J-INTECH (Journal of Information and Technology)
Accredited Sinta 4 Ministry of Higher Education, Science and Technology Republic of Indonesia SK No. 10/C/C3/DT.05.00/2025
E-ISSN: 2580-720X || P-ISSN: 2303-1425

Developing the 'Angker' Multiplayer Horror Mobile Game Using the Agile Method

Apik Banyubasa[1], Cindy Valencia[2], Hassan Nasrallah Matouq[3], R Bramaditya Ario Wirawisesa[4] , Aditya Wicaksono[5*] , Muhammad Nasir[6]

[1,2,3,4,5,6]IPB University, Software Engineering Technology, Bogor, Indonesia, Jl. Raya Dramaga, Kampus IPB Dramaga Bogor, 16680

Keywords
Agile Method; Game Development; Godot Engine; Horror Game; Multiplayer Horror

Corresponding Author:
adityawicaksono@apps.ipb.ac.id

Abstrak
This study develops a mobile-based multiplayer horror game titled *"Angker"* using the Agile method, specifically Agile Scrum, in response to the rapid growth of the mobile game industry and increasing interest in the multiplayer horror genre. The Agile method was chosen for its flexibility in managing complex game development, allowing quick adaptation to changes and strong team collaboration. The development process was divided into eight sprints, covering planning (team role assignment, project scheduling), design (UML diagrams, UI design using Figma), feature development (character movement, multiplayer system, chat, item system, visual/audio assets, in-game store, skin transactions), internal and external testing, and final deployment. A total of 15 core features were successfully developed and implemented, including 8 technical modules such as a multiplayer lobby system, coin transaction system, interactive item system, and win condition detection system. Testing results demonstrated improvements in time efficiency, team coordination, and final product quality. The novelty of this study lies in the comprehensive integration of the Agile method in multiplayer horror game development using the Godot engine—an approach still rarely implemented, especially by small teams with limited resources.

1. Introduction

The rapid advancement of information technology and mobile devices has significantly driven the growth of the digital game industry, with mobile-based games emerging as one of the most popular forms of entertainment among the public [1]. Among various genres, horror games have gained increasing popularity due to their suspenseful atmosphere, immersive storytelling, and adrenaline-pumping surprise elements [2]. Recent trends indicate that cooperative multiplayer horror games are becoming more favored, as they offer interactive, collaborative, and social gameplay experiences [3]. However, the development of multiplayer horror games presents several technical challenges, such as player synchronization, real-time system design, and multiplayer network integration [4].

To address these challenges, the Agile methodology is widely adopted in software development due to its iterative approach and adaptability to evolving user requirements [5]. Agile also fosters effective collaboration among team members with diverse roles [6]. This approach has proven successful in the development of no-code applications such as Appsheet, which require high flexibility to meet user demands [7]. In another context, Agile has been implemented in the development of educational games like Hotel Yamato, which features contextual and interactive narratives [8]. Several studies suggest that Agile improves development time efficiency in small- to medium-scale game projects [9]. Moreover, this methodology positively impacts the final product quality and user satisfaction in the context of entertainment game development [10], making it a viable option for complex mobile-based game projects [11].

Beyond methodology, the success of game projects also heavily depends on the choice of appropriate technologies. Godot Engine is a popular open-source, lightweight engine that supports cross-platform game development [6]. It offers flexibility for creating both 2D and 3D games using GDScript, a language specifically designed for efficient game development [12]. Godot is also equipped with a modular multiplayer system, which is well-suited for real-time synchronization requirements in multiplayer games [13]. Its potential has been demonstrated in educational projects such as a Covid-19 awareness campaign, which required social interaction in gameplay [14]. With these features, Godot serves as an ideal solution for small teams developing mobile-based multiplayer games [15].

Based on this background, this study aims to design and develop Angker, a mobile-based multiplayer horror game using the Agile Scrum methodology as the development framework [16]. The game emphasizes a tense and collaborative gameplay experience through a lobby-room system that utilizes unique session IDs for real-time player synchronization. One of its gameplay elements is a reward system, inspired by RPG games such as Mobile Legends, although it excludes character level progression [17].

This study distinguishes itself from previous research, which mostly focused on educational or single-player games. Budiharjo et al., for instance, developed a 2D disaster mitigation game without multiplayer features, thus avoiding discussions on synchronization and real-time communication [18]. Rahadianto's work was limited to single-player production management simulations, emphasizing resource management aspects [19]. Meanwhile, Nurindiyani et al. designed a visual novel without integrating real-time multiplayer features into its gameplay [20]. Hence, this study contributes new insights to Agile-based game development, particularly in the context of real-time multiplayer games that demand complex coordination.

In addition to product development, this research also evaluates the effectiveness of the Agile method in improving development time efficiency in small-scale projects [21]. The evaluation includes team collaboration effectiveness and improvements in final product quality through limited user playtesting [22]. The findings are expected to offer both theoretical and practical contributions to game developers, especially those employing the Godot Engine and Agile methodology in small team settings.

2. Research Method

This study adopts the Software Development Life Cycle (SDLC) approach based on the Agile methodology, with Scrum serving as the primary framework. Scrum was selected for its flexibility in accommodating changing user requirements throughout the development process [5], and its suitability for complex projects such as multiplayer game development [6].

The game development was conducted incrementally using the Agile methodology, structured into multiple sprints, which are short, iterative work phases [7]. Each sprint produced functional increments that were immediately tested and evaluated, enabling the team to perform continuous improvements effectively throughout the development cycle [23]. Previous studies employing the Godot engine in game projects have also demonstrated that such an approach facilitates efficient workflows, particularly for small teams and projects with manageable scope [19].

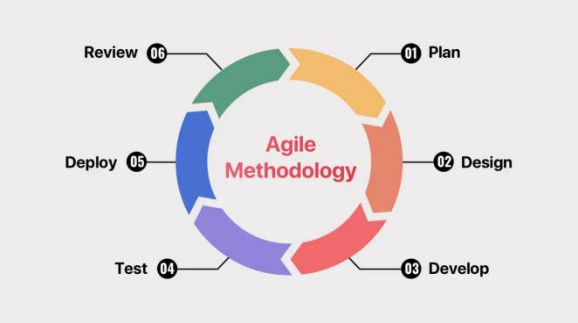

Figure 1. Agile Method

Overall, the development process comprised eight sprints covering five main stages: pre-production, design, implementation, testing, and finalization. During pre-production, the development team gathered system requirements and engaged in discussions to create the Game Design Document (GDD), which served as the primary reference for designing the core game elements [2]. The GDD functioned as a shared guideline to maintain consistency in vision and development direction. The design phase focused on outlining the game's foundational structure, including level planning, system logic, character controls, and user interface (UI). To assist in UI visualization, the team utilized Figma as a design tool, facilitating developers' understanding of the intended appearance and interaction flows [11]. The resulting designs formed the basis for efficient and consistent feature implementation by the technical team.

The implementation phase began with the preparation of a product backlog containing prioritized features based on user needs and project scope. These features were then developed incrementally according to the preplanned sprint schedule [23]. Godot was selected as the primary game engine due to its lightweight nature, cross-platform support, and suitability for small to medium-scale projects [13]. Additionally, Godot supports a modular multiplayer system fitting the game's requirements [15]. Multiplayer functionality was realized through a lobby-room system that enabled players to join game sessions via unique session IDs, while game logic and player interactions were managed using Godot's internal scripting language, GDScript [4].

Following implementation, testing was conducted through internal assessments and limited user trials. Internal testing concentrated on bug detection, basic functionality verification, and system stability during gameplay [21]. Early in development, Agile Requirements Engineering was applied to iteratively gather user needs, reducing the risk of uncontrolled requirement changes during the process [24]. Once all features passed testing and were deemed functional, development proceeded to the finalization stage. The product was subsequently released on the Android platform as the project's culmination. An evaluation was carried out to assess time efficiency, team coordination effectiveness, and product quality based on user feedback obtained during testing sessions [22].

3. Results and Discussions

The planning stage is crucial to ensuring smooth project execution. It provides a comprehensive view of development direction, enabling proportional task division across each sprint. A systematic schedule helps team members clearly understand their roles and responsibilities. Main activities include idea exploration, Game Design Document (GDD) formulation, and feature grouping into sprint cycles.

After planning, the process continues to the design phase, where the team constructs the game's technical foundation using Unified Modeling Language (UML). Two main diagrams are prepared: a Use Case Diagram to represent user interactions with system features, and a Class Diagram to show relationships between game objects—serving as implementation guidelines.

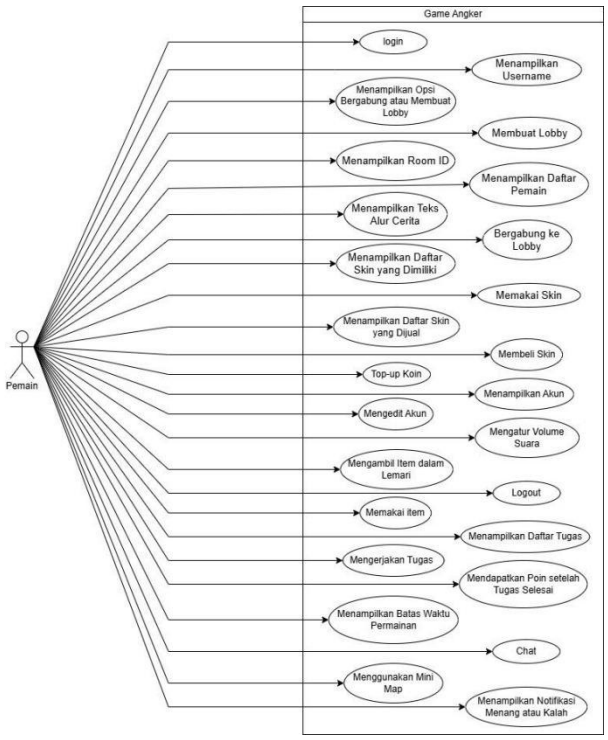

Figure 2. Use Case

This figure illustrates the Use Case Diagram for the game *Angker*, representing the system's functionalities from the user's perspective. The diagram outlines player interactions with features such as login, displaying usernames, joining or creating lobbies, showing room IDs, listing players, displaying storylines, showing owned or available skins, equipping or purchasing skins, topping up coins, managing or editing accounts, adjusting sound volume, picking up items from wardrobes, logging out, using items, completing tasks, displaying task lists, gaining points upon task completion, showing time limits, chatting, using the mini map, and displaying win/loss notifications. This diagram acts as a guide for the implementation phase to ensure efficient and well-structured system development.

The Class Diagram is used to illustrate the structure and relationships between the main classes in the game. It includes core classes such as GameManager, HUD, InputSynchronizer, GameCharacter, Room, Player, and Ghost. Each class has a specific responsibility, such as the GameManager managing the game flow and the InputSynchronizer handling input synchronization among players in multiplayer mode. The relationships between classes are visualized through associations, inheritance, and composition, facilitating a clearer understanding of the system's logical structure.

In addition to UML-based system design, this phase also involves the development of the user interface (UI) using Figma. The primary objective of the UI design is to deliver an intuitive, visually appealing, and consistent user experience throughout the game. The process begins with the creation of wireframes for key screens such as the main menu, gameplay interface, and in-game store. These wireframes serve as an initial guide to map user interaction flows before being refined into detailed and implementation-ready UI designs.

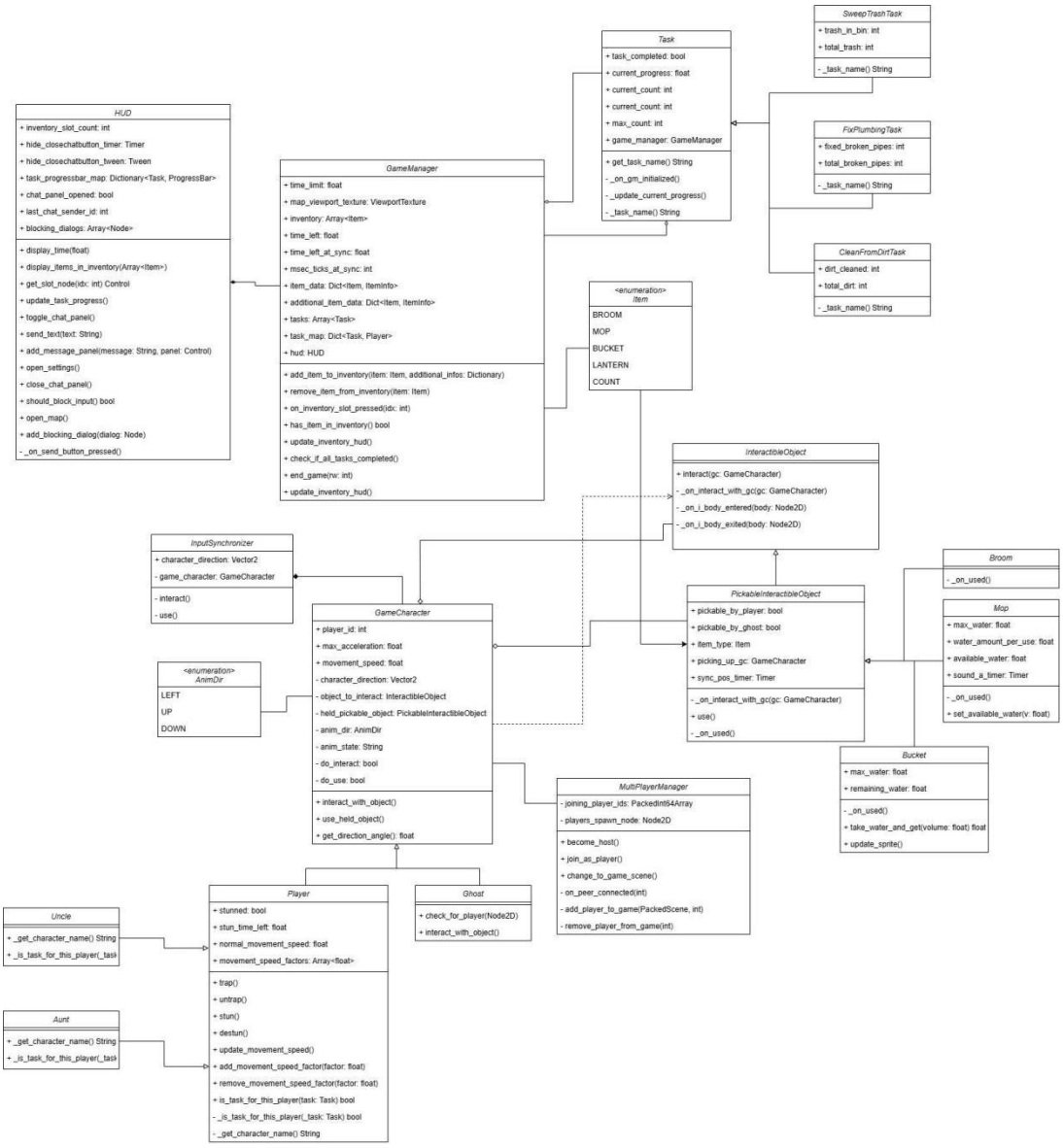

Figure 3. Class Diagram

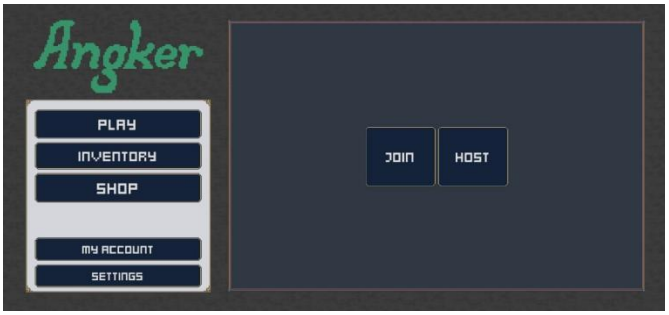

Figure 4. Main Menu

This image displays the main menu of Angker, where players can select various options such as 'Play' to start the game, 'Inventory' to view owned items, 'Shop' to purchase items or skins, 'My Account' to create or edit

usernames, and 'Settings' to adjust music and sound effects. Additionally, the 'Join' and 'Host' buttons indicate a multiplayer component, allowing users to either join others or start their own game sessions.

Figure 5. In-Game View

Figure 6. Coin Top-Up

This image shows the in-game screen during task completion in Angker. The top-left corner displays the completion percentage and the number of remaining tasks, such as '0/27 sweep trash' and '0/15 clean dirt.' Players control their characters using a joystick on the lower left and action buttons on the lower right to interact with cleaning tools. Time, map, settings, and chat features are also available to enhance the gameplay experience.

This image shows the Shop section, featuring several cosmetic items that players can purchase using coins displayed on the top button. If a player has insufficient coins, they can buy more by clicking the coin button, which will display options as shown in the image, allowing the player to select the desired coin package.

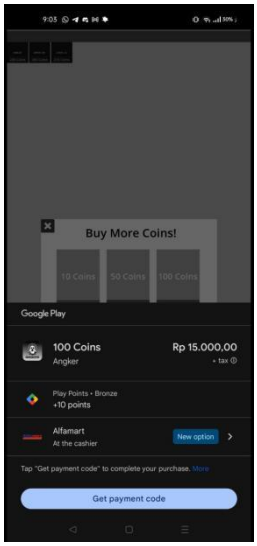

Figure 7. Payment

This image illustrates the coin top-up payment process via Google Play. Users select the coin amount (e.g., 100 Coins), view pricing details including taxes, and choose a payment method such as Alfamart, with an option to tap 'Get payment code' to complete the transaction. Subsequently, confirmation messages like 'Purchase successful!' and 'Payment successful' will be displayed, showing the token and the number of coins added (e.g., '+100').

Figure 8. Join

Figure 9. Host

This image displays the screen for joining a game in Angker. Players have three options to join: 'JOIN ROOM BY ID' allows players to directly enter a unique identification code for a lobby for quick access; 'JOIN ROOM BY NAME' provides an alternative for players to search for and join a lobby using a specific name set by the host; or 'BROWSE LOBBY' functions as a directory, enabling players to browse a list of publicly available or accessible lobbies.

This image visualizes the game lobby hosting screen in Angker, designed to facilitate players in creating their own game sessions. Its main feature is the ability to set a 'Shortcode' or unique name for the lobby to be created. After the shortcode is entered, the 'CREATE' button functions to process the formation of that lobby. Immediately after the lobby is successfully created, a 'Join' button will appear, allowing the host to directly enter the lobby they have just prepared.

3.1 Multiplayer Implementation Challenges in Our Game

We faced several challenges in converting our single-player game into a multiplayer one. The first issue was selecting the optimal connection method. We initially used a peer-to-peer approach with Noray, but it was unstable. After evaluating alternatives, we switched to Hathora, which was easier to integrate and provided a more stable connection. Next, we addressed latency, as client input had to be sent to the server and returned for processing—necessary to prevent cheating but causing noticeable input delay. To improve responsiveness, we used the Netfox plugin to process input client-side, allowing immediate character reaction. The server still performed authoritative evaluation and sent back the corrected position to ensure synchronization. We also encountered synchronization issues where object interactions by one player were not reflected on other clients, or objects couldn't be picked up at all. This was traced to position sync problems, which, once resolved, allowed consistent and reliable object interaction across all clients.

3.2 Quantitative Results from Playtesting

3.2.1 Bug Analysis and Resolution

During the testing phase, several functional issues were identified in key features of the game. These bugs were subsequently documented, analyzed, and gradually resolved to ensure a stable and optimal gameplay experience. The following summarizes the identified bugs:

Figure 10. List of Identified Bug Scenarios in Angker

During the testing phase, several critical bugs were identified and resolved to enhance the overall stability of the game. Adjustments were made to the text retrieval function to ensure storyline consistency across gameplay sessions. The lobby status detection system was refined so that the join option consistently appears when a lobby is available. The top-up process was optimized to ensure immediate coin crediting upon successful payment, supported by improved balance verification logic. Username validation was added to prevent duplication, and the chat system was adjusted to ensure that messages remain readable even when users input text without spaces.

However, certain features could not be implemented due to technical constraints. The login functionality was not developed due to the absence of a database and the limited capacity of the Hathora server used, which also prevented the inclusion of a logout feature. As an alternative, a username change function was provided to maintain system stability, particularly in lobby and multiplayer functionalities. This approach allowed the development to remain focused on core features that support real-time collaborative gameplay.

3.2.2 User satisfaction score

Evaluation Aspect	Score (1-5)	Observation Notes
Game Enjoyment	4	It's quite fun because you can play it with friends, and completing tasks in the game feels fun.
Tension/Horror Level	3	In terms of plot and appearance, it's not too tense, but the sound effects help to enhance the horror atmosphere.
Control Ease	2	The controls are still a bit tricky, especially when moving objects like the broom or throwing away trash. Beginners may find it difficult as it requires precision.
Level Design	4	The level design is quite interesting and makes players curious to know what happens next in the next level.
Interest to Replay	5	Really looking forward to the next version of this game.
Average Player Satisfaction Score:	**3.6**	

Figure 11. User Satisfaction Scores Based on Gameplay Aspects

The test results indicate that the average player satisfaction score for the Angker game reached 3.6 out of 5, with the highest rating (5) given to the interest to replay aspect, reflecting a strong desire among players to try future versions of the game. Game enjoyment and level design each received a score of 4, indicating a fun gameplay experience and engaging level structure. On the other hand, the tension/horror level scored 3, as the horror atmosphere was considered moderate despite the use of supportive sound effects. The lowest score (2) was recorded for control ease, due to unintuitive controls that challenged beginner players, especially when performing precision-based actions. These findings highlight the need for improvements in control responsiveness and horror intensity to enhance the overall player experience.

3.2.3 Player Engagement Rate

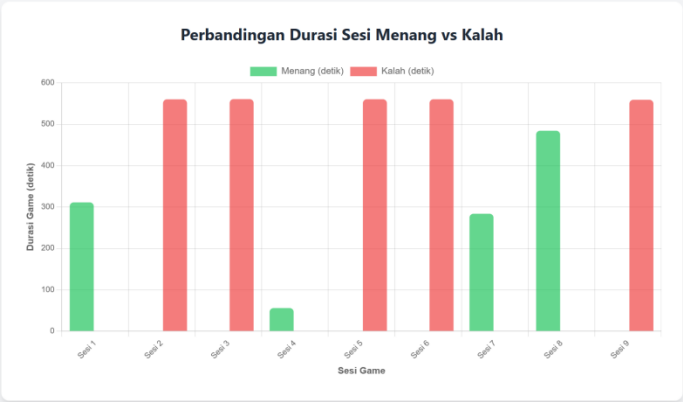

Figure 12. Comparison of Game Session Durations: Win vs. Loss

This chart illustrates the comparison of session durations between winning and losing players as a measure of game completion rate. It is clear that the majority of sessions ending in a loss (red bars) have significantly longer durations, consistently around 550 seconds, compared to winning sessions (green bars), which are more varied and generally shorter. This pattern indicates a relatively high game difficulty, as players tend to persist longer in sessions they ultimately lose. Therefore, this data highlights the need for adjustments to gameplay balance for a more balanced playing experience.

4. Conclusions and Future Works

This study developed Angker, a mobile horror game with cooperative multiplayer features, using the Agile Scrum methodology. Agile was chosen for its adaptability, collaboration, iterative process, and responsiveness to change—enabling efficient integration of user feedback over eight sprints. The project began with defined team roles and scheduling using Microsoft Project. Technical design employed UML diagrams, while UI was designed in Figma, from wireframes to full layouts. Development followed a product backlog, implementing key systems such as character movement, multiplayer, chat, items, skills, points, shop, and skin transactions. Testing involved internal and external reviews, followed by bug fixing. After stabilization, Angker was published. The novelty lies in combining multiplayer horror gameplay with full Agile implementation using the Godot Engine. This study also evaluates Agile's effectiveness in time efficiency, collaboration, and final product quality. For future work, the following recommendations are proposed: analyzing quantitative metrics of efficiency and coordination; further leveraging Godot's open-source, cross-platform capabilities; expanding features or storyline for greater replayability; comparing with other games using different engines or methods; and deepening user satisfaction evaluation post-release.

5. References

[1] Y. Fernida and N. Nurdianto, "Pembuatan Game 3D 'Uji Nyali: The Game' Berbasis Mobile," *Jav. J. Vokasi Inform.*, pp. 149–154, Feb. 2024, doi: 10.24036/javit.v3i3.155.

[2] I. M. S. A. Pande, R. A. N. Diaz, and R. A. A. Ramadhan, "PEMBUATAN HOROR GAME 3D MULTIPLAYER CO-OP 'CALONARANG' BERBASIS MOBILE," *Naratif J. Nas. Riset, Apl. dan Tek. Inform.*, vol. 5, no. 2, pp. 149–161, Dec. 2023, doi: 10.53580/naratif.v5i2.244.

[3] K. A. and G. W. Wiriasto, "Save Our Sea: Rancangan Game 2D RPG Berbasis Android dengan Metode Extreme Programming menggunakan Godot Script".

[4] S. Y. Saputra and S. Subari, "Game Multiplayer â€œMini Car Circuitâ€ Berbasis Android," *J-INTECH*, vol. 7, no. 01, pp. 58–66, Jun. 2019, doi: 10.32664/j-intech.v7i01.406.

[5] Wahyu Mustika Aji, Achmad Syahmi Rasendriya, Alya Putri Salsabila, Keysha Maulina Halimi, Muhammad Nasir, and Aditya Wicaksono, "Pengembangan Website Spherevent untuk Manajemen Event Berbasis Web dengan Pendekatan Agile," *J. FASILKOM*, vol. 15, no. 1, pp. 131–137, May 2025, doi: 10.37859/jf.v15i1.9116.

[6] Aqil Farras, Ira Puspita Sari, Jacky Junaidi, and Hustinawati, "Penerapan Metode Agile Dalam Pembuatan Aplikasi Inventory Seal Berbasis Appsheet," *J. SANTI - Sist. Inf. dan Tek. Inf.*, vol. 5, no. 1, pp. 29–37, Feb. 2025, doi: 10.58794/santi.v5i1.1297.

[7] A. Budiharjo, A. Irma Purnamasari, and R. Danar Dana, "GAME EDUKASI MITIGASI BENCANA BANJIR MENGGUNAKAN METODE AGILE DEVELOPMENT," *JATI (Jurnal Mhs. Tek. Inform.*, vol. 6, no. 2, pp. 802–810, Nov. 2022, doi: 10.36040/jati.v6i2.5741.

[8] A. R. Pramnesti, A. Lailannafisa, A. Aqilah, M. Rifky, B. T. Pramesti, and P. W. Atmaja, "Perancangan Game Edukatif Terkait Insiden Hotel Yamato di Surabaya Menggunakan Metode Agile SCRUM," *Scan J. Teknol. Inf. dan Komun.*, vol. 19, no. 2, Jun. 2024, doi: 10.33005/scan.v19i2.4732.

[9] J. L. Andre, E. Handriyantini, and C. A. Oktavia, "Pengembangan Game Virtual Reality Berbasis Android Menggunakan Unity Sebagai Media Penunjang Pengenalan Bahasa Inggris," *J-INTECH*, vol. 6, no. 02, pp. 208–213, Feb. 2019, doi: 10.32664/j-intech.v6i02.253.

[10] N. Bayhakky, "Permainan Ular Tangga Berbasis Android Menggunakan Unity".

[11] F. Diouf Ghiffary, H. Zulfia Zahro', and F. Santi Wahyuni, "PERANCANGAN GAME BERBASIS ANDROID 'KNIGHT THE DEVIL SLAYER' MENGGUNAKAN METODE PATHFINDING," *JATI (Jurnal Mhs. Tek. Inform.*, vol. 8, no. 6, pp. 12377–12384, Nov. 2024, doi: 10.36040/jati.v8i6.10536.

[12] P. Harsadi, W. L. Y. Saptomo, and C. Y. Wardhana, "Implementasi Algoritma Fisher-Yates Shuffle Pada Game Edukasi Aksara Jawa Menggunakan Godot Engine," *J. Teknol. Inf. dan Komun.*, vol. 10, no. 1, May 2022, doi: 10.30646/tikomsin.v10i1.603.

[13] M. F. A. Arkas Dyanta Ananda Kautsar, Tommy Bustomi, "Pembangunan game 'Recon Duty' dengan Godot Engine".

[14] R. K. Putra and P. Simanjuntak, "PERANCANGAN GAME EDUKASI COVID-19 BERBASIS GODOT ENGINE DENGAN METODE MDLC,".

[15] D. P. Kynta, "Implementation of Fuzzy Logic in Educational Game on Manners and Morals for Kids Using Godot Engine," *J. Artif. Intell. Softw. Eng.*, vol. 5, no. 1, p. 339, Mar. 2025, doi: 10.30811/jaise.v5i1.6458.

[16] D. A. M. and U. A. Rosyidah, "Analisis dan Pengembangan Game Edukasi 'Earth Defender'", doi: 10.35891/explorit.

[17] G. S. Tri Yusnanto, Gregorius Ronald Fanky Wicaksana, "RANCANG BANGUN ROLE PLAYING GAME (RPG) BERTEMA MOBILE LEGENDS : BANG BANG," *TRANSFORMASI*, vol. 16, no. 2, Feb. 2021, doi: 10.56357/jt.v16i2.225.

[18] I. D. Rahadianto, "Game Simulasi Manajemen Produksi Game Dengan Metode Agile Development," *J. Penelit. Pendidik.*, vol. 21, no. 2, pp. 49–65, Aug. 2021, doi: 10.17509/jpp.v21i2.37405.

[19] A. K. Nurindiyani, A. F. Dianta, H. Sa'dyah, and I. A. Riyadi, "Implementasi Agile Scrum Pembuatan Game Visual Novel Cerita Asal Usul Kota Surabaya," *JTIM J. Teknol. Inf. dan Multimed.*, vol. 4, no. 4, pp. 330–341, Feb. 2023, doi: 10.35746/jtim.v4i4.277.

[20] U. Mawaddah, E. D. Wahyuni, and A. P. Kusuma, "Penerapan Metode Agile Dalam Sistem Informasi Manajemen Asrama Santri pada Yayasan Pondok Pesantren Darul Huda Blitar Berbasis Web," *J-INTECH*, vol. 11, no. 2, pp. 188–199, Dec. 2023, doi: 10.32664/j-intech.v11i2.1004.

[21] H. Prasetyo and U. Nugraha, "Optimasi Keamanan dalam Pengembangan Aplikasi Menggunakan Metode Agile Scrum dan JSON Web Token," *JUSTINFO | J. Sist. Inf. dan Teknol. Inf.*, vol. 1, no. 1, pp. 34–41, Dec. 2023, doi: 10.33197/justinfo.vol1.iss1.2023.1228.

[22] M. I. Khalid, R. Firliana, and E. Daniati, "Manajemen Proyek Pengembangan Game Kasur Rusak dengan Menggunakan Agile Scrum," *JSITIK J. Sist. Inf. dan Teknol. Inf. Komput.*, vol. 2, no. 2, pp. 141–149, Jun. 2024, doi: 10.53624/jsitik.v2i2.252.

[23] S. Widayati, Y. I. Chandra, and D. Ruri, "Penerapan Metode Agile Process dengan Model Extreme Programming Dalam Pembuatan Game RPG 'The Realm of Unknown' Menggunakan MV RPG Maker," *J. Esensi Infokom J. Esensi Sist. Inf. dan Sist. Komput.*, vol. 6, no. 1, pp. 51–58, May 2022, doi: 10.55886/infokom.v6i1.456.

[24] K. W. Prasetyo and N. R. A. Tam, "Analisis Kebutuhan Sistem Informasi Manajemen Kegiatan Kemahasiswaan STIKI Malang Menggunakan Agile Requirements Engineering," *J-INTECH*, vol. 10, no. 1, pp. 21–29, Jun. 2022, doi: 10.32664/j-intech.v10i1.675.