

Plug and Unplugged Activities to Enhance Computational Thinking Self Efficacy of Pre-service Biology Teachers

Indah Juwita Sari^{1*}, R. Ahmad Zaky El Islami², Desi Nur Eka Fitriana¹, Dwi Ratnasari¹, Ika Rifqiawati¹, Indria Wahyuni¹, Prasart Nuangchalerm³

¹Department of Biology Education, Faculty of Teacher Training and Education, Universitas Sultan Ageng Tirtayasa, Banten, Indonesia

²Department of Science Education, Faculty of Teacher Training and Education, Universitas Sultan Ageng Tirtayasa, Banten, Indonesia

³Faculty of Education, Mahasarakham University, Thailand

*Corresponding author's email: indah.juwitasari@untirta.ac.id

Article History:

Received date: January 23 2024

Received in revised from: April 5 2024

Accepted date: April 17 2024

Available online: April 21 2024

Citation:

Sari, I.J., El Islami, R.A.Z., Fitriana, D.N.E., Ratnasari, D., Rifqiawati, I., Wahyuni, I., & Nuangchalerm, P. 2024. Plug and unplugged activities to enhance computational thinking self efficacy's pre-service biology teachers. *Jurnal Pendidikan Sains Indonesia (Indonesian Journal of Science Education)*, 12(2):393-402.

Abstract. Currently, plug-and-unplugged activities for learning statistics are needed to determine students' mindsets when solving statistical problems manually, using calculations without programming as an unplugged activity and for data analysis as a plug activity. This research analyzes computational thinking self-efficacy in pre-service biology teachers by implementing plug-and-unplugged activities in statistics learning. The research method used is a case study with a computational thinking self-efficacy questionnaire instrument. The sample used was 107 pre-service biology teachers at a university in Indonesia who received plug-and-unplugged activities in statistics learning for one semester. The programs used in plug activities are the R and SPSS applications. The research results show that 43% of pre-service biology teachers' computational thinking self-efficacy is in the good category, 26% is in the moderate category, 16% is in the very good category, 10% is in the low category, and the rest are very low. Meanwhile, if we look at each component of self-efficacy, computational thinking, decomposition, pattern recognition, abstraction, and algorithmic thinking are each in the moderate category. This indicates that plug-and-unplugged activities can initiate computational thinking skills in pre-service biology teachers. It is necessary to routinely provide activities or learning activities that can improve the computational thinking of pre-service biology teachers because of the need in this digital era.

Keywords: plug and unplugged activities; computational thinking self-efficacy; learning statistics; pre-service biology teachers

Introduction

Computers have become an irreplaceable tool in the daily life of most people. Many real-world problems are solved using computers or its science principles. For this reason, the ability of individuals to use computers effectively is a skill that can help them to manage problems in their daily activities (Booth, 2013). However, people may have difficulty solving problems using computers if they do not have a strong technological background. Such people may need to learn how to make their lives easier by using technology effectively and in ways that improve their quality of life (Wing, 2008; Computer Science Teacher

Association [CSTA], 2011). Therefore, students need to carry out unplugged and plugged activities in solving data analysis problems in statistics courses as an opportunity to gain hands-on experience in class.

Unplugged and plugged programming activities in statistics learning are needed because students need to analyze data effectively, efficiently, and validly. Unplugged programming activities are learning activities without being connected to IT devices (Tsarava et al., 2017). Plugged programming activities are learning activities that use programs that are paired with IT devices (plugged activities) (Tsarava et al., 2017). For plugged activities, namely games, using the help of a computer, you can use a PC, laptop, or smartphone. Besides that, students also get the much-needed provision for the future. According to Ramalingam et al. (2004), programming experience affects programming self-efficacy. Because there is a relationship between programming and computational thinking, these results add weight to the assumption that students' self-efficacy might impact their competence acquisition. Besides that, students also get the much-needed provision for the future. According to Ramalingam et al. (2004), programming experience affects self-efficacy because there is a relationship between coding and computational thinking. These results add weight to the assumption that students' self-efficacy in computational thinking might impact their competence acquisition in it.

The ability to think computationally is not just a thought process to solve problems but also contains processes in it, such as problems and solving each part one by one and modeling the problem coherently (Zhang et al., 2023; Astrachan, 2009). According to Brichacek (2014), computational thinking is directly related to '21st-century student standards'. If we apply computational thinking to higher-order skills such as problem-solving, creativity, and critical thinking, our innovation capacity will likely increase. Given that innovation is one of the skills students need in the 21st century, it can be argued that computational thinking has an important role in delivering the standards expected in this century. In this regard, computational thinking must be one of the critical skills of the 21st century (Philips, 2009).

The efforts made by educators in Indonesia are still limited regarding the measurement of the concept of self-efficacy, which is related to concepts such as academic success, motivation, and educational self-regulation. Although several factors influence student success, self-efficacy and attitudes are more important than others (Anastasiadou & Karakos, 2011). Belief in self-efficacy is an individual's belief about whether they can demonstrate the behavior necessary to achieve the desired target. The main goal when measuring self-efficacy is measuring task performance rather than individual personal features. Self-efficacy measurement is a component that gives an idea of the individual's success. Steps can then be taken to increase their success and performance (Askar & Davenport, 2009). Therefore, self-efficacy measurement is considered important, and several self-efficacy scales have been developed for many concepts in the field of education (Ding et al., 2023). Several scales related to computational thinking, such as programming, were developed (Ramalingam et al., 2004).

Measuring self-efficacy is an important component that can provide an overview of how successful individuals face the real world in the future. The steps used to solve problems in computational thinking can then be taken to increase the success and performance of students in the future. This is reinforced by technology that continues to develop and programming languages that continue to experience progress in versions. So, students must be introduced to the differences in data analysis using the unplugged system and plug systems such as SPSS and R programming in the statistics class. Thus, this research aims to analyze computational thinking self-efficacy in pre-service biology teachers by implementing plug-and-unplugged activities in statistics learning.

Methods

This research was conducted using the case study method, namely a research method, to understand better a subject or process (Flyvbjerg, 2011). The sample of this research is 107 in-service biology teachers. The research instrument used in this study was the computational thinking self-efficacy questionnaire developed by Kukul et al. (2017), consisting of 20 items. Kukul et al. (2017) developed 31 computational thinking self-efficacy items on a 5-point scale called computer programming self-efficacy. Based on the validity and reliability of tests results, these items have an adequate level to measure the self-efficacy of students' computer programming.

Table 1. The Category of computational thinking self-efficacy

Category	Range
Very Satisfied	85-100
Satisfied	70-84
Moderate	55-69
Disatisfied	50-54
Very Disatisfied	0-49

Results and Discussion

Plug and unplug activities are learning that is connected to the computer and learning that is not connected to the computer. This plug and unplug teaching can provide an overview of student satisfaction in learning (Erumit & Sahin, 2020). In learning statistics, plug and unplug activities are shown in the student section to analyze data using the SPSS application and unplug activities are shown in student activities through manual calculations. This plug and unplug activity are considered quite effective in training students to think computationally. This is in accordance with research conducted by Chongo et al. (2021) that plug and unplug activities are effective in developing students' CT skills, not only that, plug and unplug are also recommended in learning because they are able to improve students' abilities in solving problems.

The ability to think computationally is a thinking process that focuses on formulating problems and expressing solutions in such a way that they can be applied. Computational thinking is a trend of skills that can be developed in learning. The components in computational thinking are abstraction, generalization, decomposition, algorithmic thinking and debugging (detecting and correcting errors). Abstraction the ability to decide what information about an object is known to be stored and what information should be ignored (Wing, 2011). Generalization is the ability of students to formulate general solutions. Decomposition is the ability of students to solve complex problems into smaller and simpler components. Algorithmic is the ability to design a series of actions to be performed. Debugging means the ability to correct actions taken, erase and correct errors (Selby, 2014). Computational thinking is one of the good skills to improve students' ability to solve problems because someone will find it easy to observe problems, find solutions to problems, solve problems, and be able to develop solutions or problem solving. In addition, computational thinking sharpens us to think more effectively and efficiently.

Computational thinking ability can be seen in students' self-confidence. Research has shown that self-efficacy has a strong influence on coding achievement and computational thinking skills (Baek et al., 2019). Self-efficacy can be defined as the perceptions of students regarding their own skills and is thought to be directly associated with their performance and effort in performing a task (Bandura, 1977). Self-efficacy is an important

construct in education, as it can influence (among other aspects) perseverance, engagement, and success in educational tasks. As such, a student's CTSE can have an important influence on, and may be a predictor for, the development and use of CT skills (De Jong & Jeuring, 2022). As such, a student's CTSE can have an important influence on, and may be a predictor for, the development and use of computational thinking skills (De Jong & Jeuring, 2022)

In this study, data were obtained from the distribution of the research instrument used in this study, namely self-efficacy questionnaires for computational thinking skills as many as 20 items and distributed to the study sample, namely 107 pre-service biology teachers. The data can be seen in Figure 1.

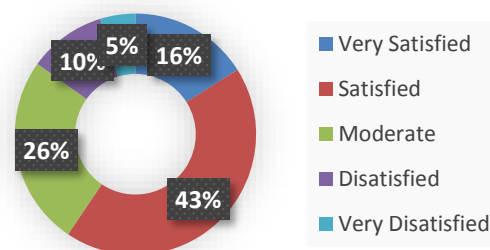


Figure 1. Computational Thinking Self Efficacy of Pre-service Biology Teachers

The results show that students for CTSE is divided into five categories, namely very satisfied, satisfied, moderate, dissatisfied, and very dissatisfied. As many as 16% of students in very satisfied category, 43% of students in satisfied category, 26% of students in moderate, 10% of students in dissatisfied category and students who have self-efficacy in very dissatisfied category as much as 5%.

Student self-efficacy is a belief in students demonstrating the behaviour needed to achieve success. As many as 16% of students have computational thinking self-efficacy in the very satisfied and as much as 43% in the satisfied category. This means that most of the total students have high self-efficacy in their computational thinking abilities. Furthermore, 26% of students have computational thinking self-efficacy which is in the quite satisfied category. This means that a small portion of the total students have computational thinking self-efficacy at a moderate level. Students who have self-efficacy are in the category of satisfied and very dissatisfied as much as 10% and 5%. This means that a small proportion of students have low computational thinking self-efficacy.

Computational thinking self-efficacy will determine students to feel, think, motivate themselves and behave computationally. Students who have high computational thinking self-efficacy have good confidence in each stage of it that has been carried out during the learning process starting from determining the most suitable knowledge data and information to use, formulating solutions, decomposing or breaking down problems into simple components, as well as designing various steps to be carried out (algorithmic), and identifying errors that occur and correcting errors regarding coding in programming. All the steps that are passed through plugging or unplugging can be done very well by students. In line with Ramadhani & Mariani (2021) opinion which states that students with computational abilities students with high self-efficacy can complete tasks, understand, and choose strategies in completing tasks (magnitude dimension), students are able to

survive and be confident in facing tasks and challenges (strength dimension), students have confidence in completing tasks in various contexts (generality dimension). High self-confidence makes students always enthusiastic and have great motivation in participating in statistics learning so that they have good performance in carrying out assignments and are reflected in the success of everyone.

On the other hand, students with low CTSE mean that these students are unsure of every step of it that is done. Students are still afraid and do not have confidence in their ability to deal with various diverse contexts and lack confidence when they have to be in an unusual situation so that students are less able to determine the most suitable knowledge data and information to use, formulate solutions, decompose or solve problems into simple components, as well as designing various steps to be carried out (algorithmic), and identifying errors that occur and correcting errors regarding coding in programming. Students with low computational thinking self-efficacy tend to see difficult assignments as threats that they must avoid. Because of this, they also tend to avoid setting goals and have a low level of commitment to the goals they set.

Computational thinking self-efficacy is a benchmark that describes student success in working and thinking computationally whose success is achieved along with high self-confidence. A higher level of self-efficacy will increase the success of individuals and the level of happiness caused by that success. Individuals who trust their talents are more advanced in coping with difficult tasks (Bandura, 2001). The factors of independence and persistence of students, the ability to analyse complex and simple programming tasks, and self-regulation are things that need to be considered in increasing the self-efficacy of computational thinking (Ramalingam and Wiedenbeck, 1998). In addition, the habit of practicing mutational thinking to solve problems can be continued because a positive correlation is found between computational thinking, self-efficacy programming and reflective thinking aimed at problem solving (Durak et al. 2019). This is in accordance with the opinion (Ramalingam et al., 2014) which says that it is also influenced by previous programming experiences carried out by students and increases as students' progress through introductory courses.

Simple research data is simply narrated in the form of complete and clear sentences, while complex data is presented in the form of tables and or pictures as needed.

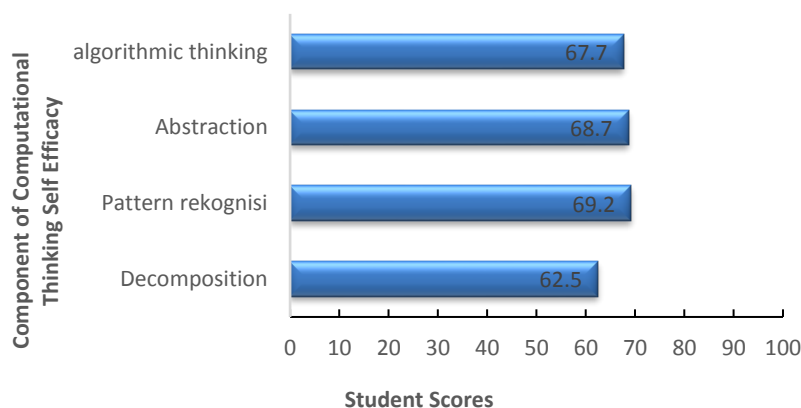


Figure 2. Component of Computational Thinking Self Efficacy

Decomposition is one of the important components of computational thinking because the decomposition ability of students greatly influences the next problem-solving step. If students already have good decomposition skills, they will find solving problems easier. As the first step, problem decomposition will affect the next steps greatly. If students have good skills in problem decomposition, students will have bigger opportunity to solve the

given problem (Rosali & Suryadi, 2021). The research results using plug and unplugged activities in statistics lessons obtained a decomposition value of 62.5, included in the moderate category. This shows that some students can solve statistical problems systematically, from a complex level to a level that is easier to understand. Students can identify problems, develop strategies, and understand problem patterns to find concrete solutions. Decomposition can involve looking at similarities within and patterns of the problem's constituent parts so they become easier to understand and work with (Mvalo & Bates, 2019).

The pattern recognition component averages 69.20 in the category where most students have developed the pattern recognition component. According to Supiarmo, et al. (2021), pattern recognition identifies patterns used to develop solutions and solve problems. Pattern recognition can be used to find the best solution for solving a problem and studying certain types of problems. In this study, indicators of pattern recognition ability were measured using a questionnaire. The highest indicator is the indicator related to knowing how to use programming variables, namely 71.58. The lowest indicator in pattern recognition is the indicator that can correct errors regarding coding in the program, namely 66.72. However, students have reached the pattern recognition component in computational thinking. This is because students can solve problems with the material and patterns found in a problem despite errors. According to Hasanah (2022), Pattern recognition in problem-solving is the primary key to determining the right solution to a problem and to knowing how to solve a particular type of problem so that recognizing the same pattern or characteristic can help students in solving problems and help students in building a problem solution.

The research results on plug and unplugged activities in statistics education courses show that abstraction ability has an average value of 68.7, which is included in the moderate category. Shows that students can choose the information needed to solve a problem. According to Suprihatiningsih (2020), the ability to abstract in computational thinking is the ability to identify patterns or information used to find different solutions to a problem. The ability to abstract will affect the ability to design algorithms in computational things because after students can choose essential and unimportant information on a problem, they will be able to determine the right solution to the problem. In this study, indicators of abstraction ability were measured using questionnaires. The highest indicator is the ability of students to choose the most suitable programming language, with a value of 75.5. In contrast, the lowest indicator of abstraction ability is the ability of students to suggest different solutions to problems. These results show that students can choose the information needed but have yet to determine different solutions based on the information obtained. Based on the results of this study, abstraction ability needs to be improved. Ways that can be done to improve abstraction skills, according to Supini (2022), are (1) making a summary, someone will choose essential and unimportant information from what is obtained, (2) retelling the day that has been passed, by retelling then someone will choose a memorable story from the many stories that happened on that day. Wafiqah (2020) also stated that classroom learning begins by giving questions to recalling previously learned material or concepts can help trigger reflective abstraction skills.

Algorithmic thinking in learning activities in this research occurs in the final phase, such as creating systematic stages in data analysis using applications, programming, or by hand. Based on the results of the CTSE questionnaire, the average score for the algorithmic thinking component of pre-service biology teachers was 67.7, which is in the moderate category. This means that pre-service biology teachers can use SPSS and R to produce valid data analysis with the highest achievement score of 72.3. Pre-service biology teachers are also able to choose the most suitable steps for solving programming problems and are also able to determine solutions to solve the problem step by step. Algorithmic thinking

needs to be further improved because it will be helpful in the future careers of pre-service biology teachers who are becoming biology teachers at the high school level. They need to analyze data and draw conclusions quickly regarding student learning outcomes, learning processes involving calculations in class, or in making research or development of programming and digital media. According to Juškevičienė (2020), algorithmic thinking is very important in this digital era for the programming needs of educators. However, the challenge is that there is still limited learning to improve this ability. Therefore, plug-and-unplugged activities can be used to initiate students' algorithmic thinking, which is generally computational thinking skills. As is the case with research conducted by Sari et al., (2022) by conducting learning using computational inquiry-based teaching (CibT) the results of which can improve students' computational thinking skills. Milkova (2015) also explains that multimedia applications can develop algorithmic thinking, which provides excellent opportunities for teachers to demonstrate and visualize lesson material more clearly and comprehensively and allows them to prepare learning materials for students that optimize their study habits. Along with large software products developed by professional teams, there are also various small programs dealing with objects appropriate to the subject matter that are created based on scripts provided by teachers, considering the needs of students.

Conclusion

In this study, most students had high computational thinking self-efficacy because students believed in each stage of it, starting from determining the most suitable knowledge data and information to use, formulating solutions, decomposing, or breaking down problems into simple components, and designing various the steps to be performed (algorithmic), and identifying errors that occur and correcting errors regarding coding in programming which are passed both plug and unplug. The research results show that 43% of pre-service biology teachers' computational thinking self-efficacy is in the good category, 26% is in the moderate category, 16% is in the very good category, 10% is in the low category, and the rest are very low. Meanwhile, if we look at each component of self-efficacy, computational thinking, decomposition, pattern recognition, abstraction, and algorithmic thinking are each in the moderate category. This indicates that plug-and-unplugged activities can initiate computational thinking skills in pre-service biology teachers. It is necessary to routinely provide activities or learning activities that can improve the computational thinking of pre-service biology teachers because of the need in this digital era.

References

- Anastasiadou, S.D. & Karakos, A.S. 2011. The beliefs of electrical and computer engineering students regarding computer programming. *The International Journal of Technology, Knowledge, and Society*, 7(1):37–51. <https://doi.org/10.18848/1832-3669/CGP/v07i01/56170>
- Askar, P., Davenport, D. 2009. An investigation of factors related to self-efficacy for java Programming among engineering students. *The Turkish Online Journal of Educational Technology*, 8(1):26–32. http://dx.doi.org/10.22495/jgr_v4_i4_c5_p6
- Astrachan, O. 2009. *A New Way of Thinking About Computational Thinking*. ACM SIGCSE. Chattanooga, ABD.

- Baek, Y., Yang, D.Z., & Fan, Y.B. 2019. Understanding second grader's computational thinking skills in robotics through their individual traits. *Information Discovery and Delivery*, 47(4):218–228. <https://doi.org/10.1108/IDD-09-2019-0065>
- Bandura, A. 1977. Self-efficacy: Toward a unifying theory of behavioral change. *Psychological Review*, 8:191–215, [https://doi.org/10.1016/0146-6402\(78\)90002-4](https://doi.org/10.1016/0146-6402(78)90002-4)
- Bandura, A. 2001. Social cognitive theory: An agentic perspective. *Annual Review of Psychology*, 52(1):1–26. <https://doi.org/10.1146/annurev.psych.52.1.1>
- Booth, A.W. 2013. Mixed-Methods Study of the Impact of a Computational Thinking Course on Student Attitudes about Technology and Computation. *Doktora Tezi, Baylor University, Department of Educational Psychology, Baylor*. <http://dx.doi.org/10.1145/3017680.3017750>
- Brichacek, A. 2014. Computational thinking boosts students' higher-order skills. <https://www.iste.org/explore/articleDetail?articleid=232&category=Featuredvideos&article=Computational%20thinking%20boosts%20students%E2%80%99%20higher-order%20skillsadresinden21.05.2015tarihindeulařılmıştır>
- Chongo, S., Osman, K., & Nayan, N.A. 2021. Impact of the Plugged-in and Unplugged chemistry computational thinking modules on achievement in chemistry. *Eurasia Journal of Mathematics, Science and Technology Education*, 17(4):1-21. <https://doi.org/10.29333/ejmste/10789>
- CSTA, ISTE. 2011. Computational thinking in K–12 education leadership toolkit. *Computer Science Teacher Association*: <http://csta.acm.org/Curriculum/sub/CurrFiles/471.11CTLeadershipToolkit-SPvF.pdfadresindenalındı>
- De Jong, I. & Jeuring, J. 2022. Developing a Self-efficacy Scale for Computational Thinking (CT-SES). In *Proceedings of the 22nd Koli Calling International Conference on Computing Education Research* (pp. 1-2). <https://doi.org/10.1145/3564721.3565954>
- Ding, Y., Yang Hansen, K., & Klapp, A. 2023. Testing measurement invariance of mathematics self-concept and self-efficacy in PISA using MGCFA and the alignment method. *European Journal of Psychology of Education*, 38(2):709-732. <https://doi.org/10.1007/s10212-022-00623-y>
- Durak, H.Y., Yilmaz, F.G.K., & Yilmaz, R. 2019. Computational thinking, programming self-efficacy, problem solving and experiences in the programming process conducted with robotic activities. *Contemporary Educational Technology*, 10(2):173-197. <https://doi.org/10.30935/cet.554493>
- Erumit, A.K. & Sahin, G. 2020. Plugged or unplugged teaching: a case study of students' preferences for the teaching programming. *International Journal of Computer Science Education in Schools*, 4(1):23-32. <http://dx.doi.org/10.21585/ijcses.v4i1.82>

- Flyvbjerg, B. 2011. Case study. *The Sage Handbook of Qualitative Research*, 4:301–316.
- Hasanah, U. & Haryadi, H. 2022. Pendampingan mahasiswa dalam berpikir secara komputasi (computational thinking). *Abdinesia: Jurnal Pengabdian Kepada Masyarakat*, 2(1):7-14. <http://dx.doi.org/10.24071/altruism.2020.030204>
- Juškevičienė, A. 2020. Developing algorithmic thinking through computational making. *Data science: new issues, challenges, and applications*: 183-197.
- Kukul, V., Gökçeşlan, Ş., & Günbatır, M.S. 2017. Computer programming self-efficacy scale (CPSES) for secondary school students: Development, validation, and reliability. *Eğitim Teknolojisi Kuram ve Uygulama*, 7(1):158-179. <http://dx.doi.org/10.17943/etku.288493>
- Milkova, E. 2015. Multimedia application for educational purposes: Development of algorithmic thinking. *Applied Computing and Informatics*, 11(1):76-88. <https://doi.org/10.1016/j.aci.2014.05.001>
- Mvalo, S. & Bates, C. 2019. Students' Understanding of Computational Thinking with a Focus on Decomposition in Building Network Simulations. *Proceedings of the 10th International Conference on Computer Supported Education (CSEDU 2018)*: 245-252. <https://doi.org/10.5220/0006693302450252>
- Phillips, P. 2009. *Computational Thinking a Problem-Solving Tool for Every Classroom*. Computer Science Teacher Association: <http://csta.acm.org/Resources/sub/Resourcefiles/comphinking.pdfadresindenalındı>.
- Ramadhani, L.I.P. & Mariani, S. 2021. Kemampuan Komputasional Siswa Dalam Memecahkan Masalah Matematika SMP Melalui Digital Project Based Learning Ditinjau Dari Self Efficacy. *PRISMA, Prosiding Seminar Nasional Matematika* 4:289-297.
- Ramalingam, V. & Wiedenbeck, S. 1998. Development and validation of scores on a computer programming self-efficacy scale and group analyses of novice programmer self-efficacy. *Journal of Educational Computing Research*, 19(4):367-381. <https://doi.org/10.2190/C670-Y3C8-LTJ1-CT3P>
- Ramalingam, V., LaBelle, D., & Wiedenbeck, S. 2004. Self-efficacy and mental models in learning to program. In *Proceedings of the 9th annual SIGCSE conference on Innovation and technology in computer science education* (pp. 171-175). DOI: 10.1145/1007996.1008042
- Rosali, D.F. & Suryadi, D. 2021. An analysis of students' computational thinking skills on the number patterns lesson during the covid-19 pandemic. *Formatif: Jurnal Ilmiah Pendidikan MIPA*, 11(2):217-232 <http://dx.doi.org/10.30998/formatif.v11i2.9905>
- Sari, I.J., Vongsangnak, W., & Pongsophon, P. 2022. The effect of bioinformatics module on molecular genetics concepts on senior high school students' computational thinking skills. *Shanlax International Journal of Education*, 10(2):9-17. <http://dx.doi.org/10.34293/education.v10i2.4680>

- Selby, C.C. & Woollard, J. 2013. Refining an Understanding of Computational Thinking. *Special Interest Group on Computer Science Education (SIGCSE)*, 1–23.
- Supiarmo M, Turmudi, & Susanti, E. 2021. Proses berpikir komputasional siswa dalam menyelesaikan soal pisa konten change and relationship berdasarkan self-regulated learning. *Numeracy*, 8(1):58-72. <https://doi.org/10.46244/numeracy.v8i1.1378>
- Supini, E. 2022. *Konsep Abstraksi dalam Computational Thingking dan Penerapannya*. <https://blog.kejarcita.id/konsep-abstraksi-dalam-computational-thinking-dan-penerapannya>.
- Suprihatiningsih, Harmini, T., & Annurwanda, P.S. 2020. Computational thinking ability students based on gender in calculus learning. *Jurnal Program Studi Pendidikan Matematika*, 9(4):977–986. <http://dx.doi.org/10.24127/ajpm.v9i4.3160>
- Tsarava, K., Moeller, K., Pinkwart, N., Butz, M., Trautwein, U., & Ninaus, M. 2017. Training computational thinking: Game-based unplugged and plugged-in activities in primary school. *European conference on games based learning* (pp. 687-695). Academic Conferences International Limited.
- Wafiqoh, R. 2020. Reflective abstraction: How can you find out in mathematics learning? *International Journal of Scientific and Technology Research*, 9(2):43–47. <http://dx.doi.org/10.1088/1742-6596/1280/4/042039>
- Wing, J. 2008. Computational thinking and thinking about computing. *Philosophical Transactions of The Royal Society*, 3717–3725.
- Wing, J.M. 2011. Computational thinking. *Communications of the ACM*, 3–3. DOI: 10.1109/vlhcc.2011.6070404.
- Zhang, J.H., Meng, B., Zou, L.C., Zhu, Y., & Hwang, G.J. 2023. Progressive flowchart development scaffolding to improve university students' computational thinking and programming self-efficacy. *Interactive Learning Environments*, 31(6):3792-3809. <https://doi.org/10.1080/10494820.2021.1943687>