CHAPTER V
CONCLUSIONS AND RECOMENDATIONS

5.1 Conclusions

Based on the results of the implementation, testing, and analysis that has been carried out, several conclusions can be obtained as follows.

1. The Graph Grammar method has been successfully implemented as a level generation mechanism in 2D platformer games. Through the application of the Linear Expansion, Branching, and Key-Lock Mechanism rules, the system is able to form a graph structure that represents the gameplay consistently. These rules produce a level structure that has a main path, a branching path, and a lock and lock mechanism as a supporting element for the game's progress.
2. The developed system successfully translates the graph structure into a physical representation of the platformer's 2D game levels using a chunk-based approach. Each node in the graph is mapped into chunks according to the type of room, while the relationships between nodes are translated into physical connections through the mechanism of opening walls between chunks. The approach allows the logical structure formed by Graph Grammar to be realized into a playable level layout.
3. Validation using the Breadth-First Search (BFS) algorithm was successfully used to evaluate the structure of the generated level. A level is declared valid if it successfully meets two criteria, namely the execution of the Key-Lock rule and the success of BFS in finding the path from the Start node to the Goal node. Based on the test results in four scenarios, the system obtained a validity performance of 95% in scenario 10 iterations, 99% in scenario 20, and 100% in scenarios 30, and 50 iterations. These results show that Graph Grammar has been able to produce levels that meet the criteria consistently.

Overall, this study shows that the Graph Grammar method can be used as an effective approach in generating levels of 2D platformer games based on graph structure. In addition to producing a consistent level structure, the system is also able to translate the structure into a physical level and validate its connectivity using

the BFS algorithm so that the resulting level meets the predetermined playability criteria.

5.2 Recommendations

Based on the results of the implementation and testing conducted, there are several areas that can be explored in future research, namely:

1. The Graph Grammar algorithm can be further developed by increasing the number of production rules with more diverse mechanisms. Adding these rules will increase the complexity and variety of the generated levels.
2. Level structure validation can be further developed by adding parameters such as jump distance, difficulty level, and level accessibility, so that level validation is not only based on the existence of a path but also considers the quality of the generated path. This development will help the system generate more robust levels that users can play immediately.
3. The system interface can be further developed by providing clearer tutorials or user guides. This can help transform the system not only into a level generator but also into a more practical tool in the level design process.