

Development of the ITSJava Application as a Computer-Assisted Learning Media in Object-Oriented Programming courses

ABSTRACT

This research aims to develop ITSJava application as an interactive learning media for object-oriented programming (OOP) course in information technology education programme at Gorontalo State University. ITSJava was developed using the Personal Xtreme Programming (PXP) development method which consists of six stages: needs identification, planning, initial iteration, design, implementation, and system testing. The results of system testing show that the ITSJava application is suitable for use as a medium for learning OOP. This system can help students learn OOP in an interactive and adaptive way to students' learning styles. Hopefully, this system can be further developed by adding a unit test function to ensure the accuracy of the program code written by students and ensure that the code runs according to the expectations and logic expected by the lecturer.

Keywords: *Interactive learning media, intelligent tutoring system, Object-oriented programming (OOP), Personal extreme programming, ITSJava application.*

INTRODUCTION

Teaching programming in higher education, especially information technology and related study programs, providing the best programming teaching is still a question. Many learning models in programming teaching have been implemented, such as problem-based learning, project-based learning, and pair programming. The learning model applies the 4C approach (Creativity and innovation, Critical thinking and problem-solving, Communication, and Collaboration) and High Order Thinking Skills (HOST). Several programming paradigms are known by a programmer for creating computer software, one of which is the object-oriented programming paradigm (OOP). OOP is the best way to write computer programs to solve various problems (Stroustrup, 1988). The object-oriented paradigm describes problem domains in the real world in the form of objects and the interactions of these objects. The software that will be created is then written in program code based on the object model and object interactions (Poo & Ashok, 2008). The OOP paradigm offers the concept of modularity, code reuse, and ease of modifying program code. Java is the programming language that uses objects as the basis for writing code.

In the Information Technology Education study program at the Faculty of Engineering, Gorontalo State University, the Object-Oriented Programming (OOP) course uses JAVA, taught to 3rd-semester students. This course discusses the concept of object-oriented programming, problem analysis using object-oriented concepts, variables (declarations, initialization, and use of variables), operators, decision-making, looping, arrays, exception handling, creating methods, encapsulation, abstraction, polymorphism, and inheritance. The results of initial observations in the learning process carried out in the classroom; it was found that the lecturer teaching the OOP course, even though he had used the problem-based learning model in the practice of making programs, the learning approach in delivering material on programming concepts was still centered on the lecturer[14]. So, the learning method is still the lecture method, and program code examples are written directly following what the lecturer exemplifies. The learning media used are still e-book modules and PowerPoint slides. It causes the average student who takes OOP courses: (1) An average of 35% of students' learning results are below the expected results, especially for mid-semester and final semester exam scores; (2) Students do not understand and implement the material provided by the lecturer; (3) Students are less active in asking and consulting lecturers about material they do not understand; (4) In online learning, if students are given the task of reading material and structured assignments independently, students tend to be passive due to the absence of interactive and adaptive learning media that provides material according to student understanding; (5). Lecturers do not understand students' learning styles in the learning process, so the way of learning and delivering material to all students is still the same.

One solution to help students learn programming outside of lecture time is to develop interactive learning media using an intelligent tutoring system or an Intelligent Tutoring System (ITS). ITS is a computer-based adaptive learning system developed to compete with human tutors using artificial

intelligence, pedagogy, and psychology. It can accurately analyze students' knowledge, structure, and learning styles, decide what steps to take next, and provide responses in the form of feedback, as stated by Chughtai et al. (2015) and Elham et al. (2018). The last two studies were conducted by Schez et al. (2020), who developed a desktop application-based ITS that can assist the programming learning process by using notations based on road figures and traffic signs represented by 3D images in an augmented reality environment and research conducted by Castro et al., (2021) who developed ITS that is adaptive in giving assignments according to student's learning styles and abilities. According to Almasri et al. (2019), the ITS component consists of 4 main parts: the domain/expert model, student model, teacher/pedagogical model, and communication/user interface model. The domain/expert model contains concepts, rules, and problem-solving strategies. Domain/expert models can perform several tasks, such as being a source of expert knowledge, a standard for evaluating student performance, or detecting errors made by students. The teacher/pedagogical model receives information from the domain and learner models and makes choices about learning strategies and necessary actions. Figure 1 is an overview of the ITS conceptual model in general.

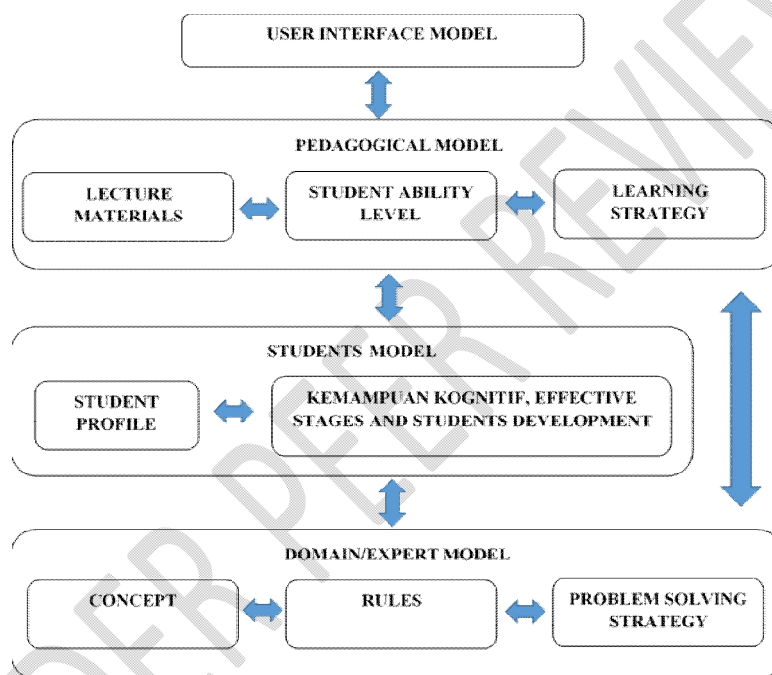


Figure 1. ITS conceptual model

Several studies on ITS have been carried out in recent years, such as those by Eryilmaz and Adabashi (2020), which developed a fuzzy Bayesian ITS that supports students' adaptive learning environment. The fuzzy Bayesian method is used in the domain model to detect student errors and evaluate student abilities. This system is compared with traditional e-learning, which is then evaluated to see whether there are differences in student learning performance. His research shows that students who use ITS get high average academic scores in a shorter time on the post-test compared to students who use traditional e-learning. Castro et al. (2021) develop and implement ITS applications that can encourage students to learn to experiment through tasks that require all the skills and abilities possessed by students, where these tasks are appropriate to the level of students' abilities. More specifically, in programming, Sanchez et al. (2020) created an ITS to help the learning process by using notation based on road figures and traffic signs represented by 3D images in an augmented reality environment on the system interface model. Students can utilize the feedback provided by ITS, and tutors can also use it to explain programming concepts in class. In its development, it can be integrated with collaborative learning; apart from enabling students to improve their learning experience, it can also help improve student learning outcomes. Intelligent tutoring systems based on collaborative learning provide students with a cognitive and collaborative assistance platform. Olsen et al. (2014) conducted research comparing collaborative and individual learning methods in receiving knowledge both procedurally and conceptually in ITS. The research

results show that students using the collaborative learning method get the same learning results as students using the individual method. However, students using the collaborative method can solve problems more quickly. Therefore, integrating collaborative learning program code into an intelligent tutoring system can make learning more effective, even for younger students.

ITS is a type of computer-based learning (CBL). CBL is similar to the computer-based learning environment (CBLE), which refers to the software students use and their interactions during learning. The main characteristic of CBL is its interactivity, so CBL is different from other learning media, such as presentations, video and audio, and so on. Furthermore, CBL places great emphasis on independent learning through the active involvement of students in a set of pedagogically designed digital activities. The function of CBL replaces the role of teachers outside the classroom, and in many situations, CBL is carried out in the classroom as part of the additional curriculum (Grizioti&Kynigos, 2019).

Based on the abovementioned problems, this research aims to develop an ITS application as a computer-assisted learning medium for object-oriented programming in the information technology education major at Gorontalo State University. The ITS application will later be named ITSJava and implemented as a web-based application that students can access anywhere and at any time.

METHOD

Develop ITSJava applications using the Personal Xtreme Programming (PXP) system development model. According to Dzurov et al. (2009), the PXP system development model is a software development model modified from the extreme programming (XP) development model, with the advantage of being able to carry out software development individually. According to Iyawa (2020), the PXP development model is a development of the XP method and Personal Software Process (PSP) to develop quality software in a short time even though the software is done individually. The stages in developing ITS applications based on collaborative learning that pay attention to student learning styles are shown in Figure 2 below:

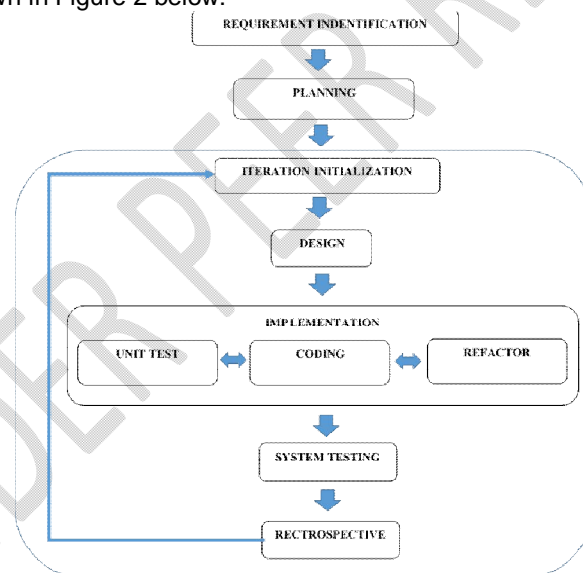


Figure 2. Stages of personal extreme programming (Dzurov et al. 2009)

1. At the ITSJava requirements identification stage, the system's functional and non-functional requirements are documented. The identification of needs is obtained through interviews and questionnaires. The results of documenting these needs are written as user stories (Ulfi et al., 2020).
2. At the planning stage, the tasks to be carried out are sorted based on the sequence of system requirements documents obtained previously.
3. Initialization iteration is a stage that indicates the beginning of program creation.
4. The design stage involves modeling the system's modules and classes that will be applied in the current iteration.
5. The implementation stage is the core stage where program creation is carried out. This activity consists of three activities: unit testing, coding, and refactoring.
6. The system testing stage is the system testing stage that was created during the overall implementation stage. The system testing method uses black box testing.

- The retrospective stage is the end of the program creation iteration. If the user still has new system requirements during program creation, the process will return to the initialization iteration stage.

After the application was developed, two lecturers who taught the OOP course tested its functionality as a learning medium. This test assesses whether the ITSJava application is suitable for implementation.

RESULTS AND DISCUSSION

Identify Needs

Functional and non-functional requirements for the ITS Java application were obtained based on needs identification. The functional requirements of the system obtained are written in the form of a user story with the format "As <type of user>, I want to <perform a certain action> so that I can <benefit from that action>". After carrying out the requirements process, the developer gets eight user stories. User stories and non-functional system requirements that have been obtained can be seen in Table 1 and Table 2.

Table 1. List of ITSJava Application User Stories

User Code	Stories	Students User Stories
Story-01		As a student, I want to be able to register in the system and use the application.
Story-02		As a student, I want to fill out a learning style questionnaire and see the results so the system can automatically use its expert system algorithm to determine learning material according to my learning style.
Story-03		As a student, I want to be able to read the material provided, answer quizzes, and do exercises independently or collaboratively with classmates to complete programming practice questions so that the system can display the results of analyzing my behavior while using the application.
Story-04		As a student, I want a system with artificial intelligence methods that can automatically determine my level of knowledge and behavior while using it so that it can decide whether or not I am worthy of continuing to the next material.
User Code	Stories	Lecturer User Stories
Story-5		As a lecturer, I want to manage everything related to the material so that I can enter, delete, and edit learning material.
Story-6		As a lecturer, I want to manage student profiles so that I can change, delete, and edit student profiles.
Story-7		As a lecturer, I want to be able to manage practice questions so that I can enter, delete, and edit them.
Story-8		As a lecturer, I want to be able to see student behavior analysis so that I can see students' understanding of the material provided

Table 2. List of non-functional requirements for the ITSJava application

Software	Windows/Linux/macOS Browser chrome/firefox/windows edge/safari Web server Xampp
Hardware	Minimum processor core i3 Minimum RAM 8 GB Minimum Hardisk 256 GB

Planning

After the requirements were identified, the planning stage was carried out. This process involves estimating user stories, determining user story priorities, and planning iterations. The priorities, estimated completion times, work order for each application user story, and the number of iterations required during system development were obtained from this stage.

The story point is an estimate of the completion time, which is directly proportional to the length of time required. The estimated completion time for one story point is three days. User story sorting determines the priority of user stories based on value and risk. The value of a user story is determined based on the business value of that user story. This value is divided into critical, significant business, and friendly. Meanwhile, the risk from a user story is determined based on the risk index, which is

based on completeness, volatility, and complexity. The total risk index determines the level of risk of each user story, which ranges from low (0 to 1), medium (2 to 4), and high (5 to 6). Iteration planning contains the user stories that will be worked on during the process. The value of 6 for velocity in each iteration is determined based on the results of discussions with the client. The value of 6 indicates a completion time of 18 days. The total number of user stories is 8, which will be worked on for 144 days. The results of iteration planning can be seen in Table 3.

Table 3. Results of ITSJava Application Iteration Design

Iteration	User Stories Code	Value	Risk	Story Point
Iteration 1	Story-02	Critical	(5)High	6
Iteration 2	Story-03	Critical	(5) High	6
Iteration 3	Story-04	Critical	(5) High	6
Iteration 4	Story-01	Critical	(1) Low	1
	Story-05	Critical	(3) Medium	1
	Story-06	Critical	(1) low	1
	Story-07	Critical	(3) Medium	2
	Story-08	Critical	(1) Low	1

Design

After the planning, the design was carried out. Based on the planning results, the developer used Class Responsibility Collaboration (CRC) Cards and a spike solution prototype. CRC cards describe the tasks of each class and its interaction with other classes. The spike solution prototype is a prototype design scheme based on CRC cards from user stories. Table 4 shows the list of CRC cards and spike solutions from the ITSJava application.

Table 4. List of CRC Cards and Spike Solution for the ITSJava Application

Iteration	User Story Code	CRC Cards	Spike Solution Prototype
Iteration 1	Story-02	CRC Learning Style Card	Prototype for Managing Learning Styles
Iteration 2	Story-03	CRC Learning Card	Prototype for Reading Material and Completing Exercises
Iteration3	Story-04	CRC Knowledge Level Automation Card	Prototype for Knowledge Level Automation
Iteration 4	Story-01	CRC Registration and Login Card	Prototype for Registration and Login
	Story-05	CRC Material Card	Prototype for Managing Content
	Story-06	CRC Students profile Card	Prototype for Student Profiles
	Story-07	CRC Exercise profile Card	Prototype for Managing Exercises
	Story-08	CRC Analysis Card	Prototype for Student Behavior Analysis

After the CRC Cards and spike solution are created, a use-case diagram of the ITSJava application is made. A use case diagram is a diagram that illustrates what users can do in the system. The use case diagram is also used to model all business processes from the perspective of system users. This diagram consists of actors/users and use cases. The actors of the ITSJava application consist of two users, namely students and lecturers, who are admins. Meanwhile, the use cases in this ITS application consist of the use case of filling out a learning style questionnaire, the use case of reading material, the use case of doing collaborative practice, the use case of managing student profiles, the use case of answering quizzes, the use case of viewing the results of behavior analysis while using the application, and the use case of managing materials and practice questions. The use case diagram of the ITSJava application is shown in Figure 3.

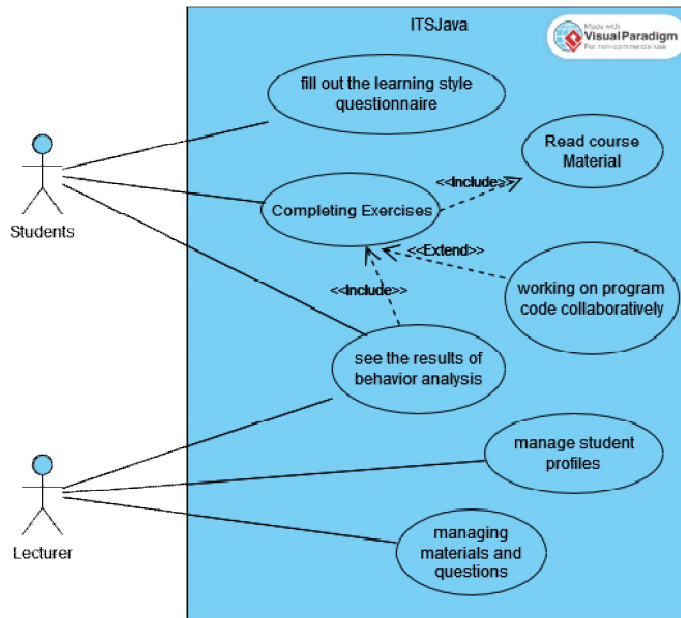


Figure 3. ITSJavaApplicationUse-case diagram

Implementation

The implementation phase was carried out over four iterations, each planned to be completed within 18 days. This phase is further divided into three stages: Unit Testing, Code Generation, and Code Refactoring. Here are the results for unit testing, code generation, and code refactoring.

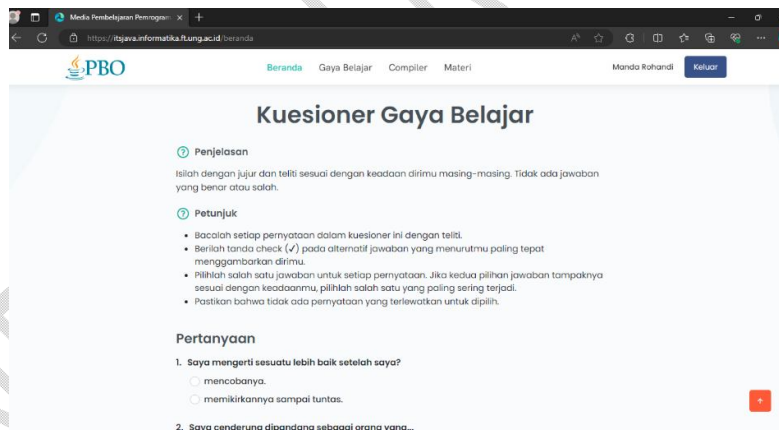


Figure 4. Learning style questionnaire page

Students must answer all questions on this learning style questionnaire page. If there are any unanswered questions, the system will not save the questionnaire, and the student will not be allowed to proceed to view the material. After the student completes the learning style questionnaire, the system will display the student's learning style results, as shown in Figure 5.

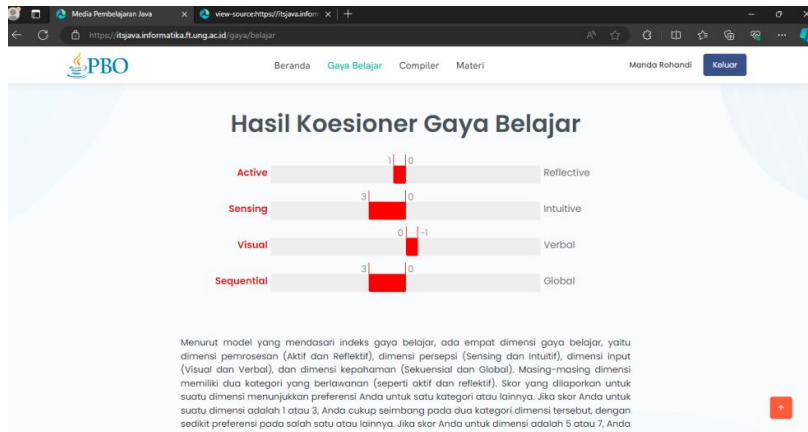


Figure 5. Student learning style questionnaire results page
 After the student completes the learning style questionnaire, the system directs them to the learning material page, as shown in Figure 6.

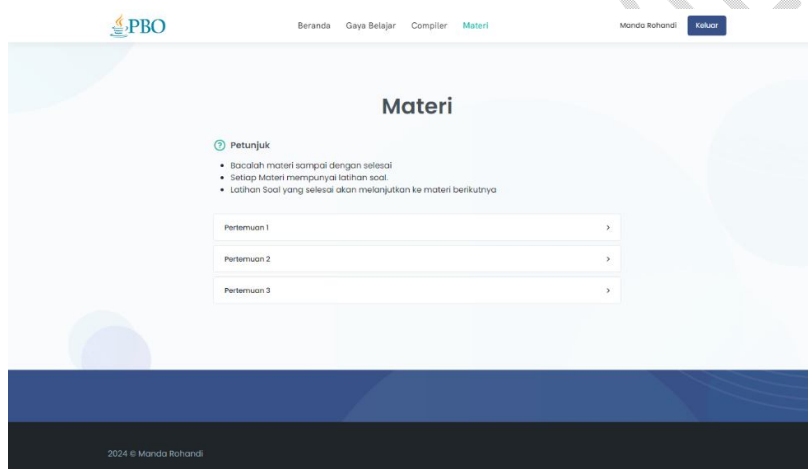


Figure 6. Learning material menu page

Next, the system will automatically evaluate and determine the form of material to be given to students based on the learning style results obtained. The material presented by the system is displayed in the form of presentation files and videos. It is presented in stages, where students can only start viewing material from the first meeting and can only proceed to the next meeting if the material has been read and each practice question worked on has met the minimum standard. The first material page, which is in the form of presentation files and videos, is shown in Figure 7.

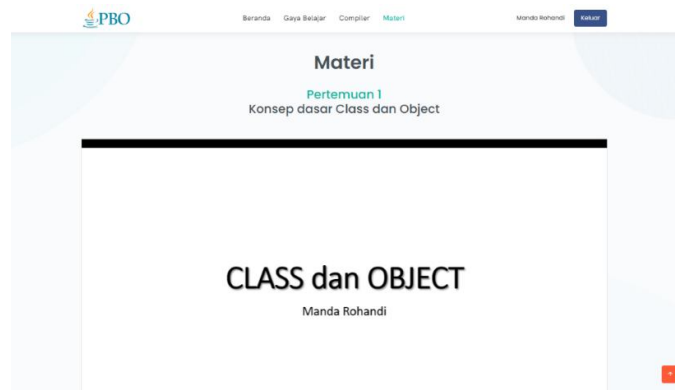




Figure 7. Learning material page

After reading the material or watching the learning video, students are directed by the system to work on practice questions/quizzes in the form of multiple choice. The practice question/quiz page is shown in Figure 8.

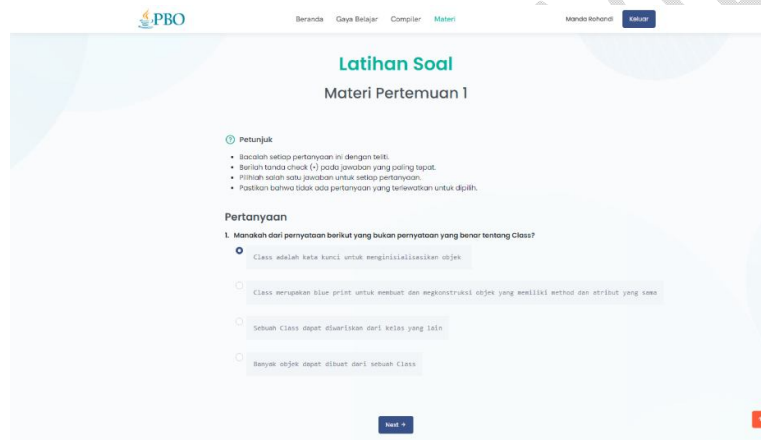


Figure 8. Practice question/quiz page

After completing the practice questions/quiz, students can see the results of the practice questions/quiz they have worked on. The practice question/quiz work results page is shown in Figure 9.

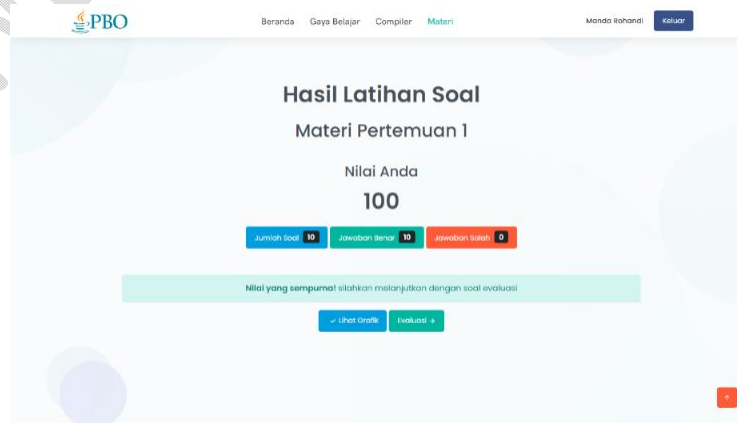


Figure 9. Practice question/quiz results page

The form of student collaboration on this page is the interaction between two students in the form of chat to communicate in solving problems on practice questions and a "live coding" editor where the first student can immediately see changes to the program code written by the second student, and vice versa. The collaboration page for program creation practice is shown in Figure 10.

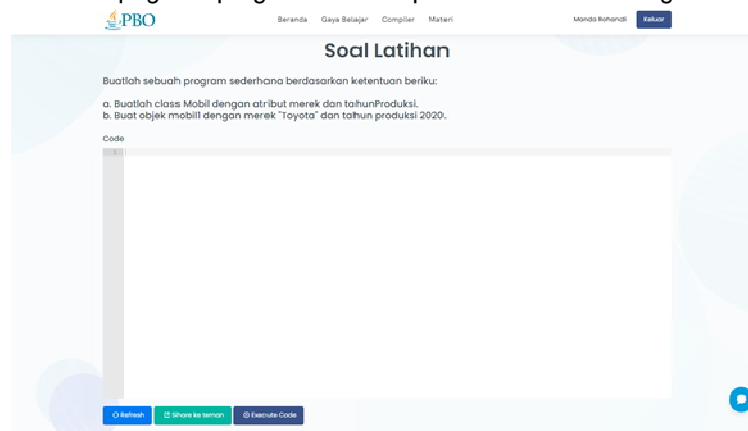


Figure 10. Collaboration page for program creation practice

After students work on the program creation practice questions, the system evaluates and decides whether they should repeat the material being studied or proceed to the next material. The system also displays an analysis of the student's behavior during the learning process. The page for evaluating learning outcomes and analyzing student behavior during the learning process is shown in Figure 11.

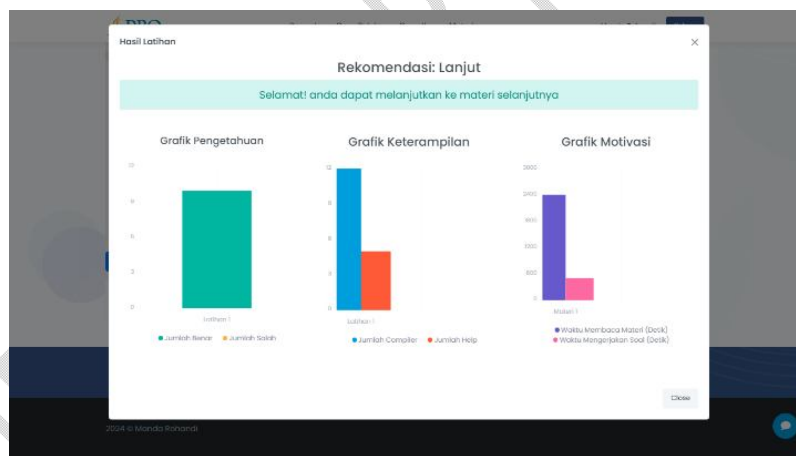


Figure 11. Page for evaluating learning outcomes and analyzing student behavior

System testing

After the program code has been created, the functionality of all implemented features that have been carried out during the iteration process will be tested by the client. The system is hosted so that it can be tested by clients located in different locations. A remote system connects The user's device to the developer's device. This aims to allow the developer to monitor the user's activities on their device and make it easier for the developer to direct the user to the system functions worked on during the iteration process. System testing uses black box testing to see if each system functionality works appropriately so the client can accept it. The test results show that the client can accept all iterations tested. The test results can be seen in Table 5.

Table 5. Results of ITSJava Application Testing

Iteration	User Stories Code	Test Result
Iteration 1	Story-02	Accepted

Iteration 2	Story-03	Accepted
Iteration 3	Story-04	Accepted
Iteration 4	Story-01	Accepted
	Story-05	Accepted
	Story-06	Accepted
	Story-07	Accepted
	Story-08	Accepted

CONCLUSION

Based on the research conducted during the development of ITSJava, the ITSJava application is suitable for use as a learning medium in the Object-Oriented Programming (OOP) course. This is evident from the results of the system tests, which show that the users accept all the required system functionalities. ITSJava can assist students in learning OOP, as the application provides collaborative learning and adapts to students' learning styles. Hopefully, This system can be further developed by adding a unit test function to check the accuracy of the code written by the students, ensuring that the code runs as expected and conforms to the logic expected by the lecturer.

Disclaimer (Artificial intelligence)

Option 1:

Author(s) hereby declare that NO generative AI technologies such as Large Language Models (ChatGPT, COPILOT, etc) and text-to-image generators have been used during writing or editing of manuscripts.

Option 2:

Author(s) hereby declare that generative AI technologies such as Large Language Models, etc have been used during writing or editing of manuscripts. This explanation will include list the name, version, model, and source of the generative AI technology and as well as the all input prompts provided to a generative AI technology

Details of the AI usage are given below:

- 1.
- 2.
- 3.

REFERENCES

1. Almasri, A., Ahmed, A., Al-Masri, N., Sultan, Y.S., Mahmoud, A.Y., Zaqout, I., Akkila, A.N., & Abu-Naser, S.S. (2019). Intelligent Tutoring Systems Survey for the Period 2000- 2018.
2. Castro-Schez, J.J., Glez-Morcillo, C., and Vallejo, D. (2021), An Intelligent Tutoring System for Supporting Active Learning: A Case Studi on Predictive Parsing Learning. *Journal of Information Science*, Vol. 544, pp. 446-468.
3. Chughtai, R., Zhang, S. and Craig, S. (2015). "Usability evaluation of intelligent tutoring system". *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 59, No. 1, pp. 367-371. <https://doi.org/10.1177/1541931215591076>.
4. Dzhurov, Yani & Krasteva, Iva & Ilieva, Sylvia. (2009). *Personal Extreme Programming—An Agile Process for Autonomous Developers*.
5. Elham Mousavinasab, Nahid Zarifsanaiey, Sharareh R. NiakanKalhori, Mahnaz Rakhshan, Leila Keikha & Marjan Ghazi Saeedi (2018): Intelligent tutoring systems: asystematic review of characteristics, applications, and evaluation methods, *Interactive Learning Environments*, DOI: 10.1080/10494820.2018.1558257
6. Eryilmaz, M. and Adabashi, A. (2020). Development of an Intelligent Tutoring System Using Bayesian Networks and Fuzzy Logic for a Higher Student Academic Performance. *Appl. Sci.* 10, no. 19: 6638. <https://doi.org/10.3390/app10196638>
7. Grizioti, Marianthi & Kynigos, Chronis. (2020). *Computer-Based Learning, Computational Thinking, and Constructionist Approaches*. 10.1007/978-3-319-60013-0_75-1.

8. Iyawa, G.E. (2020). Personal Extreme Programming: Exploring Developers' Adoption. AMCIS.
9. Olsen, J. K., Belenky, D. M., Alevan, V., & Rummel, N. (2014). Using an intelligent tutoring system to support collaborative as well as individual learning. In International conference on intelligent tutoring systems (pp. 134-143). Springer, Cham. Chicago.
10. Poo, D., Kiong, D. and Ashok, D. (2008). Object-Oriented Programming and Java Second Edition. London. Springer-verlag.
11. Stroustrup, Bjarne. (1988). What is "Object-oriented Programming"? Software, IEEE. 5. 10 - 20. 10.1109/52.2020.
12. Schez-Sobrino, Santiago, Cristian Gmez-Portes, David Vallejo, Carlos Glez-Morcillo, and Miguel Á. Redondo. (2020). "An Intelligent Tutoring System to Facilitate the Learning of Programming through the Usage of Dynamic Graphic Visualizations" Applied Sciences 10, no. 4: 1518. <https://doi.org/10.3390/app10041518>
13. Ulfi, Muhammad &Marthasari, Gita &Nuryasin, Ilyas. (2020). Implementation of the Personal Extreme Programming Method in Developing a Company Transaction Management System (Case Study : CV. TodjoeSinar Group). Journal Repositor. 2. 261. 10.22219/repositor.v2i3.619.
14. Bajcsy P, Kooper R, Lee SC. Understanding documentation and reconstruction requirements for computer-assisted decision processes. Decision support systems. 2010 Dec 1;50(1):316-24.