

# PENERAPAN METODE *FINITE STATE MACHINE* (FSM) PADA GAME *AGENT LEGENDA ANAK BORNEO*

Ekawati Yulsilviana<sup>1)</sup> dan Hanifah Ekawati<sup>2)</sup>

<sup>1,2</sup>Manajemen Informatika, STMIK Widya Cipta Dharma Samarinda  
<sup>1,2</sup>Jl. M. Yamin No.25 Samarinda – Kalimantan Timur 75123  
 E-mail : ekawicida@gmail.com<sup>1)</sup>, hanifah@wicida.ac.id<sup>2)</sup>

## ABSTRAK

*Finite State Machine* merupakan salah satu metodologi perancangan sistem kontrol yang menggambarkan tingkah laku atau prinsip sistem dengan menggunakan tiga hal yaitu *state* (keadaan), *event* (kejadian) dan *action* (aksi). Pada penelitian ini dibuat *game* yang berjudul “Legenda Anak Borneo” menceritakan petualangan pahlawan Legenda Kalimantan. Sebuah permainan bergenre *Tactical Role Playing Game* (*Tactical RPG*). Menggunakan metode FSM (*Finite State Machine*) dikembangkan pula agen cerdas di dalamnya, dengan tujuan untuk membuat musuh yang dapat menjadi lawan dari pemain. Didalam RPG, agen permainan yang berperan dapat musuh yang dapat berpikir sendiri ini biasa disebut dengan *non-player character* (NPC). Metode pengembangan sistem yang digunakan yaitu Metode Pengembangan Multimedia. Dengan menerapkan tahapan metode tersebut, maka dihasilkan NPC yang dapat memberi respon atau memiliki tingkah laku sesuai dengan keadaan yang terjadi pada pemain atau NPC lainnya. Hadirnya NPC dengan logika FSM membuat permainan “Legenda Anak Borneo” menjadi menantang.

**Kata Kunci:** *Finite State Machine, NPC, Game Agent, Tactical, Role Playing Game*

### 1. PENDAHULUAN

Seiring dengan kemajuan teknologi, *game* modern atau *game* yang disajikan pada suatu piranti atau perangkat teknologi dan dimainkan secara *virtual* juga semakin berkembang (Afriany dkk,2018). Perangkat *mobile* atau *smartphone* yang makin maju sekarang dapat menjalankan *game* dari yang sederhana sampai *game* yang memiliki fitur dan Tampilan yang tidak kalah dengan *game* yang ada di komputer (Wijayanti dkk, 2018). *Game* pada umumnya menggunakan *artificial intelligence* pada sistemnya agar membuat *game* tersebut lebih menarik untuk dimainkan (Andrea dkk, 2014). *Artificial intelligence* diterapkan pada agen *game* tersebut yang berupa *Non Player Character* NPC dan musuh atau lawan yang harus dihadapi didalam *game* (Karamian, 2016)

Salah satu penerapan kecerdasan buatan di *game* untuk pengambilan keputusan yang cerdas adalah *Finite State Machine* (FSM) (Jones, 2015). Metode *finite state machine* sendiri dipilih karena merupakan metode yang cukup sederhana namun sangat baik dalam memodelkan perilaku agen. Metode *finite state machine* menggunakan *state* (kondisi), *event* (kejadian) dan *action* (aksi). yang saling berkaitan. Penerapan metode *finite state machine* pada agen digunakan agar agen dapat terlihat seperti mengambil keputusan saat *state* yang terdapat pada agen tersebut terpenuhi (Andrea & Palupi, 2018). Didalam RPG, agen permainan yang berperan dapat sebagai musuh ini biasa disebut dengan NPC.

Berdasarkan uraian diatas maka tim peneliti tertarik untuk mengembangkan *mobile game* khusus pada *smartphone* berbasis *android* yang berjudul “Legenda

Anak Borneo”. Sebuah permainan yang menceritakan petualangan pahlawan Legenda Kalimantan. *Game* tersebut akan memakai metode *finite state machine* dan akan menggunakan genre *Tactical RPG*. Genre *Tactical RPG* sendiri diambil karena sangat cocok untuk menguji keefektifan FSM di dalam logika berpikir NPC.

### 2. RUANG LINGKUP

Permasalahan difokuskan pada:

1. Penerapan metode FSM pada *game* Agen “Legenda Anak Borneo”.
2. Penerapan FSM hanya pada NPC *enemy* dari *game* “Legenda Anak Borneo”.
3. Dengan diterapkannya NPC dapat mengambil keputusan terhadap kondisi yang terjadi di dalam *game*.

### 3. BAHAN DAN METODE

Adapun bahan dan metode yang digunakan dalam membangun *game* ini yaitu:

#### 3.1 *Tactical RPG*

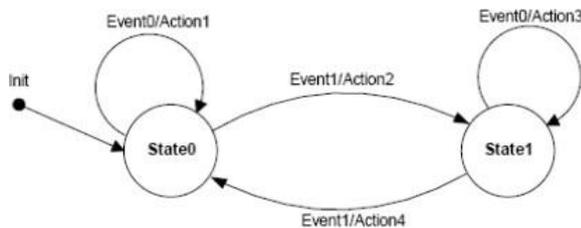
Menurut Jasson, 2009, *Role Playing Game* merupakan permainan dimana pemain dapat melakukan kendali terhadap satu atau lebih karakter yang dirancang sendiri oleh pemain dan membantu pemain untuk menyelesaikan misi (*quest*) yang diberikan oleh komputer. Keberhasilan dalam menyelesaikan *game* bergenre *role playing game* bergantung pada diselesaikannya semua *quest* yang ada. *Tactical RPG* atau sering dikenal dengan *strategy* RPG, percampuran antara elemen *game* *strategy* pada dengan proses

perkembangan karakter seperti RPG. Seperti RPG sederhana, *game* ini mengontrol beberapa karakter, baik personal atau grup/party, untuk melawan musuh namun perlu diperhatikan secara isometric pergerakan karakter/grup dalam bertempur baik dari segi arah (isometric grid) dan dari segi pergerakan strategical *game* (manual *game*). 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.

### 3.2 Finite State Machine

Millington dan Funge, 2009, *Finite State Machines* (FSM) adalah sebuah metodologi perancangan sistem kontrol yang menggambarkan tingkah laku atau prinsip kerja sistem dengan menggunakan *state* (Keadaan), *event* (kejadian) dan *action* (aksi).

Dalam *state machine* sistem menempati satu *state* (keadaan). Sistem akan beralih atau bertransisi menuju ke *state* lain jika mendapatkan masukan *event* tertentu. Sistem akan tetap melakukan aksi yang sama pada suatu *state* sampai sistem menerima *event* tertentu baik yang berasal dari perangkat luar atau komponen dari sistem itu sendiri. Setiap *state* terhubung oleh transisi dan setiap transisinya mengarah ke satu *state* lainnya. Transisi keadaan ini umumnya juga disertai oleh aksi yang dilakukan oleh sistem ketika menanggapi masukan yang terjadi. Aksi yang dilakukan tersebut dapat berupa aksi sederhana yang melibatkan rangkaian proses yang relatif rumit.



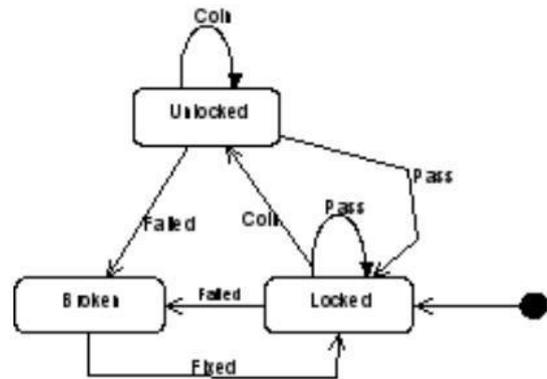
Gambar 1. Diagram *state* sederhana

Diagram pada gambar 1, memperlihatkan FSM dengan dua buah *state* dan dua buah *input* serta empat buah aksi *output* yang berbeda : seperti terlihat pada gambar, ketika sistem mulai dihidupkan, sistem akan bertransisi menuju *state0*, pada keadaan ini sistem akan menghasilkan *Action1* jika terjadi masukan *Event0*, sedangkan jika terjadi *Event1* maka *Action2* akan dieksekusi kemudian sistem selanjutnya bertransisi ke keadaan *State1* dan seterusnya.

Ada beberapa teknik pemodelan abstrak yang bisa digunakan untuk membantu definisi atau pemahaman dan desain dari FSM.

1. Diagram Transisi *State*
2. Diagram Pengambilan Keputusan *State*-Aksi.
3. Diagram Grafik *State*
4. Analisa Hirarki Perintah

Menurut Asmiatun & Putri, 2017, Pada dasarnya implementasi FSM bisa dibagi menjadi tiga (3) cara, dimana masing-masing cara memiliki kelebihan dan kekurangan masing-masing.



Gambar 2. FSM mesin minuman dalam kaleng menggunakan koin

Sebagai contoh sistem yang digunakan untuk melihat perbedaan antara cara satu dengan yang lain adalah sistem mesin minuman dalam kaleng menggunakan koin. Desain FSM sistem tersebut tampak pada gambar 2.

#### 1. Cara tradisional

Menurut Asmiatun & Putri, 2017, Kelebihan dari cara tradisional adalah mudah untuk diimplementasikan. Cara ini adalah cara klasik yang paling mudah untuk menerapkan FSM pada perangkat lunak sekaligus merupakan cara klasik yang biasanya masih banyak digunakan pada mikro komputer dengan sumber daya pemroses yang terbatas (*limited hardware resources*). Contoh cara tradisional dapat dilihat pada script dibawah ini.

```

switch (state) {
  case Broken:
    //kode state Broken disini;
    break;
  case Unlocked:
    // kode state Unlocked disini;
    break;
  default:
    // kode Lock disini break;
}

```

#### 2. Lookup table

Menurut Wibowo, 2018, Cara lain mengimplementasikan FSM dalam perangkat lunak adalah dengan menggunakan *lookup table*. Tabel ini berisi semua transisi *state* yang mungkin terjadi pada sistem. Tabel ini direpresentasikan sebagai matriks pada kode program dimana tiap baris merepresentasikan *event* atau transisi *state* dan kolomnya merepresentasikan *state* sedangkan elemennya merepresentasikan *next state*.

Salah satu contoh *lookup* tabel yang digunakan adalah seperti pada tabel 1

**Tabel 1. Contoh *Lookup table* FSM**

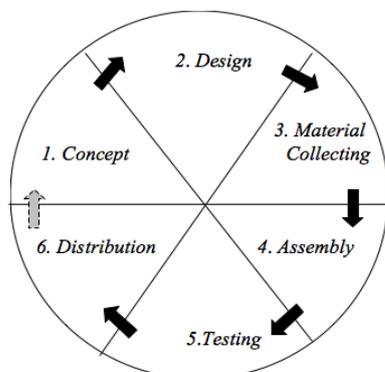
	Unlock	Lock	Broken
Coin		Unlock	
Pass	Lock		
Failed	Broken	Broken	
Fixed			Lock

### 3. *Object Oriented*

Menurut Wibowo, 2018, Kelebihan penggunaan OOP pada FSM adalah fleksibilitasnya yang tinggi dan pemeliharaannya yang mudah baik pada sistem yang sederhana, menengah, maupun sistem yang kompleks (Yacoub, 1998: 1998). Selain itu juga mendapatkan manfaat dari salah satu kelebihan OOP yaitu penggunaan kembali kode yang telah diketik (*code reusability*) sehingga pengetikan kode menjadi lebih sedikit.

### 3.3 Metode Pengembangan Multimedia

Metodologi pengembangan multimedia terdiri dari enam tahap, yaitu *concept* (pengonsepan), *design* (pendesainan), *material collecting* (pengumpulan materi), *assembly* (pembuatan), *testing* (pengujian), dan *distribution* (pendistribusian). Keenam tahap ini tidak dapat bertukar posisi. Meskipun begitu, tahap *concept* memang harus menjadi hal yang pertama kali dikerjakan (Agarwal, 2010)



**Gambar 3. Tahapan Pengembangan Multimedia**

Tahapan Pengembangan Multimedia pada gambar 3 meliputi :

#### 1. *Concept*

Tahap *concept* (pengonsepan) adalah tahap untuk menentukan tujuan dan siapa pengguna program (identifikasi *audiens*). Tujuan dan pengguna akhir program berpengaruh pada nuansa multimedia sebagai pencerminan dari identitas organisasi yang menginginkan informasi sampai pada pengguna akhir. Karakteristik pengguna termasuk kemampuan pengguna juga perlu dipertimbangkan karena dapat memengaruhi pembuatan desain.

Selain itu, tahap ini juga akan menentukan jenis aplikasi (presentasi, interaktif, dan lain-lain). Dasar aturan untuk perancangan juga ditentukan pada tahap ini, misalnya ukuran aplikasi, target, dan lain-lain. *Output* dari tahap ini biasanya berupa dokumen yang bersifat naratif untuk mengungkapkan tujuan proyek yang ingin dicapai.

#### 2. *Design*

*Design* (perancangan) adalah tahap pembuatan spesifikasi mengenai arsitektur program, gaya, tampilan, dan kebutuhan material/bahan untuk program. Spesifikasi dibuat serinci mungkin sehingga pada tahap berikutnya, yaitu *material collecting* dan *assembly*, pengambilan keputusan baru tidak diperlukan lagi, cukup menggunakan keputusan yang sudah ditentukan pada tahap ini. Meskipun demikian, pada praktiknya, pengerjaan proyek pada tahap awal masih akan sering mengalami penambahan bahan atau perubahan bagian aplikasi, atau perubahan-perubahan lain.

Tahap ini biasanya menggunakan *storyboard* untuk menggambarkan deskripsi tiap *scene*, dengan mencantumkan semua objek multimedia dan tautan ke *scene* lain dan bagian alir (*flowchart*) untuk menggambarkan aliran dari satu *scene* ke *scene* lain. Pembuatan *storyboard* dapat menggunakan cara pembuatan *storyboard* film/animasi.

#### 3. *Material Collecting*

*Material Collecting* adalah tahap pengumpulan bahan yang sesuai dengan kebutuhan yang dikerjakan. Bahan-bahan tersebut, antara lain gambar *clip art*, foto, animasi, video, audio, dan lain-lain yang dapat diperoleh secara gratis atau dengan pemesanan kepada pihak lain sesuai dengan rancangannya. Tahap ini dapat dikerjakan secara paralel dengan tahap *assembly*. Namun, pada beberapa kasus, tahap *material collecting* dan tahap *assembly* akan dikerjakan secara linear dan tidak paralel.

#### 4. *Assembly*

Tahap *assembly* adalah tahap pembuatan semua obyek atau bahan multimedia. Pembuatan aplikasi didasarkan pada tahap *design*, seperti *storyboard*, bagan alir, dan/atau struktur navigasi.

#### 5. *Testing*

Tahap *Testing* (pengujian) dilakukan setelah menyelesaikan tahap pembuatan (*assembly*) dengan menjalankan aplikasi/program dan melihatnya apakah ada kesalahan atau tidak.

#### 6. *Distribution*

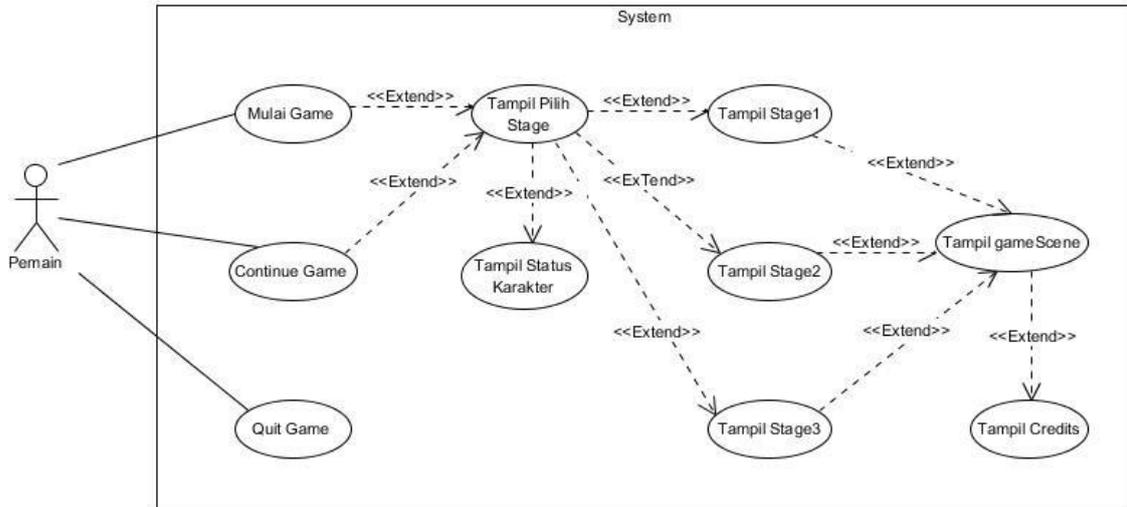
Pada tahap ini, aplikasi akan disimpan dalam suatu media penyimpanan. Jika media penyimpanan tidak cukup untuk menampung aplikasinya, kompresi terhadap aplikasi tersebut akan dilakukan. Tahap ini juga dapat disebut tahap evaluasi untuk pengembangan produk yang sudah jadi supaya menjadi lebih baik. Hasil evaluasi ini dapat digunakan sebagai masukan untuk tahap *concept* pada produk selanjutnya.

**4. KONSEP APLIKASI**

Konsep *game* “Legenda Anak Borneo” ini menggunakan *Unified Modeling Language* (UML). *Game* berbasis android ini menggunakan alat bantu UML sebagai salah satu cara untuk mempermudah dalam pembuatan aplikasi ini.

1. Diagram *Use Case Game* “Legenda Anak Borneo”

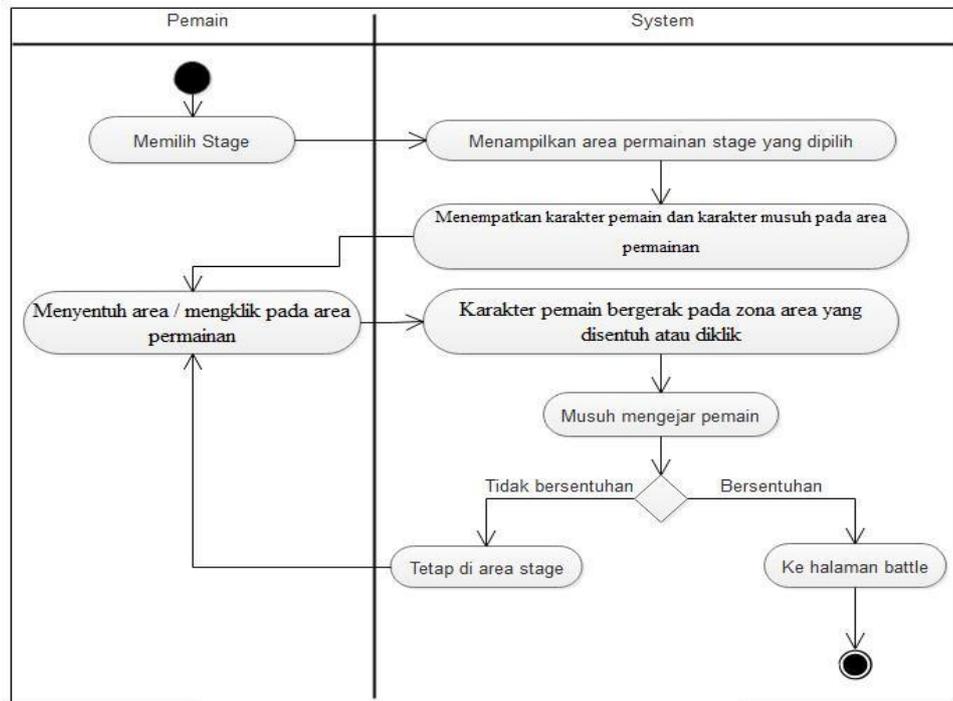
*Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem yang akan dibuat (Nugroho, 2101). Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Ada dua hal utama pada *use case* yaitu pendefinisian apa yang disebut aktor dan *use case*. Pada diagram *use case* “Legenda Anak Borneo” terdapat 10 *use case* yang dapat dilihat pada gambar 4.



**Gambar 4. Diagram Use Case game “Legenda Anak Borneo”**

2. *Activity Diagram* “Legenda Anak Borneo”

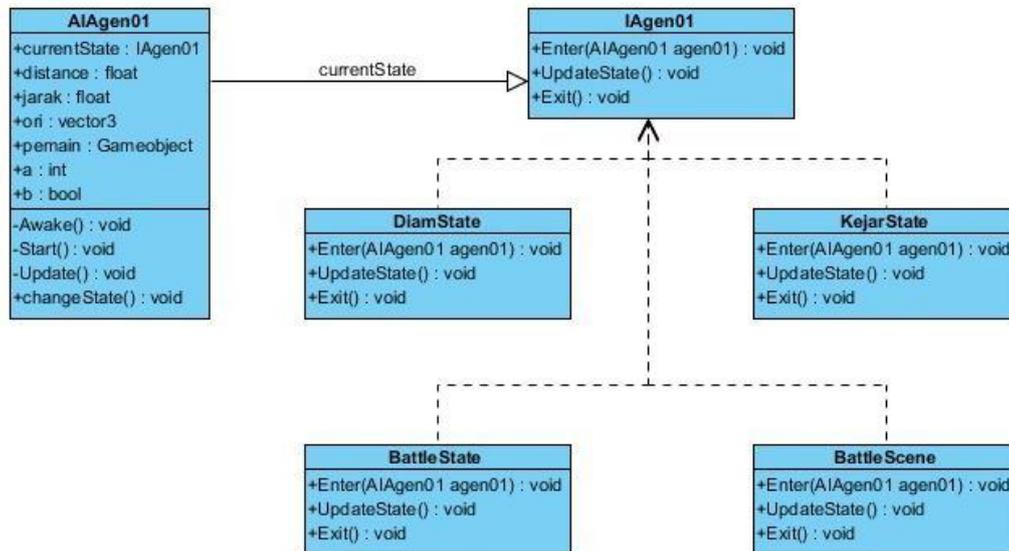
Diagram pada gambar 5, menggambarkan tentang aktifitas yang terjadi pada sistem. Dari pertama sampai akhir, diagram ini menunjukkan langkah – langkah dalam proses kerja sistem dan jalur-jalur aktifitas dari *level* atas secara umum pada *game* “Legenda Anak Borneo”. Sebuah aktifitas dapat direalisasikan oleh satu *use case* atau lebih.



**Gambar 5. Activity Diagram Tampil Stage**

3. *Class Diagram Game* “Legenda Anak Borneo”

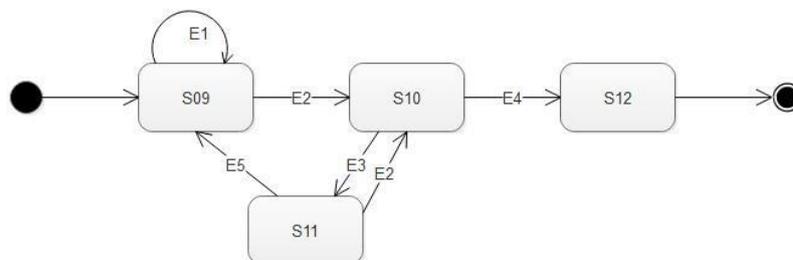
Perancangan *Artificial Intelligence* pada penelitian ini diterapkan pada pengaturan perilaku NPC dengan metode *finite state machine*. Pada artikel ini dibahas salah satu NPC yang dibangun dengan inisial agen 01. *Class diagram* agen 01 seperti pada gambar 6.



Gambar 6. *Class Diagram* Agen 01

4. *Finite State Machine* Agen “Legenda Anak Borneo” Agen 01 hanya terdapat pada area *stage*.

Agan 01 tidak memiliki atribut *health*, *attack*, *defense* seperti Agan 02. Tujuan Agan 01 adalah mengejar pemain yang berada dalam jarak tertentu dari Agan 01. Pada FSM Agan 01 State digambarkan berbentuk kotak dan *event* yang digambarkan berbentuk panah. Proses perpindahan *state-state* pada Agan 01 dapat dilihat pada gambar 6.



Gambar 6. *Diagram Finite State Machine* Agen 01

Keterangan gambar 6:

- S09 – NPC diam
- S10 – NPC mengejar
- S11 – NPC kembali ke zona awal
- S12 – NPC menangkap pemain

- E1/E5 – tidak ada pemain mendekat
- E2 – pemain memasuki zona agen
- E3 – pemain menjauhi zona agen
- E4 – pemain diam

5. *Diagram Transisi NPC Game* “Legenda Anak Borneo”

Pada tabel 2 dijelaskan transisi tiap *state* agen 01. *State* awal berada pada kolom sebelah kiri dengan menggunakan warna abu-abu dimulai dari S9 dan berurutan ke bawah sampai S12. *Event* berada pada kolom bagian atas yang berwarna abu-abu, dimulai dari E1 dan berakhir di E5. *State* akhir berada pada kolom tengah yang berwarna putih dan tidak beraturan karena *state* akhir ditentukan oleh *state* awal dan *event*.

Tabel 2. *Tabel Transisi Agen 01*

S	INPUT				
	E1	E2	E3	E4	E5
S09	S09	S10			
S10			S11	S12	
S11		S10			S09
S12					

## 5. PEMBAHASAN

Hasil desain dan *assembly* berdasarkan konsep adalah sebagai berikut :

### 1. Desain Objek Karakter 3D

Objek-objek yang terdapat di dalam *game* berbentuk 3 dimensi. Dibuat menggunakan *software* Blender. Karakter utama adalah karakter yang dikendalikan oleh pemain (Kopel & Hajas, 2018). Karakter ini berbentuk kesatria legenda Kalimantan. Terdapat 5 karakter dengan perbedaan terletak pada senjata yang digunakan pemain. Contoh karakter utama dapat dilihat pada gambar 7.



Gambar 7. Karakter Utama

### 2. Assembly Tampilan Antarmuka Game

Pada tahap ini pembuatan antarmuka yang terdapat pada *game* "Legenda Anak Borneo". Di tampilan awal *game* (Gambar 8) terdapat 3 buah tombol, yaitu tombol *start game*, tombol *continue game* dan tombol *quit game*. Tombol *start game* akan memulai baru permainan dan menampilkan halaman pemilihan *stage*. Tombol *continue game* akan menampilkan halaman pemilihan *stage* berdasarkan data yang telah disimpan oleh pemain. Tombol *quit game* akan menutup permainan..



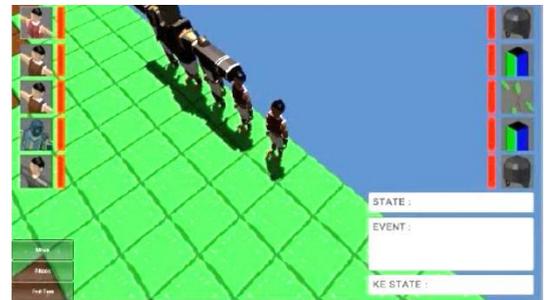
Gambar 8. Tampilan Awal Game

Tampilan Area *Stage 1* (Gambar 9) dimana pemain dapat menyentuh pada area tertentu dan karakter pemain akan bergerak ke arah tersebut. Terdapat 1 tombol di kanan atas yang dapat membawa pemain ke tampilan status karakter.



Gambar 9. Tampilan Area Stage 1

Ditampilkan *scene battle* (Gambar 10) terdapat tombol *move*, *attack*, *end turn*, *skill 1* dan *skill 2*. Dikiri terdapat status *Health Point* (HP) dan gambar karakter agen *enemy*. Disebelah kanan terdapat gambar dan status HP karakter pemain.



Gambar 10. Tampilan Scene Battle

### 3. Assembly Finite State Machine Agen

*Finite state machine* disini untuk melihat perubahan perilaku NPC *enemy* terhadap permainan yang sedang berjalan. Pembuatan *state - state* 3 agen dengan setiap perubahan *state* yang terjadi pada masing-masing agen. Agen 01 memiliki 4 *state* dan 5 *event*, yaitu :

#### 1) State Diam

Pada awal halaman *stage* dimulai, agen 01 akan diam pada posisinya atau koordinat awal seperti pada gambar 11. Dapat dilihat tanda lingkaran pada gambar 11 perwujudan NPC agen 01 dalam bentuk monster batu.



Gambar 11. Agen 01 State Diam

## 2) State Kejar

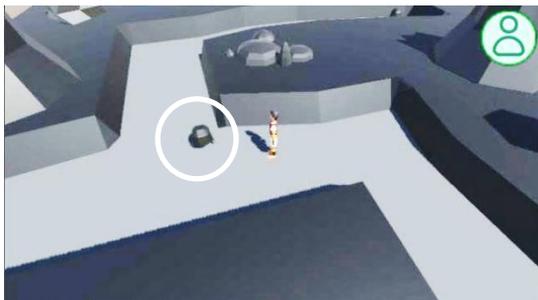
Ketika pemain berada dalam jarak zona agen, maka aksi agen 01 akan berganti ke *state* kejar seperti pada gambar 12.



**Gambar 12. Agen 01 State Kejar**

## 3) State Kembali

Ketika pemain berada diluar zona yang telah ditentukan maka agen 01 akan berganti ke *state* kembali, yaitu bergerak menuju posisi awal agen 01 seperti pada gambar 13.



**Gambar 13. Agen 01 State Kembali**

## 4) State Menangkap

Ketika pemain dan agen 01 bersentuhan seperti gambar 14, maka aksi akan berganti ke menangkap pemain. Permainan akan menuju ke *scene battle* seperti pada gambar 10 di halaman sebelumnya.



**Gambar 14. Agen 01 State Battle**

## 6. KESIMPULAN

Pengembangan *game* “Legenda Anak Borneo” melalui 6 tahap yaitu konsep, desain, pengumpulan

bahan, *assembly*, pengujian dan pendistribusian. Penerapan metode *Finite State Machine* dengan membuat *state*, *action* dan *event* pada tiap NPC yang dibuat. Fungsi *Finite State Machine* pada NPC untuk menentukan respon perilaku NPC terhadap perubahan kondisi yang terjadi pada pemain dan pada lingkungan *game*. Di dalam *game* ini pemain dapat melawan agen *enemy* dan menjelajah tiap stage-stage yang tersedia.

## 7. SARAN

Di masa yang akan datang metode *finite state machine* ini, dapat diterapkan pada berbagai tipe *genre game*, yang nantinya akan memberikan keunikan tersendiri didalamnya. *Game* menjadi lebih seru dan lebih menantang dengan bantuan metode *finite state machine*. Dengan memperbanyak *state-state* dan *event-event* diharapkan dapat membuat pergerakan atau perilaku agen lebih bervariasi dan lebih baik lagi. Memperbanyak tipe agen kedalam *game* juga akan memperbanyak variasi agen yang bisa digunakan dengan kelebihan dan kekurangan masing-masing agen.

## 8. DAFTAR PUSTAKA

- Afriany, J., Andrea, R., Yulsilviana, E., Wijayanti, S., & Nabile, D. 2018. Program Pengabdian Masyarakat: Alat Bantu Belajar Ilmu Batuan Berbasis Android pada Kelas Geologi Tambang Smk Negeri 1 Sendawar. *Sebatik*, 22(2), 202-210.
- Agarwal, B. B., Tayal, S. P., dan Gupta, M., 2010. *Software Engineering and Testing*, Burlington, USA: Jones & Bartlett Learning.
- Andrea, R., & Palupi, S. 2018. Membangun Edugame “Boni Kids–Borneo Animal Kids” Permainan Match-up dengan Teknik Pengacakan Shuffle dan Pengembangan Agen Cerdas dengan Model Finite State Machine (Fsm). *Sebatik*, 19(1), 6-10.
- Andrea, R., Akbar, R. I., & Fitroni, M. 2014. Developing battle of Etam earth game agent with finite state machine (FSM) and sugeno fuzzy. *ICCS Proceeding*, 1(1), 184-187.
- Asmiatun, S., & Putri, A. N. (2017). *Belajar Membuat Game 2D dan 3D Menggunakan Unity*. Deepublish.
- Jasson. 2009. *Role Playing Game (RPG) Maker*. Yogyakarta : Andi.
- Jones, M. T. 2015. *Artificial Intelligence: A Systems Approach: A Systems Approach*. Jones & Bartlett Learning.
- Karamian, V. 2016. *Building an RPG with Unity 5. x*. Packt Publishing Ltd.
- Kopel, M., & Hajas, T. 2018. Implementing AI for Non-player Characters in 3D Video Games. In *Asian Conference on Intelligent Information and Database Systems* (pp. 610-619). Springer, Cham.

- Millington, I., & Funge, J. 2009. Artificial intelligence for games. CRC Press.
- Nugroho, A. 2010. Rekayasa perangkat lunak berorientasi objek dengan metode USDP. Penerbit Andi.
- Wibowo, F. W. 2018. FPGA DAN VHDL: Teori, Antarmuka dan Aplikasi. Deepublish.
- Wijayanti, S., Nurhuda, A., & Andrea, R. 2018. Edugame “Etam-Tainment” Pembelajaran Bahasa Kutai dengan Shuffle Random dan Agen Cerdas.

Jurnal Nasional Teknik Elektro dan Teknologi Informasi (JNTETI), 7(3).

**UCAPAN TERIMA KASIH**

Publikasi ini dibiayai oleh:

Direktorat Riset dan Pengabdian Masyarakat  
Direktorat Jenderal Penguatan Riset dan Pengembangan  
Kementerian Riset, Teknologi, dan Pendidikan Tinggi  
sesuai dengan Kontrak Penelitian Tahun Anggaran 2019