

PEMBUATAN GAME DANGEROUS ESCAPE DENGAN MENGGUNAKAN METODE FSM (FINITE STATE MACHINE)

Jaka Rahmadi

Program Studi Teknik Informatika S1, Fakultas Teknologi Industri
Institut Teknologi Nasional Malang, Jalan Raya Karanglo km 2 Malang, Indonesia
aegyssenigma@gmail.com

ABSTRAK

Computer adalah salah satu media yang paling banyak digunakan para pengembang industri *game* untuk mengeluarkan produknya, selain *computer* terdapat media lain untuk bermain *game* yaitu konsol *game* dan *smartphone*. Dalam *game*, musuh dan karakter merupakan salah satu elemen yang paling penting dalam *video game*, untuk menjadikan *game* semakin menarik, karakter dan NPC (*Non Playable Character*), salah satunya musuh dalam *game* harus dapat meniru sifat atau perilaku manusia dalam dunia nyata. Untuk melakukan hal tersebut diperlukan kecerdasan buatan pada karakter NPC (*Non Playable Character*) pada musuh yaitu kecerdasan buatan FSM (*Finite State Machine*) yang menggunakan kecerdasan buatan FSM (*Finite State Machine*) pada karakter musuh agar dapat mendeteksi keberadaan *player*.

Dalam perancangan *game* ini, penulis menggunakan aplikasi berupa *Unity3D* dengan bahasa pemrograman C#. *Game* ini menerapkan metode kecerdasan buatan *Finite State Machine* yang menggunakan 3 *state* yaitu keadaan, kejadian, dan aksi, untuk menentukan aksi dengan jarak *player* ke musuh.

Hasil pengujian *game Dangerous Escape* yang telah dilakukan menunjukkan bahwa fungsi yang terdapat pada *game* dari menu utama hingga level 1, level 2 dan level 3 dapat berhasil. Metode kecerdasan buatan *Finite State Machine* yang diterapkan pada musuh, dapat mendeteksi keberadaan *player* dan menghasilkan *output* yang dapat menentukan aksi apa yang akan dilakukan oleh musuh.

Dari hasil pengujian *game Dangerous Escape* pada *user*, dapat diketahui bahwa secara umum rata-rata nilai dari responden menganggap 70% *game* baik, 23% menganggap *game* cukup baik, dan 7% menganggap *game* masih kurang.

Kata kunci : *Game, Unity3D, Finite State Machine, Non Player Character*

1. PENDAHULUAN

Permainan atau Permainan atau yang sering disebut dengan *game* merupakan suatu sarana hiburan yang diminati oleh banyak orang, baik dari kalangan anak-anak, remaja maupun orang dewasa, *platform* yang digunakan untuk bermain *game* saat ini pun pada saat ini tidak hanya sebatas pada konsol *gaming* saja, mulai dari *computer* hingga *smartphone* bisa digunakan sebagai media memainkan *game*. Dari sekian banyak jenis genre *game* yang ada, *side-scrolling 2D* bisa dibilang termasuk salah satu yang tertua dan terpopuler, seperti *game Contra* yang dirilis pada tahun 1987. Genre ini menarik karena konsep dasarnya sederhana, namun bisa dikembangkan terus sehingga tetap menarik dan asyik untuk dimainkan. Saat ini memang sudah banyak *game* dengan berbagai macam genre yang dirilis dengan kualitas grafis yang lebih bagus, tetapi tidak sedikit dari *game-game* tersebut yang hanya mengandalkan keunggulan dari segi grafis saja namun memiliki AI yang tidak responsif, seperti *game Dynasty Warrior, Sengoku Basara, dan Samurai Warrior*. Pada *game-game* tersebut AI yang diterapkan pada karakter musuh hanya mendekati *player* dan menunggu untuk diserang oleh *player* tanpa melakukan perlawanan apapun terhadap *player*.

Padahal dalam sebuah *game*, kecerdasan buatan digunakan untuk memberikan pengalaman yang interaktif pada *player*, tidak hanya dari segi grafis, pengalaman bermain (*gameplay experience*) juga bergantung pada kecerdasan buatan yang ada pada sebuah *game* sehingga dapat membuat *game* menjadi lebih menarik.

Dari uraian tersebut penulis kemudian tergagas untuk mengembangkan sebuah *game* yang menarik untuk dimainkan, dengan kemampuan AI yang lebih baik. Maka dari itu penulis berusaha mengembangkan sebuah *game* berjudul *Dangerous Escape*, yang menggunakan metode *finite State Machine* yang diterapkan pada *non playable character* dalam *game*. *Non playable character* tersebut akan berperan sebagai musuh dalam *game* yang harus dikalahkan oleh *player*.

Dalam *game Dangerous Escape*, musuh di implementasikan dengan AI (*Artificial Intelligence*) *Finite State Machine*. Penggunaan kecerdasan buatan pada *game Dangerous Escape* ini agar musuh dapat berinteraksi dengan pemain dan memberikan pengalaman bermain yang lebih menarik. Terdapat banyak metode yang dapat digunakan untuk pembuatan AI dalam sebuah *game*, namun metode yang digunakan dalam *game Dangerous Escape* ini adalah metode *Finite State Machine*. *Finite State*

Machine sendiri adalah salah satu jenis *artificial Intelligence* yang merupakan sebuah metodologi perancangan *system control* yang menggambarkan tingkah laku atau prinsip kerja dengan menggunakan tiga hal yang meliputi: *State* (keadaan), *Event* (Kejadian) dan *Action* (aksi). Dimana pada satu saat dalam periode waktu yang cukup signifikan, *system* akan berada pada salah satu *state* yang aktif. *System* dapat beralih atau bertransisi menuju *state* lain jika mendapatkan masukan atau *event* tertentu.

2. TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Metode pembuatan kecerdasan buatan pada *game* dengan menggunakan Metode *Finite State Machine* juga pernah digunakan oleh peneliti sebelumnya. Maka peneliti juga perlu untuk mempelajari penelitian-penelitian sebelumnya sebagai dasar acuan dalam mengerjakan penelitian ini.

Rujukan penelitian pertama yaitu penelitian oleh Tito (2016) dengan judul *Pemodelan Perilaku Musuh Menggunakan Finite State Machine (FSM) Pada Game Pengenalan Unsur Kimia*. Dalam penelitian tersebut peneliti menggunakan metode *finite state machine* pada pemodelan perilaku musuh. Dengan menggunakan metode *Finite State Machine* musuh akan bereaksi pada *player* berdasarkan jarak dari *player* terhadap musuh. Rujukan penelitian kedua yaitu penelitian oleh Rahadian (2016) dengan judul *Penerapan Metode Finite State Machine Pada Game "The Relationship"*. Dalam penelitian tersebut peneliti menggunakan metode *finite state machine* untuk menentukan respons karakter *non-player character* yang berbeda tergantung dari interaksi yang dilakukan oleh pemain. Saat pertama dijalankan NPC akan berada pada *State idle*, kemudian NPC akan mulai bergerak dan melakukan interaksi setelah diberi perintah oleh *player*. Rujukan penelitian ketiga yaitu penelitian oleh Rostianingsih (2013) dengan judul *Perancangan Dan Pembuatan Game Simulasi Pertanian Dan Peternakan*. Dalam penelitian tersebut peneliti menggunakan metode *finite state machine* untuk menentukan respons karakter *non-player character* yang menjadi pemain lain dalam *game*. NPC tersebut termasuk dalam karakter utama, yang kemudian akan berinteraksi satu sama lain, dan metode *Finite State Machine* diterapkan pada aktivitas bercocok tanam dari setiap karakter.

2.2 Game

Game merupakan salah satu hiburan karena mampu mengurangi tingkat kepenatan seseorang dari rutinitas pekerjaan setiap hari. *Game* juga mampu meningkatkan kecerdasan seseorang ketika *game* tersebut memerlukan tingkat ketangkasan dari seorang pemain. (Praja, 2016)

Game atau lebih tepatnya disebut *video game* adalah permainan yang menggunakan interaksi dengan antarmuka pengguna melalui gambar yang

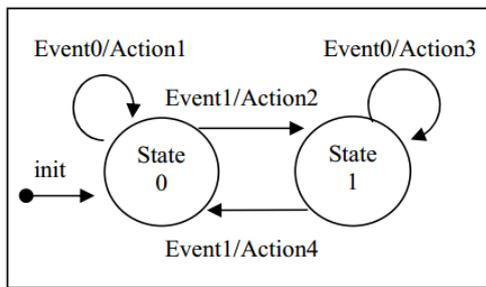
dihasilkan oleh piranti *video*. Permainan *video* umumnya menyediakan system penghargaan yang dicapai dalam menyelesaikan tugas-tugas yang ada dalam permainan. (Naomi, 2016)

2.3 Artificial Intelligence

Menurut McCarthy (1956), AI adalah untuk mengetahui dan memodelkan prose-proses berpikir manusia dan mendesain mesin agar dapat menirukan perilaku manusia. Cerdas, berarti memiliki pengetahuan ditambah pengalaman. Penalaran (bagaimana membuat keputusan dan mengambil tindakan), moral yang baik. manusia cerdas dalam menyelesaikan permasalahan karena manusia mempunyai pengetahuan dan pengalaman. Pengetahuan diperoleh dari belajar. Semakin banyak bekal pengetahuan yang dimiliki tentu lebih mampu menyelesaikan permasalahan. Tapi bekal pengetahuan tidak cukup, manusia juga diberi akal untuk melakukan penalaran, mengambil keputusan berdasarkan pengetahuan dan pengalaman yang dimiliki. Tanpa memiliki kemampuan untuk menalar dengan baik, manusia dengan segudang pengalaman dan pengetahuan tidak dapat menyelesaikan masalah dengan baik. Demikian juga dengan kemampuan pengalaman yang memadai manusia tidak bisa menyelesaikan masalah dengan baik (Dahria, 2008).

2.4 Metode Finite State Machine

Finite state machine adalah sebuah metodologi perancangan *system control* yang menggambarkan tingkah laku atau prinsip kerja dengan menggunakan tiga hal yang meliputi: *State* (keadaan), *Event* (Kejadian) dan *Action* (aksi). Pada satu saat dalam periode waktu yang cukup signifikan, *system* akan berada pada salah satu *state* yang aktif. *System* 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 (misal interupsi *timer*). (Setiawan, 2006). Transisi keadaan ini umumnya juga disertai aksi oleh aksi yang dilakukan oleh system ketika menanggapi masukan yang terjadi. Aksi yang dilakukan tersebut dapat berupa aksi yang sederhana atau melibatkan rangkaian proses yang relative kompleks. Berdasarkan sifatnya metode FSM ini sangat cocok untuk digunakan sebagai basis perancangan perangkat lunak pengendalian yang reaktif dan *real time*. Salah satu keuntungan nyata penggunaan FSM adalah kemampuannya dalam mendekomposisi aplikasi yang relatif besar dengan hanya menggunakan sejumlah kecil *item state*, seperti ditunjukkan pada Gambar 1. (Setiawan, 2006).



Gambar 1. Diagram State sederhana FSM

Selain untuk bidang *control*, penggunaan metode ini pada kenyataannya juga umum digunakan sebagai basis untuk perancangan protokol-protokol komunikasi, perancangan perangkat lunak *game*, aplikasi *WEB* dan sebagainya, dalam bahasa pemrograman procedural seperti bahasa C, FSM umumnya direalisasikan menggunakan *statemen control switch case* atau *if ... Then*. Dengan menggunakan statemen-statemen kontrol ini, aliran program secara praktis akan mudah dipahami dan dilacak jika terjadi kesalahan logika (Setiawan, 2006).

3. ANALISIS DAN PERANCANGAN

3.1 Story Line

Game ini menceritakan tentang robot yang mencoba melarikan diri dari sebuah tempat penelitian. Untuk alur cerita *game* ini yaitu :

1. Di ceritakan robot yang bernama Aegys, dia adalah sebuah robot humanoid yang memiliki kemampuan yang lebih dari robot yang telah ada sebelumnya.
2. Namun karena kelebihannya itu, dia tidak diperbolehkan berinteraksi dengan dunia luar.
3. Dengan berbekal rasa penasarannya itu, dia memutuskan untuk mencoba melarikan diri dari tempat penelitian tersebut.
4. Agar dapat keluar dari tempat penelitian tersebut, terdapat 3 pintu utama yang harus dilewati oleh Aegys.
5. Namun ternyata para penjaga telah mengetahui rencana Aegys, dan mencoba menghadang rute pelarian Aegys.
6. Untuk dapat melewati pintu tersebut, Aegys harus mengalahkan para penjaga yang menghadang.
7. Setelah mengalahkan para penjaga tersebut, akhirnya Aegys dapat keluar dari tempat penelitian tersebut.
8. Perjuangan Aegys tidak sia-sia, dia akhirnya dapat melihat dunia luar yang sangat ingin dia ketahui.

3.2 Desain Level

Game ini terbagi menjadi dua level utama. Masing-masing level memiliki latar, musuh dan tingkat kesulitan yang berbeda-beda. Beberapa aturan dari *game* ini yaitu :

- a. Player memiliki nyawa dalam bentuk *+ / plus*.
- b. Player memiliki HP (*health point*) dalam bentuk bar.

- c. Player mempunyai tiga aksi utama, yaitu *primary attack* (menembak super), *secondary attack* (menembak), dan *jump*.
- d. Player mati jika darah / HP Bar habis
- e. Terdapat item koin disepanjang perjalanan sebagai poin tambahan.
- f. Terdapat item untuk menambah *health point* (HP)
- g. Terdapat item untuk menambah nyawa.
- h. Terdapat item untuk mendapat senjata baru.
- i. Terdapat 5 musuh yang berupa *Guard Dog*, *Guard Dog Captain* Robot 1, *Robot Captain*, *Robot General*.
- j. Pada setiap level setiap musuh yang ada memiliki varian yang berbeda, yaitu dari jumlah *damage* dan *health point* yang dimiliki.
- k. Setiap membunuh musuh mendapat poin.
- l. Poin ditentukan dari jumlah musuh yang dibunuh dan jumlah koin yang dikumpulkan.
- m. Untuk menyelesaikan permainan ini, pemain harus melewati semua level dan mengalahkan boss utama.
- n. Apabila HP habis pemain harus kembali ke awal level.
- o. Apabila nyawa habis pemain harus mengulang ke level 1.

Level 1 adalah level pertama yang harus dimainkan pada *Game Dangerous Escape*, berikut adalah beberapa aturan pada Level 1:

- a. Player pertama diberikan dua nyawa.
- b. Terdapat 1 item penambah HP.
- c. Terdapat 1 item senjata baru.
- d. Terdapat 8 item koin
- e. Terdapat 5 musuh *Guard Dog*.
- f. Terdapat 1 musuh *Guard Dog Captain* yang akan menambah poin jika dibunuh.

Level 2 adalah level kedua yang harus dimainkan pada *Game Dangerous Escape* setelah berhasil melewati level 1, berikut adalah beberapa aturan pada Level 2:

- a. Player pertama diberikan dua nyawa.
- b. Terdapat 1 item penambah nyawa.
- c. Terdapat 2 item penambah HP.
- d. Terdapat 1 item senjata baru.
- e. Terdapat 10 item koin.
- f. Terdapat 4 musuh *Guard Dog*, 4 musuh *Robot 1*, 1 musuh *Robot Captain* dan 1 musuh *Guard Dog Captain*.

Level 3 adalah level terakhir yang harus dimainkan pada *Game Dangerous Escape* setelah berhasil melewati level 2, berikut adalah beberapa aturan pada Level 3:

- a. Player pertama di berikan dua nyawa.
- b. Terdapat 1 item senjata baru
- c. Terdapat 1 musuh *Robot General*.

3.3 Desain Karakter

Perancangan karakter merupakan pembahasan mengenai karakter yang terlibat dalam *Game Dangerous Escape*. Karakter pada *Game Dangerous Escape* pada Tabel 1.

Tabel 1. Penjelasan Karakter

No	Desain	Keterangan
1		Karakter utama pada game.
2		Musuh A
3		Musuh B
4		Musuh C

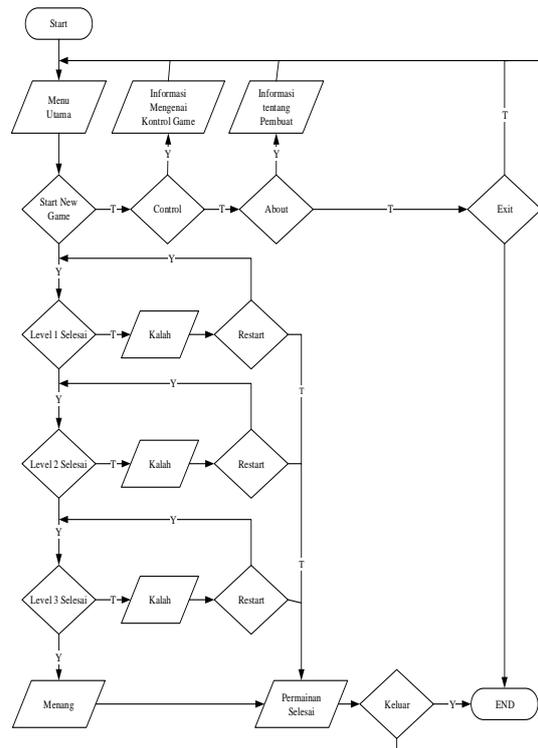
3.4 Item dalam Game

Dalam game *Dangerous Escape* ini terdapat 6 macam *item* tambahan untuk menambah daya tarik user untuk memainkan game ini. Berikut adalah *item game* yang akan dijelaskan pada Tabel 2 berikut ini.

Tabel 2. Item Dalam Game

No	Item	Keterangan
1		Gambar disamping adalah item penambah <i>Health</i> yang akan menambah <i>Health poin</i> pada <i>player</i>
2		Gambar disamping adalah <i>item koin</i> yang akan menambah <i>score</i> pada game
3		Gambar disamping adalah <i>item level up</i> yang akan mengaktifkan senjata spesial pada <i>player</i> .
4		Gambar disamping adalah <i>item Lives</i> (Nyawa) yang akan menambah <i>Lives</i> (nyawa) pada <i>player</i> .
5		Gambar disamping adalah <i>item rintangan</i> yang akan mengurangi <i>health point</i> yang ada pada <i>player</i> .
6		Gambar disamping adalah <i>item pintu</i> yang merupakan syarat yang harus dilewati <i>player</i> untuk memenangkan game.

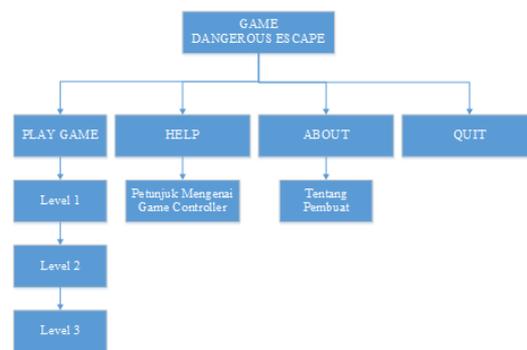
3.5 Flowchart Sistem



Gambar 2. Flowchart sistem

Flowchart berfungsi untuk mengetahui alur proses dari program dimulai dari *start* program hingga *end* program. Program dimulai dari *start* kemudian masuk ke 4 menu utama yaitu *Play*, *Help*, *About*, dan *Exit*. Jika permainan memulai *Play* maka akan langsung saja menuju level 1. Jika permainan memilih *Help* maka akan muncul bantuan kontrol dalam game untuk bermain. Jika permainan memilih *About* maka akan muncul tampilan profil pembuat game. Jika pemain memilih *Exit* maka akan keluar dari game.

3.6 Struktur Menu

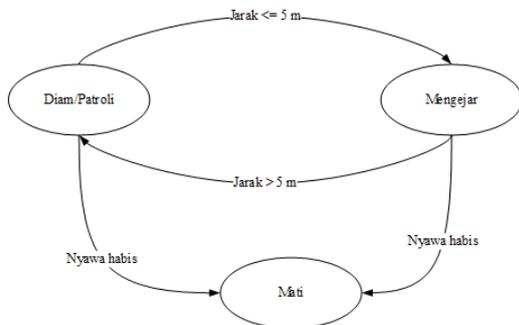


Gambar 3. Struktur Menu

Pada game *Dangerous Escape* ini terdiri dari beberapa menu yaitu *Start Game*, *Help*, *About us*, dan *Exit*. Pada menu *start game* akan menuju Level 1, Level 2 dan Level 3 dalam *game*, pada menu *Help* akan ditampilkan panduan untuk memainkan *game*, pada menu *about us* akan menampilkan informasi dasar dari pembuat *game*, sedangkan pada menu *exit* akan menutup aplikasi *game* tersebut.

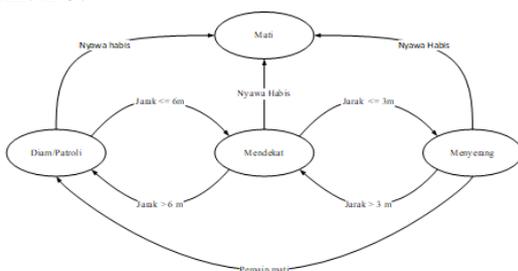
3.7 Diagram Finite State Machine pada Musuh

Game ini menggunakan 1 metode yaitu *Finite State Machine (FSM)* yang digunakan untuk memberikan *State (Keadaan)*, *Event (Kejadian)* dan *Action (Aksi)*. Sifat perilaku musuh *Guard Dog*, dan *Guard Dog Captain (Musuh A)* hanya mengejar karakter. *Health poin player* akan berkurang jika Musuh A menyentuh *player*. Musuh A akan mengejar *player* jika *player* masuk dalam jangkauan. *Player* dan musuh A akan mati jika *Health poin* sudah habis, seperti ditunjukkan pada Gambar 4.



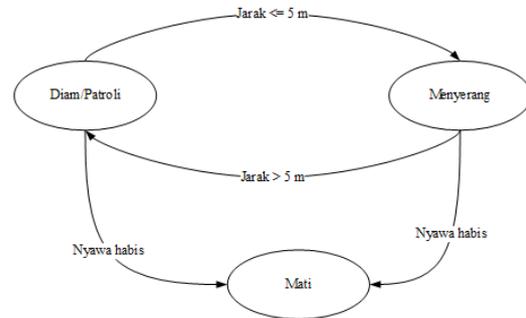
Gambar 4. Konsep FSM pada Musuh A

Sedangkan sifat perilaku musuh Robot 1 hanya mengejar dan menyerang karakter. *Health poin player* akan berkurang jika Robot 1 (Musuh B) menyentuh *player* dan terkena serangan dari Musuh B. Musuh akan mengejar *player* jika *player* masuk dalam jangkauan 1 musuh Alien A, dan akan diam jika keluar jangkauan. Kemudian musuh akan menyerang *player* jika *player* masuk dalam jangkauan 2. *Player* dan musuh B akan mati jika *Health poin* sudah habis, seperti ditunjukkan pada Gambar 5.



Gambar 5. Konsep FSM pada Musuh A

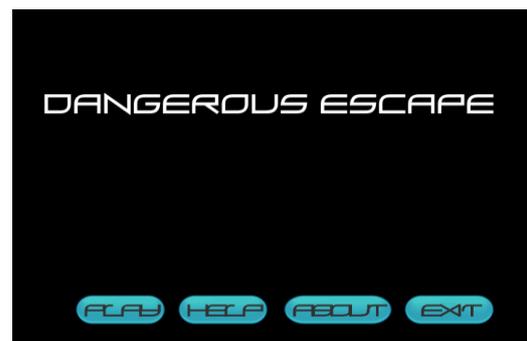
Sifat perilaku musuh Robot *Captain* dan Robot *General (Musuh C)* hanya menyerang karakter. *Health poin player* akan berkurang jika Musuh C menyentuh *player*. Musuh C akan langsung menyerang *player* jika *player* masuk dalam jangkauan musuh Musuh B. *Player* dan musuh A akan mati jika *Health poin* sudah habis, seperti ditunjukkan pada Gambar 6.



Gambar 6. Konsep FSM pada Musuh C

4. IMPLEMENTASI DAN PENGUJIAN

4.1 Menu Utama



Gambar 7. Menu Utama

Menu utama dari *game Dangerous Escape*. Menu utama dari *game* ini dapat diakses oleh *user* ketika permainan akan dimulai. Pada menu utama ini *user* dapat memilih menu yang diinginkan. Jika *user* memilih menu *play* maka *user* akan memulai permainan, sebelum *user* memulai permainan, disarankan terlebih dahulu untuk memilih menu *help* yang berisi petunjuk permainan, jadi *user* tidak akan kesulitan dalam memainkan *game* ini.

4.2 Menu About

Pada bagian ini merupakan menu *about* untuk menampilkan tentang pembuat *game* seperti nama, jurusan dan e-mail dari pembuat. Halaman *About* tersebut seperti ditunjukkan pada Gambar 8.



Gambar 8. Halaman About

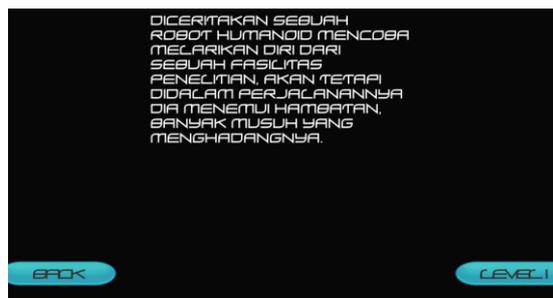
4.3 Menu Help Game



Gambar 9. Menu Help

Menu *help*, yang merupakan menu yang berisi petunjuk permainan yang disediakan kepada *user* ketika akan memulai permainan seperti ditunjukkan pada Gambar 9.

4.4 Opening Game



Gambar 10. Opening Game

Pada bagian ini merupakan tampilan *opening* untuk menampilkan cerita dan pilihan lanjut ke level selanjutnya. seperti ditunjukkan pada Gambar 10.

4.5 Tampilan Game Over



Gambar 11. Tampilan Game Over

Pada bagian ini merupakan tampilan game over yang muncul apabila player kalah dan nyawa nyawa yang dimiliki sudah habis, untuk mengulang pada level tersebut atau kembali ke menu utama. seperti ditunjukkan pada Gambar 11.

4.6 Tampilan Game Over Menang



Gambar 12. Game Over Menang

Pada bagian ini merupakan tampilan game over yang muncul setelah berhasil memenangkan level 1, untuk melanjutkan ke level 2 dan Level 3 pada game atau kembali ke menu utama. seperti ditunjukkan pada Gambar 12.

4.7 Implementasi Level 1



Gambar 13. Level 1

Pada *game level* pertama pemain harus melewati sebuah ruangan yang penuh dengan musuh dengan kecerdasan mengejar dan menyerang player supaya tidak bisa menyelesaikan level 1 dari musuh yang telah dilengkapi dengan pertahanan tubuh yang kuat dan halangan berupa bebatuan tajam kecil yang memotong jalan sehingga dapat mengalahkan *player*. Jika *player* dapat menyelesaikan misi pada *level* pertama maka *player* dapat melanjutkan ke *level* kedua. *Level* pertama pada *game* ini bersifat *easy*, artinya tantangan yang diberikan tidak terlalu kuat. Tantangan tersebut terletak pada musuh. Jika pada *level* pertama musuh akan menyerang dengan mengejar *player* dan bos yang akan mengejar kemudian menyerang *player*. Seperti ditunjukkan pada Gambar 13.

4.8 Implementasi Level 2



Gambar 14. Level 2

Pada *game level* kedua pemain harus melewati ruangan yang dengan musuh dengan kecerdasan mengejar dan menyerang player supaya tidak bisa menyelesaikan level. Dan musuh serangan yang kuat dan halangan berupa batu tajam yang dapat mengurangi *Health player* sehingga dapat mengalahkan *player*. Jika *player* dapat menyelesaikan misi pada *level* kedua maka *player* dapat melanjutkan ke *level* tiga. *Level* kedua pada *game* ini bersifat *hard*, artinya tantangan yang diberikan sangat kuat. Tantangan tersebut terletak pada *damage enemy* dan kecepatan serangan pada musuh. Jika pada *level* pertama hanya mengejar maka pada level 2 musuh dapat mengejar serta menembak, jika *player* mendekat dan masuk jangkauan musuh. Seperti ditunjukkan pada Gambar 14.

4.9 Implementasi Level 3



Gambar 15. Level 3

Pada *game level* ketiga pemain harus melewati ruangan terakhir dimana terdapat musuh bos akhir. Dengan serangan yang kuat dapat mengurangi *Health player* sehingga dapat mengalahkan *player*. Jika *player* dapat menyelesaikan misi pada *level* ketiga maka *player* dapat memenangkan *game* ini. *Level* kedua pada *game* ini bersifat *hard*, artinya tantangan yang diberikan sangat kuat. Tantangan tersebut terletak pada *damage enemy*. Pada level 3 musuh dapat menembak, jika *player* mendekat dan masuk jangkauan musuh. Seperti ditunjukkan pada Gambar 15.

4.10 Pengujian Finite State Machine Pada Musuh

Pengujian *artificial intelligence* adalah pengujian mengenai fungsi yang berkaitan dengan *artificial intelligence* yang ada pada *Game Dangerous Escape*. Hasil pengujian ditunjukkan pada Tabel 3.

Tabel 3. Pengujian Metode

Keadaan	Kejadian	Tindakan	Hasil
Idle	Jika <i>player</i> berada pada pada jarak 5m (Skala <i>Unity3D</i>)	Musuh mengejar player	Sesuai
Idle	Jika <i>player</i> berada pada jarak 3m (Skala <i>Unity3D</i>)	Musuh menyerang player dengan senjata jarak jauh	Sesuai
Idle	Jika <i>player</i> berada pada jarak sangat dekat $\leq 5m$ (Skala <i>Unity3D</i>)	Musuh memberikan damage dengan menabrak player	Sesuai
Mengejar / menyerang	Jika <i>player</i> keluar dari jangkauan $> 5m$ (Skala <i>Unity3D</i>)	Musuh akan kembali idle	Sesuai
Idle	Jika <i>player</i> tetap berada pada jarak pandang musuh $> 5m$ (Skala <i>Unity3D</i>)	Musuh tetap diam.	Sesuai

Dari Tabel 3 menunjukkan bahwa kecerdasan buatan yang diterapkan pada musuh sudah berhasil diimplementasi sesuai dengan rancangan kecerdasan buatan yang sudah dibuat.

4.11 Pengujian Sistem Operasi

Pada fungsionalitas ini game dicoba dijalankan pada Windows 7 & 10. Hasil pengujian ditunjukkan pada Tabel 4.

Tabel 4. Pengujian Sistem Operasi

No	Item Uji	Windows 7		Windows 10	
		Berhasil	Gagal	Berhasil	Gagal
1	Menu Utama	√	-	√	-
2	Button Start	√	-	√	-
3	Button Help	√	-	√	-
4	Button About	√	-	√	-
5	Button Exit	√	-	√	-
6	Masuk Level 1 Game	√	-	√	-
7	Masuk Level 2 Game	√	-	√	-
8	Bar Health Point karakter	√	-	√	-
9	Bar Health Point Musuh	√	-	√	-
10	Loncat dan animation state loncat pada karakter	√	-	√	-
11	Attack dan animation state attack pada karakter	√	-	√	-
12	HP musuh berkurang ketika terkena attack	√	-	√	-
13	Collision Detection pada Musuh	√	-	√	-
14	Item tambah HP pada player	√	-	√	-
15	Item Koin	√	-	√	-
16	Musuh mengikuti pada daerah yang ditentukan	√	-	√	-
17	Musuh menyerang pada jarak yang ditentukan	√	-	√	-
18	Game tamat ketika player menyentuh pintu	√	-	√	-

Dari Tabel 4 menunjukkan bahwa pengujian fungsionalitas di Windows 7 dan 10 berhasil dijalankan sesuai dengan yang diuji.

5. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil analisa dan pengujian yang telah dilakukan, maka dapat diambil beberapa kesimpulan sebagai berikut:

1. Implementasi kecerdasan buatan FSM (*Finite State Machine*) berhasil diterapkan pada *game Dangerous Escape* dengan aksi musuh dapat mengejar dan menyerang *player* dengan kondisi *player* masuk dalam jarak serang musuh pada musuh 1 (6m), pada musuh 2 (3m & 6m), dan pada musuh 3 (5m).
2. Semua aksi pada *player* berhasil diimplementasikan sesuai dengan yang dirancang sebelumnya.

3. Fungsi tombol pada menu utama sudah berhasil berjalan sesuai yang dirancang pada struktur menu.
4. Fungsi yang diterapkan pada karakter *game* pada level 1, level 2, dan level 3 berhasil seluruhnya sesuai dengan desain level yang sudah dibuat.
5. Pengujian sistem operasi pada *game* berhasil dijalankan pada versi *windows 7* maupun *windows 10*.
6. Sebagian besar responden yang melakukan pengujian pada *game* menyatakan jika *game Dangerous Escape* termasuk baik sebesar 70%.

5.2 Saran

Adapun saran yang diberikan untuk penelitian selanjutnya antara lain:

1. Dikembangkan sebagai *mobile game* berbasis *android* dan *IOS* atau versi webnya.
2. Dapat ditambah jumlah *level* dan ditingkatkan kualitas grafiknya.

DAFTAR PUSTAKA

- [1] Iwan Setiawan. 2006. Perancangan *Software Embedded System* Berbasis FSM. Universitas Diponegoro, 1-2.
- [2] Muhammad Dahria. 2008. Kecerdasan Buatan (*Artificial Intelligence*). Jurnal SAINTIKOM, 5(2), 185-196.
- [3] Naomi Budi Mawarni. 2016. Perancangan Game Tentang Wisata Kebun Binatang Semarang Untuk Anak Usia 4-6 Tahun. *Skripsi*, Semarang : Universitas Dian Nuswantoro.
- [4] Praja Irwandi, Aan Erlansari, Rusdi Effendi. 2016. PERANCANGAN GAME *FIRST PERSON SHOOTER* (FPS) “BOAR HUNTER” BERBASIS VIRTUAL REALITY. Jurnal Rekursif, 4(1), 68-79.
- [5] Rahadian, M. F., Suyatno, A., & Maharani, S. 2016. PENERAPAN METODE FINITE STATE MACHINE PADA GAME “THE RELATIONSHIP”. *Jurnal Informatika Mulawarman*, 11(1), 14-22.
- [6] Rostianingsih, S., Satiabudhi, G. and Wijaya, H.K., 2013. *Game Simulasi Finite State Machine Untuk Pertanian dan Peternakan* (Doctoral dissertation, Petra Christian University), Surabaya : Universitas Kristen Petra, 1-6.
- [7] Tito, B., 2016. Pemodelan Perilaku Musuh Menggunakan Finite State Machine (FSM) Pada Game Pengenalan Unsur Kimia. *Skripsi*, Semarang : Universitas Dian Nuswantoro.