Universal Journal of Educational Research 8(9A): 1-9, 2020 http://www.hrpub.org
DOI: 10.13189/ujer.2020.082001

Development of Instructional Design Models Based on PBL Model for Software Modeling Course at the Information Technology College in Indonesia

Bahar[1,2], Basuki Wibawa[1,*], Robinson Situmorang[1]

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

Received December 10, 2019; Revised June 20, 2020; Accepted July 29, 2020

Cite This Paper in the following Citation Styles

(a): *[1] Bahar, Basuki Wibawa, Robinson Situmorang , "Development of Instructional Design Models Based on PBL Model for Software Modeling Course at the Information Technology College in Indonesia," Universal Journal of Educational Research, Vol. 8, No.9A, pp. 1 - 9, 2020. DOI: 10.13189/ujer.2020.082001.*

(b): *Bahar, Basuki Wibawa, Robinson Situmorang (2020). Development of Instructional Design Models Based on PBL Model for Software Modeling Course at the Information Technology College in Indonesia. Universal Journal of Educational Research, 8(9A), 1 - 9. DOI: 10.13189/ujer.2020.082001.*

Abstract The Conventional instructional design methods using teacher-centered learning approaches provide few opportunities for students to be actively involved during the learning process. This process is not relevant to the characteristics of the Software Modeling course which ideally emphasizes student-centered learning. As a result, instructors cannot control the learning system efficiently, so students cannot reach the minimum competency standards planned in the learning design. In this work, we propose an instructional design model that integrates Problem-Based Learning (PBL) into the instructional deign of Software Modeling course. Three main elements of the PBL learning system model are proposed, namely: Elementary Curriculum that emphasizes the use of problems as the starting point of student learning; Element Group that emphasizes collaboration systems (group discussion-based learning); and Student Elements that emphasizes the Student-Directed Learning (SDL) system. The three main elements of PBL are applied thoroughly and proportionally to each particular learning topic in Software Modeling course. This concept is different from PBL concepts which have been widely applied as learning models, which do not apply the three main PBL elements to each learning topic. This concept is also designed to improve the weaknesses of the PBL model which only places individual learning in small portions, or even none at all. This research uses the Research and Development (R&D) method which consists of three main stages, namely: analysis of system requirements, system development, and formative evaluation. In formative evaluation we do the expert validation, one-to-one student evaluations, and small group evaluation to test the effectiveness of the system. The result of expert validation (related to content, instructional design, and media design) shows that the instructional media product based on PBL that is developed is very suitable to be used as a learning resource, with an average validation percentage of 83.2%. The students' responses to the one-to-one student evaluation and small group trials show that the instructional design products that are developed are feasible to make student actively involved during the learning process, with an average value (related to product display, Usability, and benefits aspects) reach 82.6%. The average level of mastery of students on certain topics reaches more than 96%.

Keywords Instructional Design Model, Problem-based Learning, Software Modeling, Formative Evaluation

1. Introduction

Software Engineering (SE) education 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 education 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 (Kazimov, Bayramova, 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 (Oliveira, dos Santos, & Garcia, 2013; Krusche, von Frankenberg, & Afifi, 2017). Su, Jodis, & Zhang (in Fertalj, Milašinović, & Kosović, 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.

It is essential for software developers to have the ability to understand user needs, cooperate in teams, and participate in the overall software engineering development process. Related to this, students feel that software engineering is a complex science because, in addition to mastering technical skills, it also requires social aspects. Žagar (2008) suggests that software engineers are required to have soft skills, including the ability to present knowledge, learn from independent sources, and listen to what others have to say.

The characteristics possessed by software as described previously, require an approach in the learning process of Software Engineering which emphasizes active learning, leads to the concept of lifelong learning, emphasizes the resolution of problems in the real world and unstructured problems, and emphasizes collaboration skills. 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 learning model (direct instruction) 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, Gunawan, & Sahidu, 2017)

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.

PBL is one of the learning models that can be applied to engineer disciplines. PBL is one of the best exemplars of a constructivist-learning environment. PBL is an influential way for inquiry-based learning in which students use an authentic problem as the context for an in-depth investigation of what they need and what to know. Problem simulation is used to activate students' curiosity before learning a subject (Akcay, 2009). 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, Rubini, 2016). PBL correlates with cognitive functions that contain various types of thinking and creative acts in the learning phase (Nuswowati et al., 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 (Dickens, Arlett, 2009).

There have been many studies on engineering and learning software courses. Murphy, Phung, & Kaiser (2008) reported the results of their investigation on The COMS W4156 Advanced Software Engineering course at Columbia University. The study only focused on programming topics (programming work in pairs) with distance learning methods. The results of the investigation also do not clearly explain the PBL method used, and only present a number of challenges faced when teaching extreme Programming (XP) with a distance learning system using PBL for students who are not physically in one area, such as students' aversion to aspects of XP and difficulties in scheduling.

Richardson & Delaney (2009) reported the results of their investigation regarding the use of Problem Based Learning in Software Engineering Classroom. The study focused on the stages of system analysis and design in the cycle of Software Engineering. The application of PBL in the learning cycle begins with presenting authentic problems at the beginning of the learning sequence with the aim of developing a set of skills to not only solve problems in the future but also to bring these skills to the workplace. Furthermore, interaction is more focused on collaborative learning systems through small group discussions in the classroom to share knowledge from their work experience in the school. The instructor, in this case, does not act as a facilitator as it does in the normal PBL environment. So, student independence is reduced. There is little individual learning done by students in this model. In this experiment, learning is only done in the classroom. This investigation also does not present the impact of PBL on improving students’ competence in learning.

Shim, Choy, & Kim (2009) introduced collaborative-learning in Software Engineering by implementing PBL strategies to help students understand the importance of social aspects and a systematic framework for improving teamwork skills. The concept offered is to integrate 12 stages of PBL into the four main phases of the Pressman model (Communication, Planning, Modeling & Construction, and Deployment). However, the 12 PBL phases are not implemented in any particular stage in the Pressman model. The initial three steps of PBL (Introducing PBL, introducing role instructors/learners, and identifying the problem) were integrated into the initial phase of RPL (communication phase). The next three PBL phases (establishing activity plans, learning goals settings, assigning tasks) were combined into the second phase of RPL (Planning). The next PBL Phase (scanning and collecting information, analyzing the collection information, and deriving a solution) was integrated into the third stage of RPL (modeling & construction). The final three phases of PBL (presenting a solution, evaluating the process, promoting adaptation to similar environments) were integrated into the fourth step of deployment. The implementation of the PBL model into the learning design offered by Shim does not apply all three basic things (problem orientation, independent learning, group collaboration) PBL on each particular topic in the course. Shim only uses some of the three basic things, so that the potential cause problems if a specific topic of discussion requires the three elements of the PBL (problem orientation, students' independent activities, and student collaboration activities) in learning.

Tanner & Scott (2015) uses the flipped classroom approach in software analysis and design at the University of South Africa 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 are not only theoretical understandings 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 student-centered (Wibawa, Kardipah, 2018; Syakdiyah, Wibawa, & Muchtar 2018), encourages students' critical thinking (Moravec, Williams, & Aguilar-Roca, 2010), and allows students to learn theoretical concepts outside of classroom settings and apply these concepts in the classroom while getting help from facilitators and other students (Warter-Perez, Dong, 2012). However, the flipped classroom approach is fully controlled by the teacher, so students are not free to apply their knowledge to solve more complex problems. Other side, the concept of modeling should be carried out in various ways and must be focused on modeling options and assessing their validity. This situation makes some students break away from discussion because they feel they are not free to be creative.

Fakhriyah (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. However, 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 application of the three main elements of PBL (problem orientation, independent learning, group collaboration) on each learning topic, is intended to perfect weaknesses in the Shim (2009) model which does not implement 3 PBL main elements on each learning topic. This concept is also designed to refine the weaknesses of Richardson & Delaney (2009) and Fakhriyah (2014) models which only place individual independent learning in tiny portions, or even none at all.

2. Methods

In this work we use Research and Development (R&D) methods that refer to Gall et al. (2015) in developing and testing PBL-based Instructional Design Modeling Software Models. The main stages of the study consisted of:

1) Analysis of instructional needs, aims to formulate the main competencies in software modeling. The Focus Group Discussion (FGD) method was adopted in formulating key competencies. FGDs involve learning designers, management of study programs, college graduates, and users of graduates.
2) Development of instructional designs, including: developing assessment instruments, developing instructional strategies, and developing instructional materials. The proposed PBL concept is implemented in the stages of preparing the learning strategy
3) Model validation. Validation activities consist of: validation by experts (related to media design and instructional design) and one-on-one evaluations of students. At the end of each trial phase, data analysis and model revisions are carried out based on input obtained from the results of the trials.
4) Testing the effectiveness of instructional design through trials on small groups of students. A total of 20 students who have the characteristics of high, medium, and low ability to become respondents in the testing phase of a small group of students.
5) Evaluation of learning outcomes using items of test instruments available in instructional design documents. Learning outcomes are guided by the value of the ideal Minimum Mastery Standards (Rahma, Aulia, 2013). which are set nationally by the Indonesian government (minimum 75 on the assessment scale 0-100). At the end of each trial session a Focus Group Discussion is conducted to assess the constraints of each lecture session. If the evaluation results have not yet reached the minimum Mastery Criteria, the learning trial is repeated in a different class. Before the learning trials are carried out, a revision of the learning design is done first. The evaluation process is declared over and the learning design of PBL-based Software Modeling is declared effective if the acquisition of test results for all study participants has reached the ideal Minimum Mastery Standards.

3. Result and Discussion

3.1. The Proposed Model of Development Instructional Design

The Proposed model of development instructional design based on PBL model for Software Modeling course presented in Figure 1.

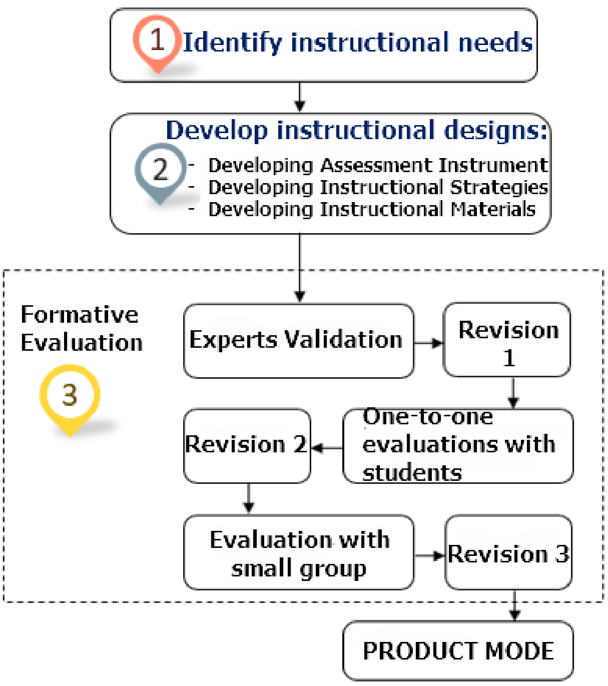

Figure 1. Stages of Development of Instructional Design Models

In Figure 1, there are three main stages in the development of instructional design, namely: identifying instructional needs, developing instructional designs, and evaluating instructional designs.

Identifying Instructional Needs

The activity of identifying instructional needs is aimed at formulating instructional objectives. Formulation of instructional objectives involves stakeholders in the educational environment (active students, graduates, educators, and graduate users). Stakeholder involvement to ensure the achievement of Instructional needs that reflect the needs of the parties in the educational environment.

Develop Instructional Designs

The instructional design development activities are aimed at preparing learning outcomes assessment instruments, developing learning strategies, and developing instructional materials. Development of learning outcomes assessment instruments is aimed at measuring the level of mastery of students in each competency that has been set at the Instructional Objectives. The design of the learning outcomes assessment instrument is intended to measure the area of cognitive taxonomy (knowledge), the area of psychomotor taxonomy (practice), and the affective region (attitude of behavior). Cognitive assessment of students' responses using instruments in the form of tests, both in written and oral form. Psychomotor responses of students are measured by means of observation in the form of a check list combined with an attitude scale, while a description of the attitudes, perceptions or opinions of students is measured using rating scales.

The learning strategies design is adapted to the proposed PBL model. Ten steps recommended in the PBL syntax, which are tailored to the learning characteristics of Software Modeling (Bahar, Wibawa & Situmorang, 2019) consisting of: PBL Orientation; Describe the problem & clarify the term; Organizing study groups; Learn independently; Formulate and present problems; Designing field investigations; Carry out field investigations; Small group discussions; Making final project documents; General discussion forum. There are three main aspects in PBL Concept proposed in Software Modeling Course, namely: (1) Curriculum-level, the use of problems as a starting point for student learning; (2) Group-level, there is a collaboration (group discussion); and (3) individual Student-level, independent learning (Student-Directed Learning/SDL). Integration PBL levels with the software modeling phase in software modeling course presented in Figure 2.

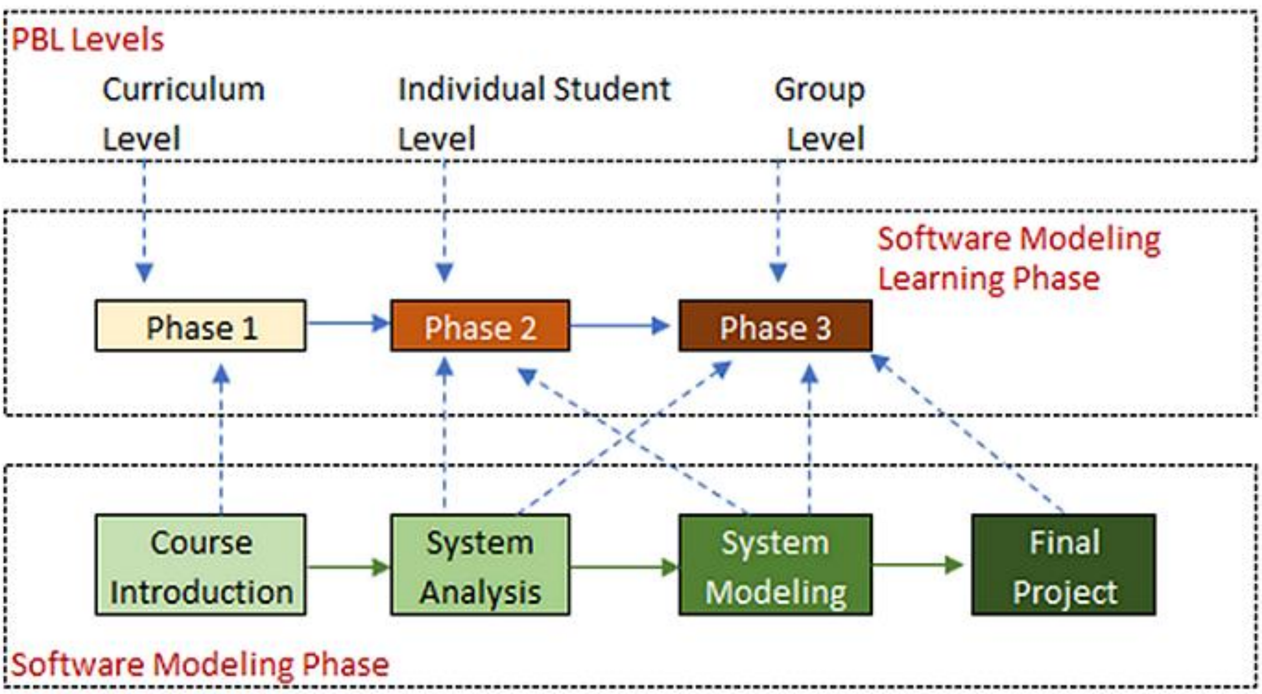

Figure 2. The Concept of PBL Levels Integration with the Software Modeling Phase in Software Modeling Course

In Figure 2, there are three phases in the software modeling proposed learning concept model, which are the results of alignment between the main PBL components (level) and the software modeling phase:

1) Stage one (initial phase)

The initial period is a learning phase with a face-to-face system in class, with presentation and brainstorming methods. In the software modeling course, this initial phase was used for introducing lectures that explain the basic concepts of analysis and modeling of software and the objectives and objectives of analysis and modeling. Direct face-to-face models in class (direct learning) were used to convey conceptual and structured matters. In PBL, this phase is at the curriculum level, namely the use or explanation of the problem as the starting point of student learning. Three PBL syntax is implemented in this phase, namely: PBL orientation; describe the problem & clarify the term; organizing study groups.

2) Stage two (mid phase)

Phase two is the phase of independent learning by individual learning participants. The second phase in PBL is called the period at the Student level, which is independent learning (Student-Directed Learning). In learning Modeling Software, this activity is identical to some of the events of problem analysis and system requirements and software system modeling activities, which emphasizes the activeness of students in independently reviewing things that are not structural and real in the field related to the concept of software requirements model that will developed. Two PBL syntax is implemented in this phase, namely: learn independently and carry out field investigations.

3) Stage three (final stage)

Stage three is a collaborative learning phase involving students in small groups, both in class and outside the classroom. The third phase in PBL is called the period at the Group level, characterized by collaboration (group discussion), while in learning modeling software, this activity is identical to several events in system analysis and system modeling which emphasizes the partnership of students (small groups) in formulating and synthesizing the results of studies previously carried out independently. This 3rd phase in learning software modeling is also identical to the stages of software modeling preparation that will be accounted for in groups in the general discussion forum in the class, as a medium to reflect on all conceptual issues that have been studied both individually and in small groups. In this stage, Five PBL syntax is implemented in this phase, namely: formulate and present problems; designing field investigations; small group discussions; making final project documents; general discussion forum

Evaluating Instructional Design

Formative evaluation models that refer to the Dick and Carey (2014) models are used to validate instructional designs that have been developed. Dick and Carey (2014) divided the stages of formative evaluation into four stages, namely: one-on-one evaluations by experts, one-on-one evaluations by students, small group evaluations, and field trials. But in its implementation, Keller (Suparman, 2014) holds that the designer can choose and use only two or three steps in accordance with certain considerations (availability of time and cost). The formative test in this study only went through three stages of formative testing, namely: validation by experts (instructional design experts, Information Technology media experts, content modeling software experts), one-to-one student evaluations, and small group evaluations.

In the model validation process by the experts, each expert (instructional design expert, information technology expert, and modeling software content expert) was asked to examine the instructional design that was developed. Input from each expert is used to perfect instructional design before conducting one-to-one evaluation with students.

One-to-one evaluations by students involved three respondents (students) who had the characteristics of the target population. The three selected students represent high, medium, and low abilities. First the instructional material is given to students to learn, then ask the students' understanding of certain parts of the instructional system. Students provide a view of the material and learning activities undertaken. Student input is used to make improvements to instructional design before testing small groups.

Small group testing involved 20 students who represented high, medium and low abilities. The initial step is to prepare the learning environment, learning facilities, tools needed in instructional strategies and forms of instructional activities that have been determined. Instructional activities begin by measuring the students' initial competency through pretest. The next step is organizing learning activities in accordance with the PBL strategy that has been formulated. Posttest is carried out after the learning activities are over. At the end of the learning session, students again provide input regarding instructional activities. Input is needed as a guideline for perfecting instructional design before becoming a final product. The effectiveness of instructional activities is measured by comparing the significance of the value of student learning outcomes at Posttest with Pretest, as well as the achievement of predetermined competency standards. The effectiveness of instructional activities is also measured based on the attitudes of students and teachers towards instructional activities, as well as the feasibility of instructional activities in terms of time relevance.

3.2. Testing the Effectiveness of Instructional Design

Expert Validation

Validation by experts and students of instructional design products is carried out through questionnaire sheets in the assessment range 1 - 4. Value 4 = correct, 3 = quite accurate, 2 = inaccurate, and 1 = inaccurate. Each value given is transformed into a percentage (%): 85 - 100 = very feasible (does not need to be revised), 75 - 84 = feasible (does not need to be revised), 55 - 74 = not enough (needs to be revised), and 0 - 54 = not feasible (needs to be revised). On the questionnaire sheet, experts can provide suggestions, criticisms, and input on the learning media products developed.

Instrument items for evaluating instructional design are items related to aspects: the accuracy of the formulation of learning objectives, evaluation tools, and learning strategies. To evaluate learning content, instrument items related to aspects of learning material relevance and the quality of learning material are used. The Evaluation of Information Technology-based Learning Media, uses instrument items related to the aspects of visual, functional, and Usability communication. Table 1 presents the results of evaluations by experts on the instructional design products developed.

Table 1. Expert Validation Results

Expert	Average Score (%)
Software Modeling Content Expert	83.20
Instructional Design Expert	82.40
Information Technology Expert	84.30
Average	83.20

Table 1 presents the results of the average assessment by instructional content experts, which is 83.20%. This value indicates the learning content in the category is feasible to use. The results of the instructional design expert assessment reached an average of 82.40%, indicating the instructional design was in the appropriate category for use. Likewise, the results of the Information Technology-based instructional media expert evaluation reached 84.30%, indicating that the learning media for Modeling Software based on online technology is also feasible to use. The average value of the results of expert validation on instructional design products developed reached 83.20%, indicating that the products in the effective and feasible category were used.

One-to-One Evaluation by Students, and Small Group Tests

One-to-one Evaluation by Students and Small Group Tests, done by comparing the significant or not the value of student learning outcomes in Posttest with the value of the results of the Pretest when testing on students in small groups. The effectiveness of instructional design development was also confirmed to students through one-to-one student test. In the one-to-one student test, students are asked to provide responses related to Visual Communication aspects, usability aspects and benefits aspects. Students' responses are also asked at the end of the field test. Evaluation results as presented in table 2.

Table 2. Student Evaluation Results

Pretest Completeness Average (%)	Posttest Completeness Average (%)	Instructional Product Assessment Results (%)
54.60	95.50	Visual communication display: 84.50 Usability: 81.80 Benefits aspects: 81,60 Average: 82.60

The results of assessments by students of the Visual Communication Display aspect reached an average of 84.50%. These results indicate the instructional design is very feasible to use. The Usability aspect reaches an average of 81.80%, indicating instructional design is appropriate to use. The assessment of students on Benefits Aspect averaged 81.6%, indicating that the instructional design was feasible to use. Overall, the average value of students' scores on instructional design products for Modeling Software that was developed reached an average of 82.60%, indicating that instructional design products in the appropriate category were used.

The results of instructional design tests on students in small groups showed there were about 85% of 20 respondents who stated that the learning methods applied in instructional designs greatly motivated students to learn, so that the target of learning achievement could be achieved. These results include face-to-face learning in class and independent learning processes. This statement is supported by data on the percentage of mastery learning in the field of competency tested (table 2), which on average reaches more than 95% in the posttest results. This result is higher than the results obtained at pretest which only reached 55%. This finding is in line with the findings of Jones et al. (2013) that many elements in PBL-based learning strategies provide opportunities to motivate student learning. This finding is also consistent with the findings of Fatima and Abdullah (2013) who concluded that PBL creates a learning environment that trains students to think, guides students to critically ask questions, and facilitates students' level of understanding to become deeper.

4. Conclusion

The PBL strategy model, which is applied comprehensively in the learning phases of software modeling, is effectively used as a learning strategy to reach the minimum competency standards planned in the learning design. However, in the trial model we found that some students seemed confused following each learning session, even though the lecture procedure was explained at the beginning of the lecture meeting, and several modules were provided as a guideline for lectures. Some students ask instructors to give more concrete explanations, even though they are at the end of the lecture session. Even after refining the strategy in the initial description of this PBL-based learning procedure, the results of the trial show that some students still do not understand the PBL concept in depth, so that their scores are still less than the ideal minimum adequacy standard. This finding is in line with the findings of Jones (2013) who questioned how long it took for PBL orientation to students so that students really understood and followed the PBL-based learning process correctly? This finding also supports the results of Schneider's study (2014) which has caused some open debate regarding PBL-based learning, namely: how many modules must be created for learning guidelines in PBL so that PBL can run effectively, students have the potential to experience confusion if there is not enough framework in early learning , and students who do not have relevant and adequate initial knowledge tend to be left behind.

The transition from conventional teaching methods (direct instruction) to PBL models instills a negative mindset towards PBL for students who are not familiar with inductive learning methods. We found some students to be passive after attending the learning session. Some students get bored following PBL's long procedures, and others feel burdened when they are asked to fill out learning forms in an orderly manner, which is new to them. This finding is in line with the statements of Harun, Yusof, Jamaludin, & Hassan (2012) that student motivation is the key to success in the application of problem-based learning (PBL). In this regard, we recommend examining the integration of the PBL model into instructional designs based on motivation in future studies.

Application support for Information Technology-based Content Management Systems / Moodle applications is not sufficient as an effective communication medium to support PBL-based learning strategies. Students need more effective 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. Technology features are also needed to support independent learning resources, as stated by Bahar & Soegiarto (2020) that Information technology-based learning media products are very suitable to be used as a source of independent learning. In this connection, further discussion is still needed on how a truly effective Information Technology (Information System) design model can support PBL-based learning strategies.

Application support for Information Technology-based Content Management Systems / Moodle applications is not sufficient as an effective communication medium to support PBL-based learning strategies. Students need more effective 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. Technology features are also needed to support independent learning resources, as stated by Bahar & Soegiarto (2020) that a good Information technology-based learning media product design is very suitable to be used as a source of independent learning.

The limitation of this study is that partial testing has not been carried out to determine which components have a significant impact on the application of instructional design, whether the instructional design product or PBL model is integrated in the instructional design. Future studies can be carried out in depth to examine this, in order to determine the right type of intervention for the two components that make up the instructional design system, if under certain conditions the results of implementing the instructional system do not have an effective impact.

Finally, we conclude that a long discussion that leads to different views on the application of PBL concepts in learning is still needed to provide opportunities for further research to find PBL strategies that are truly effective in learning, especially in learning of Modeling Software.

REFERENCES

[1] Akcay, B. (2009). Problem-based learning in science education. *Journal of Turkish science education, 6*(1), 28-38.

[2] 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.

[3] Bahar, Soegiarto. (2020). Development of Instructional Media Based on Mobile Technology to Enriching Teaching Material for Primary School Students in Indonesia Post-Learning in the Classrooms. *International Journal of Scientific & Technology Research*, 9(1), 94-98

[4] Bahar, Wibawa, B., & Situmorang, R. (2019). The Problem-Based Learning Procedural Model in the Software Modeling Course at the Information Technology College in Indonesia. Universal Journal of Educational Research, 7(9A), 99-105

[5] Dick, W., Carey, L., & Carey, J.O. (2014). *The Systematic Design of Instruction*, Eight Edition. USA: Pearson.

[6] Dickens, J. & Arlett, C. (2009). *A Hand Book for Teaching*

and Learning in Higher Education. New York: Routledge.

[7] 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.

[8] Ewing, B. (2011). Direct Instruction in Mathematics: Issues for Schools with High Indigenous Enrolments: A Literature Review. *Australian Journal of Teacher Education*, 36(5), 64-91.

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

[10] Fatima, S. & Abdullah, S. (2013). Improving Teaching Methodology in System Analysis and Design Using PBL for ABET. *International Journal of Modern Education and Computer Science*, (7), 60-68.

[11] 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.

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

[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. (2004). *Software Engineering 2004: Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering, The joint Task Force on Computing Curricula*, Association for Computing Machinery.

[15] 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.

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

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

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

[19] Moravec, M., Williams, A., & Aguilar-Roca, N. (2010). Learn Before Lecture: a Strategy that Improves Learning Outcomes in a Large Introductory Biology Class. CBE—Life Sciences Education, 9(4), 473-481.

[20] Murphy, C., Phung, D., & Kaiser, G. (2008). A distance learning approach to teaching extreme programming, *ITiCSE Proceedings of the 13th annual conference on Innovation and technology in computer science education,* Madrid, Spain, 199-203

[21] 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.

[22] 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).* IEEE, 267-272.

[23] Pressman, R.S. (2002). *Software Engineering* (Book 1). Yogyakarta: Penerbit Andi..

[24] Rahma, A., & Aulia R. (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.

[25] Richardson, I., & Delaney, Y. (2009). Problem based learning in the software engineering classroom. In *2009 22nd Conference on Software Engineering Education and Training*. India: IEEE, 174-181

[26] Schneider, D.K. (2006). *Instructional Design Models and Method*, http://tecfa.unige.ch/tecfa/teaching/formcont/webmaster2004/tie/pdf/files/tie-talk06.pdf; Accessed: October 21, 2019.

[27] Shim, C.Y., Choi, M. & Kim, J.Y. (2009). Promoting Collaborative Learning in Software Engineering by Adapting the PBL Strategy, *World Academy of Science, Engineering, and Technology*, (29), 1161-1164

[28] Suparman, M.A. (2014). *Desain Instructional Modern*. Jakarta: Erlangga.

[29] Syakdiyah, H, Wibawa, B., & Muchtar, H. (2018). The Effect of Flipped Classroom and Problem Based Learning Strategies in High School Chemistry Education. The Journal of Social Sciences Research, (2), 505-511.

[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] Wahono, S. (2012). *Strategi dan Model Pembelajaran*. Jakarta: Permata Puri Media.

[32] Warter-Perez, N., Dong, J. (2012). Flipping the classroom: How to Embed Inquiry and Design Projects into a Digital Engineering Lecture. In Proceedings of the 2012 ASEE PSW Section Conference, Washington DC, (39), 1-17.

[33] 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.

[34] Žagar, M., Bosnić, I., & Orlić M. (2008, May). Enhancing Software Engineering Education: A Creative Approach. In *Proceedings of the 2008 International Workshop on Software Engineering in East and South Europe.* ACM, 51-58.