Defense behavior of real time strategy games: comparison between HFSM and FSM

Rahmat Fauzi[1], Mochamad Hariadi[2], Supeno Mardi Susiki Nugroho[3], Muharman Lubis[4]
[1,4]School of Industrial Engineering, Telkom University (Tel-U), Indonesia
[2,3]Faculty of Industrial Engineering, Institut Teknologi Sepuluh November (ITS), Indonesia

Article Info

Article history:

Received Jul 24, 2018
Revised Nov 21, 2018
Accepted Dec 3, 2018

Keywords:

Comparison
Defense behavior
Finite state machine
Real time strategy

ABSTRACT

RTS Game is one of the popular genre in PC gaming, which has been played by various type of players frequently. In RTS game, NPC Defense Building (Tower) has attacking behavior to the closest enemy without considering certain enemy parameters. This causes the NPC Tower to be more predictable by the opponent and easily defeated if NPC attacked by enemies in the group. Thus, this research simulates NPC Tower using Hierarchical Finite State Machine (HFSM) method compared with Finite State Machine (FSM). In this study, NPC Tower detects enemies by seeing at four parameters namely NPC Tower Health, Enemy's Health, Enemy Type, and Tower Distance to enemies. NPC Tower will attack the most dangerous enemy according to the 'Degree of Danger' parameter. Then use the decision-making logic of the rule-based system. The output of NPC Tower are three type of behaviors namely Aggressive Attacking, Regular Attacking, and Attack with Special Skill. From the test results of 3 NPC Tower, Kamandaka NPC Tower with HFSM method is winning 8.92% compare to Kamandaka Tower with FSM method. For Gayatri Tower NPC obtained equal results using both HFSM and FSM. Meanwhile, Adikara NPC with HFSM method is 4.62% superior to Adikara Tower with FSM method.

Corresponding Author:

Rahmat Fauzi,
School of Industrial Engineering,
Telkom University (Tel-U),
Jalan Telekomunikasi No. 1, Bandung, 40257, Indonesia.
Email: rahmatfauzi@telkomuniversity.ac.id

1. INTRODUCTION

Real-Time Strategy (RTS) is a genre game that has a typical war game consisting of building construction and troop strength. Resource collection is used for building construction, defense and strengthening combat troops. The RTS game considers aspects of economy, resources possessed, troops' strength, attacking and defensing strategies. This kind of genre has become extremely popular due to the rising of famous Age of Empire, Warcraft and Dota. Currently, one famous example of RTS games is Clash of Clans (CoC), which was developed by Supercell besides other known game, Boom Beach (BB) that played by thousands of people in mobile phone platform. In CoC games, there are Non-Playable Character (NPC) Defense Buildings like NPC Cannon, Mortar, Archer Tower and Wizard Tower. Canon and Mortar attack the enemy on the ground while Archer Tower and Wizard Tower shoot enemies in the ground and water (air) area. In both CoC and BB, NPC behavior have defensive pattern in which do strike through attack nearest. Unfortunately, this type of strike is easily penetrated or anticipated by opponent. Moreless, if they are united by attacking in the group. Therefore, this kind of fixed model for NPC Tower behavior creates the fight or course in the game become less vary or predictable to the opponent. To overcome this issues, one solution is through dynamic NPC behavior with more diverse and less predictable attack pattern, which can be achieved by the adaptive and intelligent nature of the behavior pattern. Therefore, dynamic behavior is a type of pattern that takes into account several factors that influence the decision making of NPC. For example NPC combat defense will not always use strong attacks that require enormous energy if the health conditions are in good condition or do not always use normal attacks that require less energy if health conditions are weak. In short, the system should exhibit the whole range of flexibility simply as a result of changes in the NPC attributes for particular games that might raise certain possibilities.

In general, Finite State Machine (FSM) is common method applied in the game to determine the behavior because its simplicity to be implemented in various type of games, although at certain point it got hindered if the agent has multiple approaches in the NPC behavior [1]. It also has negative points due to it has a large number of states and transitions so that representation and analysis become difficult [2, 9]. In order to design the dynamic behavior of NPC that has certain characteristic of behavior, it is suggested to use Hierarchical Finite State Machine (HFSM) method to improve the process to determine the behavioral response in the fluctuate changes of condition, which used certain kind of rule system. Meanwhile, HFSM as a developmental method of FSM has the advantage of reducing the complexity of designing agent behavior, which can simplify the computational process and keep the constraint by minimizing a number of coding bits and adjusting the codes for the states [2, 3]. A detailed and realistic simulation of dynamic behavior requires a precise modeling of decision-making process regarding action prioritization and selection, which is capable to develop goal directed behavioral tendencies and tactical model based on demands and input parameter [4]. Importantly, information is a basic requirement for the development process as a part of the ecosystem [5], which can be utilized lead to the substantial savings in power consumption [6], differentiate distinctive attributes of behavior [7], develop collaboration at tactical level [8, 20], etc. In this study, the researcher developed RTS game entitled DWIPA YUDHA, in which its concept has same type and refers to the RTS genre like Clash of Clans. It has several components, which are (1) Troops (2) Combat Defense Building and (3) Resources. Likewise, this research also develop DWIPA YUDHA to have those components, but this research emphasized more on behavior of NPC Combat Defense Building.

2. RESEARCH METHOD

Importantly, the method stages determine the success of the research to assure its validity and reliability [16-19, 24-31]. Figure 1 is a flowchart that describe the detail of the steps undertaken to complete this research, which is include NPC Agent Design, Design of Environment, Design of Enemy, Design of HFSM for NPC Agent and Design of FSM for Enemies, then continue with various test scenarios.

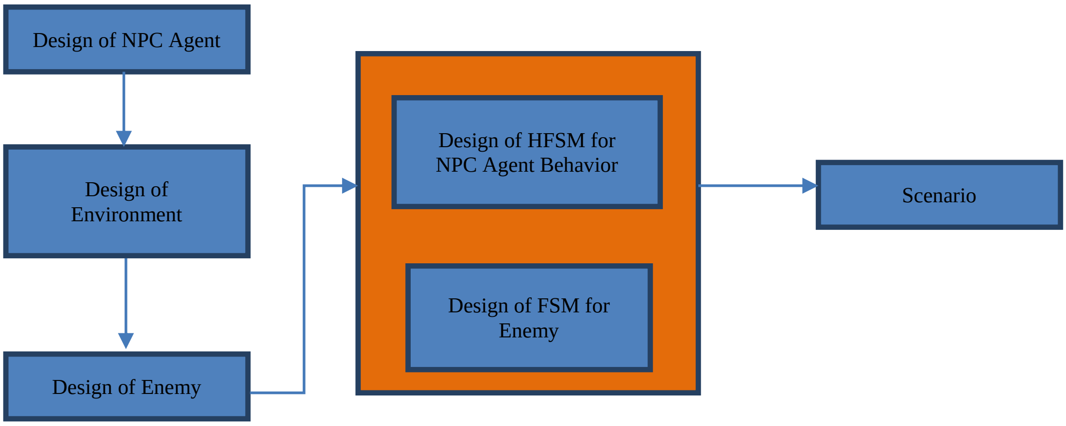

Figure 1. Effects of selecting different switching under dynamic condition

2.1. Design of NPC Tower Agent

The four NPC Combat Defense are NPC Tower Kamandaka, Gayatri NPC Tower, NPC Gandewa Tower and NPC Adikara. Each NPC Tower has several parameters that characterize them such as the value of *Damage*, *Attack Speed*, *Attack Range*, *Hitpoint*, *Attack Type* and *NPC Behavior*. The values of each parameter of each NPC Tower are described in Table 1.

Table 1. NPC Tower Statistics

No	Parameter NPC	Kamandaka	Gandewa Tower	Gayatri Tower	Adikara
1	Damage	10	7	13	18
2	Attack Speed	0.9 s	0.5 s	1.2 s	4 s
3	Attack Ranged	10	11	8	15
4	Attack Type	Single Attack	Single Attack	Splash Area	Splash Area
5	Hit Point	450	325	570	400
6	NPC Behavior	Normal Attack, Aggressive Attack, and Unique Skill	Normal Attack, Strong Attack, and Unique Skill	Normal Attack, Strong Attack, and Unique Skill	Normal Attack, and Strong Attack

2.2. Design of NPC Enemy Agent

To create NPC Tower as a Combat Defense Building, this study also designed the Enemy Agent to test the strength of NPC Tower to survive in specific situation. There are three (3) types of Enemy for NPC Agents, which are NPC *Enemy of Swords* that belong to the category of strong enemies, NPC *Enemy of Arrows* are categorized as moderate enemies and NPC *Enemy of Magic* which is a weak type of enemy. Agent NPC Agent has several parameters such as damage, movement speed (MS), attack range, Hitpoint. Full statistics of Agent NPC Agent parameter values are attached (See Table 2).

Table 2. NPC Enemy Statistics

No	Parameter	Enemy of Swords	Enemy of Arrows	Enemy of Magic
1	Damage	10	7	5
2	Armor	5	2	2
3	Hitpoint	60	40	40
4	Attack speed	1/s	1/s	1/s
5	MS	1	2	4
6	Training time	30 s	20 s	30 s
7	Attack range	1	4	4

2.3. Design of HFSM NPC Tower

This research uses Hierarchical Finite State Machine (HFSM) method to design NPC behavior. Besides that, it also used Rule Based System as a decision making for NPC when fighting with various types of enemies who attempt to enter the range of the NPC Tower. In addition, HFSM NPC Tower is symbolized by a circle as a sign of a state, which the arrow line as a sign of inter-state transfer as shown in Figure 2.

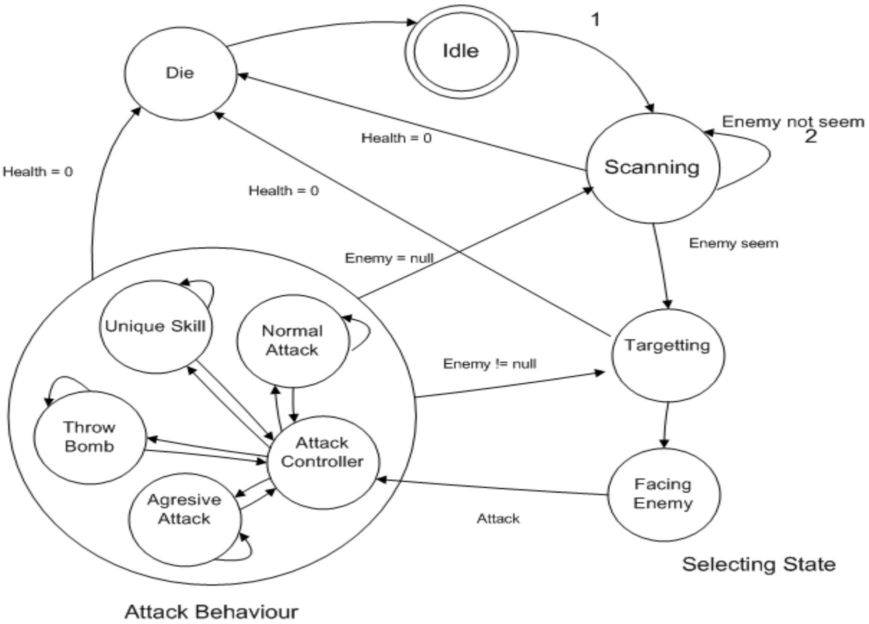

Figure 2. HFSM Agent NPC Tower

2.4. NPC Tower Behavior Design

The behavior of NPC Combat Defense is designed using rule-based decision-making methods [9-13]. In this case, the rule-based has several parameters that influence the decision making process such as the *distance*, the *enemy type*, the *NPC's life* and the *enemy's life*. Meanwhile, the NPC behavior are *Strong Attack*, *Normal Attack* and *Throwing Trap*. In addition, the NPC Combat Defense Behavior consist of:

1. Strong Attack. It is the attacking behavior of NPC with its full of available power in which NPC Tower can destroy enemy forces on a large scale or reduce the lives of enemy's hero in great damages.
2. Normal Attack. It is a NPC attack behavior with moderate strength, which NPC Tower can only destroy enemies on a small scale or lower damage compare to strong attack.
3. Unique Skill. It is the attacking behavior with special skills that only certain NPC has, which this kind of behavior becomes secret weapon in specific condition and to specific target [14, 15].
4. Silence. It is the NPC's behavior when its condition is in precarious condition or be squeezed that is unlikely to win. This particular behavior only for NPC *Archer Tower* and *Wizard Tower*.

2.5. Design of Rule-based NPC Tower

Attack Controller has a function for controlling the attack behavior of NPC Tower. In this study, it used rule-based logic [21-23] to determine the NPC Tower behavior, which is embedded in State Controller Attack. By given the rule-based knowledge, each NPC can respond to the changes in input variables into behaviors that have been designed to use HFSM. From each building combat defense, the NPC has 4 (four) type parameters that become rule-based input, which are NPC *building health*, *enemy health*, *distance* and *enemy status*. Meanwhile, the output from rule-based of NPC Combat Defense behavior through 3 (three) type of responses towards the enemy in the form of Aggressive (Strong Attack), Ordinary (Normal Attack) and Special (Unique skill). Each of those parameters has an interval range of values, which is divided into three parts namely *Weak, Moderate* and *Strong* for the Tower and Enemy Life's life parameters. Meanwhile the distance parameter has only two parts namely the short and long distance parameters. The parameters of intervals shown in Table 3, 4 and 5.

Table 3. Parameter Interval Tower Life

Interval Value	Notation
0 – 33	Weak
34 – 66	Moderate
67 – 100	Strong

Table 4. Parameter Interval Enemy Life

Interval Value	Notation
0 – 33	Weak
34 – 66	Moderate
67 – 100	Strong

Table 5. Parameter Interval Enemy Distance

Interval Value	Notation
0 – 50	Near
51 – 100	Far

3. RESULTS AND ANALYSIS

NPC Combat Defense Buildings that have been designed and described in previous chapter are implemented in simulations using Unity Game Engine. This test is aimed to evaluate the developed approach method and analyze the result obtained. Therefore, the experiment was done by comparing FSM method and HFSM method, which obtained the time results and the number of enemies killed. There are several test scenarios that have been done ie.

NPC Kamandaka against Weak team, Moderate team and Strong team.
NPC Gandewa Tower against Weak team, Moderate team and Strong team.
NPC Adikara against Weak team, Moderate team and Strong team.

The first scenario by testing 1 (one) NPC Tower against strong troops obtained the average time of killing 1 enemy troops 3.81s on testing with HFSM and 3.845s with FSM. Meanwhile, by adding 1 (one)

NPC Tower as a second scenario for testing 2 (two) NPC Tower against strong troops obtained the average of killing 1 enemy troops 2.78s on HFSM and 3.1145s with FSM. In this test NPC Tower with HFSM method can kill a maximum of 10 enemy troops while the NPC Tower with FSM have maximum of 9 enemy troops. Furthermore, in the third scenario, by testing 1 (one) NPC Tower against moderate troops obtained on average time killing 1 enemy troop 1.005s on HFSM and 1.0075s on FSM test. Then, by adding 1 (one) NPC Tower as the fourth scenario, with testing 2 (two) NPC Tower against moderate troops had the average time of killing 1 troop 1.69s on HFSM and 1.95s on FSM testing. In this test NPC Tower with HFSM method can kill a maximum of 11 enemy troops while the NPC Tower with FSM maximum of 10 enemy troops.

For the fifth scenario, this study was testing 1 (one) NPC Tower against the weak troops which resulted the average time to kill 1 enemy 1.6s on HFSM and 1.605 on FSM. On the other hand, by adding other 1 (one) NPC Tower as the sixth scenario, with testing 2 (two) NPC Tower against the weak troops gained the average time of killing 1 troop 2.27s on HFSM testing and 2.34s on FSM testing. In this test NPC Tower with HFSM method have capability to kill a maximum of 13 enemy troops while the NPC Tower with FSM can reach until 12 enemy troops. On the other hand, in the seventh scenario, testing 1 (one) NPC Gandewa Tower against strong troops have obtained the average time of killing 1 enemy troop 5.63s with HFSM and 5.60s with FSM. Then, increase the number with 1 NPC Tower as the eighth scenario, then testing 2 (two) NPC Tower against strong troops got an average result of killing 1 enemy troop 3.93s on HFSM testing and 4.52s on testing with FSM. In this test NPC Tower with HFSM method can kill a maximum of 8 enemy troops while the NPC Tower with FSM maximum of 8 enemy forces.

In the next scenario, which is ninth, this study was testing 1 (one) NPC Gandewa Tower against moderate troops resulted the average time of killing 1 (one) enemy troop 4.27s on testing with HFSM and 4.14s with FSM. Moreover, by supplementing 1 (one) NPC Tower as the tenth scenario, then testing 2 (two) NPC Tower against moderate troops gained the average result of killing 1 enemy troop 2.79s on HFSM testing and 3.23s with FSM. In this test NPC Tower with HFSM method can kill a maximum of 10 enemy troops, which have equal value with NPC Tower with FSM. In the next step, the eleventh scenario was testing 1 (one) NPC Gandewa Tower against the weak troops gained the average time of killing 1 enemy troop 4.45s on testing with HFSM and 4.45s on testing with FSM. Additing 1 (one) NPC Tower as the twelfth Scenario, then testing 2 (two) NPC Tower against weak troops obtained the average result of killing 1 enemy troop 2.92s on HFSM testing and 3.93s with FSM. In this test, NPC Tower with HFSM method can kill a maximum of 10 enemy troops, so did the NPC Tower with FSM.

In the thirteenth scenario, this study was testing 1 (one) NPC Tower Adikara against strong troops obtained the average of time to kill 1 (one) enemy troop 0.03s with HFSM and 0.13s with FSM. Adding 1 (one) NPC Tower as the fourteenth scenario, then testing 2 (two) NPC Tower against strong troops obtained the average result of killing 1 enemy troop 0.2155s on HFSM and 0.498s with FSM. In this test NPC Tower with HFSM method can kill a maximum of 16 enemy troops while the NPC Tower with FSM 15 troops. Meanwhile, in the fiftenth scenario, again testing 1 (one) NPC Tower Adikara against the moderate troops gained the average result of killing 1 enemy troop 0.07s with HFSM and 0.107s with FSM. Once again, by adding 1 (one) NPC Tower as the sixteenth scenario, then testing 2 (two) NPC Tower against moderate troops obtained the average result of killing 1 enemy troop 0.42s on HFSM test and 0.53 with FSM. In this test NPC Tower with HFSM and FSM method can kill a maximum of 21 enemy troops.

In the seventeenth scenario, the test also use one NPC Tower Adikara against the weak troops, which obtained the average time of killing 1 enemy troops 0.64s with HFSM and 0.99s with FSM. Adding 1 (one) NPC Tower as the eighteenth scenario, then by testing 2 (two) NPC Tower against the weak troops obtained the average time of killing 1 enemy troops 0.75s on HFSM testing and 1.14s with FSM. In this test NPC Tower with HFSM method can kill a maximum of 28 enemy troops while the NPC Tower with FSM also 27 troops. Furthermore, the result of comparison test between FSM and HFSM method from 18 (eighteenth) scenarios of each 3 NPC Tower against 3 (three) type of troops were presented in graph forms as shown in the Figure 3, Figure 4 and Figure 5.

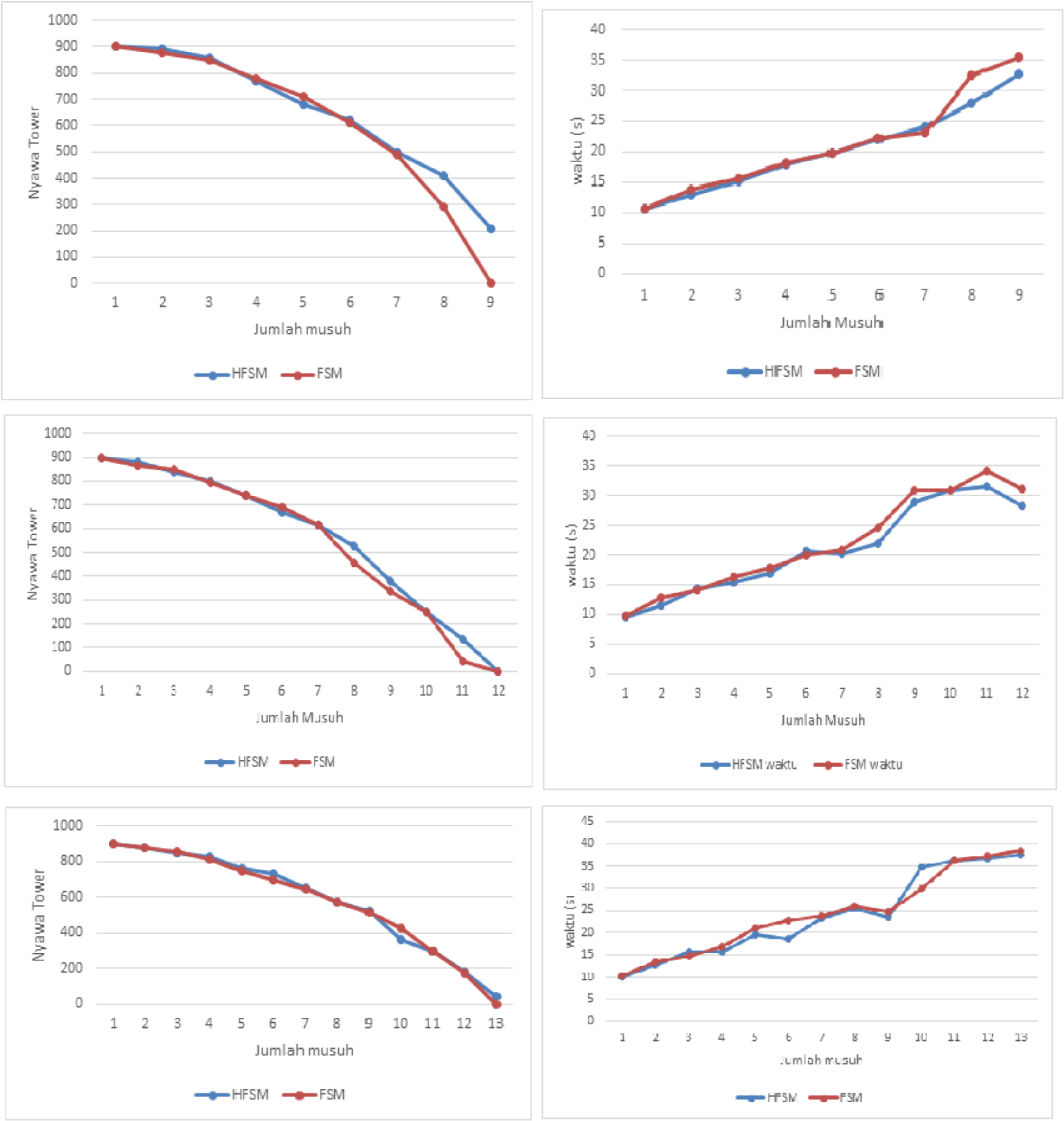

Figure 3. Simulation Results of 2 NPC Kamandaka against 3 Type of Troops

The first test using 1 NPC Tower kamandaka and 3 types of enemy troops that have different attack powers namely strong troops (sword), moderate troops (arrows) and weak troops (magic). In the combat defense conditions that only use 1 (one) Tower obtained a balanced result between HFSM and FSM method. It was found that single NPC Tower was able to withstand 6 (six) strong troop attacks. For moderate troops, single NPC Tower was able to withstand a moderate attack by maximum 8 (eight) troops. As for the weak troops, again single NPC Tower was able to survive from 9 (nine) troops. Meanwhile, by adding one NPC Kamandaka Tower provides better combat defense results in the HFSM method compared to the FSM method. In the combat defense condition with 2 (two) NPC Tower with HFSM can withstand attack from 10 strong troops, which is 10% more superior rather than Tower with FSM. On the other hand, Tower with HFSM can withstand an attack of 11 moderate troops, which is 9.09% more superior compare to Tower with FSM. At last, Tower with HFSM able to hold 13 weak troops that is 7.69% more superior compared to Tower with FSM. From these 3 (three) tests, an average score of 8.92% was obtained for HFSM more superior compared to FSM method.

In the next test, the second one using 1 (one) Gayatri Tower NPC and 3 types of enemy forces that have different attack power, which are strong (sword), troops (arrows), troops weak (magic). In the combat defense conditions that only use 1 Tower, it has obtained an equal result between HFSM methods and FSM method. Afterwards, it was found that 1 NPC Tower was able to withstand the attack of 3 (three) strong troops. For the moderate troops, 1 NPC Tower is able to withstand the attacks from 4 (four) troops. As for the weak troops, 1 NPC Tower able to survive from 5 troops. The addition number of NPC Gayatri Tower provided balanced defense results between the HFSM compared to the FSM. As a result, Gayatri tower with HFSM method is superior in terms of the parameter of the rest of life after fighting against the enemy compare to the Gayatri Tower with FSM method. At last, the third test using 1 NPC Adikara Tower and 3 types of enemy troops that also have different attack power like previous test. In the combat defense condition that only use 1 Tower Adikara presented HFSM method is more superior to FSM method. It was found that 1 NPC Tower with HFSM able to withstand strong troop attacks by 8 troops, meant 12.5% more superior than the FSM Tower. For the moderate troops, 1 NPC Tower is able to withstand 12 troop attacks by 8.33% more superior than the FSM Tower. Meanwhile, as for the weak troops, 1 NPC Tower with HFSM able to survive from 13 troops, 15.38% more superior compare to the FSM Tower.

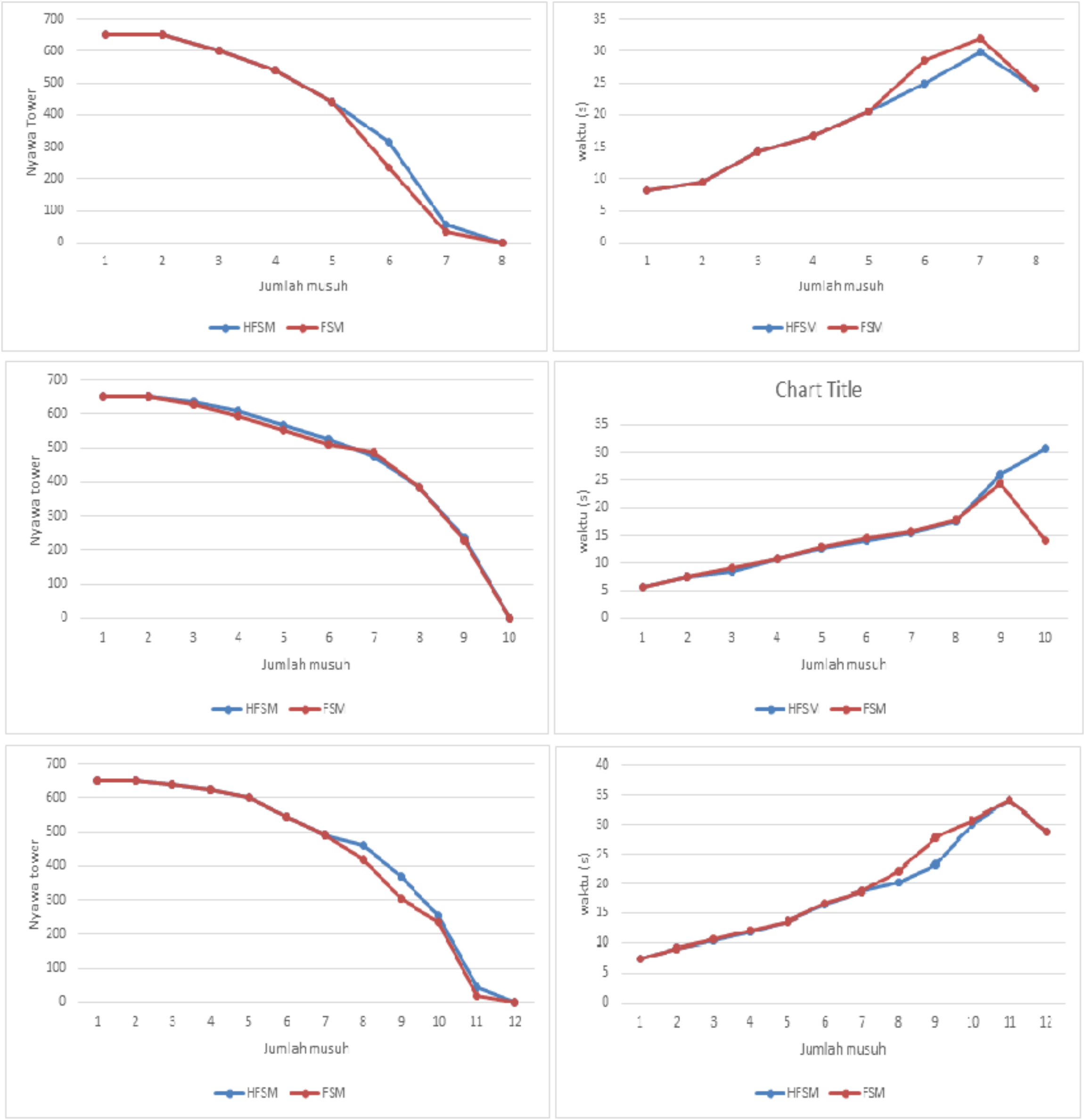

Figure 4. Simulation Results of 2 NPC Gandewa Tower against 3 Type of Troops

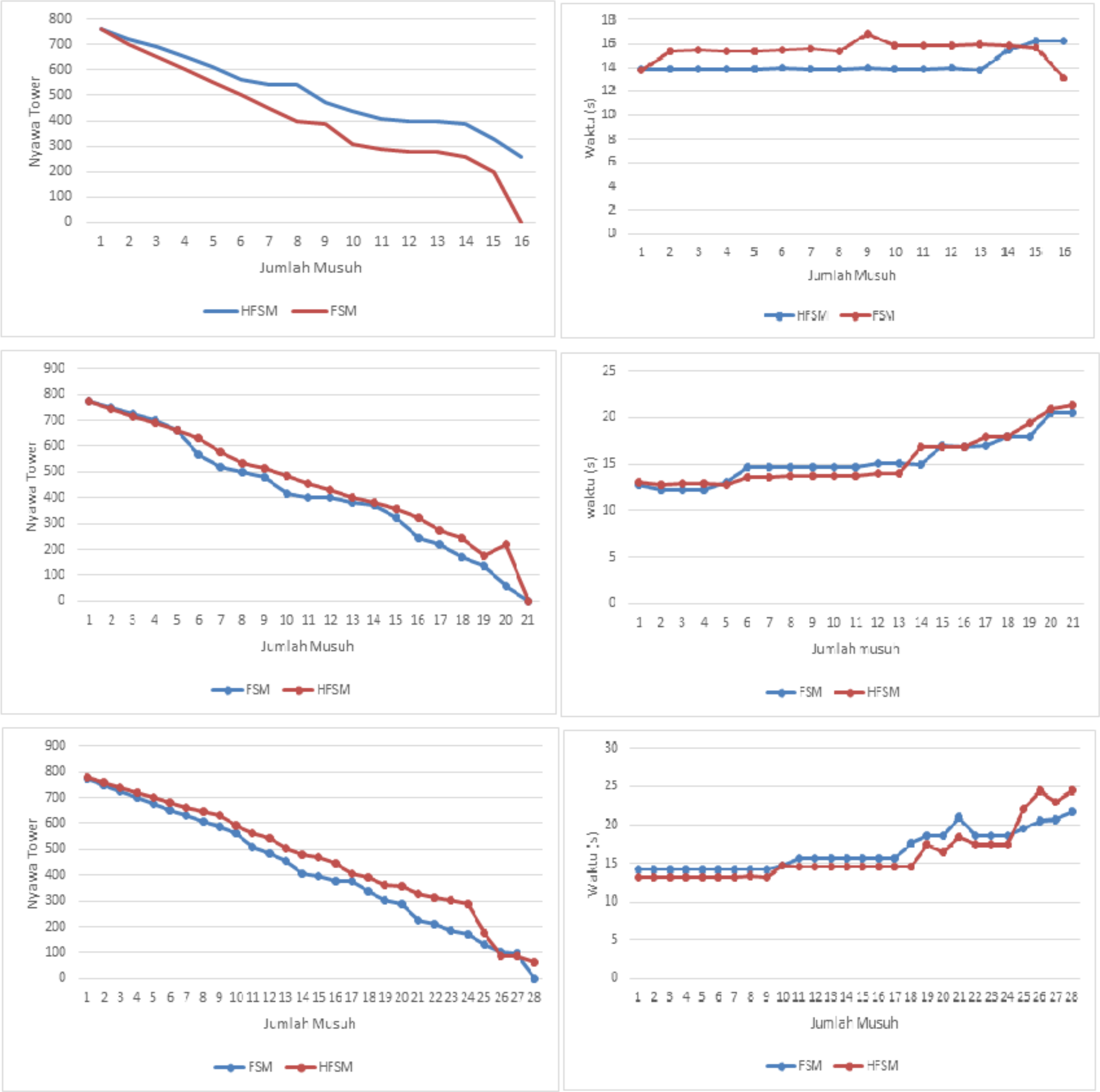

Figure 5. Simulation Results of 2 NPC Adikara Tower against 3 Type of Troops

4. CONCLUSION

The increases total number of NPC Kamandaka Tower provided better result of combat defenses compare to HFSM methods, if using FSM methods. Therefore, with 2 (two) NPC Tower with HFSM can protect the attack from 17 strong troops with leading score 5.88 % compare to the tower with FSM. Meanwhile, the tower with HFSM can endure the attack from 22 medium troops with leading score 4.54 % compare to the tower with FSM. On the other hand, the tower with HFSM can handle 29 weak troops with leading score 3.44 % compare to the tower with FSM. From 3 (three) simulations, the average score of HFSM is 4.62 % more superior compare to FSM. It can be concluded based on the evidence that HFSM method can make NPC Combat Defense Tower perform better result in 18 type of scenarios compare to FSM method. In this study the use of HFSM methods implemented in the NPC Combat Defense Building (Tower) still has the possibility to be developed further for different categories of components or other variant type of game. For example, by adding time parameters into RTS game such as in the DoTA game and the number of the NPC bullets. On the other hand, for the NPC Combat Defense Building (Tower) also can be implemented differently such as the use of Fuzzy Logic method or the optimilization with Genetic Algorithm (GA), NSGA - II method.

REFERENCES

[1] Millington I. *Artificial Intelligence for Games*. San Francisco: Morgan Kaufman Publisher. 2006.

[2] Girault A, Lee B, Lee EA. Hierarchical Finite State Machines with Multiple Concurrency Models. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 18(6), 742-760, 1999.

[3] Czerwinski R, Kania D. Synthesis Method of High Speed Finite State Machines. Bulletin of the Polish Academy of Sciences, *Technical Sciences*, 58(5), 635-644, 2010.

[4] Kielar PM, Handel O, Biedermann DH, Borrmann A. Concurent Hierarchical Finite State Machines for Modeling Pedestrian Behavioral Tendencies. *Transportation Research Procedia*, 2, 576-584, 2014.

[5] Lubis M, Kusumasari TF, Hakim L. The Indonesia Public Information Disclosure Act (UU-KIP): Its Challenges and Responses. *Int. J. of Electrical and Computer Engineering*, 8(1), 94-103, 2018.

[6] Chow S-H, Ho Y-C, Hwang T. Low Power Realization of Finite State Machines-A Decomposition Approach. *ACM Transaction on Design Automation of Electronic Systems*, 1 (3), 315-340, July 1996.

[7] DeDeo S. Conflict and Computation on Wikipedia: A Finite-State Machine Analysis of Editor Interactions. *Future Internet*, 8(x), 2016.

[8] Arif YM, Hariadi M, Nugroho SMS. Integrasi Hierarchy Finite State Machine dan Logika Fuzzy untuk Desain Strategi NPC Game. *Matics* 4(3), Maret 2011.

[9] Lee B, Lee EA. Interaction of Finite State Machines and Concurrency Models. *Circuits, Systems and Computers* 2, 1998.

[10] Mlakic D, Nikolovski S, Alibasic E. Designing Automatic Meter Reading System using Open Source Hardware and Software. *Int. J. of Electrical and Computer Engineering*, 7(6), 3282-3291, Dec. 2017.

[11] Mallouk W, Clua E. An Object-Oriented Approach for Hierarchical State Machine. *Proceedings of the SBGames Conference in Computing*. Recife, Brazil. 2006.

[12] Majid NK, Hariadi M, Nugroho SMS. *Variasi Perilaku Tarung Non Player Character Prajurit Berbasis Distribusi Gaussian pada Real Time Strategy*. Surabaya: Jurusan Teknik Elektro, Institut Teknologi Sepuluh Nopember. 2010.

[13] Arif YM, Hariadi M, Nugroho SMS. *Strategi Menyerang Pada Game FPS Menggunakan Hierarchical Finite State Machine dan Logika Fuzzy*. Surabaya: Jurusan Teknik Elektro, Institut Teknologi Sepuluh Nopember. 2010.

[14] Tremblay J, Dragert C, Verbrugge C. Target Selection for AI Companions in FPS Games. *Int. Con. on FDG*, 2014.

[15] Tremblay J, Torres PA, Verbrugge C. Measuring Risk in Stealth Games. *Int. Con. on FDG*, 2014.

[16] Doherty D, O'Riordan C. *The Design Goals and Implementation of AI in Modern Computer Games*. Galway: D. of IT, National University of Ireland. 2006.

[17] Wahono RM. *Pengantar Software Agent: Teori dan Aplikasi*. IECI Chapter Japan Series, 3(1), 2001.

[18] Adisusilo AK. *Optimization Agent Behaviour on Boxing Game Using Genetic Algorithm*. Surabaya: Jurusan Teknik Elektro, Institut Teknologi Sepuluh Nopember. 2013.

[19] Widyastuti I. *Pemilihan Perilaku NPC Pada Game Pertarungan Jarak Dekat Menggunakan Fuzzy Coordinator*. Surabaya: Jurusan Teknik Elektro, Institut Teknologi Sepuluh Nopember. 2012.

[20] Benveniste A, Berry G. The Synchronous Approach to Reactive and Real-Time Systems. *Proceedings of the IEEE*, 79 (9), September 1991.

[21] Alur R, Kannan S, Yannakakis M Communicating Hierarchical State Machines. *Int. Colloqium of Automata, Languages and Programming*, Springer pp. 169-178, 1999.

[22] Risler M. *Behavior Control for Single and Multiple Autonomous Agents Based on Hierarchical Finite State Machines*. Doctoral Dissertation, Technischen Universitat Darmstadt, 2009.

[23] Kurt A, Ozguner U. Hierarchical Finite State Machines for Autonomous Mobile Systems. *Control Engineering Practice*, 21 (2), pp. 184-194. February 2013.

[24] Hakim L, Kusumasari TF, Lubis M. Text Mining of UU-ITE Implementation in Indonesia. *J. of Physics: Conference Series* 1007 (1).

[25] Pinheiro AC, Adenilso S, Ana Maria A. FSM-Based Test Case Generation Methods Applied to Test the Communication Software on Board the ITASAT University Satellite: A Case Study. Sao Jose dos Campos. *J. Aerospsace Technology Management*, 6(4), 447-461, Oct-Dec 2014.

[26] Chow TS. Testing Software Design Modelled by Finite-State Machines. *IEEE Transactions on Software Engineering*, 4(3), 178-187. 1978

[27] Fujiwara S, van Bochmann G, Khendek F, Amalou M, Ghedamsi A. Test Selection based on Finite State Models. *IEEE Transactions on Software Engineering*, 17(6), 591-603. 1991.

[28] Santiago V, Vijaykumar NL, Guimaraes D, Amaral AS, Ferreira E. An Environment for Automated Test Case Generation from Statechart and Finite State Machine-based Behavioral Models. Proceedings of the IEEE International Con. on Software Testing Verification and Validation Workshop. Washington. DC, USA. *IEEE Computer Society*, pp. 63-72, 2008.

[29] Salauyou V and Grzes T. FSM state assignment methods for low-power design. *Int. Con. Computer Information Systems and Industrial Management Appliations*, pp. 345-350, 2007.

[30] Lala PK. An Algorithm for the State Assignment of Synchronous Sequential Circuits. *Electronics Letters* 14(6), pp. 199-201, 1978.

[31] Ahlan AR, Lubis M, Lubis AR. Information Security Awareness at the Knowledge-based Institution: Its Antecedents and Measures. *Procedia Computer Science* 72, pp. 361-373, 2015.