An Overview of the Teaching and Learning Process Basic Programming in Algorithm and Programming Courses

Wilda Susanti1*, Jalius Jama2, Krismadinata3, Dachi Ramadhani4, Torkis Nasution5

1Institut Bisnis dan Teknologi Pelita Indonesia, Indonesia
2Universitas Negeri Padang, Indonesia
3Universitas Negeri Padang, Indonesia
4IKIP PGRI Pontianak, Indonesia
5STMIK Amik Riau, Indonesia

*wilda@lecturer.pelitaindonesia.ac.id

Article History: Received: 11 January 2021; Accepted: 27 February 2021; Published online: 5 April 2021

Abstract: In this paper, we review the literature related to computer programming learning, where Algorithms and Programming are the topic domains of the Informatics and Computer science clusters. There are 4 competencies in learning outcomes, such as: 1) understand algorithmic concepts; 2) master algorithm concepts and principles; 3) master programming language concepts; and 4) master programming languages and algorithms. The main focus of this review is on beginner programming and topics related to student difficulties in learning programming. Various problems experienced by beginners were identified from the literature to some of the solutions offered by researchers.

Keywords: Programming, Student difficulty, Competence

1. Introduction

IT (Information Technology) graduates are still the prima donna. The field of information technology or other businesses that are supported by the application of information technology at present and in the future remains the government’s concern. In 2020, the number of higher education graduates in Indonesia is around 6 million people per year, assuming 7 percent of students take the IT discipline (Tutang, 2009). Ironically, from the data of the Informatics and Computer Education Association (Aptikom) IT graduates in Indonesia only 10 percent is absorbed by the industry of the 25,000 IT graduates in Indonesia. Another case in India, 25 percent of IT workforce graduates are absorbed by the workforce of 3 million IT workers (Nair, 2020). The gap between universities and the industrial world is due to the unfulfilled competencies of graduates (Muchlis et al., 2020).

Education should be oriented to the world of work, the emphasis is not solely on cognitive aspects, but other personality aspects such as affective and psychomotor aspects are needed (Muhson et al., 2012). To face this challenge, higher education institutions must prepare an innovative learning system, and improve the competence of graduates who have 21st century skills (Learning and Innovation Skills), namely 4C skills (Critical thinking, Creativity, Collaboration, and Communication (Muhson et al., 2012; Bakar et al., 2013; Cradler, J., McNabb, M., Freeman, M., and Burchett, 2002; Verawardina et al., 2020; Nouri et al., 2020; Trilling & Fadel, 2010).

KKNI APTIKOM (2015) establishes the standard of learning outcomes (Learning Outcomes) for subjects related to the subject area in the field of Computer Science / Informatics at the undergraduate level that Algorithm and Programming with reference to learning outcomes based on the KKNI APTIKOM which aims to meet the qualifications of Bachelor of Computer graduates in the S1 Information Engineering Program. Competence Algorithms and Programming are specifically defined and relevant to competencies based on the KKNI APTIKOM. Thus Programming Algorithms include aspects of the topic domain of the subjects in the S1 Informatics Engineering study program with related subjects, namely, Programming Basics, Data Structures and Algorithms, Algorithm Design and Analysis, Declarative Programming, Automata Language Theory, Intelligent Systems, Object Oriented Programming, and Web Programming. To achieve graduate competence, the core learning outcomes of the study program are formulated which refers to learning outcomes. The core learning outcomes in the field of Information Technology / Computer Science are grouped into six competency domains, namely (1) Mathematics (2) Basic Computer Science; (3) Algorithms and Programming; (4) Software Engineering; (5) Computer Systems; and (6) Life Skills (Success Skills).

Table 1. Learning outcomes of the Information Engineering / Computer Science Study Program to meet the qualifications of Bachelor graduates according to the KKNI level by referring to the learning outcomes recommended by APTIKOM

<table>
<thead>
<tr>
<th>Topic Domain</th>
<th>Learning Outcomes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Algorithms and Programming</td>
<td>1. Understand the concepts of algorithms and complexity, covering the central concepts and skills needed to design, implement and analyze algorithms</td>
</tr>
</tbody>
</table>
2. Mastering the concepts and principles of algorithms and computer science theory that can be used in computer-based system modeling and design
3. Mastering programming language concepts, and being able to compare various solutions and various programming language models
4. Mastering programming languages and algorithms related to application programs to manipulate image, graphic and image models

Algorithms and Programming are the first programming courses given to students. This subject requires a set of cognitive processes that naturally develop through practice, writing solutions with algorithms (Francisco & Ambrosio, 2015). Programming is part of the curriculum in computer science education and is a basic skill that all computer science students should learn. Teaching programming languages aims to enable students to develop a set of skills needed to design computer programs and systems capable of solving real problems (Gomes et al., 2008). Learning programming requires students to increase creativity, teamwork, innovation and knowledge of data structures and algorithms (Nair, 2020). Computer programming skills require several types of thinking skills such as logical thinking and problem solving (Ozyurt, 2015). Developing a computer program requires the ability to translate and model one's thinking, problems and solutions in natural language into the selected programming language (Renumol et al., 2010). Computer programming requires problem-solving strategies and involves a lot of programming logic activities that pose challenges for students (Wang & Hwang, 2017).

Based on a survey between institutions around the world on research (Bennedsen & Caspersen, 2007) in 15 countries, in 63 institutions found 33% failure rate to learn computer programming or pass 67%. This research is strengthened by (Watson & Li, 2014) conducting a survey for quantitative evidence that has been done by (Bennedsen & Caspersen, 2007) whether the failure rate in computer programming increases or decreases over time. With data from 14 countries in 51 institutions, the failure rate was 32.3% or a graduation rate of 67.7%.
The educational practice of each country will be different, further (Watson & Li, 2014) giving the average percentage of students who do not pass computer programming by country, as seen in Figure 2. Portugal has the highest failure rate followed by Germany and Brazil. Meanwhile, Indonesia is ranked 7th out of 15 countries.

Figure 2. Students who do not pass by country (Watson & Li, 2014)

In Figure 3, the selection of programming languages at the student graduation level according to (Watson & Li, 2014) the first rank is seen in C language and followed by C++. Furthermore (Bennedsen & Caspersen, 2019) conducted another study on the failure rate of learning programming where previous research found that the student failure rate was an average of 33% in 2007 and in 2018 the average failure rate was 28%. Not surprisingly, learning to program can be such a difficult task, to the point where the failure rate phase in programming classes is almost the same (Bennedsen & Caspersen, 2007).

Figure 3. Students who do not pass are grouped according to programming language (Watson & Li, 2014)

From the literature, collected on the problem of programming failure is a programming language commonly used in programming classes such as C, C++, C# and Java which has a broad and complex syntax, making learning difficult for beginners (Gomes et al., 2008; O’Kelly & Gibson, 2006; Robins et al., 2003; Milne & Rowe, 2002; Watson & Li, 2014; Bravo et al., 2005). The number of students who fail when starting computer programming learning because of difficulty understanding: basic concepts, problem solving skills, by identifying problems, developing algorithms and coding algorithms with programming languages (Milne & Rowe, 2002; Jenkins, 2002; Esteves et al., 2009; Lahtinen et al., 2005; Susanti et al., 2020). Ineffective problem solving strategies (Febrian & Lawanto, 2018; Whittington, 2004). Learners also experience learning barriers because they have the wrong perspective about computers (Ben-Ari, 1998; Brennan, K., & Resnick, 2012; Lischner, 2001) or face an unwanted learning environment (for example, lack of human interaction)(Ben-Ari, 2001). Some
of the misconceptions include misconceptions related to initialization of variables, loops, conditions, pointers and recursion (Milne & Rowe, 2002).

The results of the study (Khaleel et al., 2017) show that the low value of learning programming is because students experience ineffective learning, lack of interest and lack of motivation. Another most important problem faced by students in learning programming is in the practicum section, which involves their need to practice extensively to achieve higher programming skills (Wei, 2010).

Researchers from the literature review have done a lot of research on strategies to help students learn computer programming which is a challenging problem of how to improve beginners' understanding and programming skills. Cognitive theory is offered to answer the question: why do so many students fail to learn programming. These include difficulty understanding program objectives and their relationship to computers, difficulty understanding the syntax and semantics of a particular programming language (Robins et al., 2003). Misunderstanding of programming constructs (Lane & VanLehn, 2003), inability to solve problems (McCracken et al., 2001) and inability to read and understand program code (Lister et al., 2004).

Two cognitive factors that make learning to program difficult are learning styles and motivation (Milne & Glenn, 2002; Tandon & Ravikumar, 2013). Traditional teaching methods, usually based on lectures and specific programming language syntax, fail to motivate students to engage in programming (Berlin & Bennedsen, 2006; Lahtinen et al., 2005; Esteves et al., 2007). Therefore, it is important to incorporate concept knowledge and strategies for its use in the learning process (Lahtinen et al., 2005).

2. 3. Programming Learning Solutions

Teaching programming languages aims to make students develop a set of skills needed to design computer programs and systems capable of solving real problems. The 36 literature reviewed showed eight failure rates in computer programming. The first is not applying the correct algorithm (4 articles). Second, programming is difficult (14 articles). Third, ineffective problem-solving strategy skills (18 articles). Four, the inability to solve the problem (6 articles). Five, wrong perspective about computers (3 articles). Six, lack of human interaction (2 articles). Seven, learning styles and motivation (4 articles). Eight, lecturer-centered learning (3 articles). Figure 1 shows the percentage of literature review of students' failure rates in computer programming.

<table>
<thead>
<tr>
<th>Failure Reason</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Not implementing the correct algorithm</td>
<td>7%</td>
</tr>
<tr>
<td>Programming is difficult</td>
<td>26%</td>
</tr>
<tr>
<td>Ineffective solving strategy skills</td>
<td>33%</td>
</tr>
<tr>
<td>Inability to solve problems</td>
<td>11%</td>
</tr>
<tr>
<td>Lack of human interaction</td>
<td>4%</td>
</tr>
<tr>
<td>Wrong perspectives about computers</td>
<td>6%</td>
</tr>
<tr>
<td>Learning styles and motivation</td>
<td>7%</td>
</tr>
<tr>
<td>Lecturer-centered learning</td>
<td>6%</td>
</tr>
<tr>
<td>Lack of human interaction</td>
<td>4%</td>
</tr>
</tbody>
</table>

Figure 1. Percentage of failure in programming

Literature on computer programming failure, from studies there are states that the same problem is obtained so that the percentage is as above.

To be able to compete, students must have cognitive skills, ability to solve complex problems, have attitudes and motivation (Tsai & Tsai, 2018). The solution to dealing with problems that will become a trend in the 21st century is to combine mastery of knowledge with technology (Voskoglou & Buckley, 2012).
Many researchers have attempted to overcome this problem by developing various computer-based instructional tools and improving existing learning methods (O’Kelly & Gibson, 2006; Olapiriyakul & Scher, 2006; Adams, 2007; Febrian & Lawanto, 2018). Designing metacognitive related activities can be done through the use of technology that is integrated into educational activities. Designing metacognitive-related activities that focus on social and cognitive development is a theoretical and practical challenge, especially in supporting teaching and learning computer programming (Rum & Ismail, 2017).

4. METODOLOGY

The results of the search for scientific sources such as international journals found solutions that have been done by previous researchers to answer student failures in computer programming. Appropriate samples were collected and analyzed. The following are the systematic steps taken in conducting a literature study as shown in Figure 2.

![Flowchart of systematic literature review](image)

After selecting and screening, based on the criteria for articles that meet the requirements for literature review, the website database page. A total of 20 articles were used as a source of review reading as a strategy to overcome the failure of basic programming learning.

5. RESULT AND DISCUSSION

The results of the literature review explain several programming learning strategies that have been carried out by several researchers along with the results of these studies. The solution offered is learning programming with collaborative, problem-based programming with PBL or PJBL and an approach with computational thinking. More details can be seen in Table 2.

<table>
<thead>
<tr>
<th>No</th>
<th>Author</th>
<th>Research Solutions</th>
<th>Result</th>
<th>Publisher/Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>(Voskoglo u &amp; Buckley, 2012)</td>
<td>Problem solving-based learning with a computational thinking approach to synthesizing critical thinking and existing knowledge.</td>
<td>Identifying students’ critical thinking</td>
<td>Egyptian Computer Science Journal, ECS, Vol.36 No.4, September 2012</td>
</tr>
<tr>
<td>2.</td>
<td>(Esteves et al., 2011)</td>
<td>Computer learning and programming is developed in cyberspace. Observations are focused on 1) how students and teachers interact 2) virtual classroom activities 3) use of interfaces 4) challenges and constraints</td>
<td>Identify problems that hinder teacher intervention in cyberspace and detect solutions to</td>
<td>British Journal of Educational Technology, doi:10.1111/j.1467-8535.2010.01056.x</td>
</tr>
</tbody>
</table>
An Overview Of The Teaching And Learning Process Basic Programming In Algorithm And Programming Courses

<table>
<thead>
<tr>
<th></th>
<th>Study (Year)</th>
<th>Description</th>
<th>Enhanced Learning Experience</th>
<th>Conference/Ref</th>
<th>Citation</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>(Esteves et al., 2007)</td>
<td>Use a collaborative virtual environment to increase student effectiveness and motivation</td>
<td>Enhances the learning experience for computer programming students</td>
<td>International conference on multimedia and information and communication technologies in education</td>
<td>Education Tech Research Dev DOI 10.1007/s11423-017-9551-0 Springer</td>
</tr>
<tr>
<td>4</td>
<td>(Wang &amp; Hwang, 2017)</td>
<td>Problem posing based collaborative learning strategy using the C# language. Experimental group trial data of 25 students were taught using collaborative problem solving strategies and 28 students as a conventional control group.</td>
<td>Improve learning achievement and programming skills</td>
<td>Education Inquiry Volume 11, 2020 - Issue 1</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>(Nouri et al., 2020)</td>
<td>Skills development related to computational thinking concepts</td>
<td>Cognitive skills, language skills, creative problem-solving skills and attitude and collaborative attitude skills</td>
<td>Education International Conference on Multimedia and Information and Communication Technologies in Education</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>(Febrian &amp; Lawanto, 2018)</td>
<td>Identifying and investigating programming task understanding skills</td>
<td>The thought process of the participants</td>
<td>International Education Studies; Vol. 11, No. 12; 2018</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>(Saltan, 2016)</td>
<td>Perform online algorithmic visualization as an instructional approach</td>
<td>Knowing the effect of online algorithmic visualization on student achievement</td>
<td>Journal of Education and Learning; Vol. 6, No. 1; 2017</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>(Renumol et al., 2010)</td>
<td>Identify students’ cognitive processes. Eight cognitive processes identified 1) confusion 2) hypothesis 3) interrogation 4) repetition 5) monitoring 6) memory 7) relapse 8) translation</td>
<td>Make programming teachers aware of the cognitive difficulties of programming and the importance of the programming teaching process</td>
<td>ACM Transactions on Computing Education</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>(Esteves et al., 2009)</td>
<td>Learning approach with Second life is a problem-based 3-dimensional online virtual world</td>
<td>Observe and reflect on the problems that arise</td>
<td>Journal of Virtual Worlds Research</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>(Ideris et al., 2019)</td>
<td>Students work in groups to solve programming problems using the Scratch software as a teaching aid</td>
<td>Improve student scores on tests and higher order thinking skills</td>
<td>Konferensi ASEAN ke-4 tentang Psikologi, Konseling, dan Humaniora (ACPCH 2018)</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>(Serrano-Cámara et al., 2016)</td>
<td>Uses a mobile collaboration tool called MoCAS</td>
<td>For motivation</td>
<td>International Journal of Engineering Education</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>(Bravo et al., 2005)</td>
<td>Using animation and simulation programs in computer-supported collaborative learning</td>
<td>Provides educational tools to support teaching and learning of computer programming</td>
<td>Jurnal Ilmu Komputer Universal</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>(Chen, 2021)</td>
<td>Problem-based learning with a focus on...</td>
<td>Students’ ability</td>
<td>Advances in Social</td>
<td></td>
</tr>
</tbody>
</table>

---

2939
<table>
<thead>
<tr>
<th>Year</th>
<th>Authors</th>
<th>Methodology</th>
<th>Findings/Conclusion</th>
<th>Journal/Publication</th>
</tr>
</thead>
<tbody>
<tr>
<td>2018</td>
<td>Wilda Susanti, Jalius Jama, Krismadinata, Dochi Ramadhani, Torkis Nasution</td>
<td>Computational thinking approach to programming languages.</td>
<td>To think computationally in solving problems</td>
<td>Science, Education and Humanities Research (ASSEHR), volume 156. Atlantis Press</td>
</tr>
<tr>
<td>2018</td>
<td>Othman et al., 2013</td>
<td>PBL requires students to work collaboratively</td>
<td>Investigate student interests, learning styles and student learning preferences.</td>
<td>International Conference on University Learning and Teaching, Science Direct</td>
</tr>
<tr>
<td>2017</td>
<td>Da Rodrigues et al., 2017</td>
<td>Designing a prototype model for online collaborative learning systems in a virtual environment. The technique used is the collaborative learning technique “Think-Pair-Share”</td>
<td>The results indicate the need to develop online small group discussions</td>
<td>Proceding</td>
</tr>
<tr>
<td>2019</td>
<td>Peng, Yuan, et al., 2019</td>
<td>Doing collaborative learning practice computer programming</td>
<td>Students are more motivated by sharing</td>
<td>Jurnal Teknologi Pendidikan Australasia</td>
</tr>
<tr>
<td>2019</td>
<td>Peng, Wang, et al., 2019</td>
<td>Collaborative learning supported by MoCAS. Evaluation of motivation on 139 students</td>
<td>CIF’s collaborative instructional approach and MoCAS tools</td>
<td>Int. J. Smart Technology and Learning</td>
</tr>
<tr>
<td>2006</td>
<td>O’Kelly &amp; Gibson, 2006</td>
<td>Propose a visualization-based and progressive learning environment as cognitive tools to support collaborative learning to support PjBL programming.</td>
<td>Incorporating a simple approach to complex visualization-based cognitive tools is more effective in improving student programming performance</td>
<td>ITICSE’06, June 26–28, 2006</td>
</tr>
<tr>
<td>2015</td>
<td>Lovos, 2015</td>
<td>Learning programming by integrating technology</td>
<td>Critical review of how emerging technologies have been integrated in programming learning</td>
<td>Revista Internacional de Ciencia, Matemáticas y Tecnología</td>
</tr>
</tbody>
</table>

A review of 20 literature studies with problem-based, collaborative and computational thinking approaches

Applying technology in programming learning has become hot during these 40 years (Douce et al., 2005). The programming problem-solving process forces a person to think about what problems and what needs to be done to solve the problem and then various program functions and procedures are identified in the form of inputs and outputs, so solving problems is a key component for programming activities so that learning programming by embracing PBL / PjBL as one of the instructional approaches. Problem-based programming learning involves various aspects of programming knowledge, strategies, and problem-solving processes. Incorporating collaborative work in programming courses has been identified as a potential strategy for maximizing student participation and having a positive impact on learning. However, social interaction in collaborative learning does not happen automatically. Appropriate guiding strategies and supporting tools for collaborative learning are indispensable. Practice-based problem solving strategies to support collaborative learning activities in computer programming practice courses.

A literature review also shows that collaborative programming learning can increase student motivation (Esteves et al., 2007). The research supports that collaboration is an effective educational feature for programming (McDowell et al. 2002; Esteves et al. 2009). According to Roe and Queensland (2008), in their
research, there was a change in student perceptions about collaborative learning, such as: 1) a more pleasant environment; 2) more confident with a peer environment; and 3) increased skills. W. W. M (2009) also stated that collaborative learning supported by computers can improve thinking skills, social interaction, criticality and creativity. Computational thinking involves solving problems, designing systems, and understanding human behavior based on computer science concepts (Grover & Pea, 2013). Understand the cognitive processes underlying collaborative (A. Soller et al., 2004).

Computational thinking is one of the programming learning approaches to train students to solve problems to be able to think in a structured, logical, and algorithmic manner (Yeni Anistyasari, Ekohariadi, 2020). Students' skills in programming computers are in line with computational thinking which is a skill to learn and to think in a structured, abstract, algorithmic and logical manner, and to be ready to solve complex and open problems (Yasin, 2020). Computational thinking is in accordance with 21st century skills in solving problems, designing systems, and understanding human behavior by drawing on the basic concepts of computer science (Nouri et al., 2020).

6. Conclusion

In this literature review, the failure rates in computer programming are presented and the factors that cause the failure rates of students to learn programming and the solutions that have been offered by several researchers. Based on the study: 1) good knowledge of problem solving skills; 2) knowing the syntax and semantics of programming languages; 3) being able to understand existing code; 4) the ability to analyze problems; 5) being able to compile solutions; and 6) being able to express the program into a programming language and being able to test it.

References


16. Esteves, M., Morgado, L., Martins, P., Fonseca, B., Esteves, M., Morgado, L., Martins, P., & Fonseca, B. (2007). The use of Collaborative Virtual Environments to provide student’s contextualisation in programming To cite this version: INTERNATIONAL CONFERENCE ON MULTIMEDIA AND INFORMATION AND COMM- student’s contextualisation in programming


61. Wei, X. (2010). Research of Practical Course Teaching of JAVA Language Abstract-At present , many institutions of higher learning set up JAVA course , how to better integrate the characteristics of the course with students ’ hands-on practical ability , and to accumulate, 10–12.
