USEJ 8 (2) (2019)

Unnes Science Education Journal

http://journal.unnes.ac.id/sju/index.php/usej

USING PROBLEM-BASED LEARNING TO TEACH SOFTWARE MODELING IN INFORMATION TECHNOLOGY COLLEGES IN INDONESIA: A CONCEPT

Bahar [1,✉], Basuki Wibawa [2], Robinson Situmorang[3]

[1]Informatics Engineering Department, STMIK Banjarbaru, Indonesia
[2,3]Educational Technology Department, Universitas Negeri Jakarta, Indonesia

Article Info

Article History:
Received May 2019
Accepted June 2019
Published July 2019

Keywords:
Conceptual model, problem-based learning, software modeling

Abstract

The Direct-Instruction model which is still widely used in the learning of software modeling in higher education places more emphasis on classroom interactions initiated by the teacher. Generally, it only involves a small portion of student-to-student interaction. The direct learning systems that depend on the reflection ability of instructors only provide few opportunities for students to be actively involved in the learning process. Thus, it is different from software modeling which emphasizes Student-Centered Learning. Consequently, learning becomes ineffective and students cannot reach the minimum competency standard stated in the learning design. This paper proposes a Problem-Based Learning model that is integrated in software modeling learning at three segments: Curriculum segment, emphasizing the use of problems as the starting point of student learning; group segment, emphasizing collaboration systems (group discussion-based-learning); and student segment, stressing the Student-Directed Learning (SDL) system. The effectiveness test results show that the application of the PBL model in 3 segments (Curriculum, Individual, Group) in the learning design of Software Modeling, is effective in increasing the level of student mastery of a particular topic.

p-ISSN 2252-6617
e-ISSN 2252-6232

✉Corresponding author:
Bahar
Informatics Engineering Department, STMIK Banjarbaru, Indonesia
Email: bahararahman@gmail.com

INTRODUCTION

Software Engineering (SE) education (curriculum, outcomes and delivery) has received considerable attention from Institute of Electrical and Electronics Engineers (IEEE) and Association for Computing Machinery (ACM) societies. There SE is defined as "The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software". Guiding Principles from IEEE and ACM clearly state that SE education needs to combine computer science foundations with engineering, organizational, teamwork, communication and project management issues. Guidelines for curriculum delivery from the same reference clearly point to the need for practical project and team-oriented exercises with a significant capstone project (IEEE Computer Society, 2004).

Software Engineering is a core course in Software Engineering or Informatics Engineering in the Information Technology College in Indonesia. The RPL course is named System Analysis and Design/Modeling in the Information Systems Department, and Software Engineering in the Informatics Engineering Department. Software Engineering covers a vast field, so it is generally made into several subjects, and given names such as Software Project Planning, Software Analysis and Modeling, and Software Implementation and Testing (Kerangka Kurikulum, 2016).

Software Engineering has characteristics that are different from other fields of science. Software Engineering is a science whose technology is continuously changing, due to the following factors. First, software is related to the advances in technology in relation to other hardware and software that demand compatibility, as well as changes in the company's business strategy and the behavior of end-user demand support changes to the need for software. It requires high adaptation skills in the development process (Juman, 2018). Second, the development of Information and Communication Technology operating by software has caused some difficulties during the software development process. The large volume of code (program) and the complexity of the system architecture cause small software development systems to be individually abandoned, and transferred to technology development collectively and collaboratively. Third, software design and programming technology also develop dynamically. Various development models and software programming languages have encouraged the emergence of a variety of new libraries in the software development environment, thus demanding lifelong learning adaptation (Bollin, Hochmüller & Mittermeir, 2011; Kazimov, 2017).

Software Engineering applies a systematic and disciplined approach to the development, operation, and maintenance. Thus, software developers need technical and social skills in their work. Traditional teaching methods that focus on lectures and tutorials for Software Engineering students are not enough to develop the skills to solve real-world problems. In the traditional learning approach, most students must complete their assignments, and this is contrary to professional practice in the collaborative environment of software development teams (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 in college.

According to Garg and Varma (2015), software engineering requires the requirements of an effective and sustainable learning environment, which is classified into an authentic learning climate, independent learning, learning from failure and success, motivating students, etc.

Enabling problem-based learning through web 2.0 technologies can streamline the movement of students individually to find learning resources, improve students' collaboration skills 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).

The characteristics possessed by software as described previously, require an approach in the learning process of Software Engineering which emphasizes active learning (Freeman et al., 2014; Jensen, Kummer, & Godoy, 2015), leads to the concept of lifelong learning, emphasizes the resolution of problems in the real world and unstructured problems, and emphasizes collaboration skills (Karabulut-Ilgu, Jaramillo Cherrez, & Jahren, 2018). It is contrary to the learning system that is widely used in learning Software Engineering in Higher Education, namely the direct learning model (direct instruction).

Direct-instruction Model is a learning model that uses teacher demonstrations and explanations combined with training and student feedback to help students obtain real knowledge and skills needed for further learning (Wahono, 2012). Direct-Learning Model emphasizes class interaction which is primarily initiated by the teacher and generally does not involve student interaction (Ewing, 2011). The direct learning model also highlights the achievement of basic/declarative and procedural pedagogical goals (Ekasari, 2016). The direct learning system that depends on the reflection ability of instructors only provides little opportunity for students to be actively involved in the learning process, causing learning to be ineffective.

We have conducted preliminary research at several Information Technology campuses in Indonesia using questionnaires distributed to teachers and students. We also make observations in classes that hold lectures on Software Modeling. The results of the preliminary study indicate that the direct learning method in the modeling software course results in a passive learning process, the low critical nature of students, low understanding of students in certain parts of the competence taught, and dull atmosphere of learning in the classroom. Due to such factors, students cannot reach the minimum competency standards planned in the learning design.

Problem-Based Learning (PBL) is one of the learning models based on constructivist learning; namely, learning initiated by a posed problems, queries, or problems that the learner wants to solve. Complex real-world problems are used to motivate students to identify and examine concepts and principles they need to know from work through these problems. Student work independent, in small work team, and bringing together collective skills at acquiring, communicating, and integrating information (Duch, Groh, & Allen, 2001).

PBL is an educational 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 science literacy appropriately (Ardianto, 2016). PBL correlates with cognitive functions that contain various types of thinking and creative acts in the learning phase (Nuswowati, Susilaningsih, Ramlawati, & Kadarwati, 2017), including the use of existing knowledge (prior knowledge), reorganizing new knowledge in cognitive structures, analysis and synthesis, structuring and idea development, and problem solving (Pierrakos, Zilberberg, & Anderson, 2010).

There have been many studies on engineering and learning software courses. Tanner & Scott (2015) introduced the flipped classroom approach used for two second-year Information Systems courses at a South African University in the teaching of systems analysis and design using Unified Modelling Language (UML). The aim is to create a learning environment that is more student-centered to encourage class discussion and debate, which in turn functions to train students' critical thinking skills. Students are not only 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 integrated into case studies.

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

Paez (2017) conducted research at the Universidad Nacional de Tres de Febrero, Argentina, using a flipped classroom approach combined with other non-traditional teaching techniques that were adopted by the engineering course software. Teachers design relevant teaching materials and plan activities in the classroom and outside the classroom with the help of virtual classes. Virtual classes are used to share files and to expand interactions between teachers and students outside of class meetings. Additional teaching strategies involved in this approach are sustainable practice, teaching by example, and the use of real-world tools. The main pedagogy used in this study is a flipped classroom, and proven effective in learning Software Engineering. This method does not emphasize developing team work skills.

Fonseca and Gomez (2017) apply problem-based learning (PBL) to software engineering course at two different universities in Chile. The PBL application enabled students to acquire a deeper knowledge and apply it in practical ways in accordance with work plans. In this framework, students are given different project roles in real projects and were able to work in a dynamic environment using PBL. Even though this learning framework gives importance to develop independent learning skills, creative product development, and teamwork, this method is lacking in developing communication/ presentation skills

Fakhriah (2014) introduced the PBL model as an effort to develop students' critical thinking skills in the learning process. PBL begins with the orientation of the problem at the initial lecture meeting in the class guided by their instructor. Furthermore, students divided into small groups conducted field observations related to specific learning themes. Students in groups formulate problems encountered in the field observation process, then determine the right solution to solve the problem. At the end of the learning session, students reflect and conclude the results of the learning activities. This PBL model only emphasizes social skills acquired from the collaborative learning system, not individual learning skills to look for sources of learning outside the classroom are needed, so students are accustomed to being learners of all time.

This paper proposes a conceptual PBL model implemented in Software Engineering life cycle phase (the analysis phase and design phase), which Pressman (2002) called "the System Modeling phase".

Integrating PBL into learning Modeling Software is done in three segments: 1) In the Curriculum segment, emphasizing the use of problems as the starting point of student learning. The face-to-face model in class (direct instruction) with Presentation and Brainstorming methods is used to convey conceptual, structured things related to the problems to be discussed at each learning session; 2) In the Student segment, emphasizes independent learning (Student-Directed Learning). This activity underscores the activeness for students independently reviewing things that are not structural, are real in the field, about the concept of software requirements model developed. 3) In the Group segment, emphasizing collaboration. This activity highlights the active participation of students (small groups) in formulating and synthesizing the results of studies conducted independently. In learning Software Engineering, this activity is synonymous with the stages of Software Modeling Preparation which are presented in groups in the General Discussion Forum in the class, as a reflection medium for all the concepts of the problems studied, both individually and in small groups.

The overall implementation three main elements of PBL (problem orientation, independent learning, group collaboration) on each learning topic, is intended to perfect the models which is oriented toward class meetings that is fully controlled by the teacher (Tanner & Scott, 2015), models which place individual independent learning in tiny portions (Fakhriah, 2014), models that does not emphasize developing team work skills (Paez, 2017), and models that lacking in developing communication/ presentation skills (Fonseca, Gomez, 2017).

METHODS

This research uses Research and Development Methods, by adapting the stages of Research and Development proposed by Gall, Gall, & Borg (2015). The steps of research consist of:

1) Problems and needs analysis. Focus Group Discussion Method that involves learning designers, department leaders, college graduates, and graduates, applying to formulate competencies in the field of software modeling needed in learning design.
2) Development of learning designs, consisting of: (1) determining competency standards, (2) determining essential competencies, (3) analyzing learning, (4) formulating indicators, (5) developing assessment instruments, (6) developing learning materials, (7) compile learning strategies, and (8) design evaluations. The advanced PBL concept was implemented in the 7th stage (preparation of learning strategies).
3) Formative Evaluation, consisting of (1) material expert validation (learning aspects and aspects of learning material), (2) individual trials (one to one evaluation), (3) small group evaluation, (4) field trials/large groups. At the end of each trial phase, data analysis and product revisions are carried out based on the input obtained from the experiment.

The model effectiveness is tested using pseudo-effectiveness evaluation, which is examining the model in the real situation that is conditioned (not in the actual field condition). Assessment of the effectiveness of the model was carried out in field trials / large groups. The stages of model effectiveness validation consist of:

1) Implement the PBL model according to the learning stages in the learning design document for Software Modeling. The implementation was carried out on 30 students in one class with varying levels of student intelligence.
2) Evaluating learning outcomes, carried out through the test instrument items available in the learning design document. Learning outcomes are guided by the value of the ideal Minimum Completion (Rahma, Lutfi, 2013) which is nationally determined by the Indonesian government (minimum 75 in the range of assessments 0-100).
3) The focus group discussion method is carried out at the end of each trial session, to examine the constraints of each lecture session
4) If the evaluation results have not reached the Minimum Completion Criteria standard, learning trials are repeated in different classes. Before the learning trial is conducted, first a revised learning plan is carried out.
5) The evaluation process is declared over, and the PBL-based Modeling learning software design is declared effective if the acquisition of test results for all study participants has reached the ideal Minimum Completion Standard.

RESULTS AND DISCUSSION

Proposed PBL Concept

The PBL concept developed includes three segments, consisting of:

1) Curriculum Segment, which is the use of problems as the starting point of student learning.
2) Group Segments, namely the collaboration of students.
3) Student Segments, namely independent learning conducted by students (Student-Directed Learning)

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, consisting 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 3 segments (curriculum, individuals and groups) in the PBL environment, as shown in figure 1.

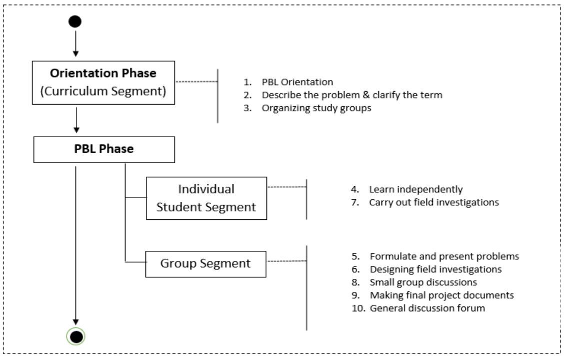

Figure 1. Implementation of PBL Syntax into the Learning Phase

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

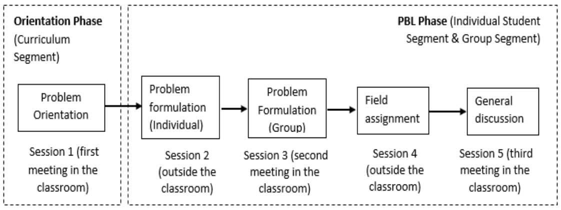

Figure 2. PBL Strategy in the proposed Software Modeling Learning Design

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 small group discussion, group members reflect individually to fill the knowledge gap that has been obtained by each group member. The teacher acts as a motivator and facilitator 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, filling 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, then presents it at a public meeting in the class room.

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 one 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.

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

Experiment	Respondents	Mastery learning (%)	General Constraints / Revised Item
I	Group I	63	- 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	- 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	- There is not enough time to complete all lecture sessions in one material - Test items are too difficult

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 following/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 concrete 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.

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 easily follow PBL learning procedures. The percentage of students who get the minimum >= Minimum Completion Standard is better. 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 is in line with the findings of ones 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) 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 are 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 from conventional (direct-instruction) teaching methods to PBL instills a negative mindset towards PBL for students who are not familiar 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 the PBL model was that there was an increase in the number of students who achieved the competency test results that met the ideal Minimum Completion Standard, which was 96.6% in the 3rd trial. Minimum completeness in direct-instruction, which is <50% (the results of preliminary research in 3 RPL classes in 3 semesters).

CONCLUSION

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

REFERENCES

Ardianto, D., Rubini, B. (2016). Comparison of Students' Scientific Literacy in Integrated Science Learning through Model of Guided Discovery and Problem Based Learning. *Jurnal Pendidikan IPA Indonesia*, *5*(1), 31-37.

Bollin, A., Hochmüller, E., & Mittermeir, R. T. (2011, May). Teaching software project management using simulations. In2011 24th IEEE-CS Conference on Software Engineering Education and Training (CSEE&T) (pp. 81-90). IEEE.

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.

Ekasari, R. R., Gunawan, G., & Sahidu, H. (2017). Pengaruh Model pembelajaran langsung berbantuan media laboratorium terhadap kreatifitas fisika siswa SMA. *Jurnal Pendidikan Fisika dan Teknologi*, *2*(3), 106-110.

Ewing, B. F. (2011). Direct instruction in mathematics: Issues for schools with high indigenous enrolments: A literature review. *Australian Journal of Teacher Education*, *36*(5), 63-91.

Fakhriyah, F. (2014). Penerapan Problem Based Learning dalam Upaya Mengembangkan Kemampuan Berpikir Kritis Mahasiswa. *Jurnal Pendidikan IPA Indonesia*, *3*(1), 95-101.

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.

Fonseca, V. M. F., & Gómez, J. (2017). Applying active methodologies for teaching software engineering in computer engineering. *IEEE Revista Iberoamericana de Tecnologias del Aprendizaje*, 12(3), 147-155.

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.

Gall, M. D., Gall, J. P., Borg, W. R. (2015). *Applying Educational Research* (Seventh Edition). USA: Pearson.

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).

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 educational use of web 2.0 technologies. *In Proceedings of the 7th International Conference on Networked Learning* (Vol. 2010, pp. 934-94).

Goel, S. (2010). Design of Interventions for Instructional Reform in Software Development Education for Competency Enhancement. *Online Submission*.

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.

LeBlanc, R. J., Sobel, A., Diaz-Herrera, J. L., & Hilburn, T. B. (2006). *Software engineering 2004: curriculum guidelines for undergraduate degree programs in software engineering*. IEEE Computer Society.

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.

Jones, B. D., Epler, C. M., Mokri, P., Bryant, L. H., & Paretti, 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.

Juman, K. K. (2018). Pengembangan Model Pembelajaran Rekayasa Perangkat Lunak Berbasis Multimedia Pada Perguruan Tinggi. *Jurnal Forum Ilmiah.* 15(1), 84-94.

Karabulut-Ilgu, A., Jaramillo Cherrez, 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.

Kazimov, T., & Bayramova, T. (2017). Problems of teaching software engineering in Azerbaijan. *Problems of information society*, *8*(1), 92-96.

Kerangka Kurikulum Berdasarkan KKNI Bidang Ilmu Informatika dan Komputer - Naskah Akademik. (2016). Depok: APTIKOM Curriculum Team.

Krusche, S., von Frankenberg, N., & Afifi, S. (2017). Experiences of a Software Engineering Course based on Interactive Learning. In *SEUH* (pp. 32-40).

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.

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. *Jurnal Pendidikan IPA Indonesia*, *6*(2), 221-228.

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.

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.

Pierrakos, O., Zilberberg, A., & Anderson, R. (2010). Understanding undergraduate research experiences through the lens of problem-based learning: Implications for curriculum translation. *Interdisciplinary Journal of Problem-based Learning*, *4*(2), 4, 35-62.

Pressman, R. S. (2002). *Rekayasa Perangkat Lunak* (Buku 1). Yogyakarta: Penerbit Andi.

Rahma, A., & Aulia, R. I. Z. K. Y. (2013). Pemanfaatan Media Permainan Tradisional Selibur Sebagai Media Pembelajaran Untuk Meningkatkan Hasil Belajar Struktur Atom (The Utilization Of Traditional Game Selibur As Instructional Media For Increasing Study Result Of Atomic Structure). *Unesa Journal Of Chemical Education*, *2*(1), 59-63.

Schneider, D. K. (2016). Educational (instructional) Design Models. E*ducation Technology -Wiki*: 94.

Standar Nasional Pendidikan Tinggi. (2014). Jakarta: Directorate General of Higher Education of the Republic of Indonesia.

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.

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.

Wahono, S. (2012). Strategi dan Model Pembelajaran. Jakarta: Permata Puri Media, pp.363.

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.