

Pemetaan Perilaku *Non-Playable Character* Pada Permainan Berbasis *Role Playing Game* Menggunakan Metode *Finite State Machine*

Matahari Bhakti Nendya¹, Samuel Gandang Gunanto² dan R. Gunawan Santosa³

¹Jurusan Teknik Elektro, Fakultas Teknologi Industri,
Institut Teknologi Sepuluh Nopember Surabaya
e-mail: dida.nendya@gmail.com

²Program Studi Animasi, Fakultas Seni Media Rekam,
Institut Seni Indonesia Yogyakarta
e-mail: sgandang@yahoo.com

³Program Studi Teknik Informatika, Fakultas Teknologi Informasi,
Universitas Kristen Duta Wacana, Yogyakarta
e-mail: gunawan@ukdw.ac.id

Abstrak

Role Playing Game merupakan permainan dimana para pemainnya memerankan tokoh-tokoh tertentu dan berkolaborasi dengan tokoh lain untuk membentuk suatu cerita. *Role Playing Game* identik dengan *Non-Playable Character* (NPC), oleh karena itu diperlukanlah sebuah metode pengambilan keputusan yang cerdas. *Finite State Machine* merupakan salah satu metode pengambilan keputusan yang cerdas sehingga bisa dilihat aspek kedinamisan dari *game*. Penelitian ini berfokus kepada pemodelan perilaku NPC menggunakan *Finite State Machine* dan implementasinya sebagai pada *game* berbasis *Role Playing Game* menggunakan *Game Engine* RPG Maker XP. Ada empat jenis NPC yang menjadi acuan, yaitu NPC partner, NPC quest, NPC enemy dan NPC pendukung cerita. Setiap NPC memiliki ciri khas tersendiri sehingga membuat *player* akan menerima respon yang berbeda-beda ketika kondisi *player* itu berbeda. Hasil penelitian menunjukkan bahwa *Finite State Machine* menghasilkan sebuah respon yang dinamis yang tentu saja membantu *player* untuk memahai jalur cerita dan pembentukan karakter *player* pun dapat terjadi.

Kata kunci: *non-playable character, role playing game, finite state machine.*

Abstract

Role Playing Game is a game where players play certain figures and collaborate with other characters to form a story. *Role Playing Game* synonymous with *Non-Playable Character* (NPC), therefore it requires the a method of making an intelligent decision. *Finite State Machine* is one method pengambilakan smart decision so that it can be seen the dynamic aspect of the game. This study focuses on modeling the behavior of NPCs using *Finite State Machine* and its implementation as the *Role Playing Game* based games using the *Game Engine* RPG Maker XP. There are four types of NPC as the reference, the NPC partner, NPC quests, NPC NPC enemy and supporting the story. Each NPC has its own characteristics that make the player will receive a different response when conditions were different player. The results showed that the *Finite State Machine* generates a dynamic response which of course helps the player to memahai story line and the player character formation can occur.

Keywords: *non-playable character, role playing game, finite state machine.*

Pendahuluan

Game merupakan salah satu tipe animasi interaktif yang membuat *player* dapat berinteraksi dengan permainan. *Game* haruslah dapat memberikan tantangan kepada *player* agar *player* tidak merasa bosan dan bersemangat untuk melanjutkan *game* hingga tamat. *Game* akan disebut menarik jika *game* tersebut tidak terlalu mudah dan tidak terlalu sulit. Salah satu *genre game* adalah *Role Playing Game*. *Role Playing Game* atau yang biasa disebut dengan RPG merupakan permainan dimana *player* memerankan tokoh-tokoh tertentu dan berkolaborasi dengan tokoh lain untuk membentuk suatu cerita. Para pemain dapat memilih aksi tokoh-tokoh mereka berdasarkan karakteristik tokoh tersebut. Keberhasilan aksi *player* dalam permainan RPG sangat bergantung pada sistem peraturan yang telah ditentukan. Asal tetap mengikuti peraturan yang ditetapkan, *player* dapat melakukan improvisasi membentuk arah dan hasil akhir permainan. Metode kecerdasan buatan *Finite State Machine* digunakan untuk memberikan kedinamisan respon dari NPC yang akan membuat permainan lebih menarik dan membuat NPC seolah-olah berpikir.

Game

Game merupakan sebuah jenis aktivitas bermain yang situasinya dibuat seolah-olah lebih dari kehidupan nyata, di mana pesertanya berusaha mempertahankan satu atau lebih kemampuan bertahan untuk mendapatkan tujuan yang dibatasi dengan beberapa aturan tertentu. *Game* didesain dan diciptakan berdasarkan acuan kehidupan sehari-hari. (Adams & Rowling, 2007).

Role Playing Game

Role Playing Game merupakan permainan dimana pemain dapat melakukan kendali terhadap satu atau lebih karakter yang dirancang sendiri oleh pemain dan membantu pemain untuk menyelesaikan misi (*quest*) yang diberikan oleh komputer. Keberhasilan dalam menyelesaikan

game ber-genre *role playing game* bergantung pada diselesaikannya semua *quest* yang ada. Perkembangan karakter seperti kekuatan dan kemampuan (*abilities*) merupakan kunci utama dalam permainan RPG. *Quest* yang diberikan dapat berupa taktik pertarungan, *logistic*, pertumbuhan ekonomi, eksplorasi dan penyelesaian teka-teki. Tantangan yang berupa fisik sangat jarang diberikan kecuali pada gabungan antara *genre* permainan RPG dan *action*. (Adams & Rowling, 2007).

Non-Playable Character

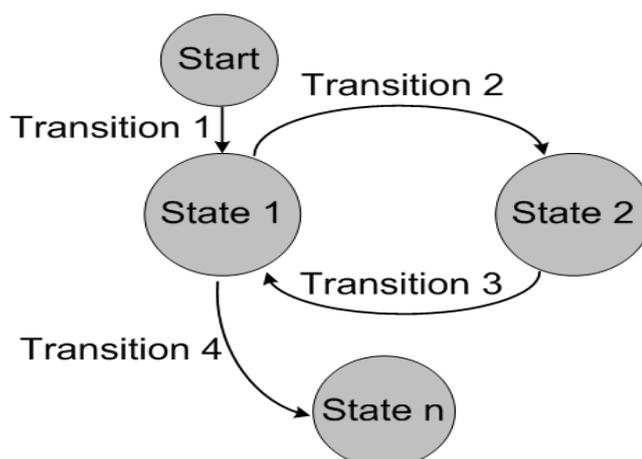
Non-Playable Character merupakan jenis *autonomus agent* yang ditunjukkan untuk penggunaan komputer animasi dan media interaktif seperti *games* dan *virtual reality*. Agen ini mewakili tokoh dalam cerita atau permainan dan memiliki kemampuan untuk improvisasi tindakan mereka. Ini adalah kebalikan dari seorang tokoh dalam sebuah film animasi, yang tindakannya ditulis di muka, dan untuk “*avatar*” dalam sebuah *game* atau *virtual reality*, tindakan yang diarahkan secara real time oleh *player*. (Reynolds, 1999).

Finite State Machine

Finite State Machine (FSM) merupakan pemodelan dari perilaku (*behavior*) sebuah sistem atau obyek yang kompleks dengan beberapa kondisi atau mode yang terdefiniskan dimana mode transisi berubah sesuai dengan keadaan.

FSM terdiri dari empat elemen utama:

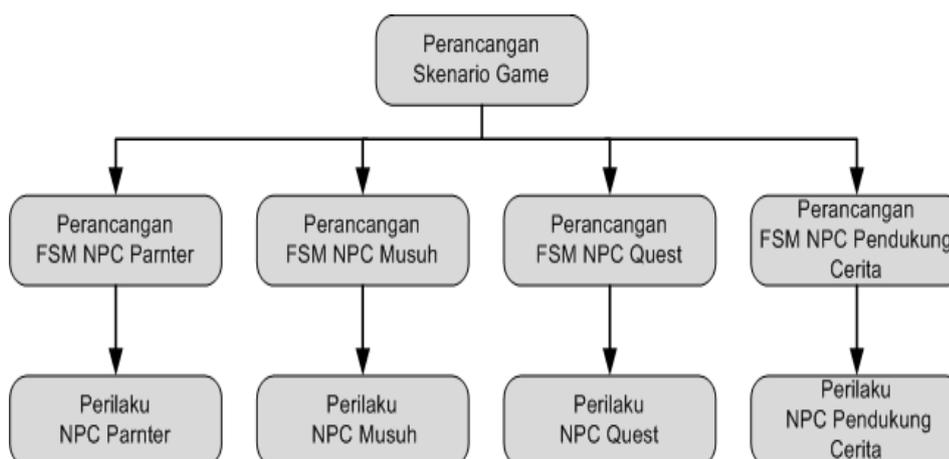
1. *State* yang mendefinisikan kelakuan dan mungkin menghasilkan aksi.
2. Transisi *state* dimana merupakan perpindahan dari satu *state* ke *state* lain.
3. Aturan atau kondisi yang harus dipenuhi supaya ada transisi *state* kejadian (*events*).
4. Input yang terjadi baik *internal* maupun *external*, yang memungkinkan *trigger* aturan dan mengacu ke transisi *state*.



Gambar 1. Contoh Diagram *State* Sederhana

Metode Penelitian

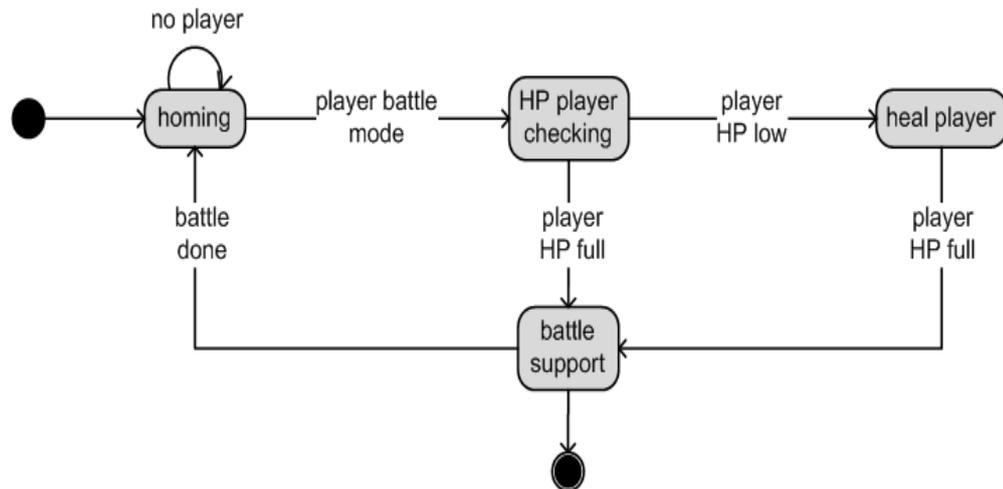
FSM digunakan untuk menentukan perilaku NPC berdasarkan kondisi (*state*) yang dimiliki oleh *player*. Gambar 2 menunjukkan tahapan-tahapan dalam penelitian yang meliputi perancangan skenario *game*, perancangan FSM NPC partner, perancangan FSM NPC musuh, perancangan FSM NPC quest, dan perancangan FSM NPC pendukung cerita.



Gambar2. Blok Diagram Metode Penelitian

Perancangan FSM NCP Parter

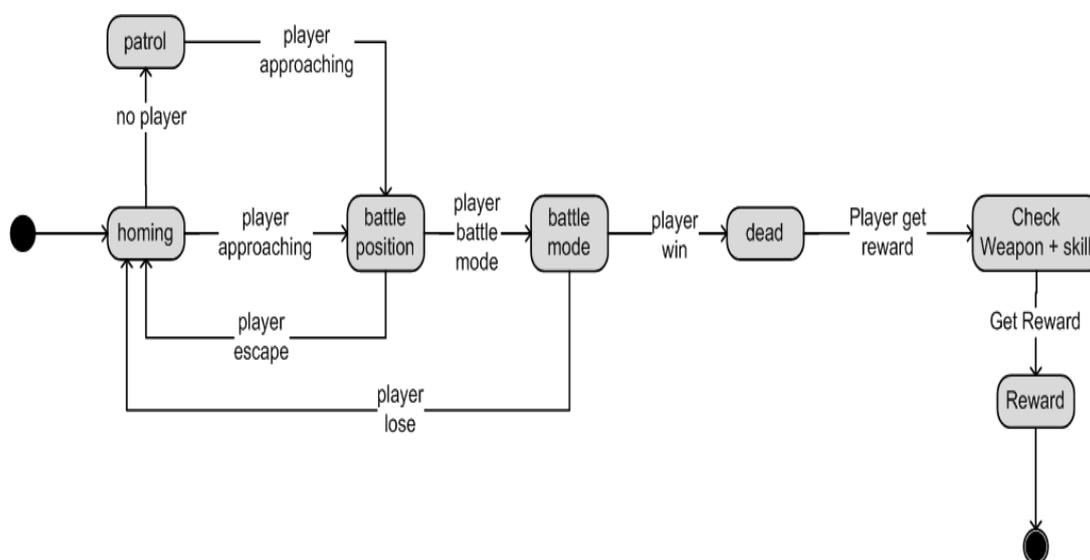
NPC Partner merupakan NPC yang bertugas membantu *player* dalam *battle system*. Perilaku yang dimiliki oleh NPC partner adalah *HP player checking*



Gambar 3. State Diagram FSM NPC Partner

State awal dari NPC Partner adalah state “*homing*”. Dalam state ini NPC tidak memberikan respon kepada *player*, akan tetapi NPC tetap mengikuti pergerakan *player* kemanapun. *Player* akan masuk kedalam *player battle mode* saat *player* bertemu dengan NPC musuh. Saat *player* masuk dalam *battle mode*, NPC akan masuk kedalam state “*HP player checking*”. NPC bertugas untuk melakukan pengecekan terhadap *HP player* sebagai tanda bahwa *player* layak untuk bertarung atau tidak. Ada dua kemungkinan yang dapat terjadi. Yang pertama *HP player* dinyatakan *low* oleh perhitungan NPC, sehingga NPC akan melakukan state “*heal player*”. Saat *HP player* telah dinyatakan *full*, maka NPC akan masuk kedalam state “*battle support*”. Yang kedua *HP player* dinyatakan *full*, sehingga NPC partner langsung masuk kedalam state “*battle support*” dan membantu *player* bertarung dengan NPC lain.

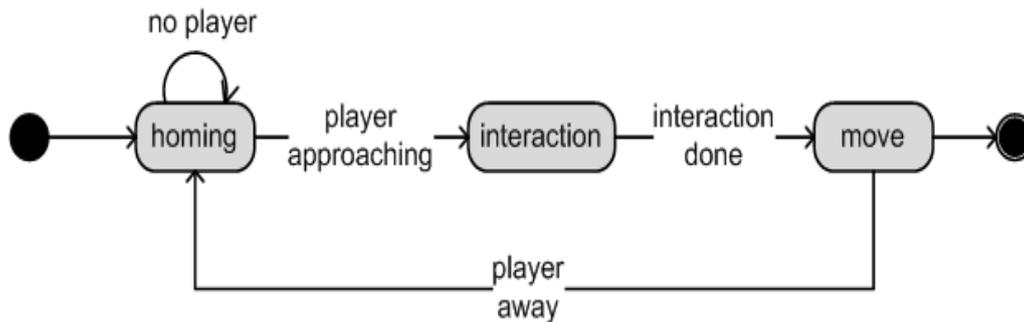
Perancangan FSM NCP Musuh



Gambar 4. State Diagram FSM NPC Partner

State awal dari NPC Musuh adalah state “*homing*”. Dalam state “*homing*” ini, ada dua kondisi yang transisi yang mempengaruhi, yaitu kondisi “*player approaching*” dan “*no player*”. Saat *player* mendekat, NPC akan masuk kedalam state “*battle position*”, sedangkan saat tidak ada *player*, maka NPC akan masuk kedalam state “*patrol*”. Dalam kondisi state “*patrol*” juga akan berubah menjadi state “*battle position*” saat *player* datang. Ada dua kemungkinan yang terjadi dalam state “*battle position*”, yaitu *player* masuk kedalam *battle mode* atau *player* kabur. Ketika *player* masuk kedalam *battle mode*, NPC masuk kedalam state “*battle mode*” dimana akan ada pertarungan antara NPC dan *player*. Sedangkan saat *player* kabur, maka NPC akan kembali ke state “*homing*”. Ada dua kemungkinan lagi yang terjadi saat NPC berada dalam state “*battle mode*”, yaitu *player win* dan *player lose*. Ketika *player win*, maka NPC akan masuk dalam state “*dead*” dan bisa di pastikan NPC tersebut mati. Sedangkan ketika *player lose*, NPC akan masuk dalam state “*homing*” dan ini menandakan *game* masuk kedalam *Game Over* karena *player* mengalami kekalahan saat *battle* dengan NPC.

Perancangan FSM NCP Pendukung Cerita



Gambar 6. *State Diagram* FSM NPC Pendukung Cerita

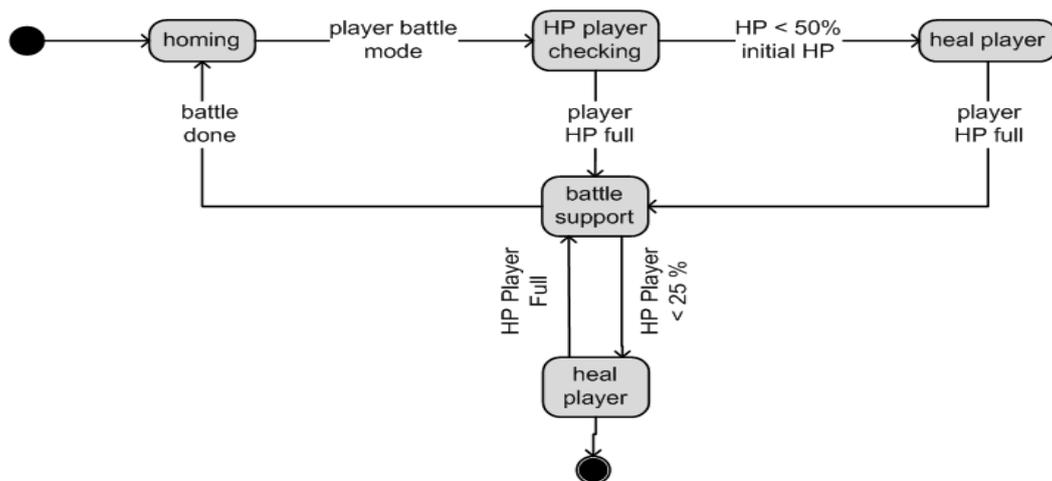
State awal dari NPC pendukung cerita adalah *state* “*homing*” hingga *player* datang mendekati. Saat *player* mendekati maka NPC akan masuk kedalam *state* “*interaction*” dimana terjadi interaksi antara *player* dan NPC. Interaksi yang terjadi bisa berupa sebuah petunjuk untuk melangkah ke *stage* berikutnya. Setelah interaksi selesai, NPC akan masuk kedalam *state* “*move*” dan menunggu hingga *player* menjauh untuk kembali kepada *state* awal.

Hasil Dan Pembahasan

Pengujian respon NPC yang menggunakan *Finite State Machine* dilakukan dengan cara *player* mendekati NPC dengan berbagai macam variasi *state* (kondisi) yang meliputi *mission*, HP, *skills*, *item*, *weapon* dan lainnya. NPC akan memberikan respon yang kemudian secara langsung *player* akan bertindak dengan memberikan respon balasan kepada NPC. Pengujian FSM sendiri hanya akan mengambil 1 buah NPC dari tiap-tiap jenis NPC yang ada.

Pengujian Respon NPC Partner

Prosedur simulasi respon NPC Partner mengambil Aluxes sebagai karakter *player* dan *Hilda* sebagai karakter NPC Partner. Proses pengujian dilakukan pada saat terjadi *battle system*. NPC Partner akan memberikan respon yang telah diatur pada *state diagram* berikut ini:



Gambar 7. *State Diagram* Respon NPC Partner

Dalam pengujian ini ada 3 *state-transition* yang sangat berpengaruh terhadap variasi respon yang diberikan oleh NPC. Variasi respon yang terjadi mengacu pada parameter *HP player*.

1. Pengujian ketika *HP Player* maksimal (*full*).

Condition : Turn 0

Program akan melakukan pengecekan nilai *HP player* yang sesuai dengan nilai awal sebelum terjadinya *battle system*. Pengecekan nilai *HP player* ini tentunya akan mengubah kondisi *state* “*HP player full*” → *state* “*battle support*”.

```

    @>Script: $game_system.face_graphic = "Hilda"
    @>Script: $game_system.face_graphic = "Hilda"
    @>Change Party Member: Add[Hilda], Initialize
    
```

Program akan memberikan informasi kepada *player* dan menambahkan NPC *Hilda* sebagai partner *player* dalam *battle system*.



Gambar 8. Hasil Pengujian Respon NPC Partner ketika HP *Player* maksimal

2. Pengujian ketika HP *Player* $\leq 50\%$ nilai awal HP sebelum terjadinya *battle system*.

Condition : Actor [Aluxes]'s HP 50% or below

Program akan melakukan pengecekan nilai HP *player* yang sesuai dengan nilai awal sebelum terjadinya *battle system*. Ketika nilai HP *player* $\leq 50\%$ dari nilai awal sebelum terjadinya *battle system*, maka akan terjadi perubahan kondisi *state* "HP $\leq 50\%$ initial HP" \rightarrow state "Heal Player".

```
@>Recover All: [Aluxes]
@ShowAnimation: Player, [HP Restoration 1]
@>Change Party Member: Add[Hilda], Initialize
```

Program akan memberikan informasi kepada *player*, melakukan *recovery* HP *player* menjadi maksimal dan menambahkan NPC *Hilda* sebagai partner *player* dalam *battle system*.



Gambar 9. Hasil Pengujian Respon NPC Partner ketika HP *Player* $\leq 50\%$ nilai awal HP sebelum terjadinya *battle system*

3. Pengujian ketika HP *Player* $\leq 25\%$ saat terjadinya *battle system*.

Program akan melakukan pengecekan apakah *Hilda* (NPC partner) sudah berada di dalam *battle system*. Dalam hal ini NPC partner sudah berada dalam *state "battle support"*.

```
@>Conditional Branch: [Hilda] is in the party
  @>Text: Aluxes, lukamu cukup parah
  @>Recover All: [Aluxes]
  @>Show Animation: Player, [HP Restoration1]
  @>
: Else
  @>
: Branch End
@>
```

Terjadinya perubahan *state "battle support"* \rightarrow *state "heal player"* dipicu pada :

Condition : Actor [Aluxes]'s HP 25% or below

Ketika terjadi nilai HP *player* $\leq 25\%$ saat *battle system*, NPC akan langsung masuk kedalam *state "heal player"* dan melakukan *recovery* HP *player* melalui :

Recover All : [Aluxes]



Gambar 10. Hasil Pengujian Respon NPC Partner ketika HP *Player* $\leq 25\%$ saat terjadinya *battle system*

Analisis Evaluasi Kuisisioner Game

Prosedur pengujian kuisisioner ini dilakukan dengan menguji *game* pada 30 responden. Prosedur pengujian kuisisioner ini juga mengambil masing-masing 1(satu) NPC dari tiap-tiap jenis NPC yang ada.

Pengujian Kuisisioner NPC Partner

Proses pengujian terhadap NPC Partner terjadi ketika *player* mendapatkan *quest* untuk mengalahkan *goblin* yang datang mengganggu ketenangan *Ormad Village*. *Player* berperan sebagai *Aluxes* dan NPC Partner diwakili oleh *Hilda*. *Player* dan NPC Partner bersama-sama melawan *goblin*. Parameter yang digunakan untuk menguji kondisi ini adalah *state "Heal"*. Dari pengujian terhadap 30 orang responden, didapatkan hasil sebagai berikut :

Tabel 1. Hasil Pengujian Terjadinya *State "Heal"*

No.	<i>State "Heal"</i>	Jumlah	Persentase
1.	Ya	25	83 %
2.	Tidak	5	17 %
Total		30	100 %

Dari tabel 1 diperoleh 25 responden atau 83 % mengalami terjadinya *state "Heal"* dan 5 responden (17 %) tidak mengalami *state "Heal"*. Tidak terjadinya *state "Heal"* ini kemungkinan dikarenakan *player* mengalami *state "HP player full"* dimana akibat dari penggunaan *item* atau *recover HP* yang terjadi pada *quest* sebelumnya. Hasil dari 25 responden yang mengalami *state "Heal"* ini kemudian digunakan untuk pengujian selanjutnya. Pengujian selanjutnya adalah pengujian *state-transition* yang menuju *state "Heal"* yaitu *state-transition "HP Player \leq 50% initial HP"* dan *state-transition "HP Player \leq 25%"*. Pengujian *state-transition* tersebut menghasilkan data sebagai berikut :

Tabel 2. Hasil Pengujian Terjadinya *state-transition "HP Player \leq 50% initial HP"* dan *state-transition "HP Player \leq 25%"*

No.	<i>State-transition</i>	Jumlah	Persentase
1.	<i>HP Player \leq 50% initial HP</i>	19	76 %
2.	<i>HP Player \leq 25%</i>	6	24 %
Total		25	100 %

Dari tabel 2 diperoleh hasil 19 responden atau 76% mengalami terjadinya *state-transition "HP Player \leq 50% initial HP"*. Hal ini dimungkinkan terjadi karena sebelum masuk kedalam *battle system* melawan *goblin*, *player* mengalami kekurangan HP akibat dari *quest* sebelumnya, yaitu berlatih dengan *Master Vashili*. *Player* yang mengalami *state-transition "HP Player \leq 25%"* dikarenakan besarnya nilai ATK (*attack*) dari *goblin* yang mencapai 80 point.

Pengujian Kuisisioner NPC Musuh

Proses pengujian terhadap NPC Musuh terjadi ketika *player* bertarung melawan *Goblin* setelah berhasil melaksanakan *quest* dari *Master Vashili*. Parameter yang digunakan adalah *state "reward"*. Dari pengujian terhadap 30 responden didapatkan hasil sebagai berikut :

Tabel 3 Hasil Pengujian terhadap *State "reward"*

No.	<i>State "reward"</i>	Jumlah	Persentase
1.	EXP +10	3	10 %
2.	EXP +15	5	17 %
3.	EXP +20	13	43 %
4.	EXP +25	9	30 %
Total		30	100 %

Dari tabel 3 diperoleh hasil 3 responden (10 %) mendapatkan *reward* berupa EXP+10. Kondisi ini terjadi ketika *player* hanya menggunakan *weapon iron sword* dan dipastikan tidak melanjutkan untuk menyelesaikan latihan yang kedua serta tidak menggunakan *reward* yang didapatkan dari latihan pertama. *State "reward"* kedua, 5 responden (17 %) mendapatkan *reward* berupa EXP+15. Kondisi ini terjadi ketika *player* hanya menggunakan *weapon bronze sword*. *Player* memilih untuk tidak melanjutkan latihan yang kedua akan tetapi menggunakan *reward* yang didapatkan dari latihan pertama. *State "reward"* ketiga, 13 responden (47%) mendapatkan *reward* berupa EXP+20. Kondisi ini terjadi ketika *player* telah berhasil menyelesaikan latihan hingga latihan kedua. Akan tetapi dalam *battle system* tidak melakukan perubahan terhadap *weapon* yang dimilikinya. *State "reward"* keempat, 9 responden (30%) mendapatkan *reward* berupa EXP+25. Kondisi ini dapat terjadi ketika *player* telah berhasil menyelesaikan latihan hingga latihan yang kedua dan dalam *battle system* *player* melakukan perubahan terhadap *weapon* yang dimilikinya. Dari keempat hasil pengujian *state "reward"* diatas, dapat disimpulkan bahwa responden lebih memilih untuk menyelesaikan jalan cerita daripada mengoptimalkan pembangunan karakter.

Pengujian Kuisisioner NPC Quest

Proses pengujian terhadap NPC *Quest* terjadi ketika *player* bertemu dengan *Master Vashili*. Parameter yang digunakan dalam pengujian adalah *state “interaction”* dan *state “reward”*. *State “interaction”* menandakan bagaimana kondisi *player* ketika pertama kali bertemu dengan *Master Vashili* dan *state “reward”* merupakan penghargaan (*reward*) yang didapatkan oleh *player* ketika berhasil menyelesaikan *quest* dari *Master Vashili*. Pengujian pertama yang dilakukan adalah pengujian *state “interaction”*. Dari pengujian terhadap 30 responden didapatkan hasil sebagai berikut :

Tabel 4. Hasil Pengujian terhadap *State “interaction”*

No.	<i>State “interaction”</i>	Jumlah	Persentase
1.	Kembali mengambil pedang	26	87 %
2.	Memberi <i>quest</i> baru	4	13 %
Total		30	100 %

Dari tabel 4 diperoleh hasil 26 responden (77%) mendapat respon untuk mengambil pedang. Kondisi ini terjadi ketika *player* melupakan *item* yang penting, yaitu *bronze sword*. Selanjutnya 4 responden (13%) yang mendapatkan *quest* baru menandakan bahwa *player* benar-benar mendapatkan semua *item* yang dibutuhkan untuk menyelesaikan permainan. Hasil dari kedua *state “interaction”* diatas dapat disimpulkan bahwa *player* kurang melakukan eksplorasi untuk mendapatkan *item*.

Pengujian kedua merupakan pengujian *state “reward”*. Dari pengujian terhadap 30 responden didapatkan hasil sebagai berikut :

Tabel 5 Hasil Pengujian terhadap *State “reward”*

No.	<i>State “reward”</i>	Jumlah	Persentase
1.	<i>Weapons</i>	12	40 %
2.	<i>Weapons dan Skills</i>	18	60 %
Total		30	100 %

Dari tabel 5 diperoleh hasil 12 responden (40%) mendapatkan *reward* berupa *weapons*. Kondisi ini terjadi ketika *player* hanya menyelesaikan latihan yang pertama. Pada *state “reward”* yang kedua, 18 responden (60%) mendapatkan *reward* berupa *weapons* dan *skills*. Kondisi ini terjadi ketika *player* telah berhasil menyelesaikan latihan

hingga latihan kedua dan juga dianggap telah menyelesaikan *quest* dengan baik. Hasil dari kedua *state "reward"* diatas dapat disimpulkan bahwa *player* tidak mengikuti semua *quest* hingga selesai dan lebih memilih untuk membuat alternative penceritaan sendiri sesuai dengan kondisi *player* dan lingkungan permainan.

Pengujian Kuisisioner NPC Pendukung Cerita

Proses pengujian terhadap NPC Pendukung Cerita terjadi ketika *player* melakukan interaksi pertama dengan *villager* yang merupakan NPC Pendukung Cerita itu sendiri. Parameter yang digunakan dalam pengujian adalah *state "interaction"*. Dari pengujian terhadap 30 responden, diperoleh hasil sebagai berikut :

Tabel 6 Hasil Pengujian terhadap *State "interaction"*

No.	<i>State "reward"</i>	Jumlah	Persentase
1.	Lokasi rumah <i>Master Vashili</i> .	17	57 %
2.	Meminta tolong menghentikan serangan <i>goblin</i> .	9	30 %
3.	Mengucapkan terima kasih karena <i>Omrad Village</i> sudah kembali normal	4	13 %
Total		30	100 %

Dari tabel 6 diperoleh 17 responden (57 %) mendapatkan respon kepada *player* berupa lokasi rumah *Master Vashili*. Kondisi ini terjadi ketika *player* sedang berada dalam *quest* pertama. Adapun 9 responden (30 %) mendapatkan respon berupa permintaan untuk menghentikan serangan *goblin*. Kondisi ini terjadi ketika *player* sedang berada didalam *quest* kedua. Sedangkan 4 responden (13%) mendapatkan respon berupa permintaan terima kasih karena *Omrad Village* sudah kembali normal. Kondisi ini terjadi ketika *player* telah berhasil menyelesaikan *quest* kedua. Dengan adanya respon yang berbeda-beda dari NPC Pendukung Cerita, *player* akan sangat mudah untuk mengetahui kondisi saat menjalankan *quest* dan merupakan salah satu indikasi dari aspek kedinamisan sebuah cerita.

Kesimpulan Dan Saran

Finite State Machine dapat digunakan untuk memberikan variasi respon yang dinamis pada agen cerdas atau NPC. Tingkat kedinamisan respon dari NPC itu sendiri sangat bergantung pada bagaimana pola perancangan lingkungan pada *game*.

Untuk pengembangan pemetaan perilaku *Non-Playable Character* pada permainan berbasis *Role Playing Game*, dimasa yang akan datang metode *Finite State Machine* dapat dikembangkan lebih lanjut untuk mendapatkan NPC yang bisa beradaptasi dengan lingkungan *game* sehingga variasi respon menjadi lebih beragam dan *game* akan menjadi lebih menarik. Penggabungan metode *Finite State Machine* dengan metode kecerdasan lain seperti *Logika Fuzzy* menghasilkan *Fuzzy State Machine* yang dapat digunakan untuk mendapatkan variasi respon NPC yang lebih ketika berinteraksi dengan *player*.

Referensi

- Brownlee, Jason.(2004). *Finite state machines in games*. Diakses pada : 17 Desember 2011 dari World Wide Web: <http://ai-depot.com/FiniteStateMachines/FSM.html>.
- Jasson.(2009). *Role Playing Game (RPG) Maker Software Penampung Kreativitas, Inovasi & Imajinasi Bagi Game Designer*. Yogyakarta : Penerbit Andi.
- Hermasnyah, D. dan Hariadi. M. (2009). *English Vocabulary Games With Automatic Leveling* (Tesis S2, Institut Teknologi Sepuluh Nopember Surabaya), dari <http://digilib.its.ac.id>
- Kim, C.H., Jeong, S.M., Hur, G.T., dan Kim, B.G. (2006). Verification of FSM using Attributes Definition of NPCs Models, *IJCSNS International Journal of Computer Science and Network Security*, VOL.6 No.7A, July 2006.168-174.
- Lim, M.Y., Dias, J., Aylett, R., dan Paiva, A. (2009). *Intelligent NPCs for Educational Role Play Game*. Edinburgh: Heriot Watt University.
- Millington, I. (2006). *Artificial Intellegence For Games*. San Francisco: Morgan Kaufman Publishers.
- Merrick, K. dan Maher, M.L.(2009). *Motivated Reinforcement Learning Curious Character For Multiuser Games*. Springer.

- Schwab, B. (2004). *AI Game Engine Programming*. Massachusetts : Charles River Media, Inc, Hing-ham.
- Smed, J. dan Hakoneni, H. (2006). *Algorithms and Networking For Computer Games*. West Sussex: John Wiley & Sons, Ltd.
- Reynolds, C.W. (1999). *Steering Behaviors For Autonomous Characters*. Sony Computer Entertainment, America.
- Rolling, A. dan Adams, E. (2003). *Game Design*. New Riders Publishing. United States of America.
- Wijaya, S. A. (2009). *Fuzzy State Machine Untuk Menghasilkan Variasi Respon NPC (Non-Playable Character) Pada Game*. (Tesis S2, Institut Teknologi Sepuluh Nopember Surabaya), dari <http://digilib.its.ac.id>

[halaman ini sengaja dikosongkan]