The Development of CT-S Learning Module on The Linear Motion Topic to Promote Computational Thinking Thinking

Isra Khasyyatillah[*], Kamisah Osman

Faculty of Education, The National University of Malaysia, Bangi Selangor, Malaysia

ARTICLE INFO

A B S T R A C T

Computational Thinking (CT) is the main skill of the 21st century that is increasingly attracting more researchers to study how to implement CT in the learning and teaching process. Among the CT tools that can be used to develop CT is programming. Currently, availability and easily accessible programming tools have led researchers and educators to explore how to introduce CT in the context of learning and teaching in schools. Recognizing the importance of implementing CT in the classroom, this study aims to develop the CT-S (Computational Thinking and Scratch) module for the Linear Motion topic. The type of this study is research and development research to develop modules based on the ADDIE model to produce the CT-S module with validity and reliability. Data were analyzed using descriptive statistical analysis. The result showed that the CT-S module was valid. It is eligible to be used as the instructional material of Physics. This study implies that computational thinking skills can be integrated with other subjects besides computer science like physics. Therefore, teachers can design lessons that are relevant to the context and students' characteristics.

1. Introduction

The learning process needs constant innovation to align the teaching and learning goals with the modern life faced by students. However, the fact shows an irrelevant situation between school and the real world. As in the workplace, the majority of workers agreed (59%) that the skills taught in schools are not following the needs of future requirements. Most career skills are acquired and developed beyond the

school (The Pearson Foundation et al., 2013). Therefore, the issue is that the existing workforce skills do not fit the skills required in the workplace. Computational thinking is the skill that must be taught to the students early for career readiness and can effectively participate in the digital world (Csizmadia et al., 2015).

Computational thinking is a process of thought involved in formulating and solving a problem in such a way that the solution can be done by a human or machine or both (Wing, 2010). Studies on computational thinking have become increasingly popular among researchers including the importance of computational thinking in K-12 education (Cetin, 2016; Bocconi et al., 2016). These studies were driven by many factors including technological and economic demands for the future workforce that have the necessary computer skills (Chen et al., 2017).

The impact of technological advancement globally not only affects the economic and career aspects but also affects the role of teachers in teaching the skills needed by students (Gretter et al., 2016). The role of the teacher is very important to bridge the gap between current world requirement and what needs to be taught to students (Partnership for 21st Century Skills, 2005; 2014). The importance of applying and integrating computational thinking skills should be realized and implemented by teachers in the classroom. However, computational thinking skills have not been incorporated as the skills that must be applied in the teaching and learning of the curriculum 2013 in Indonesia (Kemendikbud, 2014). Furthermore, there are still limited researchers focusing on computational thinking intervention studies in the context of K-12. Such limited studies were mostly carried out after school activities (Lye et al., 2014). Hence, there is a real gap in the research field on the implementation of computational thinking among students.

Dwyer et al. (2014) found that using computational thinking and programming as a method of physics learning can help students to master the learning. Additionally, exposing students to computational-based learning can help students in mastering science content such as Kinematics and Ecology and computational thinking effectively (Basu et al., 2014). One of the physics topics that became the basis for mastering more advanced topics was the concept of kinematic (Dam-o et al., 2018). The linear motion topic is the starting point for familiarizing students with scientific thinking (Trudel et al., 2011). Based on need analysis, the linear motion topic is a topic that students choose and compatible with programming.

Computational thinking does not necessarily relate to computing and programming but focusing on thought processes. Programming is one of the tools to cultivate such skills (Wing, 2010). One way to introduce and apply computational thinking to students is to use tools, such as Scratch (Cetin, 2016; Shute et al., 2017). Scratch is a programming language application that can be used offline and is downloaded for free (Lawanto et al., 2017). Additionally, the programming class based on computational thinking like Scratch can cultivate elements such as excellence, self-esteem, interest, purpose, motivation, achievement and knowledge applications compared to conventional classes based on information technology and communication skills (Jeon et al., 2017). Therefore, based on the above issues, this

study takes part in developing a learning module based on computational thinking for students in Linear Motion topic. The objective of this study is to develop the Computational Thinking and Scratch (CT-S) module for the Linear Motion topic.

2.   Methodology

This study used research and development (R&D) research design that aimed to develop a module. Most researchers agree that studies focused on the development of educational products are categorized as a research and development research design (Gall et al., 2003; Postlethwaite, 2005; UNESCO, 2014; OECD, 2015).

R&D is the design of studies that fill the gap between research and educational practices (Gall et al., 2003). The gap in this study is the issue of incompatibility of teaching materials with the skills needed by the students. Therefore, it is necessary to develop an integrated computational thinking-based module with Physics subjects for high school students.

The development of this module was based on Analysis, Design, Development, Implementation, and Evaluation (ADDIE) model. The ADDIE model consists of five phases ie, analysis phase, design phase, development phase, implementation phase and evaluation phase.

Analysis

At this stage, the researcher needs to identify the research gap and the importance of the study. The researcher also determines the learning goals, identifying students, determining the resources needed, including cost estimates and project management plans (Allen, 2017; Branch, 2009). The following Table 1 shows the process in the analysis phase.

Tabel 1. The process in the analysis phase

Analysis phase	Item analyzed	Method
Study gap and learning constraints	Study gap from previous studies and school conditions Based on the 2013 curriculum (syllabus and textbooks)	Readings and questionnaires
Learning goal		Readings
students profile and context	Demographics Capabilities Existing experience Interest and motivation Preference	Readings and questionnaires
Required source Instructional tool	Adapted to learning goals	
Project timeframe		

Design

In this study, the ADDIE model refers to Branch (2009), The main process in the design phase is to carry out task inventory and to set learning objectives. Task inventory is a process of identifying the essential tasks required to achieve learning goals. task inventory helps to organize content logically and orderly so students can build the knowledge and skills needed to achieve learning goals. In the CT-S module, learning objectives were based on Core Competencies (KI) and Basic Competencies (KD) that had been set in the 2013 Curriculum (K13). The learning objective included 3 domains such as cognitive, affective and psychomotor.

Development

The purpose of the development phase is to generate and validate the learning module. The outcome of this phase is a valid module. The process involves generating content, selecting or developing media, developing guidance for students and teachers, formative evaluation, and pilot studies (Branch, 2009).

In this third phase, the researcher developed the CT-S module for students and guidance for teachers. The focusing topic was the linear motion topic. This topic was divided into several sub-topics comprising activities that fit the learning objectives. The experts involved in the validation of the module were three lecturers and two experienced teachers. The validity instrument of the module is an evaluation questionnaire used to assess the validity of the module. The validity instrument of the module used is a validated instrument developed by the Badan Nasional Standar Pendidikan (BNSP) Indonesia in 2014. The validity instrument of the module covers four components such as content qualification, presentation, linguistics and visualization.

3.   Results and Discussion

Analysis

The need analysis showed that the students have never known about computational thinking. Students did not know about programming and had never been involved in programming activities. The selected topic as the appropriate topic was the linear motion topic. In this study, the CT-S module was developed for 10th-grade students (age 15-16 years). Learning goals were identified based on the 2013 curriculum related to the linear motion topic. The project timeline is for 6 months. Additionally, the resources needed in the module development process such as a computer, programming application, textbook, and the syllabus of the curriculum 2013.

Design

In the design phase, the researcher made a task inventory and determined the learning objectives based on the syllabus of the 2013 curriculum. Table 2 shows an example of a task inventory.
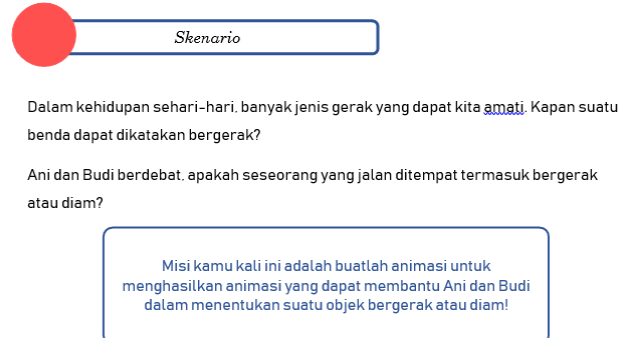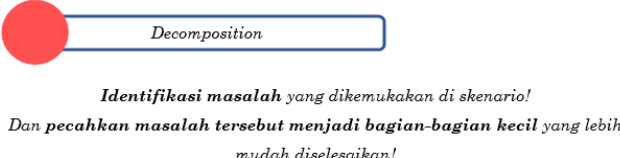
Table 2. The example of task inventory

General Objective	Activities					Note
	Step1 Engagement	Step2 Decomposition	Step3 Pattern Recognition	Step4 Abstraction	Step5 Algorithm	
Students can design, perform and present data and graphs of linear motion experiments.	Identify the problem stated in the scenario.	Solve the problem into smaller parts.	Identify the problem pattern by looking at its similarities with other problems that have been solved.	Select and sort out important and unimportant information in solving the problem.	Compile steps to solve problems. Students compile procedures of the experiment and create graphs using Scratch	Media: skydiving video and worksheet Methods: Discussionand experiment Assessement: Dr. Scratch

Development

In the development phase, the researcher generated the content of CT-S module. The teaching strategy needs in the development phase to organize the construction of knowledge, skills, and procedures during the teaching and learning. The teaching strategy is the method of organizing and arranging the sequence of learning activities (Branch, 2009). This CT-S module was designed to impact the student's computational thinking. The learning strategy used in this study was the 5E instructional model as a guide for writing lesson plans and teaching in the CT-S module. Table 3 shows an example of activities in the CT-S module.
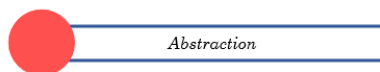
Table 3. Activities in the CT-S module

Phase of The 5E Instructional model	Learning Activities	Example
Engagement	Scenario (identifying problem)	*Skenario* Dalam kehidupan sehari-hari, banyak jenis gerak yang dapat kita amati. Kapan suatu benda dapat dikatakan bergerak? Ani dan Budi berdebat, apakah seseorang yang jalan ditempat termasuk bergerak atau diam? Misi kamu kali ini adalah buatlah animasi untuk menghasilkan animasi yang dapat membantu Ani dan Budi dalam menentukan suatu objek bergerak atau diam!
Exploration	Explore 4 elements of computational thinking to solve the problem Step 1: Decomposition	*Decomposition* *Identifikasi masalah yang dikemukakan di skenario!* *Dan pecahkan masalah tersebut menjadi bagian-bagian kecil yang lebih mudah diselesaikan!*

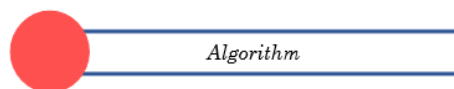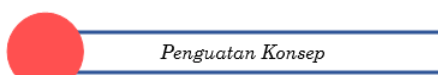|		
	Step 2: Pattern Recognition	*Pattern Recognition* *Mengenal pola atau karakteristik merupakan kunci dalam menentukan solusi yang tepat dalam menyelesaikan masalah dan mengetahui cara menyelesaikan masalah jenis tertentu.*
	Step 3: Abstraction Peniskalaan	*Abstraction* *Mengidentifikasi persamaan dan perbedaan spesifik antara masalah yang sama.* *Contoh 1c:* Fokus pada apa yang kamu perlukan dan abaikan yang tidak penting. | Slide presentasi yang Nampak rumit memiliki pola yang sederhana. **Slide presentasi**
	Step 4 Algorithm	*Algorithm* *Algorithm adalah rancangan, instruksi langkah demi langkah un menyelesaikan masalah.*
	Create animation using Scratch	*Langkah-Langkah pembuatan animasi Scratch* 1. Masukkan 3 sprite kucing yang akan menjadi objek.
Explanation	Present dan explain the solution	Students present findings of the exploration process and exchange ideas to other groups.
Elaboration	Strengthening concepts (additional activities such as challenging questions and problems)	*Penguatan Konsep* Yanka dijemput bus sekolah didepan rumahnya. Di dalam bus, Yanka duduk diku bagian kiri dan melambaikan tangan kepada ibunya. 1. Bus (diam/bergerak) terhadap rumah. 2. Yanka (diam/bergerak) terhadap bus. 3. Ibu (diam/bergerak) terhadap bus. 4. Bus (diam/bergerak) terhadap Yanka. Jelaskan jawabanmu untuk setiap jawaban di atas!
Evaluation	Conclusion and assessment	Encourage students to assess their understanding and ability. Provide teachers to evaluate student progress in achieving learning objectives.

Guidance for the teacher was developed as a guide in carrying out a learning process based on computational thinking. Guidance for the teacher contains the instructional steps and the recommended resources that can be used to help make

learning based on computational thinking. Figure 1 shows teacher guidance for using the CT-S module. computational thinking.

Figure 1. The content of guidance for the teacher

The supporting media needed in the learning and teaching process are a laptop, powerpoint slides, videos, and class notes. Additionally, students need to install Scratch programming applications and Adobe air apps. In this phase, a formative assessment was also conducted to identify the validity of the CT-S module. Development of the CT-S module based on the ADDIE model had produced the CT-S module that had good validity (r = 0.775) in Table 4.

Table 4. Result of the validity assessment of the module

Component	Assessment of expert					Total Persentage (%)
	1	**2**	**3**	**4**	**5**	
Content	88.2	88.2	80.0	92.9	95.3	88.9
Presentation	95.0	81.0	88.0	91.0	99.0	90.8
Linguistics	90.0	85.0	85.0	91.7	100.0	90.3
Visualization	90.0	82.0	90.0	88.0	98.0	89.6
Total Persentage (%)	90.8	84.05	85.75	90.9	98.075	89.9

Based on Table 4, the percentage of module validity from each expert is more than 70%, in the range of 84.1% - 98.0%, means the CT-S module has a validity coefficient of above 0.80. This shows that the CT-S module has good validity. The validity of the module is important to ensure that the module can help the student master the learning objectives (Sidek et al., 2005).

Unlike some previous studies, this study combined both computational thinking and content at once to help students in developing computational skills and mastering learning content. Some last empirical studies focused on computational thinking in the context of science or STEM but did not explicitly teach science content (Saez-Lopez et al., 2016; Hershkovitz et al., 2019). Some studies used programming to teach science content but did not explicitly teach the skills or concepts of computational thinking (Lopez et al., 2015; Mariano et al., 2019). In conclusion, there are still limited efforts in integrating computational and science thinking (Peel et al., 2019).

The CT-S module was developed for encouraging students to use technology. The study of Vallance et al. (2016) showed how technology could transform teaching practices and influenced the way students to learn and think. The findings showed that students have not only been able to identify the problem areas, but also to produce solutions using technology and in collaboration with students. The approach in this study involved computational thinking and heutagogy that included contextually relevant content produced by students, real collaboration, environmental-based learning, flexible curriculum; and flexible and negotiated assessments (Vallance et al., 2016).

The CT-S module promotes students to produce artefacts using visual programming as a computational thinking tool. The artefacts produced by students are animations related to the concept of linear motion topic. Making computational artefacts (using or relating to computers) aims to involve students in the experience of linking mathematical and scientific ideas in creating automation. Furthermore, making artefacts is one of how students can engage in collaborative inquiry. Making computational artefacts provide students to combine ideas into an integrated process, compile understanding in new ways, and debugging if the instructions are made to produce something unexpected. An important component in constructionism theory is the artefact that has been made students must be presented to the public so that students feel ownership of the construction, learn from one another, and receive criticism (Wilkerson-Jerde, 2014).

4.   Conclusion

This study developed the CT-S module with validity and reliability through the development process according to the ADDIE model. Each of these development processes can be a reflection of the teacher on how to develop modules based on computational thinking. Teachers can adapt this development process to the context and characteristics of students in their classroom. However, the topic described in this study is the linear motion topic only. Hence, future studies can be carried out on other subjects and topics.

References

Allen, M. (2017). Designing Online Asynchronous Information Literacy Instruction Using the ADDIE Model. In T. Maddison & M. Kumaran (Eds.), *Distributed Learning: Pedagogy and Technology in Online Information Literacy Instruction* (pp. 69–91). Cambridge: Elsevier Ltd. https://doi.org/10.1016/B978-0-08-100598-9.00004-0

Basu, S., Kinnebrew, J. S., & Biswas, G. (2014). Assessing Student Performance in A Computational-Thinking Based Science Learning Environment. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 476–481. https://doi.org/10.1007/978-3-319-07221-0_59

Bocconi, S., Chioccariello, A., Dettori, G., Ferrari, A., & Engelhardt, K. (2016). *Developing Computational Thinking in Compulsory Education – Implications for Policy and Practice*. EUR 28295 EN. https://doi.org/10.2791/792158

Branch, R. M. (2009). *Instructional Design: The ADDIE Approach*. Boston, MA: Springer US.

Cetin, I. (2016). Preservice Teachers ' Introduction to Computing : Exploring Utilization of Scratch. *Journal of Educational Computing*, 1–25. https://doi.org/10.1177/0735633116642774

Chen, G., Shen, J., Barth-Cohen, L., Jiang, S., Huang, X., & Eltoukhy, M. (2017). Assessing Elementary Students' Computational Thinking in Everyday Reasoning and Robotics Programming. *Computers and Education*, *109*, 162–175. https://doi.org/10.1016/j.compedu.2017.03.001

Csizmadia, A., Curzon, P., Dorling, M., Humphreys, S., Ng, T., Selby, C., & Woollard, J. (2015). *Computational Thinking: A Guide for Teachers*. London, UK: Computing at School. Retrieved from https://community.computingatschool.org.uk/resources/2324/single

Dam-o, P., Gondek, J., Karbowiak, M., & Wibig, T. (2018). Observation of the Effect of Gender on Children ' s Concept of Motion ; Sustainability Issue. *Sustainability*, *10*(9), 3076-. https://doi.org/10.3390/su10093076

Dwyer, H. A., Boe, B., Hill, C., Franklin, D., & Harlow, D. (2014). Computational Thinking for Physics: Programming Models of Physics Phenomenon in Elementary School. In *2013 Physics Education Research Conference Proceedings* (pp. 133–136). https://doi.org/10.1119/perc.2013.pr.021

Gall, M. D., Gall, J. P., & Borg, W. R. (2003). *Educational research: An*

introduction, 7th Edition. Boston: Allyn and Bacon.

Gretter, S., & Yadav, A. (2016). Computational Thinking and Media & Information Literacy : An Integrated Approach to Teaching Twenty-First Century Skills. *TechTrends*, *5*(60), 510–516. https://doi.org/10.1007/s11528-016-0098-4

Hershkovitz, A., Sitman, R., Israel-fishelson, R., Eguíluz, A., Garaizar, P., & Guenaga, M. (2019). Creativity in The Acquisition of Computational Thinking. *Interactive Learning Environments*, 1–17. https://doi.org/10.1080/10494820.2019.1610451

Jeon, Y., & Kim, T. (2017). The Effects of The Computational Thinking- Based Programming Class on The Computer Learning Attitude of Non-Major Students. *Journal of Theoretical and Applied Information Technology*, *95*(17), 4330–4339.

Kemendikbud. (2014). Konsep dan Implementasi Kurikulum 2013. Jakarta: Kementerian Pendidikan dan Kebudayaan.

Lawanto, K., Close, K., Ames, C., & Brasiel, S. (2017). Exploring Strengths and Weaknesses in Middle School Students' Computational Thinking in Scratch. In J. P. Rich & B. C. Hodges (Eds.), *Emerging Research, Practice, and Policy on Computational Thinking* (pp. 307–326). Cham: Springer. https://doi.org/10.1007/978-3-319-52691-1

Lopez, V., & Hernandez, M. I. (2015). Scratch as a computational modelling tool for teaching physics. *Physics Education*, *50*(3), 310–316. https://doi.org/10.1088/0031-9120/50/3/310

Lye, S. Y., & Koh, J. H. L. (2014). Review on teaching and learning of computational thinking through programming: What is next for K-12? *Computers in Human Behavior*, *41*, 51–61. https://doi.org/10.1016/j.chb.2014.09.012

Mariano, D., Martins, P., Helene Santos, L., & de Melo-Minardi, R. C. (2019). Introducing Programming Skills for Life Science Students. *Biochemistry and Molecular Biology Education*, 1–8. https://doi.org/10.1002/bmb.21230

OECD. (2015). *Frascati Manual -2015 Edition [Flyer]*. Retrieved from https://www.oecd.org/sti/inno/Frascati-Manual-2015-Flyer-EN.pdf

Partnership for 21st Century Skills. (2005). *A Report and Mile Guide for 21st Century Skills*. Washington, DC: Partnership for 21st Century Learning.

Partnership for 21st Century Skills. (2014). *Framework for State Action on Global Education*. Washington, DC: Partnership for 21st Century Learning.

Peel, A., Sadler, T. D., & Friedrichsen, P. (2019). Learning Natural Selection through Computational Thinking: Unplugged Design of Algorithmic Explanations. *Journal of Research in Science Teaching*, 1–25. https://doi.org/10.1002/tea.21545

Postlethwaite, T. N. (2005). Educational research: some basic concepts and terminology 1. *Comparative Education*, *1*, 1–55. https://doi.org/10.1111/j.1467-8527.2007.00388_4.x

Saez-Lopez, J.-M., Roman-Gonzalez, M., & Vazquez-Cano, E. (2016). Computers & Education Visual programming languages integrated across the curriculum in elementary school : A two year case study using " Scratch " in fi ve schools ez-L o. *Computers & Education*, *97*, 129–141. https://doi.org/10.1016/j.compedu.2016.03.003

Shute, V. J., Sun, C., & Asbell-Clarke, J. (2017). Demystifying computational

thinking. *Educational Research Review*, *22*, 142–158. https://doi.org/10.1016/j.edurev.2017.09.003

Sidek, M. N., & Jamaludin, A. (2005). *Pembinaan Modul: Bagaimana Membina Modul Latihan dan Modul Akademik*. Serdang: UPM.

The Pearson Foundation, Microsoft Partners in Learning, & Gallup. (2013). 21st Century Skills and The Workplace A 2013 Microsoft Partners in Learning and Pearson Foundation, 1–19.

Trudel, L., & Métioui, A. (2011). Conception of A Computer-Aided Physics Laboratory to Facilitate The Understanding of Kinematical Concepts. In *Proceedings of the 9th International Conference on Education and Information Systems, Technologies and Applications*. Orlando, Florida: International Institute of Informatics and Systemics.

UNESCO. (2014). *Guide to conducting an R&D survey: For countries starting to measure research and experimental development*. https://doi.org/10.15220/978-92-9189-151-1-en

Vallance, M., & Towndrow, P. A. (2016). Pedagogic transformation, student-directed design and computational thinking. *Pedagogies*, *11*(3), 218–234. https://doi.org/10.1080/1554480X.2016.1182437

Wilkerson-Jerde, M. H. (2014). Construction, categorization, and consensus: Student generated computational artifacts as a context for disciplinary reflection. *Educational Technology Research and Development*, *62*(1), 99–121. https://doi.org/10.1007/s11423-013-9327-0

Wing, J. M. (2010). Computational Thinking: What and Why? *Thelink - The Magaizne of the Varnegie Mellon University School of Computer Science*, 1–6. Retrieved from http://www.cs.cmu.edu/link/research-notebook-computational-thinking-what-and-why