

GAME FIGHTING THE BLACK BLOOD MENGGUNAKAN UNITY 3D INSTITUT TEKNOLOGI NASIONAL MALANG

Hasna Aprilia

Teknik Informatika, Institut Teknologi Nasional Malang
apriahhasna@outlook.com

ABSTRAK

Game kini sudah menjadi alternatif hiburan bagi banyak kalangan, dari anak-anak hingga orang dewasa, pria maupun wanita. *Game* kini telah dikembangkan oleh banyak negara seperti pada benua Amerika, Eropa dan Asia. Tujuan jangka panjang dari pembuatan *game* The Black Blood ini diharapkan dapat membuat pemain merasakan suasana pertarungan antara tokoh utama dan musuh-musuhnya dalam cerita yang telah diberikan. Sedangkan untuk tujuan khususnya, *game* The Black Blood ini akan diproduksi lebih banyak dan dapat dimainkan oleh pecinta *game* selain *game fighting*, agar dapat merasakan bagaimana suasana bermain *game fighting*.

Metode yang dipakai dalam pembuatan *game* The Black Blood ini adalah metode *collision detection*. Dimana metode tersebut bagian dari metode kecerdasan buatan atau *artificial intelligence* dimana ia adalah sebuah fungsi yang dapat mendeteksi tabrakan antara 2 obyek atau lebih. Metode *collision detection* ini akan diterapkan pada karakter utama dan pada musuh. Musuh atau *enemy* pada *game* The Black Blood ini berperan sebagai lawan dari tokoh utama yang harus dikalahkan pada tiap *stage* permainan. Untuk dapat menyelesaikan *game* ini, pemain harus mengalahkan para lawan di setiap *stage*.

Hasil pengujian dari metode *collision detection* yang telah diterapkan pada *game* ini menunjukkan bahwa musuh dapat menyerang pemain dan memberikan *damage* pada pemain, begitu pula dengan pemain yang dapat menyerang musuh dan memberikan *damage*. Hasil pengujian kontrol player juga sudah sesuai dan berjalan sesuai dengan yang telah dirancang.

Kata kunci: *Game, fighting, collision detection*

1. PENDAHULUAN

Permainan atau sering disebut dengan *game* merupakan suatu sarana hiburan yang diminati dan dimainkan oleh banyak orang baik dari kalangan anak-anak, remaja maupun orang dewasa. *Game* digital merupakan *game* yang disajikan pada suatu piranti atau perangkat teknologi dan dimainkan secara virtual. Media teknologi yang digunakan, yaitu *console*, komputer, *handphone* dan elektronik lainnya.

Dewasa ini, *game* digital mulai banyak dimainkan karena dianggap lebih menarik daripada *game* konvensional. Hal ini dikarenakan video *game* membuat para pemainnya dapat melakukan banyak hal yang tidak mungkin atau terlalu mahal untuk dilakukan di dunia nyata. *Game* digital saat ini memiliki banyak *genre*, seperti *adventure*, *action*, RPG, dll. Salah satu *genre* yang cukup populer adalah *genre fighting* yang pertama kali ditemukan pada *game arcade* di Jepang bernama Heavyweight Champ di tahun 1976. Sejak saat itu, *game fighting* yang semula hanya pada *arcade*, kemudian berkembang ke berbagai *platform* seperti pada *console*, komputer, dan telepon genggam. Sejauh ini, *game fighting* yang ada tidak menampilkan cerita yang bersambung pada tiap *stage*, seperti pada *game adventure*.

Berdasarkan pemaparan di atas, penulis tertarik untuk mengembangkan *game fighting* dengan judul The Black Blood. *Game* ini adalah *game* 3D yang

menggunakan metode *collision detection* sebagai kecerdasan buatanya.

1.1 Batasan Masalah

Agar pembahasan tidak meluas, maka ditentukan beberapa batasan masalah, yaitu :

1. *Game* dibuat menggunakan Unity 5.
2. Grafik dalam *game* ini adalah 3D.
3. *Game* hanya dapat dimainkan pada PC atau berbasis *Desktop*.
4. *Game* ini dibuat untuk berjalan pada sistem operasi Windows.

1.2 Tujuan

Adapun tujuan dari pembuatan *game* ini, yaitu :

1. Menerapkan metode *Collision Detection* untuk menghasilkan suatu kecerdasan buatan pada *game* The Black Blood.
2. Untuk membuat pemain menikmati alur cerita dan memainkan *game* ini.

2. TINJAUAN PUSTAKA

2.1 Penelitian Sebelumnya

Dewasa ini telah banyak *game fighting* dengan berbagai macam kecerdasan buatan yang diterapkan. Sato, N, dkk. melakukan penelitian tentang pemain AI yang didesain untuk memiliki dua desain sekaligus yaitu berbasis *rule-based* dan juga *online machine learning*. Metode ini menggabungkan kedua desain itu kemudian

pada selang waktu tertentu akan saling dialihkan. Dengan metode ini, musuh AI berusaha untuk bertindak lebih dominan terhadap pemain musuh saat itu. Untuk melakukan tes terhadap metode ini, digunakan *platform FightingICE*. Beberapa eksperimen menunjukkan keefektifan dari metode ini pada *fighting game*. Musuh AI terdiri dari 3 *rule-based computer player* dan metode yang diusulkan ini melampaui ketiga pemain tersebut. Metode ini juga sedikit meningkatkan performa dari pemain AI terhadap pemain yang menggunakan desain *online machine learning* (Sato, 2015). Yamamoto, K, dkk. melakukan penelitian tentang AI berbasis *rule-based* yang dapat memprediksi gerakan lawan selanjutnya dan melakukan serangan balik sesuai dengan yang telah diprediksi menggunakan algoritma *k-nearest neighbor*. Dengan menggunakan platform *FightingICE* untuk melakukan test, telah dibuktikan bahwa AI dengan metode yang diusulkan dapat melakukan serangan balik yang lebih unggul daripada serangan yang diterima (Yamamoto, 2014).

Bender, J dkk. melakukan penelitian tentang algoritma baru dimana dapat mendeteksi *collision detection* secara *real time*. (Bender, 2013). Hong, Weihu, dkk. melakukan penelitian tentang penghitungan *collision detection* secara matematis sehingga dapat memberikan hasil yang presisi dan solusi yang lebih efisien dalam *game*. Solusi matematis yang diusulkan adalah untuk beberapa bentuk dasar di *collision detection* pada *game* 2D maupun 3D. Metode yang diusulkan cukup mudah untuk diterapkan tanpa terdapat banyak masalah pada performa pada algoritma yang berbasis *per-pixel*, namun keakurasian dan presisi meningkat cukup signifikan jika dibandingkan dengan mekanisme *collision detection* berbasis *pixel* (Hong, 2013).

Dengan melihat dari penelitian yang sudah ada, maka penulis mencoba untuk membuat sebuah *fighting game* berbasis *rule-based* dengan menggunakan *collision detection* sebagai metodenya.

2.2 Fighting Game

Fighting game adalah *subgenre* dari *action game* yang mensimulasi pertarungan berhadapan dari dekat yang biasanya menggunakan teknik bela diri asia. Aksi dari pemain biasanya terdiri dari serangan berhadapan dan bertahan. Serangan yang sukses dari karakter (tidak terkena blok dari aksi bertahan) akan mengurangi energi dari *character hit* dan *game* akan terus berlanjut hingga salah satu energi dari karakter habis. Strategi dari *fighting game* menyerupai relasi yang terdiri dari tiga bagian seperti batu-gunting-kertas, sehingga pemain tidak dapat menggunakan aksi yang lebih dominan pada aksi lainnya berulang-ulang. Contohnya, tendangan lebih unggul terhadap pukulan, pukulan lebih unggul terhadap aksi bertahan, dan aksi bertahan lebih unggul dari tendangan di *game fighting*. (Adams, 2010).

2.2 Collision Detection

Collision detection adalah sebuah kecerdasan buatan yang mendeteksi adanya tumbukan dari dua

atau lebih objek. Tujuan dari *collision detection* adalah melaporkan secara otomatis kontak geometrik ketika sudah terjadi atau sebelum kontak tersebut terjadi. *Collision detection* adalah komponen yang penting di dalam *game*, karena dengan adanya *collision detection* dapat menjaga pemain dari berjalan melewati tembok atau jatuh dari lantai. *Collision detection* juga dapat memberitahu musuh jika mereka melihat ada pemain dan mereka dapat menyerang pemain. Teknik pendeteksian tumbukan bisa dikelompokkan menjadi dua macam yaitu *priori detection* dan *post detection*. *Priori detection* adalah pengecekan tumbukan sebelum tumbukan tersebut terjadi, sedangkan *post detection* adalah pengecekan tumbukan setelah tumbukan tersebut terjadi. Pada *fighting game*, pendeteksian yang digunakan adalah *post detection* karena tumbukan baru ditangani setelah tumbukan tersebut terjadi (Ericson, 2005).

2.3 Finite State Machines (FSM)

Finite State Machines (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 (misal interupsi *timer*). 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. Berdasarkan sifatnya, metode FSM ini sangat cocok digunakan sebagai basis perancangan perangkat lunak pengendalian yang bersifat reaktif dan *real time* (Setiawan, 2006).

Salah satu keuntungan nyata penggunaan FSM adalah kemampuannya dalam mendekomposisi aplikasi yang relatif besar dengan hanya menggunakan sejumlah kecil *item state*. Selain untuk bidang kontrol, 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 prosedural seperti bahasa C, FSM ini umumnya direalisasikan dengan menggunakan statemen kontrol *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).

2.4 Desain AI

Dari sekian banyak desain untuk AI yang ada, pada umumnya desain AI dibagi menjadi dua tipe

yaitu, *rule-based* dan *machine learning*. *Rule-based* adalah desain yang paling mudah dari semuanya. Desain ini menggunakan *if-else rule* dan bila kondisinya terpenuhi, maka AI akan mengeksekusi aksi yang sudah diterapkan di dalam sistem. Sedangkan AI yang menggunakan *machine learning* menyesuaikan aksinya sesuai dengan lingkungan dari Q-table atau jaringan syaraf tiruan. *Machine learning* dibagi lagi menjadi dua yaitu *offline learning* dan *online learning*. Pada *offline learning*, AI menggunakan data yang cukup banyak untuk memutuskan pola aksi dari AI, namun setelah itu polanya akan konsisten. Sedangkan untuk AI yang menggunakan *online learning*, akan berusaha untuk mendapatkan pola serangan dari lawan atau situasinya dalam permainan dan mempelajarinya (Saini, 2011).

3. METODE PENELITIAN

3.1 Analisis Game

Analisis yang dilakukan oleh penulis dalam proses pembuatan *game The Black Blood* dijabarkan dalam sub-bab berikut

3.1.1 Analisis Masalah

Dalam *fighting game*, pemain diharuskan untuk menyelesaikan semua *stage* yang ada dalam *game*. Setiap *stage* memiliki tingkat kesulitan tertentu. Di setiap *fighting game*, pemain diharuskan untuk melawan setiap musuh yang ada di setiap *stage* untuk dapat melanjutkan ke *stage* berikutnya. Bila pemain tidak berhasil mengalahkan musuh pada *stage* tertentu, maka pemain akan mengulangi *stage* tersebut dengan melawan musuh yang sama. Untuk mengalahkan musuh, pemain secara tidak langsung dituntut untuk mengasah ketangkasan dan strateginya.

Fighting game memiliki beberapa elemen yaitu pemain dan musuh AI. Musuh AI adalah karakter yang dikendalikan oleh komputer atau bisa disebut NPC (*Non-playable character*). Pembuatan AI dalam *fighting game* merupakan hal yang cukup rumit, dikarenakan pengembang harus menentukan serangan dari AI agar tidak monoton dan bervariasi. Desain AI yang paling sederhana adalah menggunakan *rule-based*, sehingga gerakan AI yang akan dieksekusi sudah diterapkan dalam sistem.

3.1.2 Pengenalan Game The Black Blood

Game yang akan dibangun adalah *game The Black Blood* dengan *genre fighting sideview game*. *Game* ini dibangun dengan mengaplikasikan metode *collision detection* sebagai kecerdasan bukannya. Fitur-fitur yang digunakan pada *game* ini adalah:

1. Menggunakan grafik 3D.
2. Metode yang digunakan adalah *Collision Detection*.
3. *Game* ini bergenre *fighting*

4. *Collision Detection* terletak pada karakter utama dan musuh, jika terdeteksi tumbukan antara karakter utama dan musuh, maka *health bar* akan berkurang.

Game The Black Blood ini menceritakan tentang seorang laki – laki bernama Rothern yang tiba – tiba berada di dimensi lain dari dunianya dimana di dalam dunia itu semua tempat tidak berpenghuni. Kota tempat ia tinggal di dunia nyata juga terlihat seperti kota mati. Dalam perjalanannya untuk mencoba kembali ke dimensinya, ia bertemu dengan musuh yang harus dihadapinya agar ia dapat kembali ke dimensinya lagi.

3.2 Perancangan Sistem

Perancangan sistem adalah suatu bagian dari metodologi pengembangan suatu perangkat lunak yang dilakukan untuk memberikan gambaran secara terperinci tentang *Game The Black Blood*.

3.2.1 Perancangan Karakter

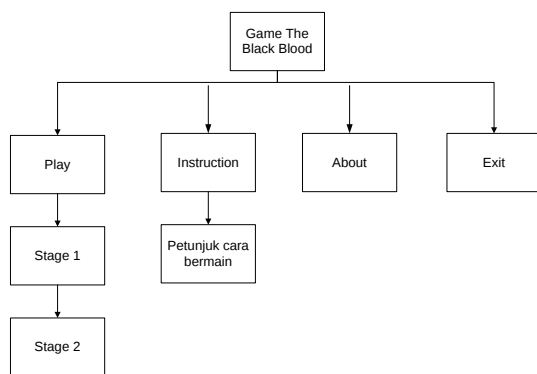
Perancangan karakter pada *Game The Black Blood* dapat dilihat pada Tabel 1.

Tabel 1. Perancangan Karakter

No	Gambar	Keterangan
1		- Karakter Utama, Rothern - Muncul pada setiap <i>stage</i>
2		- Musuh 1, Ayam - Muncul pada <i>stage</i> 1 awal dan <i>stage</i> 1 <i>fighting</i>
3		- Musuh 2, Babi - Muncul pada <i>stage</i> 2 awal dan <i>stage</i> 2 <i>fighting</i>

3.2.2 Perancangan Struktur Menu

Perancangan struktur menu adalah perancangan tata urutan menu dari *Game The Black Blood* seperti yang terlihat pada Gambar 1.

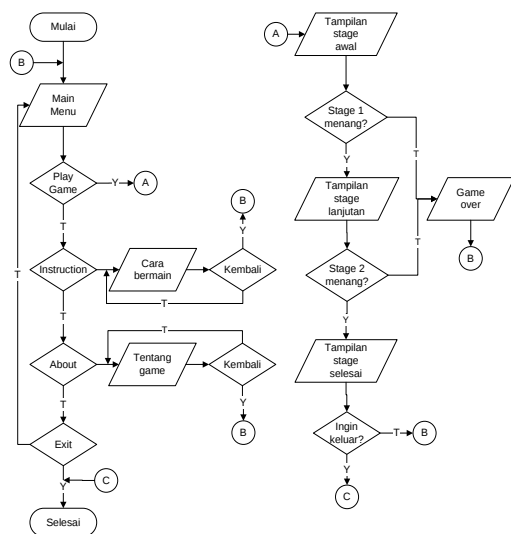


Gambar 1. Perancangan struktur menu

Pada *game The Black Blood* terdapat 4 menu utama yaitu *Play*, *Instruction*, *About* dan *Exit*. Untuk memulai *game*, pemain memilih menu *Play*. Pada menu *Instruction*, akan ditampilkan tombol – tombol yang digunakan dalam *game* ini dan beberapa tips. Menu *About* berisi tentang pembuat *game* dan menu *Exit* untuk keluar dari *game*.

3.2.3 Perancangan Alur Game

Perancangan alur *game* berfungsi untuk mengetahui alur proses dari alur program dimulai dari awal program hingga selesai seperti yang ditunjukkan pada Gambar 2.



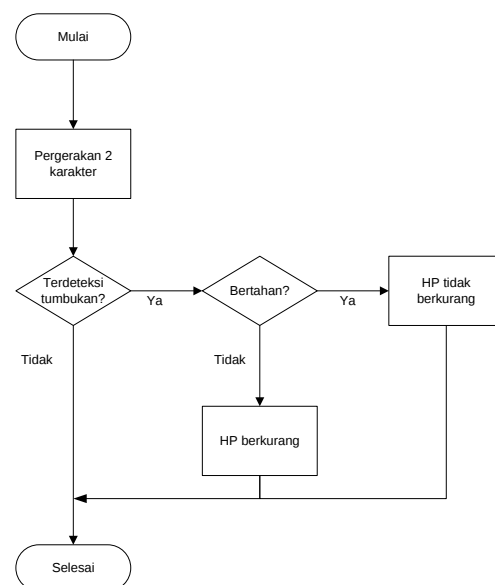
Gambar 2. Alur game *The Black Blood*

Pada Gambar 2., *game* dimulai dari *Mulai* kemudian masuk ke 4 menu utama yaitu *Play Game*, *Instruction*, *About*, dan *Exit*. Jika pemain memilih *Play Game* maka akan langsung menuju *stage 1*, dan jika selesai akan melanjutkan ke *stage 2*. Jika pemain kalah, akan di arahkan ke tampilan *Game Over* dan menuju ke *Main Menu*. Jika pemain memilih menu *Instruction*, maka pemain akan diarahkan ke tampilan cara bermain dan jika memilih kembali maka akan kembali ke tampilan menu utama. Jika pemain memilih menu *About*

maka akan diarahkan ke tampilan tentang *game* dan jika memilih kembali maka akan kembali ke tampilan menu utama. Jika pemain memilih menu *Exit*, maka *game* akan ditutup atau berakhir.

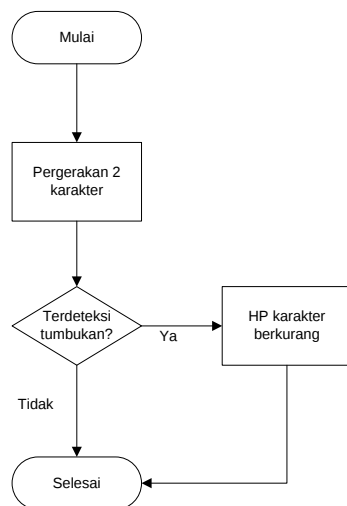
3.2.4 Alur Collision Detection

Collision Detection merupakan salah satu logika penalaran yang merupakan bagian dari *finite state machine*. Logika ini memperlihatkan perilaku system dengan berdasarkan tiga hal, yaitu *state* (keadaan), *event* (kejadian), dan *action* (aksi). Pada suatu saat, sistem akan berada pada salah satu *state* yang aktif. Sistem dapat beralih atau bertransisi menuju *state* lain jika mendapatkan masukan atau *event* tertentu. Transisi keadaan ini umumnya juga disertai oleh aksi yang dilakukan oleh sistem ketika menanggapi masukan yang terjadi. Pada *game The Black Blood* ini, metode *collision detection* diterapkan pada karakter utama dan musuh. Ketiga musuh ini memiliki alur *collision detection* yang sama.



Gambar 3. Alur diagram *collision detection* pada karakter musuh

Pada Gambar 3. memperlihatkan alur *collision detection* pada karakter musuh. Jika karakter musuh mendeteksi adanya tumbukan maka *health bar* musuh akan berkurang sesuai dengan serangan yang diterima atau musuh akan berada pada posisi bertahan yang membuat *health bar* musuh tidak akan berkurang meskipun mendapatkan serangan.

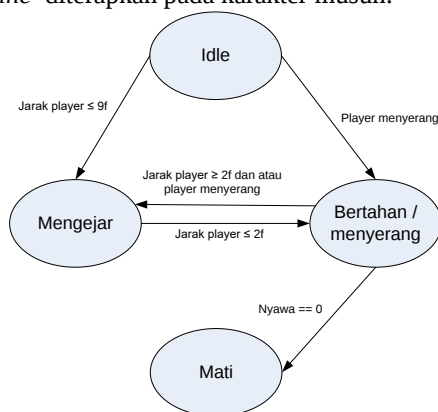


Gambar 4. Alur diagram *collision detection* pada karakter utama

Pada karakter utama, diterapkan *collision detection* yang sama, hanya saja untuk *action* dari karakter setelah menerima serangan ditentukan oleh pemain sendiri. Namun jika karakter menerima serangan dan pemain tidak dalam posisi bertahan, maka *health bar* dari karakter akan berkurang.

3.2.5 Perancangan Alur FSM (*Finite State Machines*)

Finite State Machine merupakan salah satu logika penalaran yang memperlihatkan perilaku system dengan berdasarkan tiga hal, yaitu *state* (keadaan), *event* (kejadian), dan *action* (aksi). Pada suatu saat, sistem akan berada pada salah satu *state* yang aktif. Sistem dapat beralih atau bertransisi menuju *state* lain jika mendapatkan masukan atau *event* tertentu. Transisi keadaan ini umumnya juga disertai oleh aksi yang dilakukan oleh sistem ketika menanggapi masukan yang terjadi. Pada game *The Black Blood* ini, metode *Finite State Machine* diterapkan pada karakter musuh.



Gambar 5. Perancangan alur diagram FSM pada musuh

Pada Gambar 5. memperlihatkan alur FSM (*Finite State Machine*) pada karakter musuh. Jika karakter musuh mendeteksi jarak musuh sebesar 9f, maka musuh akan mulai mengejar pemain. Kemudian jika jarak antara pemain dan musuh kurang dari atau sama dengan 2f, maka musuh akan mulai menyerang pemain. Jika pemain menyerang musuh, maka musuh akan bertahan atau menyerang pemain. Musuh akan mati jika nyawa musuh sama dengan 0.

4. HASIL DAN PEMBAHASAN

4.1 Tampilan Menu *Game*

Tampilan menu utama adalah tampilan awal yang muncul saat membuka aplikasi *Game The Black Blood*. Pada tampilan awal ini berisi tombol *Play* untuk langsung bermain, *Instruction* untuk melihat cara bermain, *About* untuk melihat tentang game ini, dan *Exit* untuk keluar dari game. Tampilan menu utama terlihat seperti pada Gambar 6. berikut:



Gambar 6. Tampilan game

4.2 Tampilan Menu *Instruction*

Pada menu *instruction* merupakan menu yang berfungsi untuk menampilkan informasi tentang bagaimana cara bermain *Game The Black Blood*. Tampilan menu *instruction* terlihat seperti Gambar 7. berikut



Gambar 7. Tampilan menu *instruction*

4.3 Tampilan Menu *About*

Tampilan menu *About* adalah tampilan menu yang berisi informasi tentang game *The Black Blood*, tampilan dari menu *About* dapat dilihat pada Gambar 8. berikut:



Gambar 8. Tampilan menu about

4.4 Pengujian Gameplay

Pengujian *gameplay* adalah pengujian bagaimana *game* tersebut berjalan sesuai dengan rancangan sistem yang telah dibuat. Tampilan pengujian *game* menggambarkan interaksi antara karakter dengan lingkungan di dalam *game*.



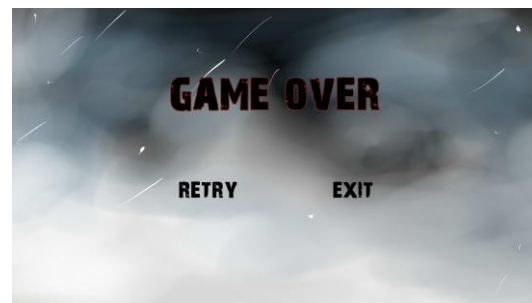
Gambar 9. Tampilan ketika pemain bertemu Ayam di stage 1 awal

Pada Gambar 9. ketika pemain bertemu musuh Ayam di *stage* 1 awal. Pada *stage* ini, musuh Ayam tidak akan menyerang pemain. Musuh hanya melakukan patrol dari titik-titik yang sudah ditentukan. Pemain harus mendekati Ayam agar dapat melanjutkan ke *stage fighting*.



Gambar 10. Pemain diserang oleh musuh

Pada Gambar 8 menampilkan ketika pemain diserang oleh musuh dan *health point* dari pemain berkurang. Jika *health bar* dari pemain habis, maka tampilan akan berganti ke tampilan *game over*. Hal ini mengindikasikan bahwa penerapan *collision detection* pada karakter sudah sesuai.



Gambar 11. Tampilan game over

4.5 Pengujian Artificial Intelligence (AI)

Pengujian AI adalah pengujian mengenai fungsi yang berkaitan dengan AI (*artificial intelligence*) yang ada dalam *game* The Black Blood. Hasil pengujian dari AI *finite state machine* dan *collision detection* dapat dilihat pada Tabel 2.

Tabel 2. Pengujian AI (Artificial Intelligence)

No	Fungsi	Output	Hasil
1	AI <i>Finite State Machine</i> pada musuh	Musuh berjalan mendekati <i>player</i> jika jarak <i>player</i> kurang dari 9 unit.	Sesuai
2	AI <i>Finite State Machine</i> pada musuh	Musuh akan mulai menyerang <i>player</i> jika jarak <i>player</i> kurang dari 2 unit.	Sesuai
3	AI <i>Collision Detection</i> pada musuh	Nyawa musuh akan berkurang ketika terdeteksi tumbukan dari <i>player</i> .	Sesuai

Berdasarkan Tabel 2., AI yang ada pada musuh sudah berfungsi dengan baik. Ketika pemain mendekati musuh, musuh bisa mengejar dan bila pemain semakin dekat, musuh bisa menyerang pemain. *Health bar* dari musuh juga sudah bisa berkurang ketika musuh menerima serangan dari pemain. Dengan demikian dapat disimpulkan bahwa semua AI (*Artificial Intelligence*) yang ada dalam *Game* The Black Blood berjalan dengan tingkat keberhasilan 100% sesuai dengan yang diharapkan.

4.6 Pengujian Fungsional

Pengujian fungsional adalah pengujian mengenai proses fungsional yang terjadi dalam *game*. Hasil dari pengujian dapat dilihat pada Tabel 3.

Tabel 3. Pengujian Fungsional

No.	Keterangan	Hasil	
		Stage 1	Stage 2
1.	Player dapat berjalan, meloncat, dan menyerang	Sesuai	Sesuai
2.	Musuh dapat menyerang player	Sesuai	Sesuai
3.	Musuh dapat mengejar player	Sesuai	Sesuai
4.	Health bar player berkurang ketika diserang	Sesuai	Sesuai
5.	Health bar musuh berkurang ketika diserang	Sesuai	Sesuai
6.	Player dapat mati ketika health bar = 0	Sesuai	Sesuai
7.	Musuh dapat mati ketika health bar = 0	Sesuai	Sesuai
8.	Tombol pergerakan player berjalan sesuai dengan yang ditentukan	Sesuai	Sesuai
9.	Timer berfungsi dengan benar	Sesuai	Sesuai

Berdasarkan Tabel 3, pengujian yang dilakukan pada karakter utama dan juga musuh. Pemain sudah dapat menggerakkan karakter utama menggunakan tombol yang telah diatur. Pemain juga dapat melakukan serangan terhadap musuh. Health bar dari pemain maupun musuh juga sudah berkurang sesuai dengan damage yang diterima. Ketika health bar sudah habis maka musuh maupun pemain akan mati. Timer di game ini juga sudah berfungsi dengan baik. Oleh karena itu, dapat disimpulkan bahwa semua fungsi berjalan dengan tingkat keberhasilan 100% sesuai dengan yang diharapkan.

4.4 Pengujian Control Player

Pengujian control player adalah pengujian fungsi dari setiap tombol yang sudah diterapkan untuk menggerakkan karakter utama. Hasil pengujian control player dapat dilihat pada Tabel 4.

Tabel 4. Pengujian Control Player

Tombol	Fungsi	Hasil
W	Menggerakkan player ke depan	Sesuai
A	Menggerakkan player ke kiri	Sesuai
S	Menggerakkan player ke belakang	Sesuai
D	Menggerakkan player ke kanan	Sesuai

Space	Menggerakkan player ke atas (loncat)	Sesuai
J	Player meninju	Sesuai
L	Player menendang	Sesuai
K	Player bertahan	Sesuai
O	Player mengeluarkan jurus spesial	Sesuai

Dari Tabel 4 ketika pemain menekan tombol W, A, S, D karakter dapat bergerak sesuai dengan arahnya yaitu, W untuk maju, A untuk bergerak ke arah kiri, S untuk bergerak mundur, dan D untuk bergerak ke arah kanan. Ketika pemain menekan tombol spasi karakter juga dapat meloncat. Ketika pemain menekan tombol L, J, K, dan O, karakter dapat melakukan serangan sesuai dengan yang telah diberikan. Tombol J untuk meninju, tombol L untuk menendang, tombol K untuk bertahan, dan tombol O untuk mengeluarkan jurus spesial. Dari penjelasan di atas menunjukkan bahwa semua fungsi dari control player berjalan dengan tingkat keberhasilan 100% sesuai dengan yang diharapkan.

4.4 Pengujian User

Pengujian dilakukan untuk mengetahui apakah sistem sudah berjalan dengan baik atau belum. Pengujian dilakukan terhadap 10 responden yang terdiri dari 2 orang siswa sekolah dasar dan 3 orang siswa sekolah menengah atas serta 5 orang mahasiswa. Kuisisioner berisi lima pertanyaan tentang game The Black Blood. Hasil dari pertanyaan terhadap responden dapat dilihat pada Tabel 5.

Tabel 1. Pengujian Terhadap User

Pertanyaan	Penilaian				
	1	2	3	4	5
Bagaimana tampilan desain karakter pada game?	0	0	3	7	0
Bagaimana tampilan desain animasi pada game?	0	0	1	9	0
Bagaimana tampilan menu pada game?	0	0	2	8	0
Bagaimana kontrol player saat memainkan game?	0	1	5	4	0
Bagaimana gerakan musuh saat melawan player?	0	0	5	5	0

Keterangan : 1 = Sangat Kurang, 2 = Kurang, 3 = Cukup, 4 = Baik, 5 = Sangat Baik.

Berdasarkan Tabel 4.5, didapatkan presentase hasil dari pengujian *user* sebagai berikut,

Sangat kurang = 0 %, Kurang = 2 %, Cukup = 32 %, Baik = 66 %, dan Sangat Baik = 0%.

Oleh karena itu, berdasarkan pengujian terhadap 10 *user* terhadap *game* The Black Blood ini menunjukkan hasil baik dengan presentase 66%.

5. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Setelah pembuatan *game* The Black Blood, maka penulis dapat mengambil kesimpulan :

1. Metode *collision detection* dapat diterapkan pada *game* The Black Blood dengan indikasi musuh dapat menyerang *player* dengan kondisi tertentu.
2. Semua fungsi dari menu, pergerakan unit pemain dan musuh berjalan dengan tingkat keberhasilan 100% sesuai dengan yang telah dirancang.
3. Pengujian terhadap 10 *user* menunjukkan hasil yang baik dengan presentase 66%.

5.2 Saran

Setelah dilakukan pengujian terhadap *game* The Black Blood maka masih ada kekurangan sehingga untuk pengembangan lebih lanjut disarankan:

1. Penggunaan metode lain pada *game* The Black Blood untuk membuat AI yang lebih pintar lagi seperti algoritma *fuzzy*.
2. Penggunaan desain *machine learning* untuk membuat AI yang memiliki pola serangan yang lebih dinamis lagi.
3. Penambahan gerakan *combo* pada *player* dan juga AI agar *gameplay* menjadi lebih menarik.
4. Menambahkan fitur *save game* agar pemain bisa melanjutkan *game* nya lagi ketika *game* dikeluarkan.

DAFTAR PUSTAKA

- [1] Adams, Ernest, 2010, *Fundamental of Game Design, Second Edition*, New Riders, Bekeley, CA
- [2] Bender, J., Dequidt, J., Duriez, C. and Zachmann, G., 2013. Parallel collision detection in constant time.
- [3] Bourg, D. M. 2002. *Physics for game developers*. "O'Reilly Media, Inc.". Sebastopol, CA.
- [4] Ericson, C., 2004. *Real-time collision detection*. CRC Press, San Fransisco, CA.
- [5] Harper, Todd, 2014, *The Culture of Digital Fighting Games Performance and Practice*, Routledge, New York.
- [6] Hong, W., Qu, J. and Wu, M., 2012, January. Mathematical Approaches for Collision Detection in Fundamental Game Objects. In *Proceedings of the International Conference on Foundations of Computer Science (FCS)* (p. 1). The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp).
- [7] Saini, S.S., Dawson, C.W. and Chung, P.W., 2011, November. Mimicking player strategies in fighting games. In *2011 IEEE International Games Innovation Conference (IGIC)* (pp. 44-47). IEEE.
- [8] Sato, N., Tamsiririrkul, S., Sone, S. and Ikeda, K., 2015, July. Adaptive Fighting Game Computer Player by Switching Multiple Rule-based Controllers. In *Applied Computing and Information Technology/2nd International Conference on Computational Science and Intelligence (ACIT-CSI), 2015 3rd International Conference on* (pp. 52-59). IEEE.
- [9] Yamamoto, K., Mizuno, S., Chu, C.Y. and Thawonmas, R., 2014, August. Deduction of fighting-game countermeasures using the k-nearest neighbor algorithm and a game simulator. In *2014 IEEE Conference on Computational Intelligence and Games* (pp. 1-5). IEEE.