Vol 13 No 1 : January 2025

JDPP

Jurnal Dimensi Pendidikan dan Pembelajaran

ISSN 2303-3800 (Online), ISSN 2527-7049 (Print)

https://journal.umpo.ac.id/index.php/dimensi/index

Utilizing Serious Games as an Interactive Platform for Programming Learning in Informatics Students

Manuel Yosia Krista Ananda Wicaksono [a,1], Noven Indra Prasetya [a,2*]

[a] Universitas Wijaya Kusuma Surabaya, Indonesia

[1] manuelyosia44@gmail.com; [2] noven@uwks.ac.id *

* corresponding author

Article Information

Article History:

Received December 2024

Accepted January 2025

Published January 2025

Keywords:

Serious game;

Game-based learning;

Interactive platform;

Programming learning;

Programming.

How to Cite:

Wicaksono, M. Y. K. A., *et al.* (2025). Utilizing Serious Games as an Interactive Platform for Programming Learning in Informatics Students. *Jurnal Dimensi Pendidikan dan Pembelajaran Universitas Muhammadiyah Ponorogo*, 13(1), pp 149-160.

Abstract

Programming is a critical skill for informatics students, enabling them to design systems, develop software, and analyze user requirements. However, many students face significant challenges in understanding programming concepts and syntax, particularly during the initial stages of learning. These difficulties often stem from limited proficiency in reading, debugging, and writing code, coupled with low motivation and engagement in the learning process. To address these challenges, it is essential to adopt engaging and enjoyable instructional approaches. Serious games have proven to be an effective instructional method, enhancing students' motivation and engagement through game-based elements. This study introduces DolananCoding, a serious game developed as an interactive platform for teaching structured programming courses. DolananCoding incorporates two key features: live coding and a leaderboard, both designed to boost students' motivation and engagement in programming learning. The platform was rigorously evaluated through two testing methods: system testing and user testing. System testing validated the fulfillment of functional requirements, including modules, materials, and experience points, all of which met the expected standards. User testing assessed the platform's user satisfaction through a questionnaire administered to 76 students, yielding an average satisfaction score of 4.3 out of 5.0. These results indicate a positive reception among students, demonstrating the platform's readiness for integration into programming learning.

INTRODUCTION

Programming learning for informatics students has become a critical component of higher education, particularly in the field of information technology. Programming is not merely a technical skill essential for the industrial world but also serves as the foundation for developing various applications and systems that support modern life (Vaca-Cárdenas et al., 2015). However, despite its substantial importance, many students encounter significant challenges in understanding programming concepts (Ekohariadi et al., 2018; Topalli & Cagiltay, 2018). The abstract and technical nature of the material often leaves students feeling disconnected, frustrated, and demotivated. These difficulties can hinder the learning process and potentially lower their academic performance (Leksonowati, 2019). Consequently, efforts to enhance students' understanding and motivation in programming learning are vital, particularly by adopting innovative learning technologies (Asmaroini, 2021; Indraswati et al., 2020).

One approach that has gained significant attention is the use of serious games in education. Serious games, designed with educational objectives in mind, have been shown to enhance student engagement and motivation across various disciplines (Caserman et al., 2020; Lee et al., 2022). In the context of programming learning, serious games offer a more interactive and enjoyable learning experience, enabling students not only to grasp programming concepts but also to apply their knowledge through simulations and challenges embedded within the games. This approach aligns with the principles of game-based learning (GBL), a pedagogical framework that is increasingly popular in higher education (Hidayat et al., 2023; Phillips et al., 2014). Recent studies indicate that GBL can significantly boost students' motivation and engagement, leading to improved learning outcomes, particularly in subjects perceived as difficult or tedious, such as programming (Shah et al., 2019; Xie et al., 2022).

Despite the significant potential of serious games in programming learning, several challenges remain to be addressed. One major issue is the low levels of student engagement and motivation, which continue to hinder the learning process. Traditional teaching methods, often theoretical and less engaging, are insufficient to fully overcome these challenges (Hijrah et al., 2020). For instance, even when students are provided with adequate theoretical material, they often struggle to connect these concepts with real-world applications. Educational games, designed with specific learning objectives, can offer a more practical and enjoyable context to address this issue (L. D. Putra et al., 2024). However, while some studies highlight the benefits of game-based learning (GBL) in education, many existing games fail to meet the specific needs of programming learning or are inadequately tailored to the characteristics and abilities of the students.

The literature highlights that implementing serious games can be an effective solution to enhance student engagement and motivation in programming learning. Existing studies suggest that serious games can create a more interactive and enjoyable learning environment (D. A. T. B. Putra et al., 2022), encouraging students to be more motivated to learn and complete programming tasks more effectively. For example, research by Khan and Madden demonstrates that students engaged in programming learning through gamified activities tend to achieve higher success rates in mastering complex programming concepts (Khan & Madden, 2016). However, despite the numerous advantages offered by game-based learning (GBL), many existing games fall short in designing learning experiences that accommodate diverse student skill levels and often fail to provide sufficient opportunities for in-depth exploration of the material.

This study aims to develop DolananCoding, a serious game designed to be more relevant and effective in the context of programming learning. By creating a game that adapts to students' skill levels and provides deeper challenges related to programming concepts, this research seeks to address gaps identified in the existing literature. Additionally, the study explores how serious games can enhance students' understanding of programming concepts and boost their motivation to continue learning. The novelty of this research lies in its comprehensive approach to designing a serious game that not only presents engaging challenges but also offers constructive feedback to help students overcome difficulties in programming learning.

The benefits of utilizing serious games for programming learning are multifaceted. Firstly, serious games enhance student engagement and motivation by incorporating game-based elements that make learning more enjoyable and interactive. This approach addresses the common issue of low motivation and engagement in traditional programming courses, which often rely on theoretical and less engaging methods. By providing an immersive and interactive learning environment, serious games can help students better understand and retain programming concepts.

The urgency of this research lies in the increasing demand for proficient programmers in the information technology industry. As programming becomes an essential skill for various applications and systems, it is crucial to find effective ways to teach and motivate students to excel in this field. The traditional methods of teaching programming have proven to be insufficient in fully engaging students and helping them overcome the challenges associated with learning programming concepts and syntax. Therefore, innovative approaches, such as serious games, are necessary to bridge this gap and improve the overall effectiveness of programming education.

Previous research has highlighted the potential of serious games in enhancing student engagement and motivation across various disciplines. Studies have shown that game-based learning (GBL) can significantly improve learning outcomes, particularly in subjects perceived as difficult or tedious, such as programming. For example, research by (Khan & Madden, 2016) demonstrated that students engaged in programming learning through gamified activities tend to achieve higher success rates in mastering complex programming concepts. However, many existing games fall short in designing learning experiences that accommodate diverse student skill levels and often fail to provide sufficient opportunities for in-depth exploration of the material.

Overall, this study aims to contribute to the development of programming learning methodologies by leveraging innovative learning technologies, particularly serious games. The primary focus of the research is to design, implement, and evaluate the platform through three testing methods: unit testing, system testing, and user testing, to assess the effectiveness of the serious game in the context of programming learning.

METHODS

Research Design

This study adopts the software development approach using the waterfall model, as described by (Abba et al., 2019; Tuakia & Prasetya, 2021). This model consists of five sequential stages that must be completed in order:

Requirement Analysis

Requirements analysis is a critical initial step in the platform development process. This stage aims to identify and understand the features that need to be implemented in the platform to effectively address user challenges. The methods employed for requirements analysis include in-depth interviews with students and lecturers, focus group discussions, and direct observations of processes or issues encountered. This approach ensures that user needs are thoroughly accommodated and that the designed features are both relevant and precisely aligned with the platform's development objectives.

System Design

In the system design phase, planning is carried out to include game flow, interface design, and database architecture. The game flow is designed using diagrams as tools to facilitate structured system implementation. Interface design is created using Figma, aiming to provide a clear visual representation of the platform's appearance before the implementation stage. Meanwhile, the database design involves planning the tables to be constructed and determining how game data will be efficiently managed and stored. This phase ensures that all system components are well-designed and aligned with the identified requirements.

Implementation

The implementation phase is the realization of the previously planned system design. This process involves the use of various software tools, including Unity, Visual Studio, XAMPP, and Visual Studio Code. The programming languages utilized for platform development are C# for game logic and PHP for server-side data processing. Unity serves as the primary software for the platform's development during this phase.

Testing

After the implementation phase is completed, the application undergoes testing to ensure that the platform operates and functions as intended. Two types of testing are conducted: system testing and user testing. System testing employs the black-box testing method to evaluate the software's functionality without examining its source code, ensuring that each feature performs as expected. Meanwhile, user testing aims to gather direct feedback from a group of students using the End-User Computing Satisfaction (EUCS) questionnaire, adapted from (Purwanto & Hedin, 2020). This questionnaire evaluates various aspects of the platform, including content, format, ease of use, timeliness, and accuracy.

Research Respondents

This study involved respondents comprising second- and fourth-semester students from the Informatics Study Program at Universitas Wijaya Kusuma Surabaya, totaling 76 students. The research was designed to evaluate student satisfaction with the platform as a tool for programming learning. Over a five-week observation period, students participated in hybrid learning activities that combined online and face-to-face instruction. During this period, students actively explored various platform features, completed interactive exercises, and tackled programming challenges designed to test their skills.

Data Collection

In the fifth week, students were asked to complete the End User Computing Satisfaction (EUCS) instrument using a 5-point Likert scale to evaluate their satisfaction with the platform. The EUCS instrument, developed by (Purwanto & Hedin, 2020), assesses five key aspects: content, format, ease of use, timeliness, and accuracy. As the basis for designing the questionnaire, this study adopted and adapted 12 questions relevant to the EUCS dimensions. Each question was translated into Indonesian to ensure respondents could better understand the context and content of the questions.

Data Analysis

This study adopted both quantitative and qualitative approaches for data collection and analysis. Quantitative data were gathered through validation sheets and questionnaires completed by students, which were then analyzed using descriptive statistics to depict satisfaction levels with the platform. Aspects such as content, format, ease of use, timeliness, and accuracy were measured using a Likert scale, providing numerical insights into students' perceptions. Additionally, qualitative data, including comments, suggestions, and feedback from students, as well as notes on the platform's performance, were utilized to identify areas requiring improvement.

RESULTS AND DISCUSSION

The results and discussion section of this study provides a detailed explanation of the findings derived from the processes of requirements analysis, system design, implementation, and testing phases.

Requirements Analysis

The development of the platform resulted in a web-based system specifically designed to support C++ programming learning. This platform provides an interactive game-based learning experience, aiming to enhance student engagement and interest, particularly in structured programming courses. Within the system, students, acting as players, can select a customizable avatar to represent them as they progress through various missions within the game. Each mission and challenge is strategically designed to encourage players to solve problems by engaging in live coding. The tasks and instructions provided require players to apply programming logic effectively to overcome the challenges, fostering a practical understanding of programming concepts through an immersive and interactive approach.

System Design

The system design phase in this study is divided into two main components: system flow design and database design.

System Flow

The system flow of this platform outlines the sequence of operations and interactions, including how users or players navigate through various components and processes, as illustrated in Figure 1. The system flow is categorized into four main stages: user registration and login, access and gameplay, code execution and feedback, and progress tracking and analysis. Each stage is designed to ensure a seamless and engaging user experience while supporting the platform’s educational objectives.

1. User Registration and Login
   a. Users or players register and log in to the platform through a dedicated user interface designed for accessibility and ease of use.
   b. The authentication service verifies user credentials to ensure authenticity and authorize access to the platform securely.
2. Access and Gaming
   a. Players select programming challenges or levels through the game interface, allowing them to navigate and engage with tasks tailored to their skill level.
   b. The game logic server processes requests, retrieves relevant game data, and presents the challenges to the players, ensuring a dynamic and responsive gameplay experience.
   c. Players write and submit programming code using an integrated code editor.

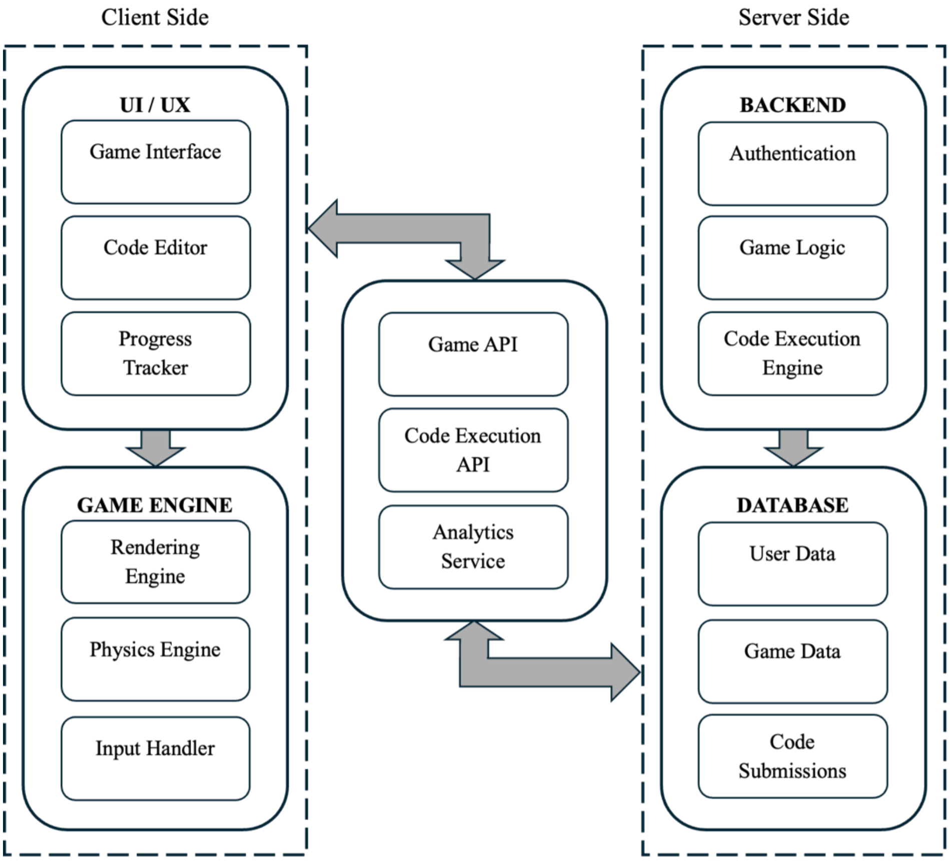


Figure 1. Platform System Flow

3. Code Execution and Feedback
   a. The code submitted by players is forwarded to the code execution engine via an API.
   b. The code is executed in a sandbox environment to ensure security and execution isolation.
   c. The execution results and feedback are sent back to the players to provide an evaluation of their submitted solutions.
4. Progress Tracking and Analysis
   a. Data related to players' progress and performance is stored in the platform's database.
   b. The analytics service processes this data to generate insights, reports, and recommendations tailored to the players' learning needs.

This workflow structure is designed to ensure an intuitive and systematic user experience while effectively supporting programming learning through the serious game platform.

Database Design

The platform's database is designed to dynamically store all data, encompassing various tables, including the user table and the challenge table. The user table plays a crucial role in storing user data, such as identities, credentials, and other relevant information. Additionally, this table is used to determine the maps and levels unlocked based on the user's progress, thereby influencing the game's interface display.

Table 1. User

Field	Data Type	Length	Information
id	int		Primary Key
username	varchar	20	
email	varchar	50	
password	varchar	20	

Field	Data Type	Length	Information
nama_depan	varchar	20	
nama_belakang	varchar	20	
poin	int		
level	int		
map	int		

The user table also records the points earned by users, which are then converted into badges. These points and badges serve as indicators of user achievement and are displayed on the leaderboard feature. This feature is designed to boost user motivation through elements of competition and recognition of their accomplishments within the platform.

Table 2. Challenge

Field	Data Type	Length	Information
id	int		Primary Key
username	varchar	20	
materi	varchar	200	
answer	varchar	200	

The challenge table is designed to store data related to the available challenges on the platform, including challenge content and correct answers. This data serves as the primary reference when users access the in-game page. On this page, the system retrieves data from the challenge table to display the challenge description, instructions, and the materials that the user must fulfill. Additionally, the stored answers are used as a reference to verify the results submitted by the user, ensuring that each challenge is evaluated automatically and accurately.

Implementation

The implementation phase aims to realize the concepts and designs that have been developed into a functional product ready for use by the users. This platform is developed using Unity as the game engine, with C# programming language for constructing the game logic. All game assets, including characters, textures, animations, and audio, are imported into Unity, and game components are added to create a more realistic and immersive gameplay experience.

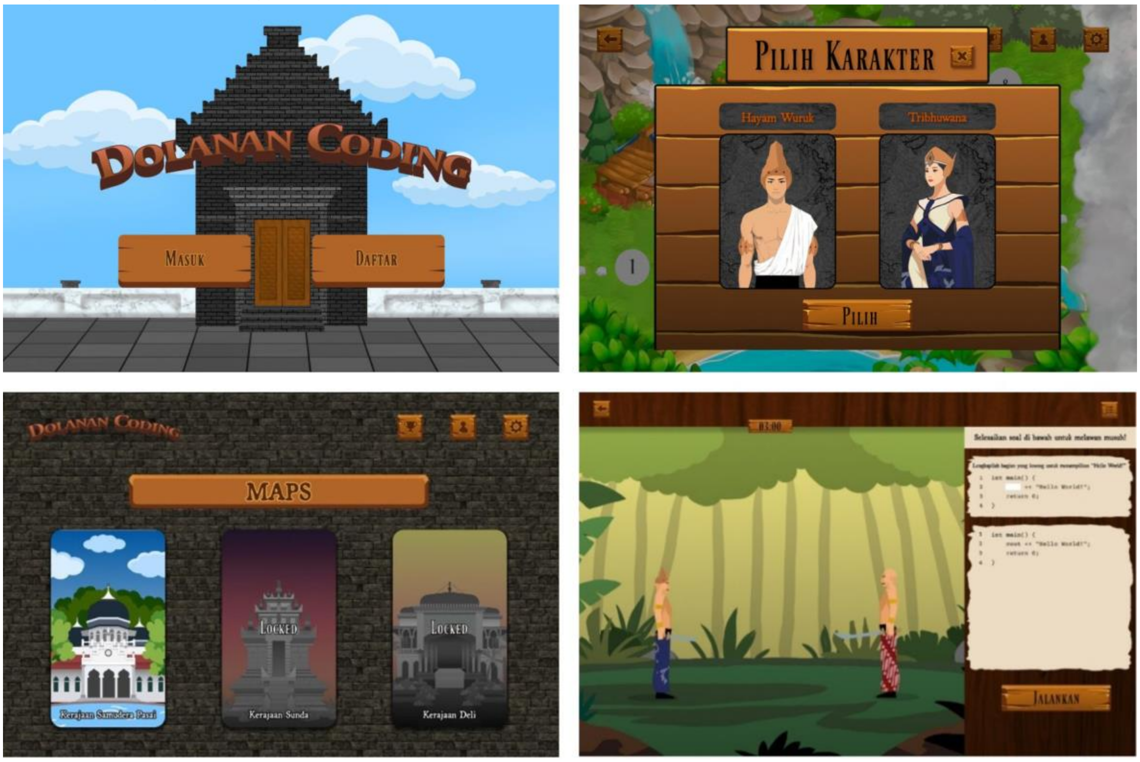
Game controls, such as movement, attacks, and interactions, are tailored to computer input devices to ensure that characters or objects in the game can be played optimally. The game logic includes setting up rules, gameplay mechanics, and game systems such as level progression, scoring, inventory, and data storage systems.

In addition, the target platform is selected before the build process to ensure that the game distribution aligns with the research objectives, which is to have a platform that can be run on the web. This step ensures that the platform is easily accessible to users and effectively supports the learning process. Table 3 presents the software specifications used during the implementation stage.

Table 3. Software Specifications

Software Name	Specification
Operating system	Windows 11 64-bit
Game Engine	Unity
Database	MySQL
Programming	C#Script
Web Component	WebGL
Code Editor	Visual Studio Code

The implementation resulted in the development of a web-based programming learning platform, as depicted in Figure 2. This platform includes several main pages, such as the homepage, login, new account registration, map selection, character selection, level selection, and the play area. Each of these pages is designed with the objective of creating an intuitive user experience while supporting an interactive programming learning process. The design ensures seamless integration between the pages, offering users an easy and responsive navigation system to enhance the overall learning experience. The interconnectedness of these elements fosters a user-friendly interface that facilitates the engagement and progression of students in their programming learning.



Selesaikan soal di bawah untuk melawan musuh!

Lengkapilah bagian yang kosong untuk menampilkan "Hello World"

```
int main() {
          << "Hello World!";
    return 0;
}
```

```
int main() {
    cout << "Hello World!";
    return 0;
}
```

Figure 2. Platform Interface

Testing

This study conducted platform validation through two types of testing: system testing and user testing. System testing was carried out to ensure that all functional requirements of the platform were met. This process employed the black-box testing method, which focuses on testing the main functions without examining the internal code structure. Meanwhile, user testing involved a group of students as participants to evaluate various aspects of the platform, including content, format, ease of use, timeliness, and accuracy. The results of both testing types provided a comprehensive overview of the platform's technical performance and user experience in supporting programming learning. This dual approach of validation offers valuable insights into the platform's effectiveness from both a functional and user-centric perspective.

Table 4. System Testing

Module	Expectation	Result
Material	The platform effectively presents C++ programming materials in an interactive and user-friendly manner, ensuring that users can easily understand and engage with the content.	Valid
Experience Points	The platform is capable of automatically calculating experience points based on the difficulty level of the challenges being completed by the user.	Valid
Progress	The system can calculate player progress as a percentage, reflecting how much of the available challenges or levels the player has completed.	Valid

The results of the system testing can be seen in Table 4, which utilizes the black-box testing method. This validation aimed to ensure that all functional requirements of the system were met in accordance with the established expectations. The system testing covered various aspects, such as the learning content, experience point calculation, and user progress tracking within the game. Each component was tested to verify that these functions operated correctly and in alignment with the predefined specifications, without any errors that could disrupt the user experience. This testing process ensures the platform’s technical integrity and its ability to provide a seamless and effective learning environment.

Table 5. User Testing

ID	Variable	N	Min	Max	Mean	Std Dev	Level	Predicate
A2	Accuracy	0.838	2	5	4.17	0.627	5	Very Strong
C1	Content	0.829	1	5	4.22	0.645	5	Very Strong
C4	Ease of Use	0.779	2	5	4.48	0.646	5	Very Strong
F1	Format	0.669	1	5	4.32	0.502	5	Very Strong
T2	Timeliness	0.731	2	5	4.31	0.505	5	Very Strong

User testing was conducted by distributing a questionnaire to 76 students involved in this study. Over a five-week period, the students explored the platform to complete programming lessons and challenges. Table 11 presents the results of user satisfaction measurements, using an instrument of variables organized in descriptive statistical form. The analysis reveals that the user satisfaction level was above a score of 4.00, consistent with a previous study by (Pradana et al., 2023), which showed a positive response to the platform. These findings suggest that the platform has effectively met the users' needs, is ready for use, and can be successfully integrated into programming learning.

The research on utilizing serious games for programming learning, as presented in the study, offers several avenues for dissemination and continuation. The findings can be shared through academic conferences, journals, and workshops to reach a broader audience of educators and researchers interested in innovative teaching methods. Additionally, the platform developed, DolananCoding, can be integrated into various educational institutions to further validate its effectiveness and gather more extensive user feedback.

However, there are some limitations in the current research that need to be addressed in future studies. Firstly, the study primarily focuses on a specific group of students from a single institution, which may limit the generalizability of the findings. Expanding the research to include diverse student populations from different educational backgrounds and institutions would provide a more comprehensive understanding of the platform's impact.

Secondly, while the study demonstrates positive results in terms of user satisfaction and functional requirements, it does not extensively explore the long-term effects of using serious games on students' programming skills and knowledge retention. Future research should include longitudinal studies to assess the sustained impact of game-based learning on programming proficiency.

Lastly, the current version of DolananCoding could benefit from further enhancements, such as incorporating adaptive learning algorithms to tailor the difficulty of challenges based on individual student performance. This would ensure that the platform remains engaging and effective for students with varying skill levels.

By addressing these limitations and continuing to refine the platform, the research can contribute significantly to the field of programming education and provide valuable insights into the use of serious games as an interactive learning tool.

CONCLUSION

Programming learning is a unique field compared to others. A student's ability to type quickly on a keyboard does not guarantee mastery of programming. Students are required to continuously practice in order to understand the diverse programming syntax. High motivation and engagement are key factors in the success of programming learning, as they encourage students to practice not only in class but also outside of it. Conventional teaching methods have proven to be less effective in achieving the objectives of programming learning; therefore, there is a need for innovative teaching methods to enhance student interactivity. One solution to increase motivation, engagement, and interactivity is by utilizing serious games as an interactive platform for programming learning, which incorporates elements such as characters, live coding, points, badges, leaderboards, challenges, and progress tracking. This approach aims to foster a more dynamic and engaging learning experience that aligns with the diverse needs of students.

This study developed DolananCoding, a serious game serving as an interactive platform for programming learning, utilizing the waterfall model as its system development methodology. Prior to distribution, DolananCoding was validated through two methods: system testing and user testing, to ensure the platform’s readiness for use. The development of DolananCoding aims to foster students' interest in practicing independently without relying solely on instructors or lecturers. Additionally, the inclusion of features such as points and leaderboards motivates students to continuously improve their programming skills by observing the achievements of their peers. This approach not only enhances individual engagement but also encourages a competitive and collaborative learning environment.

REFERENCES

Abba, S., Wadumi Namkusong, J., Lee, J.-A., & Liz Crespo, M. (2019). Design and Performance Evaluation of a Low-Cost Autonomous Sensor Interface for a Smart IoT-Based Irrigation Monitoring and Control System. *Sensors*, *19*(17), Article 17. https://doi.org/10.3390/s19173643

Asmaroini, A. P. (2021). Motivasi Belajar Mahasiswa Menggunakan Pembelajaran Daring Pada Masa Covid-19. *Jurnal Dimensi Pendidikan Dan Pembelajaran*, *9*(1), 31. https://doi.org/10.24269/dpp.v9i1.3582

Caserman, P., Hoffmann, K., Müller, P., Schaub, M., Straßburg, K., Wiemeyer, J., Bruder, R., & Göbel, S. (2020). Quality Criteria for Serious Games: Serious Part, Game Part, and Balance. *JMIR Serious Games*, *8*(3), e19037. https://doi.org/10.2196/19037

Ekohariadi, E., Anistyasari, Y., Putra, R., & Kurniawan, I. (2018). *Assessing Computational Thinking using Pseudocode Programming Instrument*. 134–138. https://doi.org/10.2991/aptekindo-18.2018.30

Hidayat, I. K., Wahyu Arizal, F., & Sutrisno, A. (2023). Primary Student and Teacher Game Based Learning Engagement: The Problem and the Challenge. *KnE Social Sciences*. https://doi.org/10.18502/kss.v8i15.13928

Hijrah, M. A., Risnasari, M., Arif, M., Cahyani, L., & Aini, N. (2020). GAME EDUKASI BERBASIS ANDROID PADA MATERI HIMPUNAN. *Jurnal Dimensi Pendidikan Dan Pembelajaran*, *8*(1), 17. https://doi.org/10.24269/dpp.v0i0.2296

Indraswati, D., Utami, P. S., . N., Sutisna, D., & Novitasari, S. (2020). Persepsi dan Motivasi

Mahasiswa Pendidikan Guru Sekolah Dasar (PGSD) Tentang Profesi Guru. *Jurnal Dimensi Pendidikan Dan Pembelajaran*, *8*(3), 131–138. https://doi.org/10.24269/dpp.v8i3.2830

Khan, A. A., & Madden, J. (2016). Speed Learning: Maximizing Student Learning and Engagement in a Limited Amount of Time. *International Journal of Modern Education and Computer Science*, *8*(7), 22–30. https://doi.org/10.5815/ijmecs.2016.07.03

Lee, M. L., Adisusilo, A. K., & Prasetya, N. I. (2022). Perancangan Multiplayer Serious Game Pengolahan Tanah Menggunakan Bajak Singkal. *Journal of Intelligent System and Computation*, *4*(1), 16–21. https://doi.org/10.52985/insyst.v4i1.191

Leksonowati, N. (2019). EFEKTIVITAS PENERAPAN MODEL PEMBELAJARAN TUTORIAL SEBAYA DAN JIGSAW TERHADAP HASIL BELAJAR DITINJAU MOTIVASI BELAJAR DAN KECERDASAN EMOSIONAL SISWA. *Jurnal Dimensi Pendidikan Dan Pembelajaran*, *7*(1), 18–24. https://doi.org/10.24269/dpp.v7i1.1530

Phillips, R. S., Horstman, T., Vye, N., & Bransford, J. (2014). Engagement and Games for Learning: Expanding Definitions and Methodologies. *Simulation & Gaming*, *45*(4–5), 548–568. https://doi.org/10.1177/1046878114553576

Pradana, F., Setyosari, P., Ulfa, S., & Hirashima, T. (2023). Development of Gamification-Based E-Learning on Web Design Topic. *International Journal of Interactive Mobile Technologies (iJIM)*, *17*(03), Article 03. https://doi.org/10.3991/ijim.v17i03.36957

Purwanto, & Hedin, P. B. D. (2020). Measurement of user satisfaction for web-base academic information system using end-user computing satisfaction method. *IOP Conference Series: Materials Science and Engineering*, *909*(1), 012044. https://doi.org/10.1088/1757-899X/909/1/012044

Putra, D. A. T. B., Adisusilo, A. K., & Prasetya, N. I. (2022). Optimasi Aset dan Karakter Permainan 3D Berbasis Tematik Sekolah Dasar. *Journal of Information System,Graphics, Hospitality and Technology*, *4*(01), 1–6. https://doi.org/10.37823/insight.v4i01.165

Putra, L. D., Arlinsyah, N. D., Ridho, F. R., Syafiqa, A. N., & Annisa, K. (2024). Pemanfaatan Wordwall pada Model Game Based Learning terhadap Digitalisasi Pendidikan Sekolah Dasar. *Jurnal Dimensi Pendidikan Dan Pembelajaran*, *12*(1), 81–95. https://doi.org/10.24269/dpp.v12i1.8749

Shah, S. I. H., Majoka, M. I., & Khan, S. I. (2019). Learning Engagement in Mathematics: A Test of an Active Learning Model. *Global Social Sciences Review*, *IV*(II), 143–150. https://doi.org/10.31703/gssr.2019(IV-II).19

Topalli, D., & Cagiltay, N. E. (2018). Improving programming skills in engineering education through problem-based game projects with Scratch. *Computers & Education*, *120*, 64–74. https://doi.org/10.1016/j.compedu.2018.01.011

Tuakia, F., & Prasetya, N. I. (2021). RANCANG BANGUN APLIKASI PENCARIAN LOKASI OBJEK WISATA DESA SUKO BERBASIS ANDROID. *Melek IT : Information Technology Journal*, *7*(2), 13–22. https://doi.org/10.30742/melekitjournal.v7i2.166

Vaca-Cárdenas, L. A., Bertacchini, F., Tavernise, A., Gabriele, L., Valenti, A., Olmedo, D. E., Pantano, P., & Bilotta, E. (2015). Coding with Scratch: The design of an educational setting for Elementary pre-service teachers. *2015 International Conference on Interactive*

Collaborative Learning (ICL), 1171–1177. https://doi.org/10.1109/ICL.2015.7318200

Xie, J., Zhang, B., Yao, Z., Peng, B., Chen, H., & Gao, J. (2022). The Relationship Between Social Mobility Belief and Learning Engagement in Adolescents: The Role of Achievement Goal Orientation and Psychological Capital. *Frontiers in Psychology*, *13*, 792108. https://doi.org/10.3389/fpsyg.2022.792108