

The Problem-Based Learning Procedural Model in the Software Modeling Course at the Information Technology College in Indonesia

Bahar^{1,2,*}, Basuki Wibawa¹, Robinson Situmorang¹

¹State University of Jakarta, Indonesia

²STMIK Banjarbaru, Indonesia

Received July 9, 2019; Revised September 9, 2019; Accepted September 16, 2019

Copyright©2019 by authors, all rights reserved. Authors agree that this article remains permanently open access under the terms of the Creative Commons Attribution License 4.0 International License

Abstract The characteristic of the Direct-Instruction system, which depends on the ability of the teacher's reflection, only provides little opportunity for students to be actively involved in the learning process. This process contrasts with the characteristics of the Software Modeling course which ideally emphasizes student-centered learning. The impact of the learning process becomes ineffective, and students cannot reach the minimum competency standards planned in the learning design. We propose a model of integrating Problem-Based Learning (PBL) into the Learning Modeling Software design on three elements of the learning system namely Elementary Curriculum, emphasizing the use of problems as the starting point of student learning; Element Group, emphasizing collaboration systems (group discussion based learning); and Student Elements, emphasizing the Student-Directed-Learning (SDL) system. We tested the effectiveness of the PBL model in small group trials in learning situations that resembled the actual situation and found PBL effectively increased student mastery of a particular topic. However, further discussions regarding 1) how many guidelines are needed in PBL; 2) potential student experience confusion if there is not enough initial learning framework; 3) students who do not have adequate initial knowledge and tend to experience underdevelopment, are still needed to find PBL models that are genuinely effective in learning the field of Software Engineering.

Keywords Procedural Model, Problem-Based Learning, Software Modeling

considerable attention at the College of Computer Science. Software engineering is a discipline related to the application of theory, knowledge, and practice to effectively and efficiently build reliable software systems that meet the requirements of customers and users. IEEE and ACM put Software Engineering into one of the main fields of study in their curriculum (IEEE Computer Society, 2013), and the curriculum has become a reference for many universities throughout the world in compiling computer science curricula.

Software Engineering has several characteristics that must be harmonized with teaching methods. Software Engineering is a field of science whose technology continues to change due to several factors. First, the software requires compatibility with other hardware and software. Progress in hardware technology, additional software, changes in the company's business strategy, and changes in end-user behavior demand alignment of software requirements. This condition requires high adaptation skills in the software development process (Juman, 2018). Second, the large volume of code (program) and the complexity of the software system architecture led to small software development systems leaving individually and turning to technology development collectively and collaboratively. Third, software design and software programming technology also develop dynamically. Various development models and software programming languages encourage the emergence of new references in the software development environment, thus demanding adaptation to lifelong learning (Bollin, Hochmüller & Mittermeir, 2011; Kazimov, 2017).

Software Engineering applies a systematic approach to development, operation, and maintenance. Thus, software developers do need not only technical skills but also social skills in their work. Traditional teaching methods that focus on lectures and tutorials for Software Engineering

1. Introduction

Software Engineering Education (SE) has received

students are not enough to develop real-world problem-solving skills. In traditional learning approaches, most students must complete their assignments, and this is contrary to professional practice in a collaborative software development team (Krusche, 2017; Oliveira, 2013). Goel (2011) and Fertalj (2013) suggest that the importance of providing students with real problems and a real teamwork environment must be a concern in software learning.

According to Garg and Varma (2015), software engineering requires an effective and sustainable learning environment, an authentic learning climate, independent learning, learning from failure and success, and motivation of students.

The characteristics of the software require a learning approach that emphasizes active learning (Freeman et al., 2014; Jensen, Kummer, & Godoy, 2015), the concept of lifelong learning, problem-solving in the real world, unstructured problems, and skills collaboration (Karabulut-Ilgu, Jaramillo Cherez, & Jahren, 2018). These conditions are contrary to the learning system that is widely used in learning Software Engineering in Higher Education, namely the direct learning model (direct instruction).

Problem-Based Learning (PBL) is a constructivist-based learning model, namely learning initiated by the problems posed, questions, and problems that students want to solve. The use of complex real-world issues aims to motivate students to identify and examine the concepts and principles they need to know from work through this problem. Students work independently and in small work teams. They bring together collective skills in acquiring, communicating, and integrating information (Duch, Groh, & Allen, 2001).

The PBL is a learning strategy that encourages students to know how to learn and work together in groups to find solutions to problems in real situations. PBL makes students think critically and analytically to get and use scientific literacy appropriately (Ardianto, 2016). PBL correlates with cognitive functions that contain various types of creative thoughts and actions in the learning phase (Nuswowati, Susilarningsih, Ramlawati, & Kadarwati, 2017), including the use of existing knowledge, reorganizing new experience in cognitive structure, analysis and synthesis, structuring and developing ideas, and solving problems (Pierrakos, Zilberberg, & Anderson, 2010).

The application of problem-based learning through web 2.0 technology can simplify the movement of students individually to find learning resources, improve skills in collaborating in solving problems through flexible communication facilities, and access to shared resources (bookmarks, websites, articles, materials, etc.). PBL-based Web 2.0 also supports collaboration between students and teachers and offers connections to various external resources (Glud, Buus, Ryberg, Georgsen, & Davidsen, 2010; Tambouris et al., 2012).

This paper proposes a conceptual model of PBL

implemented in the two phases of the life cycle of Software Engineering which Pressman (2002) calls "the System Modeling phase."

2. Literature Review

There are many studies on software modeling learning methods. Tanner & Scott (2015) introduced the flipped classroom approach used in teaching system analysis and design using the Unified Modeling Language (UML) at the University of South Africa. The aim is to create a learning environment that is more student-centered to encourage class discussion and debate, which in turn serves to train students' critical thinking skills. Students not only have theoretical understandings of UML and design concepts but can also apply these concepts at the beginning of their learning. This finding concludes that the flipped classroom approach can improve students' understanding and ability to apply theoretical concepts and focus on solving real-world problems that are integrated into case studies.

The flipped classroom approach creates a learning environment that is more student-centered (Wibawa, Kardipah, 2018), increases student involvement, and encourages students' critical thinking (Moravec, Williams, Aguilar-Roca & O'Dowd, 2010). The flipped classroom approach also allows students to learn about theoretical concepts outside classroom settings and apply these concepts in the classroom while getting help from facilitators and other students (Water-Perez, Dong, 2012). However, the flipped classroom approach is oriented towards class meetings and is fully controlled by the teacher, so students are not free to apply their knowledge to solve more complex problems than those specified in training. Naturally, the concept of modeling should be carried out in various ways and often focuses on modeling options and assessing their validity. This situation seems to confuse some students who then break away from discussion because they feel they are not free to be creative.

Paez (2017) researched the "Universidad Nacional de Tres de Febrero," Argentina, using a flipped classroom approach and combined it with other non-traditional teaching techniques in software learning. The teacher designs relevant teaching materials and plans activities in the classroom and outside the school with the help of virtual classes. Virtual classes are used to share files and to expand interactions between teachers and students outside class meetings. Additional teaching strategies involved in this approach are a sustainable practice, teaching by example, and the use of real-world concepts. The primary pedagogy used in this study is a flipped classroom. This approach proved useful in learning Software Engineering. However, this approach does not emphasize developing teamwork skills.

Fonseca and Gomez (2017) apply problem-based learning (PBL) to software engineering courses at two different universities in Chile. The PBL application allows

students to gain more in-depth knowledge and use it in practical ways to follow work plans. Within this framework, students are given different project roles in real projects and can work in dynamic environments using PBL. Although this learning framework provides importance of developing independent learning skills (development of creativity and teamwork), this method is bad at developing communication skills.

Fakhriah (2014) introduced the PBL model as an effort to develop critical thinking skills for students in the learning process. The PBL application begins with the orientation of the problem at the initial lecture meeting in the class guided by the instructor. Next, students form small groups and make field observations related to specific learning themes. Students in groups formulate the problems they encounter in the field observation process, and then determine the right solution to the problem. At the end of the learning session, students reflect and conclude the results of the learning activities. This PBL concept emphasizes social skills in collaborative learning, but few train students to become learners of all time by emphasizing individual learning skills to find necessary learning resources outside the classroom.

This paper proposes a conceptual model that implements PBL into the two phases of the life cycle of Software Engineering (analysis phase and design phase) which Pressman (2002) calls "the System Modeling phase." The application of PBL into learning Modeling Software is carried out on three elements in the learning system. First, the Curriculum element emphasizes the use of problems as the starting point of student learning. It delivers conceptual and structured matters related to the discussed issues in each learning session, using a face-to-face model in class (direct instruction) with the Presentation and Brainstorming method. Second, the Student element emphasizes independent learning (Student-Directed Learning). This activity underlines the activity of students independently learning things that are not structural and tangible in the field. 3) Third, Group elements emphasize collaboration. This activity highlights the active participation of students (small groups) in formulating and synthesizing study results independently.

The overall implementation of the three main elements of PBL (problem, independent learning, and group collaboration) at each learning session aims to perfect the Tanner & Scott (2015) model that is oriented towards class meetings that are entirely under teacher's control, the Fakhriah (2014) model that emphasizes small individual learning, and Paez (2017) model that does not emphasize the development of teamwork skills. The PBL concept also aims to perfect the Fonseca, Gomez (2017) model which only develops small-scale communication skills.

3. Methodology

This study uses the Research and Development Method

(R & D), and adapts the stages of Research and Development proposed by Gall at al. (2015). The steps of the study consist of: First, analysis of system requirements involving learning designers, institutional management, and college graduates or the business world as graduate users. They are included in the formulation of competencies in the field of software modeling needed in learning design. Second, development of learning design, following the procedure for developing learning designs proposed by Dick, Carey, and Carey (2015). The development stage consists of two main steps, namely identifying learning needs and developing learning designs. The advanced PBL concept is implemented at the scene of preparing the learning strategy. Third, Formative Evaluation to test the effectiveness of the model.

The effectiveness of the PBL model is tested on one of the learning content modeling software (need assessment). Before being tested, learning content was validated by Software Engineering content experts and Instructional Design experts. Trials are carried out at the formative evaluation stage (field trial). Different forms of test questions (as an instrument to measure the effectiveness of the PBL model in the Software Modeling learning design) were tested on a group of students who programmed Software Engineering courses or Systems Analysis and Design courses in the 3rd year of their lectures. At the end of each trial phase, data analysis and product revisions are carried out based on input obtained from the results of the trial.

4. Results and Findings

Proposed PBL Concept

The application of PBL in the proposed Software Modeling learning design consists of two main phases, namely learning orientation phase and PBL implementation phase. There are ten steps recommended in the PBL syntax, which are tailored to the learning characteristics of Software Modeling, consist of:

1. PBL Orientation
2. Describe the problem & clarify the term
3. Organizing study groups
4. Learn independently
5. Formulate and present problems
6. Designing field investigations
7. Carry out field investigations
8. Small group discussions
9. Making final project documents
10. General discussion forum

Ten PBL steps are implemented in three segments (curriculum, individuals and groups) in the PBL environment, as shown in figure 1.

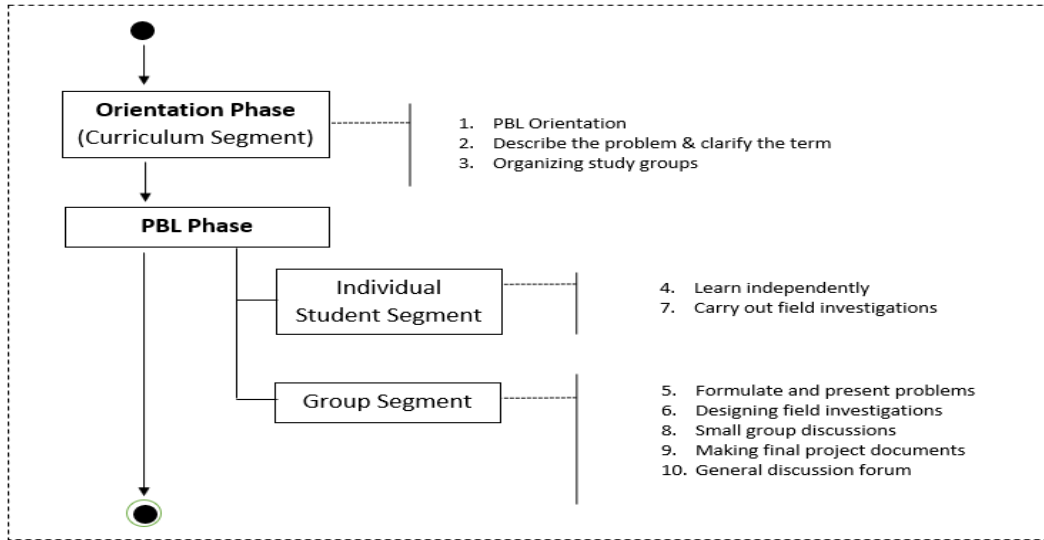


Figure 1. Implementation of PBL Syntax into the Learning Phase

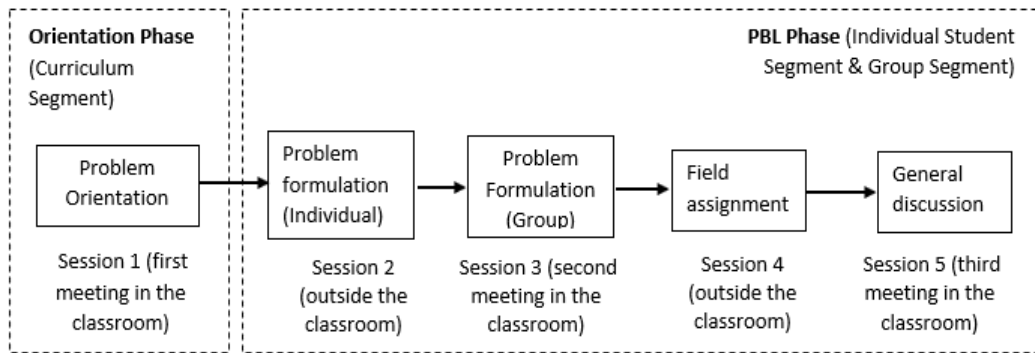


Figure 2. PBL Strategy in the proposed Software Modeling Learning Design

In detail, the Learning Modeling Software strategy in every discussion of a particular theme, presented in Figure 2.

In Figure 2, 1-course topic consists of 5 sessions of activities held in one week. Three activity sessions are held in the classroom involving the interaction of teachers and students. Two meetings are held outside the class that requires students independently or in groups.

Session 1 (Problem Orientation): bring together teachers and students directly in the classroom. There are 3 PBL steps implemented in this session, namely: 1) PBL Orientation (step 1). The instructor explains the learning objectives and achievement plan at the end of the learning session. This session presents PBL mechanisms; 2) Description of the Problem and Clarification of Terms (step 2). The teacher explains the topics or problems discussed in the learning theme. The open discussion is held to clarify basic terms in the learning topic so that students can easily follow learning activities, both individually and in groups; 3) Organizing Learning Groups (step 3). Students form small groups, consisting of 3 to 5 students in one group, and each group prepares to carry out PBL stages, both individually and in groups.

Session 2 (Formulating Individual Problems): is an activity carried out by students individually outside the class. This activity takes place between the 1st meeting and the 2nd meeting. The PBL step that was carried out in the second session was: Self-Study (step 4). Supported by online-based learning media, each student individually conducts a more in-depth study of the fundamental problems described by the Teacher in step 2 using a variety of learning resources.

Session 3 (Formulating Problems in Groups): are activities carried out by students who are members of small groups. This activity was held at the 2nd meeting in the classroom. 3 PBL steps are implemented in the 3rd session, namely: (1) Formulating and Presenting Problems (step 5). Small groups that are formed discuss the formulation of the problem obtained by each group member. Small group discussions produce a problem statement that is ready to be studied further through field assignments. In each little group discussion, group members reflect individually to fill the knowledge gap that has been obtained by each group member. The teacher acts as a facilitator and motivator to motivate students in PBL; (2) Designing Field Investigations (step 6). Small groups prepare a field

investigation strategy related to the problems formulated in step 5. The activities begin independently, and then the results are expressed at small group meetings. Each group prepares a portfolio as an instrument for field investigation.

Session 4 (Field Assignment): is an activity carried out by students individually or in groups. This activity is carried out outside the classroom, between the 2nd meeting and the 3rd meeting. There are 3 PBL steps implemented in the 4th session, namely: 1) Conducting Field Investigations (step 7). Each group member independently conducts field investigations, guided by a prepared portfolio and adapted to the theme of the problem that was carried out in the learning session. Online-based media is available to reach relevant learning resources. Online-based media are used to interact online between fellow small group members, or interact with teachers; 2) Small Group Discussions (step 8). Small group members gather to formulate their findings and discuss to produce the final formulation of their conclusions. Each group member carries out individual reflections, and fills in the gaps in the knowledge gained from each. If group members cannot meet in person, there are online-based media to be used for online discussion; 3) Prepare the Final Project Document (step 9). The formulation of the results of field assignments is stated in the Report on Field Investigation Project, referring to the existing portfolio. Members of the group are accustomed to working collaboratively to complete the Final Project Report. In the end, the small group prepares the presentation material, and then presents it at a public meeting in the classroom.

Session 5 (General Discussion): is an activity carried out by small groups. This activity was held in the classroom at the 3rd meeting. There is the last step PBL implemented in the 5th session, namely: General Discussion Forum (step 10). Small groups present their final findings at a general discussion forum in the classroom facilitated by the instructor. Each member of a little group, practice in

collaboration to show their group's ultimate results. Teachers reinforce each finding submitted by each group. Each group member also makes a final reflection to fill the gaps in knowledge gained by each.

Assessing the Effectiveness of the PBL Model

The effectiveness of the PBL model in Figure 2 was tested on one of the learning content modeling software (need assessment). Before being tested, learning content was validated by Software Engineering content experts and Instructional Design experts. Trials are carried out at the formative evaluation stage (field trial). Various forms of test questions (as an instrument for measuring the effectiveness of applying the PBL model in the Learning Modeling Software design) were tested to a group of students who programmed Software Engineering courses or Systems Analysis and Design courses in the 3rd year of their lectures. The test results are presented in table 1.

There are several problems found in the testing of the application of the PBL model in the Learning Modeling Software, as presented in table 1.

In the first trial of group I respondents, some of the group students seemed confused when implementing each learning session described at the beginning of the lecture meeting, although several modules were provided as guidelines for conducting lectures. Some of them asked the instructor to give a more detailed explanation, even though they were at the end of the lecture session. Even after refining the strategy in the initial description of PBL-based learning procedures, the results of the trials in the group II respondents showed that some students still did not understand the PBL concept in depth. Students always find it difficult to follow the lecture process, so the scores of some students are still less than ideal minimum completeness standards.

Table 1. Effectiveness Test Results for the Application of the PBL Model

Experiment	Respondents	Mastery learning (%)	General Constraints / Revised Item
I	Group I	63	<ul style="list-style-type: none"> - PBL procedures in lectures are not understood - There is not enough time to complete all lecture sessions in one material - Some Passive Students attend lecture sessions
II	Group II	80	<ul style="list-style-type: none"> - PBL procedures in lectures are still not understood - There is not enough time to complete all lecture sessions in one material - Media support in the form of Information Technology is not sufficient for independent learning and online discussion.
III	Group III	96,6	<ul style="list-style-type: none"> - There is not enough time to complete all lecture sessions in one material - Test items are too difficult

In the third trial of group III respondents, the learning modules were designed to be more informative, an explanation of the concept of PBL was given in a more significant portion. As a result, students quickly follow PBL learning procedures. The results of this trial raise several fundamental questions: 1) how many modules are prepared as guidelines for students, so students can easily follow all PBL-based learning sessions. This situation is in line with the findings of Jones et al. (2013) who found many students who were pre-established and needed a lot of guidance for their projects; 2) how long it takes for PBL orientation to students so that students really understand and follow PBL-based learning processes correctly. These questions are in line with the results of Schneider's (2014) study which also raises open debate about 1) how many guidelines are needed in PBL; 2) potential students experience confusion if there is not enough initial learning framework; 3) students who do not have relevant and adequate initial knowledge tend to lag behind.

The problem of inadequate availability of time to complete all lecture sessions on a particular topic consistently appears in several trials conducted. If the time provided for learning is 1 credit within 1 week (Standar Nasional, 2014) and is 160 minutes (50 minutes face to face, 50 minutes structured assignments, 60 minutes independent appointments), then the total time needed to teach 1 lecture topic is equivalent to 3 credits is 480 minutes (8 hours) in 1 week, both for meetings in the classroom, and for learning outside the classroom. Some students stated that the time provided for completing all lecture sessions on one particular topic (8 hours) was inadequate.

The passive attitude of students following the learning session also found in trials. Some students feel bored following the PBL's lengthy procedure, and others think troubled and burdened when they are asked to fill out orderly learning forms, which is new to them. This finding is in line with the statement of Harun, Yusof, Jamaludin, & Hassan (2012) that students' motivation is the key to success in problem-based learning (PBL) implementation. The transition of use direct-instruction teaching methods to PBL instills a negative mindset towards PBL for students who are not familiar yet with inductive learning methods.

Another problem that arises in PBL trials is Information Technology support that is not yet sufficient as an effective communication medium for PBL. Students need effective and efficient technology (Information and Communication Technology) features to support their learning outside the classroom, for example, technology features to support group members' virtual inspiration, and technology features to support group members accessing resources or resources independent learning.

Another finding obtained in the trial of the application of this PBL model is that there is an increase in the number of students who reach the minimum completeness standard, namely 96.6% in the 3rd trial, far higher than the

achievement in Direct-Instruction systems which only reach less than 50%.

5. Conclusions

The application of the PBL model in 3 segments (Curriculum, Individual, and Group) in the learning design of Software Modeling is useful for increasing the level of students' mastery of a particular topic. A lengthy discussion that led to different views regarding the application of the PBL concept in learning offered further research opportunities to find PBL models that were truly effective in learning the field of Software Engineering.

Acknowledgements

Thanks to all staff in the academic field, especially the Chair of the STMIK Banjarbaru Informatics Engineering Department who helped provide data and information, and was willing to facilitate researchers during the trial period for implementing the PBL model in the Software Engineering class.

REFERENCES

- [1] Ardianto, D., Rubini, B. (2016). Comparison of Students' Literacy in Integrated Science Learning through Model of Guided Discovery and Problem Based Learning. *Journal Pendidikan IPA Indonesia*, 5(1), 31-37.
- [2] Bollin, A., Hochmüller, E., & Mittermeir, R. T. (2011, May). Teaching software project management using simulations. In 2011 24th IEEE-CS Conference on Software Engineering Education and Training (CSEE&T) (pp. 81-90). IEEE.
- [3] Dick, W., Carey, L., Carey, J.O. (2015). *The Systematic Design of Instruction*, Eight Edition. USA: Pearson.
- [4] Duch, B.J., Groh, S.E., & Allen, D.E. (2001). *The Power of Problem-Based Learning - A Practical "How To" for Teaching Undergraduate Courses in Any Discipline*. Virginia: Stylus Publishing.
- [5] Fakhriyah, F. (2014). Penerapan Problem Based Learning dalam Upaya Mengembangkan Kemampuan Berpikir Kritis Mahasiswa. *Jurnal Pendidikan IPA Indonesia*, 3(1), 95-101.
- [6] Fertalj, K., Milašinović, B., & Kosović, I. N. (2013). Problems and experiences with student projects based on real-world problems: a case study. *Technics Technologies Education Management-TTEM*, 8(1), 176-186.
- [7] Fonseca, V. M. F., & Gómez, J. (2017). Applying active methodologies for teaching software engineering in computer engineering. *IEEE Revista Iberoamericana de Tecnologías del Aprendizaje*, 12(3), 147-155.
- [8] Freeman, S., Eddy, S. L., McDonough, M., Smith, M. K., Okoroafor, N., Jordt, H., & Wenderoth, M. P. (2014). Active

- learning increases student performance in science, engineering, and mathematics. *Proceedings of the National Academy of Sciences*, 111(23), 8410-8415.
- [9] Gall, M. D., Gall, J. P., Borg, W. R. (2015). *Applying Educational Research* (Seventh Edition). USA: Pearson.
- [10] Garg, K., & Varma, V. (2015, February). Systemic requirements of a software engineering learning environment. In *Proceedings of the 8th India Software Engineering Conference* (pp. 147-155). ACM.
- [11] Glud, L. N., Buus, L., Ryberg, T., Georgsen, M., & Davidsen, J. (2010, May). Contributing to a learning methodology for web 2.0 learning—Identifying central tensions in the educational use of web 2.0 technologies. In *Proceedings of the 7th International Conference on Networked Learning* (Vol. 2010, pp. 934-94).
- [12] Goel, S. (2010). Design of Interventions for Instructional Reform in Software Development Education for Competency Enhancement. *Online Submission*.
- [13] Harun, N. F., Yusof, K. M., Jamaludin, M. Z., & Hassan, S. A. H. S. (2012). Motivation in problem-based learning implementation. *Procedia-Social and Behavioral Sciences*, 56, 233-242.
- [14] IEEE Computer Society (2013, December, 20). *Computer Science Curricula 2013: Curriculum Guidelines for Undergraduate Degree Programs in Computer Science*. The Joint Task Force on Computing Curricula, Association for Computing Machinery.
- [15] Jensen, J. L., Kummer, T. A., & Godoy, P. D. D. M. (2015). Improvements from a flipped classroom may simply be the fruits of active learning. *CBE—Life Sciences Education*, 14(1), ar5.
- [16] Jones, B. D., Epler, C. M., Mokri, P., Bryant, L. H., & Paretto, M. C. (2013). The effects of a collaborative problem-based learning experience on students' motivation in engineering capstone courses. *Interdisciplinary Journal of Problem-based Learning*, 7(2), 34-71.
- [17] Juman, K. K. (2018). Pengembangan Model Pembelajaran Rekayasa Perangkat Lunak Berbasis Multimedia Pada Perguruan Tinggi. *Jurnal Forum Ilmiah*. 15(1), 84-94.
- [18] Karabulut-Ilgu, A., Jaramillo Cherez, N., & Jahren, C. T. (2018). A systematic review of research on the flipped learning method in engineering education. *British Journal of Educational Technology*, 49(3), 398-411.
- [19] Kazimov, T. H., Bayramova, T. A. P. (2017). Problems of Teaching Software Engineering in Azerbaijan, *Problems of Information Society*, (1): 92-96
- [20] *Kerangka Kurikulum Berdasarkan KKNi Bidang Ilmu Informatika dan Komputer - Naskah Akademik*. (2016). Depok: APTIKOM Curriculum Team
- [21] Krusche, S., von Frankenberg, N., & Afifi, S. (2017). Experiences of a Software Engineering Course based on Interactive Learning. In *SEUH* (pp. 32-40).
- [22] Moravec, M., Williams, A., Aguilar-Roca, N., & O'Dowd, D. K. (2010). Learn before lecture: a strategy that improves learning outcomes in a large introductory biology class. *CBE—Life Sciences Education*, 9(4), 473-481.
- [23] Nuswowati, M., Susilaningsih, E., Ramlawati, R., & Kadarwati, S. (2017). Implementation of Problem-Based Learning with Green Chemistry Vision to Improve Creative Thinking Skill and Students' Creative Actions. *Journal Pendidikan IPA Indonesia*, 6(2), 221-228.
- [24] Oliveira, A. M. C. A., dos Santos, S. C., & Garcia, V. C. (2013, October). PBL in teaching computing: An overview of the last 15 years. In *2013 IEEE Frontiers in Education Conference (FIE)* (pp. 267-272). IEEE.
- [25] Paez, N. M. (2017, May). A flipped classroom experience teaching software engineering. In *2017 IEEE/ACM 1st International Workshop on Software Engineering Curricula for Millennials (SECM)* (pp. 16-20). IEEE.
- [26] Pierrakos, O., Zilberberg, A., & Anderson, R. (2010). Understanding undergraduate research experiences through the lens of problem-based learning: Implications for curriculum translation. *Interdisciplinary*
- [27] Pressman, R. S. (2002). *Rekayasa Perangkat Lunak* (Buku 1). Yogyakarta: Penerbit Andi.
- [28] *Standard Nasional Pendidikan Tinggi*. (2014). Jakarta: Directorate General of Higher Education of the Republic of Indonesia
- [29] Tambouris, E., Panopoulou, E., Tarabanis, K., Ryberg, T., Buus, L., Peristeras, V., ... & Porwol, L. (2012). Enabling problem-based learning through web 2.0 technologies: PBL 2.0. *Educational Technology and Society*, 15(4), 238-251.
- [30] Tanner, M., & Scott, E. (2015). A flipped classroom approach to teaching systems analysis, design, and implementation. *Journal of Information Technology Education: Research*, 14, 219-241.
- [31] Warter-Perez, N., & Dong, J. (2012, April). Flipping the classroom: How to embed inquiry and design projects into a digital engineering lecture. In *Proceedings of the 2012 ASEE PSW Section Conference* (Vol. 39). Washington, DC: American Society for Engineering Education.
- [32] Wibawa, B., Kardipah, S. (2018). The Flipped-Blended Model for STEM Education to Improve Students' Performances, *International Journal of Engineering & Technology*, 7(2.29), 1006-1009.