

PENERAPAN METODE *FINITE STATE MACHINE* PADA GAME “THE RELATIONSHIP”

Miftah Fauzan Rahadian¹⁾, Addy Suyatno²⁾, Septya Maharani³⁾

^{1,2,3)}Program Studi Ilmu Komputer, FMIPA, Universitas Mulawarman

Jalan Barong Tongkok No. 4 Kampus Gunung Kelua Samarinda, Kalimantan Timur

Email : fauzan.rahadian@gmail.com¹⁾, addysuyatno@yahoo.com²⁾, septyamaharani@gmail.com³⁾

Abstrak

Finite State Machine (FSM) adalah sebuah metodologi perancangan sistem kontrol yang menggambarkan tingkah laku atau prinsip kerja sistem dengan menggunakan tiga hal berikut: *State* (Keadaan), *Event* (kejadian) dan *action* (aksi). Sebagai sebuah metodologi perancangan sistem kontrol, penerapan FSM telah banyak diterapkan pada perangkat lunak, khususnya pada *game*. Metode FSM dapat diterapkan untuk nilai-nilai batas perpindahan yang pasti dan sangat mudah diterapkan pada *game* “*The Relationship*”. *Game* ini merupakan sebuah *game* berbasis *desktop genre* petualangan dengan *sub-genre* simulasi dan menggunakan grafis 2 dimensi. Pengembangan *game* ini menggunakan *tools* Unity3D dan bahasa pemrograman *C#(Sharp)*. Hasil dari penelitian ini adalah terealisasinya suatu *game* dengan menerapkan metode *finite state machine* untuk menentukan respon karakter *non-player character* yang berbeda tergantung dari interaksi yang dilakukan oleh pemain.

Kata kunci : *Game, Finite State Machine, C#, Unity3D*

PENDAHULUAN

1.1 Latar Belakang

Industri *game* yang semakin maju menyuguhkan sebuah dunia maya yang baru yang terus menerus berkembang untuk mencoba mendekati kehidupan nyata. *Game* atau permainan dibuat untuk digunakan sebagai sarana menghibur diri saat penat maupun untuk menghilangkan rasa penat. *Game* memiliki dua pengertian, pertama sebuah aktifitas bermain yang murni mencari kesenangan tanpa mencari hasil menang atau kalah dan yang kedua dapat diartikan sebagai aktifitas bermain yang dilakukan dalam rangka mencari kesenangan dan kepuasan.

Salah satu unsur yang berperan penting dalam sebuah *game* adalah kecerdasan buatan atau *Artificial Intelligence (AI)* yang merupakan suatu program komputer yang bertindak dan berpikir seperti manusia dan juga bertindak dan berpikir secara rasional pada saat yang bersamaan. Dengan kecerdasan buatan, elemen-elemen dalam *game* dapat berperilaku seperti manusia. *Game playing* (permainan *game*) merupakan bidang AI yang sangat populer berupa permainan antara manusia melawan mesin yang memiliki intelektual untuk berpikir. Komputer dapat bereaksi dan menjawab tindakan-tindakan yang diberikan oleh lawan mainnya [6].

Dalam sebuah penelitian “Membangun *Game* Simulasi *Finite State Machine* untuk Pertanian dan Peternakan” menerangkan bahwa salah satu jenis *game* yang digemari adalah *game* dengan *genre* simulasi yang didukung dengan kecerdasan buatan yang menggunakan *Finite State Machine (FSM)* untuk mendukung interaksi dengan pemain. FSM terdiri dari serangkaian *state* yang menentukan keputusan. Setiap *state* dapat berpindah ke *state*

lainnya jika memenuhi kondisi yang telah ditentukan sebelumnya [5], selanjutnya untuk membuat *game* dapat menggunakan beberapa cara, salah satunya dengan menggunakan *game engine* yang menawarkan paket pengembangan visual dan elemen-elemen *software* yang dapat digunakan kembali untuk memungkinkan efisiensi pengembangan *game* karena sudah disediakan fitur-fitur seperti penyisipan gambar, tombol, bahkan grafik 3D [2].

Penerapan FSM pada *game* berguna untuk menentukan berbagai macam respon NPC berdasarkan interaksi yang dilakukan oleh pemain, hal ini disebabkan karena FSM dapat digunakan untuk mendesain dan menentukan respon perilaku yang dilakukan terhadap perubahan kondisi. Dari hasil penelitian di atas, maka penulis tertarik untuk membuat penelitian dengan mengembangkan sebuah *game* dengan menerapkan metode *finite state machine (FSM)* pada *non-playable character (NPC)*. Adapun penelitian tersebut dituangkan dalam penyusunan skripsi dengan judul “Penerapan Metode *Finite State Machine* Pada *Game The Relationship*”.

1.2 Batasan Masalah

Agar pembahasan dalam skripsi ini tidak menyimpang dari topik permasalahan, maka penulis melakukan pembatasan-pembatasan sebagai berikut ini:

1. Membuat sebuah *game* “*The Relationship*” yang dimainkan secara *singleplayer*.
2. NPC pada *game* “*The Relationship*” memberikan interaksi berupa obrolan kepada player.

3. *Game* “*The Relationship*” hanya mempunyai lima lokasi yaitu di kamar pria, kelas, perpustakaan, lapangan dan aula sekolah.
4. Hasil akhir dari permainan ini tergantung dengan interaksi yang dilakukan pemain selama permainan berlangsung.
5. *Game* yang dibangun dengan menggunakan *Unity* dengan bahasa C#.

1.3 Tujuan Penelitian

Mengimplementasikan metode *finite state machine* dalam membangun *game* “*The Relationship*” menggunakan *Unity* yang diharapkan dapat memberikan *gameplay* yang menarik beserta hasil yang *opportunity* membuat pemain harus mengambil keputusan disetiap interaksi.

1.4 Manfaat Penelitian

Adapun manfaat secara umum dari *game* “*The Relationship*” ini adalah:

1. Mampu menerapkan metode *finite state machine* dalam penentuan respon NPC.
2. Mengasah kemampuan dan kreatifitas pemain dalam mengambil setiap keputusan.
3. Memberikan rasa keingintahuan kepada para pemain ketika bermain

TINJAUAN PUSTAKA

2.1 Game

Game adalah sebetuk karya seni di mana peserta, yang disebut pemain, membuat keputusan untuk mengelola sumber daya yang dimilikinya melalui benda di dalam *game* demi mencapai tujuan [1].

Sebuah *game* yang baik akan memberikan pengalaman bermain yang optimal, sebuah proses pengembangan *game* yang baik juga akan memberikan pengalaman yang istimewa serta proses pembelajaran yang luar biasa bagi mereka yang terlibat di dalamnya. Proses pengembangan sebuah *game* secara baik secara garis besar terbagi menjadi beberapa tahapan berikut [3] :

1. Merancang *Game*

Hal yang dilakukan pertama kali adalah menentukan jenis *game* yang akan dikembangkan. Pada tahap ini ide dasar, tema, teknologi, media (*platform*), serta berbagai batasan lain dirumuskan. Tahapan riset ini menjadi sebuah tahapan krusial, berbagai elemen dasar dari sebuah *game* disusun di sini.

2. Perumusan *Gameplay*

Gameplay adalah pola, aturan, atau mekanisme yang mengatur bagaimana proses interaksi pemain dengan *game* yang diciptakan. *Gameplay* juga mengatur bagaimana seorang pemain bisa memenuhi objektif dari *game* dan mendapatkan pengalaman bermain yang menyenangkan.

3. Penyusunan *Asset* dan *Level Design*

Tahapan ini berfokus pada penyusunan konsep dari semua karakter serta *asset* (termasuk suara atau musik) yang diperlukan. Pada saat yang sama tim juga mulai

melakukan *level design* atau pengelompokkan tingkat kesulitan serta berbagai *asset* yang tepat pada tiap *level* (jika ada lebih dari 1 *level*) agar *game* tersebut bisa menghadirkan pengalaman bermain yang optimal.

4. *Test Play (Prototyping)*

Pada tahapan ini sebuah *prototype/dummy* dihadirkan untuk menguji *gameplay* serta berbagai konsep yang telah tersusun, baik dalam tiap *level* maupun secara keseluruhan, serta melakukan berbagai perbaikan yang diperlukan. Tahapan ini juga berfungsi untuk memberikan gambaran lengkap bagi seluruh tim sehingga bisa memudahkan proses pengembangan selanjutnya.

5. Development

Pada tahap ini seluruh konsep (karakter dan *asset*) yang sebelumnya telah tersusun mulai dikembangkan secara penuh.

6. Percobaan

Tahap ini dilakukan untuk mengetahui apakah semua komponen utama dari *game* telah mampu memberikan pemain pengalaman seperti yang diharapkan sekaligus juga untuk mendeteksi adanya masalah teknis yang belum terdeteksi pada tahapan sebelumnya.

7. Rilis

Pada tahap ini *game* sudah siap untuk dirilis dan diperkenalkan pada target pemainnya. Ketika sebuah *game* telah dirilis untuk publik bukan berarti proses pengembangan telah selesai, karena umumnya *game* akan terus dioptimalkan/*di-update*. Hal ini untuk memastikan bahwa *game* yang dihadirkan benar-benar mampu memberikan pengalaman bermain yang maksimal.

2.2 Kecerdasan Buatan

Definisi Kecerdasan Buatan (AI) merupakan cabang dari ilmu komputer yang dalam merepresentasi pengetahuan lebih banyak menggunakan bentuk simbol-simbol daripada bilangan, dan memproses informasi berdasarkan metode heuristic atau dengan berdasarkan sejumlah aturan.

Ada tiga tujuan kecerdasan buatan, yaitu, membuat komputer lebih cerdas, mengerti tentang kecerdasan, dan membuat mesin yang lebih. Kecerdasan adalah kemampuan untuk belajar dan mengerti dari pengalaman, memahami pesan yang kontradiktif dan ambigu, menanggapi dengan cepat dan baik atas situasi yang baru, menggunakan penalaran dalam memecahkan masalah serta menyelesaikannya dengan efektif. Adapun lingkup utama dalam kecerdasan buatan adalah [6] :

1. Sistem Pakar

Komputer digunakan sebagai sarana untuk menyimpan pengetahuan para pakar. Dengan demikian komputer akan memiliki keahlian untuk menyelesaikan permasalahan dengan meniru keahlian yang dimiliki oleh pakar.

2. Pengolahan Bahasa Alami (*Natural Language Processing*)

Komputer dapat mengolah bahasa alami yang diharapkan pengguna dapat berkomunikasi dengan komputer dengan menggunakan bahasa sehari-hari.

3. Pengenalan Ucapan (*Speech Recognition*)

Melalui pengenalan ucapan diharapkan pengguna dapat berkomunikasi dengan komputer menggunakan suara.

4. Robotika dan Sistem Sensor

Robot adalah sebuah alat mekanik yang dapat melakukan tugas fisik, baik menggunakan pengawasan dan kontrol manusia, ataupun menggunakan program. Robot biasanya digunakan untuk tugas yang berat, berbahaya, pekerjaan yang berulang dan kotor. Robot juga dilengkapi dengan sensor untuk pendeteksi terhadap sesuatu hal, misalnya seperti sensor panas.

5. *Computer Vision*

Computer Vision mencoba untuk dapat menginterpretasikan gambar atau objek-objek tampak melalui komputer.

6. *Intelligent Computer-aided Instruction*

Komputer dapat digunakan sebagai tutor yang dapat melatih dan mengajar.

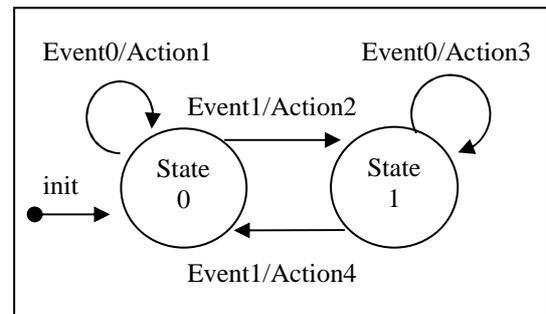
7. *Game Playing*

Game playing (permainan game) merupakan bidang kecerdasan buatan yang sangat populer berupa permainan antara manusia melawan mesin yang memiliki intelektual untuk berpikir. Komputer dapat bereaksi dan menjawab tindakan-tindakan yang diberikan oleh lawan mainnya.

2.3 Finite State Machine

Finite state machine (FSM) adalah sebuah metodologi perancangan sistem kontrol yang menggambarkan tingkah laku atau prinsip kerja sistem dengan menggunakan tiga hal berikut: *State* (Keadaan), *event* (kejadian) dan *action* (aksi). Pada satu saat dalam periode waktu yang cukup signifikan, sistem akan berada pada salah satu *state* yang aktif. Sistem dapat beralih atau bertransisi menuju *state* lain jika mendapatkan masukan atau *event* tertentu, baik yang berasal dari perangkat luar atau komponen dalam sistemnya itu sendiri. Transisi keadaan ini umumnya juga disertai oleh aksi yang dilakukan oleh sistem ketika menanggapi masukan yang terjadi. Aksi yang dilakukan tersebut dapat berupa aksi yang sederhana atau melibatkan rangkaian proses yang relatif kompleks (Setiawan, 2006).

Gambar 1 memperlihatkan FSM dengan dua buah *state* dan dua buah *input* serta empat buah *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.



Gambar 1. Diagram *state* sederhana (Sumber: Setiawan, 2006)

Salah satu alternatif implementasi FSM adalah menggunakan pemrograman berorientasi objek (*Object Oriented Programming*) atau yang sering disingkat sebagai OOP (Wijaya, 2010). Kelebihan penggunaan OOP pada FSM adalah fleksibilitasnya yang tinggi dan pemeliharaannya yang mudah baik pada sistem yang sederhana, menengah, maupun sistem yang kompleks. 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 (Wijaya, 2010).

HASIL PENELITIAN DAN PEMBAHASAN

3.1 Deskripsi Sistem

Game "The Relationship" adalah permainan petualangan dengan *sub genre* simulasi yang dimainkan pada komputer (PC) atau laptop. *Game "The Relationship"* menggunakan sudut pandang ketiga, yang artinya pemain dapat melihat karakter yang dimainkan maupun karakter NPC. Pemain harus menyelesaikan objektif-objektif yang ada di dalam *game* untuk mendekati karakter perempuan yang telah ditentukan.

3.1.1 Jalan Cerita

Game ini mempunyai cerita utama tentang karakter laki-laki bernama Mizan yang telah naik ke kelas 3 SMA di salah satu sekolah negeri. Pada kelas 3 Mizan berkeinginan untuk membuat akhir masa sekolahnya menjadi salah satu bagian yang tak terlupakan dalam hidupnya. Oleh karena itu Mizan berkeinginan untuk dikenal guru maupun teman satu sekolahnya. Selain itu di kelas 3 ini, Mizan juga berambisi untuk mendapatkan pasangan. Terdapat tiga karakter perempuan yang dapat didekati oleh Mizan, yaitu Tara yang gemar belajar, Fina yang suka berolahraga, dan Mira yang populer di sekolah.

3.1.2 Gameplay

Perumusan *gameplay* pada *game "The Relationship"* berupa aturan, yaitu :

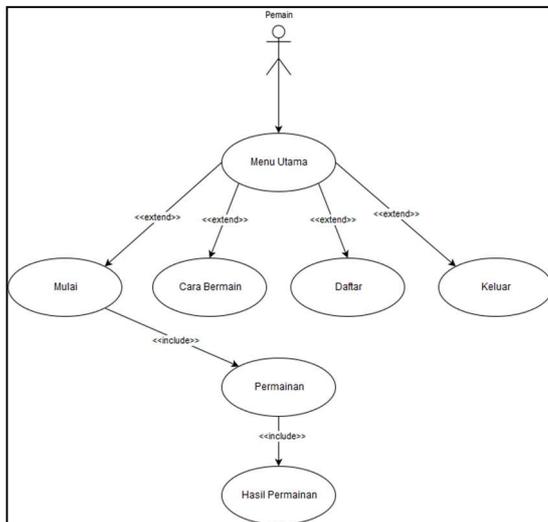
1. Permainan bersifat *single player* yaitu hanya dimainkan oleh satu pemain.
2. Permainan hanya dapat dimainkan dengan *touchpad* atau *mouse*.

dapat bertransisi menuju *state* pertanyaan 5 atau pertanyaan 6 jika *variable* kemungkinan terpenuhi.

Ketika permainan telah berlangsung selama 30 hari permainan maka NPC akan bertransisi menuju *state* perasaan akhir. Pada *state* ini pemain diharuskan untuk menyatakan perasaan kepada NPC untuk menentukan hasil akhir permainan. Hasil akhir ditentukan oleh akumulasi *variable* NPC yang dilakukan oleh pemain selama permainan berlangsung.

Setiap *state* teman, *state* teman baik, dan *state* teman dekat akan membuat NPC memberikan obrolan kepada karakter utama yang berbed-beda. Begitu juga dengan *state* pertanyaan dan *variable* kemungkinan yang akan berbeda disetiap karakter NPC.

3.1.6 Use Case Diagram Sistem “The Relationship”



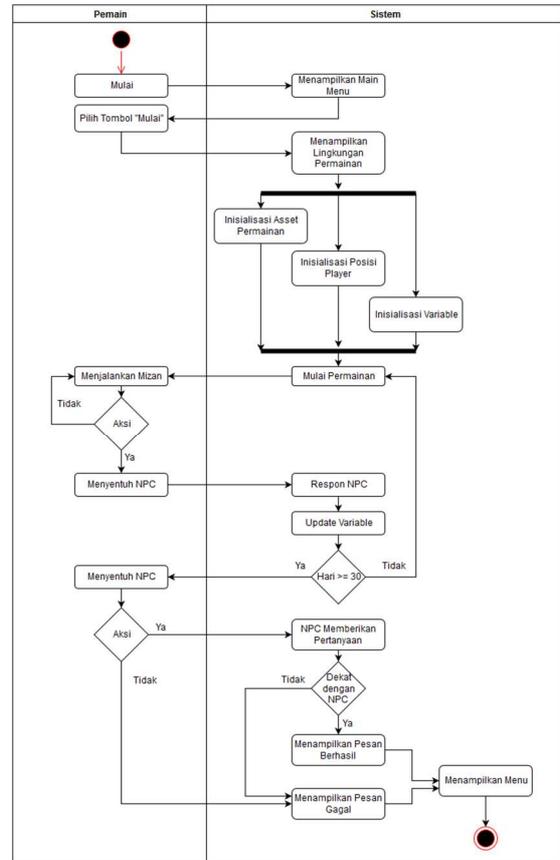
Gambar 4. Use Case Diagram Permainan “The Relationship”

Gambar 4. menjelaskan bahwa pemain dapat melakukan beberapa aksi sebelum memulai permainan. Dalam menu utama terdapat beberapa pilihan untuk pemain yaitu, Mulai, Cara Bermain, Daftar, dan Keluar. Jika pemain memilih Mulai maka *scene* akan langsung berpindah menuju *scene* permainan yang di dalamnya termasuk *scene* hasil permainan. Tombol Cara Bermain menampilkan instruksi bagaimana cara untuk bermain yang meliputi menjalankan karakter utama, cara untuk berinteraksi dengan NPC. Tombol Daftar menampilkan nama-nama yang berjasa dalam pembuatan *game* “The Relationship”. Tombol Keluar untuk keluar dari permainan atau aplikasi.

4.2.6 Activity Diagram untuk Mulai

Gambar 5 menjelaskan alur aktifitas yang terjadi saat pemain memilih tombol Mulai pada menu utama hingga pemain menyelesaikan permainan. Alur dimulai dari pemain memulai permainan, sistem akan menampilkan *menu* utama dimana pemain memilih Mulai untuk memulai

permainan, kemudian sistem akan menampilkan lingkungan permainan yang di dalamnya termasuk inialisasi *asset*, posisi *player*, dan *variable* awal game.



Gambar 5. Activity Diagram untuk “Play”

Selanjutnya pemain dapat menjalankan karakter utama untuk melakukan interaksi kepada NPC. Ketika karakter utama dan NPC bersentuhan maka sistem akan memberikan respon yang diikuti dengan update *variable* tingkat kedekatan NPC kemudian sistem akan mengecek *variable* hari. Jika *game* belum melewati 30 hari permianan maka pemain masih dapat melanjutkan permainan, jika tidak maka ketika pemain menyentuh NPC sistem akan memberikan pertanyaan untuk pemain. Selanjutnya sistem juga akan mengecek tingkat kedekatan NPC dan karakter utama, jika iya maka sistem akan menampilkan pesan berhasil menyelesaikan objektif *game*, begitu juga sebaliknya. Setelah itu sistem akan kembali menampilkan *menu*.

3.3 Implementasi dan Pengujian Sistem

Implementasi merupakan tahap realisasi setelah perancangan sistem yang menghasilkan *game* berjudul “The Relationship”.

3.3.1 Perhitungan Tingkat Kedekatan Karakter NPC

Perhitungan tingkat kedekatan dilakukan pada saat pemain melakukan interaksi kepada karakter NPC selama permainan berlangsung. Sedangkan

poin yang mempengaruhi karakter NPC dapat berkurang atau bertambah menyesuaikan dengan interaksi yang dilakukan oleh pemain. Terdapat 4 tingkat kedekatan antara karakter utama dengan karakter NPC.

Tabel 1 Tingkat Kedekatan NPC

No	Tingkat	Nilai Variable
1	Perkenalan	0
2	Teman	1 s/d 20
3	Teman Baik	21 s/d 50
4	Teman Dekat	> 50

1. Tingkat Kedekatan Tara

Struktur kondisi untuk 4 tingkat kedekatan karakter utama dengan karakter Tara, semakin tinggi nilai *variable* Tara semakin dekat hubungannya. Tingkat pertama adalah perkenalan saat *variable* Tara masih bernilai kosong. Tingkat kedua adalah teman saat *variable* Tara bernilai dari 1 sampai dengan 20. Tingkat ketiga adalah teman baik saat *variable* Tara bernilai dari 21 sampai dengan 50. Sedangkan tingkat terakhir adalah teman dekat saat *variable* Tara bernilai lebih dari 50. Struktur kondisi tingkat kedekatan Tara dapat dilihat pada potongan *script* berikut ini :

```
if(GameState.Instance.getTara() == 0) {
    //State Perkenalan
} else if(GameState.Instance.getTara() <= 20) {
    //State Teman
} else if(GameState.Instance.getTara() <= 50) {
    //State Teman Baik
} else { //State Teman Dekat }
```

2. Tingkat Kedekatan Mira

Struktur kondisi untuk 4 tingkat kedekatan karakter utama dengan karakter Mira, semakin tinggi nilai *variable* Mira semakin dekat hubungannya. Tingkat pertama adalah perkenalan saat *variable* Mira masih bernilai kosong. Tingkat kedua adalah teman saat *variable* Mira bernilai dari 1 sampai dengan 20. Tingkat ketiga adalah teman baik saat *variable* Mira bernilai dari 21 sampai dengan 50. Sedangkan tingkat terakhir adalah teman dekat saat *variable* Mira telah bernilai lebih dari 50. Struktur kondisi tingkat kedekatan Mira dapat dilihat pada potongan *script* berikut ini :

```
if(GameState.Instance.getMira() == 0) {
    //State Perkenalan
} else if(GameState.Instance.getMira() <= 20) {
    //State Teman
} else if(GameState.Instance.getMira() <= 50) {
    //State Teman Baik
} else {
    //State Teman Dekat }
```

3. Tingkat Kedekatan Fina

Struktur kondisi untuk 4 tingkat kedekatan karakter utama dengan karakter Fina, semakin tinggi nilai *variable* Fina semakin dekat hubungannya. Tingkat pertama adalah perkenalan saat *variable* Fina masih bernilai kosong. Tingkat

kedua adalah teman saat *variable* Fina bernilai dari 1 sampai dengan 20. Tingkat ketiga adalah teman baik saat *variable* Fina bernilai dari 21 sampai dengan 50. Sedangkan tingkat terakhir adalah teman dekat saat *variable* Fina bernilai lebih dari 50. Struktur kondisi tingkat kedekatan Mira dapat dilihat pada potongan *script* berikut ini :

```
if(GameState.Instance.getFina() == 0) {
    //State Perkenalan
} else if(GameState.Instance.getFina() <= 20) {
    //State Teman
} else if(GameState.Instance.getFina() <= 50) {
    //State Teman Baik
} else {
    //State Teman Dekat }
```

3.3.2 Perhitungan Variable Skill

Selama permainan berlangsung pemain juga harus meningkatkan *variable skill* karakter utama. Terdapat tiga *variable skill* yang dapat ditingkatkan oleh pemain, yaitu :

1. Variable Kecerdasan

Variable kecerdasan karakter utama dapat ditingkatkan dengan melakukan kegiatan membaca buku di perpustakaan. Nilai *variable* kecerdasan akan bertambah sebanyak 1.5 poin ketika karakter utama membaca buku, semakin banyak pemain membuat karakter utama membaca buku semakin besar juga nilai *variable* kecerdasan. Perhitungan *variable* kecerdasan dapat dilihat pada potongan *script* berikut ini :

```
GameState.Instance.SetInt1(1.5f);
```

2. Variable Stamina

Variable stamina karakter utama dapat ditingkatkan dengan melakukan kegiatan olahraga di lapangan sekolah. Nilai *variable* stamina akan bertambah sebanyak 1.5 poin ketika karakter utama berolahraga, semakin banyak pemain membuat karakter utama berolahraga semakin besar juga nilai *variable* stamina. Perhitungan *variable* stamina dapat dilihat pada potongan *script* berikut ini :

```
GameState.Instance.setVit(1.5f);
```

3. Variable Populer

Variable populer karakter utama dapat ditingkatkan dengan mengobrol kepada karakter NPC pendukung. Nilai *variable* populer akan bertambah sebanyak 0.5 poin setiap karakter utama mengobrol dengan NPC pendukung, semakin banyak pemain membuat karakter utama mengobrol semakin besar juga nilai akumulasi *variable* populer. Perhitungan *variable* populer dapat dilihat pada potongan *script* berikut ini :

```
GameState.Instance.setSiswa(id,
0.5f);
```

3.3.3 Perhitungan Akhir Karakter NPC

Pada akhir permainan pemain diharuskan untuk menyatakan perasaan kepada salah satu karakter NPC. Akhir permainan adalah hari ke-30 di dalam *game* dan pada saat itu tingkat kedekatan karakter NPC selama permainan akan diakumulasikan untuk menentukan keberhasilan pemain.

1. Akumulasi Karakter Tara

Perhitungan akumulasi karakter Tara adalah hasil akumulasi yang diambil dari 60% *variable* Tara dan 40% *variable* kecerdasan. Hasil akumulasi akan menjadi *variable* penentu ketika karakter utama menyatakan perasaan kepada Tara. Pemain hanya akan berhasil jika hasil akumulasi *variable* bernilai lebih dari 65 dan akan gagal jika dibawah 65. Nilai 65 adalah nilai minimum hasil akumulasi Tara selama 30 hari permainan.

2. Akumulasi Karakter Mira

Perhitungan akumulasi karakter Mira adalah hasil akumulasi yang diambil dari 60% *variable* Mira dan 40% *variable* populer. Hasil akumulasi akan menjadi *variable* penentu ketika karakter utama menyatakan perasaan kepada Mira. Pemain hanya akan berhasil jika hasil akumulasi *variable* bernilai lebih dari 65 dan akan gagal jika dibawah 65. Nilai 65 adalah nilai minimum hasil akumulasi Mira selama 30 hari permainan.

3. Akumulasi Karakter Fina

Perhitungan akumulasi karakter Fina adalah hasil akumulasi yang diambil dari 60% *variable* Fina dan 40% *variable* populer. Hasil akumulasi akan menjadi *variable* penentu ketika karakter utama menyatakan perasaan kepada Fina. Pemain hanya akan berhasil jika hasil akumulasi *variable* bernilai lebih dari 65 dan akan gagal jika dibawah 65. Nilai 65 adalah nilai minimum hasil akumulasi Tara selama 30 hari permainan.

3.3.4 Scene Menu Utama



Gambar 6. Scene Menu Utama

Gambar 6 merupakan *scene* yang pertama kali ditampilkan pada saat pemain memasuki sistem. Terdapat logo "The Relationship" yang merupakan judul *game* serta terdapat 4 tombol utama yaitu Mulai, Cara bermain, Daftar, dan Keluar. Setiap tombol pada *scene* utama memiliki fungsinya

masing-masing untuk menghubungkan satu *scene* dan *scene* lainnya.

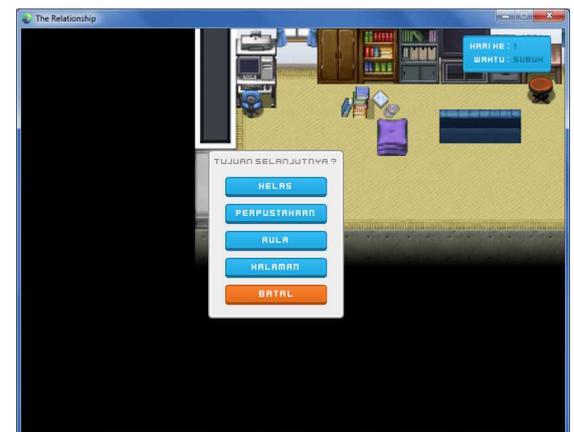
3.3.5 Scene Mulai

Scene Mulai dapat diakses melalui tombol Mulai pada *scene* menu utama. *Scene* mulai adalah *scene* utama untuk memainkan *game*. Saat permainan pertama kali dimulai karakter utama akan berada di ruang kamar dan setelah itu pemain dapat mengontrol karakter utama untuk pindah ke ruangan lain dan juga berinteraksi dengan NPC. Total terdapat 5 (lima) *scene* ruangan yang dapat didatangi oleh pemain yaitu *scene* ruang kamar, *scene* ruang kelas, *scene* aula sekolah, *scene* perpustakaan, dan *scene* halaman sekolah. Tampilan *scene* ruang kamar dapat dilihat pada gambar 7.



Gambar 7. Scene Ruang Kamar

Ketika pemain mengarahkan karakter utama menuju pintu untuk berpindah ruangan, maka akan muncul kotak dialog untuk memberikan pemain pilihan ruangan yang akan dituju. Tampilan kotak dialog pindah ruangan dapat dilihat pada gambar 8.



Gambar 8. Kotak Dialog Pindah Ruangan

Karakter utama yang diarahkan berjalan menuju karakter NPC hingga bersentuhan akan menampilkan kotak dialog obrolan antara karakter NPC dan karakter utama yang akan berpengaruh terhadap *variable* karakter NPC. Tampilan kotak dialog mengobrol dapat dilihat pada gambar 9.



Gambar 9. Kotak Dialog Mengobrol

Selain menampilkan kotak dialog mengobrol, ketika karakter utama menyentuh karakter NPC juga akan menampilkan kotak dialog pertanyaan. Setiap jawaban dari pertanyaan yang diberikan oleh NPC akan berpengaruh terhadap *variable* karakter NPC. Tampilan kotak dialog pertanyaan dapat dilihat pada gambar 10.



Gambar 10. Kotak Dialog Pertanyaan NPC

KESIMPULAN

Berdasarkan hasil dari penerapan metode *finite state machine* pada game "The Relationship" diperoleh kesimpulan antara lain:

1. Penerapan metode *finite state machine* menghasilkan respon NPC yang berbeda menyesuaikan apa yang telah dimainkan oleh pemain.
2. Hasil akhir game yang ditentukan oleh pemain karena metode *finite state machine* menawarkan pilihan sepanjang permainan berlangsung.

SARAN

Dari hasil penelitian yang dilakukan, disadari masih banyak kekurangan dalam sistem, beberapa saran yang dapat berguna untuk mengembangkan sistem, antara lain:

1. Memperbanyak interaksi NPC yang lebih kompleks agar semakin banyak kemungkinan respon yang didapatkan oleh pemain.
2. Menambahkan lebih banyak karakter NPC yang bisa didekati oleh karakter utama

sehingga lebih banyak pilihan yang ditawarkan.

3. Perhitungan tingkat kedekatan karakter NPC dan karakter utama yang lebih baik agar membuat pemain lebih tertantang.

DAFTAR PUSTAKA

- [1] Costikyan, Greg. 2013. *Uncertainty in Games (Playful Thinking)*. MIT Press.
- [2] Ekaputra, Valentino. 2014. Analisis dan Perancangan Aplikasi Game Multiplayer Life Simulation "Pet Master" Menggunakan Game Engine Unity 3D. Jakarta: Universitas Bina Nusantara.
- [3] Hikam, Arif Rahman. 2013. Pengembangan Game Edukasi Visual Novel berbasis Pembangunan Karakter pada Materi Pelestarian Lingkungan. Semarang: Universitas Negeri Semarang.
- [4] Setiawan, I. 2006. *Perancangan Software Embedded System Berbasis FSM*. Semarang: Universitas Diponegoro.
- [5] Rostianingsih, Silvia. 2013. *Game Simulasi Finite State Machine Untuk Pertanian dan Peternakan*. Surabaya: Universitas Kristen Petra.
- [6] Utari, Dely. 2012. Analisis Dan Implementasi Algoritma *Backtracking* Pada Permainan *Hangman*. Medan: Universitas Sumatera Utara.
- [7] Wijaya, Surya Adi. 2009. *Fuzzy State Machine Untuk Menghasilkan Variasi Respon Npc (Non-Playable Character) Pada Game*. Tesis. Surabaya: Institut Teknologi Surabaya.