinking: A 21st century skill. Olympiads in Informatics, 11(2), (2017) 65-70, URL: https://ioinformatics.org/oi/pdf/v11si_2017_65_70.pdf

[2] V. Barr, C. Stephenson, Bringing computational thinking to K-12: what is Involved and what is the role of the computer science education community?. Acm Inroads, 2(1), (2011) 48-54, DOI: https://doi.org/10.1145/1929887.1929905

[3] J. A. Q. Figueiredo, How to improve computational thinking: A case study. Education in the Knowledge Society, 18(4), (2017) 35-51, URL: https://www.redalyc.org/pdf/5355/535554768003.pdf

[4] V. Barr, C. Stephenson, Bringing computational thinking to K-12: what is Involved and what is the role of the computer science education community?. Acm Inroads, 2(1), (2011) 48-54, DOI: https://doi.org/10.1145/1929887.1929905

[5] S. Grover, R. Pea, Computational thinking in K–12: A review of the state of the field, Educational researcher, 42(1), (2013) 38-43, DOI: https://doi.org/10.3102%2F0013189X12463051

[6] A. Yadav, N. Zhou, C. Mayfield, S. Hambrusch, J.T. Korb, Introducing computational thinking in

education courses. In Proceedings of the 42nd ACM technical symposium on Computer science education, 2011, pp. 465-470.

[7] T. Rahayu, K. Osman, Knowledge level and self-confidence on the computational thinking skills among science teacher candidates. Jurna Ilm. Pendidik. Fis. Al-Biruni, 8(1), (2019) 117-126, DOI: https://doi.org/10.24042/jipfalbiruni.v8i1.4450

[8] R. Vinayakumar, K.P. Soman, P. Menon, DB-learn: studying relational algebra concepts by snapping blocks. In 2018 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT) 2018, pp. 1-6. URL: https://ieeexplore.ieee.org/abstract/document/8494181/

[9] K. Brennan, M. Resnick, New frameworks for studying and assessing the development of computational thinking. In Proceedings of the 2012 annual meeting of the American educational research association, Vancouver, Canada , 2012. Vol. 1, pp 25.

[10] F. Wiegant, J. Boonstra, A. Peeters, K. Scager, Team-based learning in honors science education: the benefit of complex writing assignments, (2012), URL: https://digitalcommons.unl.edu/nchcjournal/364/

[11] M. Molenda, In search of the elusive ADDIE model, Performance improvement, 42(5), (2003) 34-37, URL: http://www.damiantgordon.com/Courses/DT580/In-Search-of-Elusive-ADDIE.pdf

[12] S. Arikunto, Metode Penelitian, Rineka Cipta, 2010.

[13] T. Winarsunu, Statistik dalam Penelitian Psikologi dan Pendidikan Volume 1, UMM Press, 2017.

[14] H. Hendryadi, Validitas isi: tahap awal pengembangan kuesioner. Jurnal Riset Manajemen dan Bisnis, 2(2), (2017) 259334, DOI: https://dx.doi.org/10.36226/jrmb.v2i2.47

[15] Y. Dong, V. Catete, R. Jocius, N. Lytle, T. Barnes, J. Albert ..., A. Andrews, PRADA: A practical model for integrating computational thinking in K-12 education. In Proceedings of the 50th ACM Technical Symposium on Computer Science Education, 2019, pp. 906-912, URL: https://dl.acm.org/doi/pdf/10.1145/3287324.3287431