CHAPTER I
INTRODUCTION

1.1 Background

The gaming industry has experienced rapid growth in recent decades, fuelling innovations in technology, gameplay and user experience. According to Game Revolution (2024), the gaming industry is a dynamic and evolving sector, with significant changes taking place by 2024. It is estimated that the global gaming industry market will generate revenues of approximately $282.3 billion by 2024 and potentially increase to $363.2 billion by 2027, with a compound annual growth rate of 8.76%.

One aspect that is undergoing rapid development is the control system and behavior of non-player characters (NPCs), which play an important role in creating a more immersive and realistic gaming experience. As stated by Alves Da Silva et al. (2021), NPCs are characters in the game that are not directly controlled by the player. The presence of responsive, realistic, and adaptive NPCs can improve the quality of the gaming experience by creating more interesting challenges and strengthening the impression of immersion in the game. Early in their development, NPCs had static, simple, and predictable behavior patterns. However, with the advancement of artificial intelligence technology, NPCs can now exhibit more dynamic, complex and adaptive behavior in responding intelligently to player actions.

According to Yakan (2022), artificial intelligence (AI) in games aims to improve the overall gaming experience for players. In the development of NPC behaviour, the use of AI has become an integral part of modern game design. Traditional techniques, such as Finite State Machine (FSM), are widely used due to their simplicity in designing character behaviour. As described by Bagus et al. (2024), FSM can simplify AI programming code, especially for supporting characters that need to move or react to certain conditions. FSM allow characters to move between different states, such as Idle, Patrol, or Attack, based on predefined conditions. However, as stated by Junaidi et al. (2021), although Finite State Machine (FSM) algorithms are relatively easy to implement for NPCs in computer games, managing them becomes increasingly challenging as the number of states grows. This issue, known as state explosion, arises when the number of behaviors that need to be controlled expands significantly. In addition, FSMs lack the flexibility required to handle complex and dynamic behaviors, making system maintenance difficult as the project scales. According to Iovino et al. (2024), FSMs require managing a large number of transitions whenever states are added or removed, reducing modularity and scalability. Game developers have also observed that FSMs tend to be difficult to extend, customize, and reuse in larger or more complex projects.

To address these limitations, Hierarchical Finite State Machines (HFSMs) were introduced as an improvement over traditional FSMs. HFSMs structure states hierarchically, reducing redundancy and improving modularity. However, as noted by Iovino et al. (2024), managing a large number of transitions in HFSMs remains a challenge, further limiting modularity. Additionally, while HFSMs enhance state organization, they still lack the flexibility required for highly dynamic and reactive AI behavior. Reactivity in HFSMs is limited because state transitions must be explicitly defined, making it harder to adapt to unpredictable game environments.

As an alternative, Behavior Trees (BTs) have become a widely used approach in modern game AI development. According to Iovino et al. (2024), BTs were originally developed in the gaming industry as task-switching controllers but quickly gained popularity in robotics applications due to their structured and modular nature. BTs allow character behavior to be built hierarchically, where each task or action can be broken down into smaller, easily manageable sub-tasks. This modular structure is one of BT's key advantages. According to Iovino et al. (2022), every node in a Behavior Tree has the same output arguments, allowing subtrees to function as reusable and reorganizable building blocks without compromising the overall system structure. The hierarchical nature of BTs simplifies the design and maintenance of complex AI behaviors, making them a preferred choice for handling intricate AI systems in games.

Although BTs offer many advantages, they do not entirely replace FSMs. According to Scheide et al. (2021), manually creating BTs can be time-consuming, particularly for complex tasks such as those in robotics. In certain situations, FSMs remain essential for managing high-level state transitions. Sekhavat (2017) also highlighted that as task complexity increases, BTs can become difficult to manage. The addition of new behaviors may require significant restructuring of the tree, leading to inefficiencies and increasing maintenance efforts.

To overcome the shortcomings of both approaches, this research applies a Hybrid FSM-BT approach to develop the AI system for a hunter NPC in the game *Jalak Bali Survive*. The hunter NPC was chosen due to its complex behavioral requirements, such as detecting targets, tracking and hunting prey, and dynamically responding to environmental changes. By integrating FSMs and BTs, this approach enables the NPC to make more flexible and intelligent decisions based on real-time game conditions, enhancing both the challenge and realism of the gaming experience.

According to Zutell et al. (2021), the HFSMBTH (Hierarchical Finite State Machine Behavior Tree Hybrid) approach aims to preserve the structural clarity and explicit control flow of Hierarchical Finite State Machines (HFSM) at a high level, while leveraging the modularity, reusability, and reactivity of Behavior Trees (BT) at a lower level. In alignment with this concept, the present research adopts a similar hybrid model; however, instead of using HFSM, it integrates a traditional Finite State Machine (FSM) with Behavior Trees.

This hybrid FSM-BT configuration functions in a comparable hierarchical manner: the FSM is responsible for managing high-level state transitions such as patrol, follow, or attack, while each individual state utilizes a Behavior Tree to define the specific logic and task execution relevant to that context. This layered control structure ensures both clarity in behavioral sequencing and flexibility in action implementation.

The decision to apply this hybrid approach, rather than a standalone Behavior Tree, is grounded in its superior modularity, maintainability, and control over complex behaviors. Although standalone BTs offer powerful hierarchical control and reactivity, they tend to become increasingly difficult to manage as the behavior tree grows in size and complexity. This can lead to deeply nested structures that are hard to debug and expand. By contrast, the hybrid FSM-BT model separates concerns by isolating behavior logic within each FSM state and supporting it with a dedicated BT. This separation enhances scalability, simplifies debugging, and allows for easier updates or extensions in NPC behavior.

After successfully implementing the hunter NPC's AI system, rigorous testing is required to ensure that its behavior functions as intended. This research employs a black-box testing approach to evaluate the system based on observed inputs and outputs and white-box testing to analyze the internal logic and execution flow of the code. By combining these two methods, this research ensures that the AI system meets both functional and structural quality standards, leading to a robust and well-optimized NPC behavior model.

However, technical testing alone is not enough to ensure that the AI system contributes to an optimal gaming experience. Therefore, this study also integrates the Game Experience Questionnaire (GEQ) to evaluate how players perceive the interaction with the hunter AI in *Jalak Bali Survive* Game. With the GEQ, the player experience can be measured in terms of engagement, tension, as well as the level of challenge provided by the AI. This approach allows the research to not only assess the technical performance of the AI system, but also its impact on the game experience, making the test results more comprehensive in assessing the success of the AI design in this game.

As previously mentioned, *Jalak Bali Survive* was developed to test the implementation of the proposed method. It is a conservation-themed game featuring the Bali Myna as the main character—an endemic Balinese species that faces extinction due to habitat destruction and illegal hunting. The game's storyline follows a female Bali Myna on her journey to build a nest. To accomplish this, she must fly across the environment, collecting dry grass and weeds as nest-building materials. However, poachers pose a constant threat, making it dangerous to gather resources. Players must help the Bali Myna evade the poachers while still collecting the necessary materials, balancing risk and survival in a challenging and immersive gameplay experience.

The choice of the Bali Myna as the main character has an important local value. In addition to being a symbol of fauna conservation, this game also aims to be a digital education tool. With this approach, players are expected to better understand the importance of conserving Indonesia's endemic animals. In the game, the bird character faces various challenges, including threats from poachers. These threats reflect the real situation faced by the Bali Starling species in the wild. Thus, players can feel the tension and empathize with the bird's struggle to survive.

The implementation of Hybrid FSM-BT in hunter AI aims to create complex and realistic game dynamics. This system allows hunter NPCs to behave more responsively to player actions. In addition, this approach also serves as a validation of the effectiveness of the AI system proposed in this research.

1.2 Problem Identification

Some of the problems that have been identified are as follows.

1. Finite State Machine (FSM) reduces the modularity and scalability of the system.
2. Finite State Machine (FSM) is difficult to reuse in larger or complex projects.

3. The Behavior Tree (BT) gets complicated as the character's behavior gets more complex.
4. Behavior Tree nested structures that are hard to debug and expand
5. The need for a hybrid approach to overcome the limitations of FSM and BT.

1.3 Problem Limitation

There are several problem restrictions that can be seen as follows.

1. The game used a simple game called *Jalak Bali Survive*, where the player plays the role of a Bali Myna trying to survive from hunters.
2. AI is only applied to the hunter character in the game.
3. NPC testing using black-box testing and white-box testing methods.
4. The game was developed using Unity for the PC platform.
5. The visual and graphical aspects of the game are not the main focus of this research.
6. Evaluation of code performance and resource utilization was not part of the study.
7. The development of the npc only use hybrid Finite State Machine and Behavior Tree.

1.4 Problem Statement

The problem formulations contained in this study are as follows.

1. How to implement hybrid Finite State Machine (FSM) and Behavior Tree (BT) on NPC hunters in *Jalak Bali Survive* game?

2. How to evaluate the results of the implementation of FSM and BT on NPC hunters in the *Jalak Bali Survive* game?

1.5 Research Objective

The objectives of this research are as follows.

1. To implement hybrid Finite State Machine (FSM) and Behavior Tree (BT) on hunter NPCs in *Jalak Bali Survive* game.
2. To evaluate the results of the implementation of FSM and BT on NPC hunters in the *Jalak Bali Survive* game.

1.6 Research Benefits

This research is expected to provide the following benefits.

1. For Students

Can train and apply the knowledge and skills gained during lectures, especially in the field of game development and AI implementation using hybrid Finite State Machine and Behavior Tree.

2. For the University

This research is expected to contribute to enriching research references in the field of game development and artificial intelligence (AI). In addition, the results of this research can support curriculum development that is relevant to the needs of the game industry, as well as inspire other students to conduct innovative research.

3. For Future Researchers

The results of this research are expected to be a reference in the development of games that use a combination of Finite State Machine and Behavior Tree as an AI solution, allowing for the development of additional features or the creation of new, more complex games in the future.

4. For the Game Development Industry

The hybrid implementation of Finite State Machine and Behavior Tree in NPC AI offers a more flexible and efficient solution to manage complex behaviors in games. It can enhance the gaming experience by delivering more responsive and realistic AI. In addition, this approach can reduce the cost of AI development due to its modular structure and easy to modify. Thus, the results of this research can serve as a practical reference for game developers in creating more intelligent and dynamic AI NPCs.