

# Problems of Computational Thinking, Teaching, and Learning in a STEM Framework: A Literature Review

Bayu Widyaswara Suwahyo <sup>1, \*</sup>

<sup>1</sup> *Department of Education Technology, Faculty of Education, Universitas Negeri Malang, Malang, Indonesia*

<sup>\*</sup> *Corresponding author. Email: bayu.widyaswara.1901218@students.um.ac.id*

## ABSTRACT

Thinking like a computer scientist means more than being able to program a computer. which refers to the way a computer scientist acts in. This is the overall perspective we intend to adopt in this systematic review: as many policies strongly promote CT in primary school, it cannot be reduced to simple computer science skills nor should it lead children to think like computers, but rather, it should help them to develop higher level thinking abilities in order to correctly interact with digital technologies. This study presents a 10-year (2010–2019) systematic literature review related to the way that computational thinking (CT) has grown in elementary-level education students with the intention to: (1) knowing setting CT has been implemented, (2) the learning context in education, and (3) present the learning outcomes for students who engage in educational activities.

**Keywords:** computational thinking, primary education, systematic literature review

## 1. INTRODUCTION

The concept behind Computational Thinking (CT) isn't new; the best-known author is certainly [1] who fostered its development even in elementary school children, although we've got some precedents since the 1950s with the phrase 'algorithmic thinking'. However, the development of CT in school is commonly confused with the coding process alone, which is really the programming activity, albeit using specific languages addressing younger people (the most famous is perhaps Scratch, developed by MIT). We may suppose that this can be thanks to the prevalence in policy-making of the functionalistic paradigm [2], having the aim of promoting digital skills relevant to several career paths [3].

We have along these lines to recognize activities concentrated essentially on software engineering and programming develops from the more significant level reasoning aptitudes, more important from an instructive point of view. [4], two of the Scratch creators, proposed an operational definition where they perceive 'CT Concepts' ('successions, circles, parallelism, occasions, conditionals, administrators, and information') from 'CT practices' that 'emphasis on the way toward intuition and getting the hang of, moving past what you are figuring out how to how you are learning' ('being gradual and iterative, testing and investigating, reusing and remixing, and abstracting and modularizing') and 'CT Perspectives' (communicating, interfacing, addressing). [5] makes an amalgamation of a few encounters,

expressing and formalizing a model in 5 significant advances: 'disintegration, design acknowledgment, reflection, algorithmic structure and assessment'. This sort of definition relates CT chiefly to a critical thinking measure, as indicated by the [6] definition: 'CT is a way to deal with tackling issues in a manner that can be actualized with a PC. We should accordingly give an alternate job to the 'testing and investigating' practice, that is considered in numerous definitions (separated the definition from [4] that we have reviewed, see likewise [7].

While decay, design acknowledgment, reflection and algorithmic structure are generally identified with the program plan rather than coding in the severe sense, and could likewise be created with unplugged exercises, 'testing and investigating' identifies with the genuine programming stage, exploiting the highlights of the PC to give prompt input to the students' activities, which can be valuable so as to improve self-administrative abilities [8]. Another methodology that merits referencing from an instructive viewpoint is the one that considers figuring to be an innovative human action [7], [9], [10], [4].

Inside their 'communicating' CT point of view, express that 'a computational mastermind considers calculation to be a medium and figures, "I can make." also, "I can communicate my thoughts through this new medium" [2] connect this issue to a particular worldview of coding, the 'postmodern' one: 'Imaginative coding permits specialists to address and scrutinize code and, at

the equivalent time, communicate through code. This is the general viewpoint we expect to embrace in this efficient audit: the same number of approaches emphatically advance CT in grade school, it can't be diminished to straightforward software engineering aptitudes nor should it lead kids to think like PC, yet rather, it should assist them with creating more significant level deduction capacities so as to accurately communicate with advanced advancements.

**2. METHODS**

**a. Research Question**

The principle question of this paper is its endeavor to answer the way that 'computational reasoning' has advanced in rudimentary level instruction in primary school (K-6). Accordingly, the goal of the paper is to review the writing from 2010 to 2019 on the improvement of 'computational thinking' in the essential degree of instruction, the purpose being: (1) present a diagram at the specific circumstance/settings of instruction where CT has been actualized; (2) distinguish the learning setting that CT is utilized in training; and (3) feature the manners by which CT has been assessed and present the learning results for understudies who participate in CT instructive exercises.

**b. Database Sources and Search Term**

This sources papers distributed in journals, worldwide gatherings procedures, workshops and discussions since the start of January 2010 up to the furthest limit of December 2019. To accomplish this, ten major electronic information bases that are identified with 'instructive settings', 'advanced innovation' and 'social science' were being utilized in the survey in center. All the more explicitly, it utilizes: SpringerLink, Association for Registering Machinery (ACm), Bio-Medical Library, ERIC (Education Resources Information Center), IEEE Xplore Digital Library, Taylor and Francis Online, Wiley, LearnTechLib (Learning and Technology Library), and Science Direct. This sources paper was directed utilizing the keyword 'computational thinking'. The search was restricted to the period from January 2010 to December 2019. Subsequently, 3482 papers/things were recognized.

**c. Determination of Papers to be Remembered for the Research**

The papers selected within the review were carefully chosen under specific criteria. Through a full screening of the identified papers and a detailed take a look at each paper's title, abstract and content, furthermore as an exclusion of these papers that were not relevant to the inclusion criteria, 53 papers were congregated. The inclusion criteria so as for the papers to be included in the review are reported below. the educational papers had to: (1) make exact relevancy the term 'computational thinking' within the title and/or abstract and/or keywords;

(2) be written in West Germanic; (3) have a typical kind of a scientific paper (no posters, roundtables, add progress papers, short papers, etc.); (4) specialize in CT and first-level education students (K-6); and (5) present empirical data emerged from methodologically sound empirical studies – either collected by kindergarten students or by K-6 elementary students, by providing adequate information about the research methodology, the participants and also the research procedure used (no pilot and preliminary studies, exploratory studies).

At last, utilizing 'the previously mentioned rules', from the 3482 papers, a sum of 53 papers met the models to be remembered for the survey and were assessed as 'pertinent' for full content audit. 28 of them are distributed in 'scientific journals', sixteen at 'international conferences', seven are introduced at 'international symposiums' and two at 'international workshops'. The 28 'journal papers' found were distributed in 22 different papers. All papers were distributed in PC, instruction and identified with innovation diaries.

All diaries are preoccupied/ordered in different information bases, for example, Scopus, ProQuest, EBSCO, INSPEC, ERIC, Google Scholar, PsycINFO, Gale, ERA, SCImago and ERIH PLUS. See Addition A for additional insights concerning Data Collection. It is clear that over the most recent three year there is an enlarging enthusiasm of the computational thinking network for the development to elementary school.

**Table 1 Database Journal**

No	Databases Source	Number of Papers	The Inclusion Criteria
1	Springer	1,118	9
2	ACM	753	15
3	Bio-Medical Library	17	1
4	ERIC	192	3
5	IEEE Xplore Digital Library	325	4
6	Taylor and Francis Online	273	8
7	Wiley	172	2
8	LearnTechLib	393	6
9	Science Direct	239	5

**3. RESULTS**

**a. The Instructive Setting of Computational Thinking**

Principally, the outcomes show that the instructive degree of the led examinations don't just concern the elementary school understudies (31 papers) yet additionally the understudies going to Primary–Middle degree of training (14 papers). Concerning the area of these investigations, the greater part of them were directed in the school settings (32 papers, for example, the school study hall or the school's computer lab. The looked into considers concentrated on different learning subjects yet the extraordinary dominant part of studies

(44 papers) concentrated on STEM disciplines for creating CT-skills to elementary school understudies. With respect to the term of the examinations, the vast majority of them went on for as long as multi week (18 papers), while about equivalent number of papers (17) show that the term of the examinations fluctuates between one to a half year.

#### **b. Learning setting of Computational Thinking**

The routes through which CT has been incorporated in showing rehearses for understudies' CT-abilities advancement allude predominantly to Plugged Programming Activities (15 papers), Game Programming and Game Activities (10 papers) and Robotics (7 papers). A large portion of the mediations directed depended on stopped exercises (34 papers) for understudies' CT-improvement. Unplugged exercises were likewise led less significantly (10 papers), while there were endeavors to join stopped and unplugged exercises for a similar reason (8 papers). Papers proposing stopped exercises utilize at least one than one computerized condition: Scratch is by a long shot the most famous (16 papers), trailed by Alice (4 papers), Kodu (3 papers) and Agent sheets (3 papers).

19 other distinctive computerized conditions happen on more than one occasion in 23 papers. 38 examinations utilized a programming advanced condition for understudies' CT nurture with the square based visual programming dialects, which comprises most of the programming conditions utilized (36 out of 38). The audit study acquired that the ideas of reflection (41 papers), calculations and techniques (41 papers) and control structure (33 papers) establish the most widely recognized CT-ideas and abilities drew nearer in essential instruction. Additionally, the ideas of testing and confirmation (19 papers), issue deterioration (18 papers), parallelization (16 papers) and information examination (13 papers) were drawn nearer less significantly followed by the ideas of reproduction (8 papers), information portrayal and investigation (8 papers), information assortment (6 papers) and mechanization (2 papers).

#### **c. Computational Thinking Assessment**

The audited exact examinations alluding to essential degree of instruction understudies are arranged underneath as far as: (a) 'setting of the appraisal', and (b) 'results of the investigations'. Concerning the 'setting of the assessment', it is huge to be noticed that there were scarcely any examinations pointing not exclusively to state theories however test them too utilizing exact investigations utilizing 'test research techniques' [11].

Additionally, different 'instruments for information assortment' were utilized, with pre/post evaluations (28 papers), antiquities' investigation (17 papers), polls (17 papers), and meetings/conversations (15 papers) being among the most basic ones. Also, a mix of various sort of such apparatuses was utilized in most of the

investigations so that the results are more substantial and dependable. Likewise, the investigation of the fundamental 'results on CT decided in the looked into papers' conformed to 13 subjects.

#### **d. Analyzing Computational Thinking This Papers**

There's still an absence of assessment tools ready to effectively measure CT and its development through didactic intervention, and this may limit the spread of CT programs in K-6 education [12]. We therefore intend to extract from this systematic review some useful hints regarding the definitions most ordinarily adopted and also the potential assessment tool getting used in research. To this extent, we propose here a more profound analysis of the 18 papers with both pre- and post-assessment (even if they cannot all be considered experimental, which might also mean having an effect group). Moreover, in three of them, CT is that the variable quantity instead of the variable (for instance, 2 aims to indicate to what extent CT programs increase Critical Thinking, ability and Problem Solving).

Within the following sub-paragraphs, we therefore consider only those 15 papers with pre and post assessment associated with CT. Several papers adopt the [4] definition noted within the introduction to the present paper. [4] the foremost famous coding environment for children. per our approach, this definition has the merit of clearly distinguishing the CT Concepts mostly related to applied science ('Sequences, Loops, Events, Parallelism, Conditionals, Operators, Data'), from the CT Practices ('Being incremental and iterative, Testing and Debugging, Reusing and Remixing, Abstracting and Modularizing') and the CT Perspectives ('Expressing, Connecting, Questioning'). Not surprisingly, this definition is remarked mainly within the papers reporting research with Scratch tools.

But while CT Concepts are often measured by tests, in our review, we failed to find an assessment tool focusing strictly on Brennan and Resnick CT Practices and CT Perspectives. In some cases, we did find items in wider tests specializing in some specific aspects per the research aims. Three papers adopt the essential definition stated by [6]: CT is 'an approach to solving problems in an exceedingly way that may be solved by a computer. Two of them are ascribable to the identical research group, and especially highlight the property of this problem-solving methodology to be transferred and applied across subjects. This is clearly not an operational definition, but in any case, it addresses a scenario-based assessment tool progressing to verify if CT Concepts is applied in an exceedingly different situation than the computing environment within which they were developed.

Two more papers discuss with the Operational Definition stated by the pc Science Teachers Association (CSTA) and the International Society for Technology in

Education (ISTE) (2011); this definition is really an extension of Barr and Stephenson's (actually Barr and Stephenson report the building process of the CSTA-ISTE definition). It also focuses on data analysis, organization and representation, problem decomposition, abstraction, algorithmic thinking, automation, simulation and parallelization, all properties relating more or less on to an issue solving process.

The CSTA-ISTE definition could therefore be a decent representation of the CT features addressing higher level thinking skills, but it's limited by the absence of a stable model: the initial CSTA-ISTE statement refers to 'a problem-solving process that features (but isn't limited to)' some characteristics, so both papers must do further elaboration in order to make their assessment tool, which cannot therefore be taken as a standard reference. One paper [7] adopts a definition built upon 4 progressive steps: Decomposition, Pattern recognition, Abstraction, Algorithmic design, recalling the planning process utilized by software engineers.

We will find this definition also in papers other than the 15 that we are examining here. It also somehow refers to a controversy solving process that follows the Barr and Stephenson principle, but focuses more on the outline of a top down approach instead of the precise elements of a thinking skill. Another research [3] adopts an analogous definition, specializing in a 'thought process' that utilizes the weather of abstraction, generalization, decomposition, algorithmic thinking, and debugging, bearing on [12], [13]. One paper deserves a particular mention: its CT Assessment tool involves the Computational Thinking Levels Scale (CTLs), referring implicitly to an Operational Definition, which incorporates an artless dimension; therefore, it's somehow related to the approach recalled within the introduction that sees computing as an artless act. We'll describe the Assessment tool in additional detail below.

Finally, three papers don't declare their definition explicitly, nor can it's have deduced from the Assessment Tool, as this one mainly focuses only on CT Concepts. With relevance the assessment tools, the case we found from our analysis appears to be even worse, compared with our aims: while searching for tests or scales that: (1) specialize in CT Practices or Perspectives instead of only CT or CS Concepts; (2) are easily re-usable; and (3) are valid and reliable to some extent, we are led to the conclusion that it's only 3 out of 15 papers that meet the aforementioned conditions. We can first exclude 4 papers that truly restrict themselves only to measuring the training of CT or CS Concepts.

Moreover, we should always not consider a considerable number of them, whose authors developed their own test or scale in line with their specific research aims. this is often not fair for us for 2 reasons: first, from a metric point of view, such tests or scales didn't face any validity or reliability check; second, from a theoretical

perspective, they are strongly tormented by the specificity of the research, their aims, and also their CT Operational Definition (the 6 papers refer to 4 of the aforementioned Operational Definitions). Two more papers embedded the assessment tools within the programming environment they employed in the CT Developing Program, and are very specific, though otherwise they're of some interest. they will both be connected to the identical research group, which uses an Open-Ended Learning Environment (OELE) specializing in the flexibility of learners to represent knowledge in models.

The assessment tool is predicated on the comparison of the learners' model with the expert one, and so, the CT Vital Practices are somehow indirectly measured. The instrument is also of some interest, but the specificity of the environment makes it hardly transferable to other situations. Probably, the foremost interesting assessment tool from our perspective is that the CT Test developed by Marcos Román Gonzalez and adopted by two papers involving Román-Gonzalez himself. To the extent of our review, this test can go beyond the label of a self-developed test, because it has passed a rigorous validation process, involving content validity [14], criterion validity (Román-González, Pérez-González, & Jiménez -Fernández, 2017) and convergent validity [14].

The CT Test consists of 28 multiple choice items, executed on Google Forms technology, and it recalls an OCSE-PISA test as each item presents an issue that may be solved activating higher level cognitive processes. during this case, the respondent needs to activate to a greater or lesser extent the four main cognitive processes associated with CT: decomposition, pattern recognition, abstraction and algorithmic design. For this reason, we consider it to not be such as a mere CT Concepts Test.

Finally, we mention an Indonesia research which adopts a selected approach. CT in 5th grade students is measured by a self-perception scale inside a wider test, called Computational Thinking Level Scale (CTLs), first conceived to address educational activity students [15] and so adapted for secondary school students [16]. The dimensions consist of twenty-two items (five points Likert type), grouped in 5 factors. The authors of the size seem to trace CT back to a broader problem-solving process [17], as only one of these factors explicitly refers to computation (algorithmic thinking), while the others correspond to more general cognitive processes (creativity, critical thinking, cooperation, problem solving). However, they pay much attention to metric issues, and therefore the scales appear to be transferable and reliable.

In conclusion, checking out a unified definition of CT continues to be a challenge; while many papers outline it as a controversy Solving thought process, they often restrict it to the CT Concepts while designing research. When that specialize in the Assessment Tools, the

scenario appears to be even worse, as there's no validated tool commonly adopted addressing CT at a deeper level. Within the papers we analyzed, the sole convincing solution we could find is that the scenario-based or problem-based one, because the answerer should activate the everyday cognitive processes usually adopted by computer scientists when designing software, even if, ultimately, they failed to ignore the CT Concepts, and therefore the Román-Gonzalez CT Test is currently the sole one adopting this approach that's also transferable and validated.

#### **4. DISCUSSION**

The findings of the aforementioned categories are discussed below so as for this trend on CT to be explained and future studies in primary education to be followed. Moreover, the discussion followed by some recommendations. According to the research gaps emerged within the literature. Finally, the restrictions of this study also are discussed. From the included papers broke down in this audit, the accompanying discoveries with respect to CT's consolidation in the essential level of instruction are of centrality.

It is apparent that all through the most recent five years, there is an expanding enthusiasm of the logical instructive network towards the advancement of CT-aptitudes of elementary school understudies. Subsequently, we could accept that reviews concentrating on CT cultivation of elementary school understudies will increment in the years ahead.

Majority of the examinations center around STEM disciplines and explicitly regarding the matter of CS, apply autonomy and science. Along these lines, understudies' CT-abilities development through different controls past STEM field stays a test.

Most examinations utilize the PC as a method for CT-development. Accordingly, unplugged exercises (e.g., Bibras undertakings, puzzle solving exercises) are utilized less significantly and exploration information from such investigations are restricted.

Majority of the examinations use programming as a setting for CT-improvement. Obviously, writing computer programs is a structure of CS and CT practice, and accordingly, the selective spotlight on programming for the advancement of CT isn't just an academic yet in addition a methodological blunder, since the emphasis ought to be on more elevated level ideas that ought to be educated, just as on different psychological areas/controls to which they ought to be applied [18]

Most examinations endeavor to develop understudies CT-skills through differing exercises that require the utilization of visual programming dialects with Scratch, Alice and Kodu being the most widely recognized ones. It is likewise seen that the majority of the investigations have been applied to the understudies going to the upper

evaluations of rudimentary school. In this manner, more accentuation ought to be given to understudies going to bring down evaluations even to Kindergarten understudies, so that we have a more thorough view on how CT could be coordinated in essential instruction as a rule.

Majority of the investigations center around understudies' CT-abilities improvement by featuring and looking at their ancient rarities, without, notwithstanding, examining the impact of explicit free factors –, for example, the impact of past programming experience, sexual orientation, age, pair programming – on the development of CT sub-aptitudes, rehearses or even viewpoints.

CT-estimation stays a test and an open-finished issue. In this way, the field of CT requires precise assessment methods [19] in order to dependably quantify the various parts of CT that will conquer the appraisal of basic and nearby programming develops [20]

Most examinations center around CT-abilities development in a momentary period, while neglecting the drawn-out impacts that CT has on understudies' profession. No examinations have been embraced looking at whether the instructing/learning of CT has a direct or aberrant effect on understudies' scholastic presentation, vocation, personnel decision, or even long-haul critical thinking aptitudes [21].

Teaching and learning issues on CT ideas abilities despite everything stay open. Because of the examination field of CT being in its baby arranges, the beneath open-finished questions about CT-instructing and learning defined quite a long while prior by other analysts [6], [22] ought to be replied. The most significant of them respect: (a) the ideas and abilities that understudies can best learn at each evaluation in elementary school, just as (b) the successful sequencing of ideas in training kids as their learning ability develops throughout the long term.

#### **5. CONCLUSION**

The point of this paper was to make accessible a preview of the momentum examination and work around the way that CT has developed in basic level instruction understudies. For this reason, an efficient writing survey was directed from 2010 to 2019 dependent on 53 papers, discovered 'applicable for consideration', subsequent to looking through ten huge electronic information bases by utilizing exact watchwords. To this end, the exploration addresses made in this paper were picked with the expectation to give a diagram to specialists and teachers of how CT could be better installed into the school study hall.

The discoveries uncovered that most of the investigations utilize the system of programming for both stopped and unplugged exercises so as to develop understudies' CT-aptitudes. It was likewise affirmed that

most investigations center around the subject of CS and STEM field all in all.

Obviously, there are different instruments and ways proposed – advanced being the most widely recognized – with the expectation to consolidate CT in instructing practice. It was likewise demonstrated that the most successive CT-ideas/abilities moved toward respect ‘deliberation, calculations and strategies, control structures, testing and confirmation, issue decay and parallelization’. Besides, it ought to be noticed that further work ought to be done on the production of more substantial and solid assessment apparatuses in which CT can be evaluated.

It is trusted that this writing survey be helpful for future approach producers, educational program creators, analysts and educators who want to execute CT into the showing practice, just as structure a premise and an agreement on how CT could be instructed in essential training. It would add to the compelling educators’ training by furnishing them with essential data about CT and giving explicit rules and directions on the most proficient method to instruct CT to the perfect understudies at the perfect age level.

## REFERENCES

- [1] S. Papert, “Mindstorms,” New York: Basic Books, vol. 607, 1980.
- [2] T. Dufva and M. Dufva, “Metaphors of code—Structuring and broadening the discussion on teaching children to code,” *Thinking Skills and Creativity*, vol. 22, pp. 97–110, 2016.
- [3] J. Janssen, S. Stoyanov, A. Ferrari, Y. Punie, K. Pannekeet, and P. Sloep, “Experts’ views on digital competence: Commonalities and differences,” *Computers & Education*, vol. 68, pp. 473–481, 2013.
- [4] K. Brennan and 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, p. 25.
- [5] N. D. Anderson, “A call for computational thinking in undergraduate psychology,” *Psychology Learning & Teaching*, vol. 15, no. 3, pp. 226–234, 2016.
- [6] V. Barr and C. Stephenson, “Bringing computational thinking to K-12: what is involved and what is the role of the computer science education community?,” *Acm Inroads*, vol. 2, no. 1, pp. 48–54, 2011.
- [7] S. Grover and R. Pea, “Computational thinking in K–12: A review of the state of the field,” *Educational researcher*, vol. 42, no. 1, pp. 38–43, 2013.
- [8] R. Trincherò, “Problem solving e pensiero computazionale. Costruire sinergie tra concettualizzazione e codifica a partire dalla scuola primaria,” 2019.
- [9] D. E. Knuth, “Computer programming as an art,” in *ACM Turing award lectures, 2007*, p. 1974.
- [10] G. Olimpo, “Dal mestiere dell’informatico al pensiero computazionale,” *Italian Journal of Educational Technology*, vol. 25, no. 2, pp. 15–26, 2017.
- [11] L. Cohen, L. Manion, and K. Morrison, *Research methods in education*. routledge, 2013.
- [12] S. Grover, R. Pea, and S. Cooper, “Systems of assessments” for deeper learning of computational thinking in K-12,” in *Proceedings of the 2015 annual meeting of the American educational research association, 2015*, pp. 15–20.
- [13] C. Selby and J. Woollard, “Computational thinking: the developing definition,” 2013.
- [14] M. Román-González, J.-C. Pérez-González, and C. Jiménez-Fernández, “Which cognitive abilities underlie computational thinking? Criterion validity of the Computational Thinking Test,” *Computers in Human Behavior*, vol. 72, pp. 678–691, 2017.
- [15] Ö. Korkmaz, R. Çakir, and M. Y. Özden, “A validity and reliability study of the computational thinking scales (CTS),” *Computers in Human Behavior*, vol. 72, pp. 558–569, 2017.
- [16] Ö. Korkmaz, R. Çakir, and M. Y. Özden, “Computational thinking levels scale (CTLS) adaptation for secondary school level,” *Gazi Journal of Educational Science*, vol. 1, no. 2, 2016.
- [17] Ö. Korkmaz and X. Bai, “Adapting Computational Thinking Scale (CTS) for Chinese High School Students and Their Thinking Scale Skills Level,” *Online Submission*, vol. 6, no. 1, pp. 10–26, 2019.
- [18] J. Voogt, P. Fisser, J. Good, P. Mishra, and A. Yadav, “Computational thinking in compulsory education: Towards an agenda for research and practice,” *Education and Information Technologies*, vol. 20, no. 4, pp. 715–728, 2015.
- [19] I. Lee et al., “Computational thinking for youth in practice,” *Acm Inroads*, vol. 2, no. 1, pp. 32–37, 2011.
- [20] M. Armoni, *COMPUTING IN SCHOOLS: Computer Science, Computational Thinking, Programming, Coding: The Anomalies of Transitivity in K-12 Computer Science Education*. *ACM Inroads* 7, 4 (Nov. 2016), 24–27. 2016.
- [21] J. Lockwood and A. Mooney, “Computational Thinking in Education: Where does it fit? A systematic literary review,” *arXiv preprint arXiv:1703.07659*, 2017.
- [22] J. M. Wing, “Computational thinking,” *Communications of the ACM*, vol. 49, no. 3, pp. 33–35, 2006.
- [23] Wiyono, B. B., Kusumaningrum, D. E., Triwiyanto, T., Sumarsono, R. B., Valdez, A. V., & Gunawan, I. (2019, October). *The Comparative Analysis of Using Communication Technology and Direct Techniques in Building School Public Relation*. In *2019 5th International Conference on Education and Technology (ICET)* (pp. 81-86). IEEE.
- [24] Gunawan, I., Triwiyanto, T., & Kusumaningrum, D. E. (2018, December). *The Mentoring of Textbooks Writing for the Facilitators of Junior High School Red Cross*. In *International Conference on Education and Technology (ICET 2018)*. Atlantis Press.