Perilaku Raja Virus Dalam Game Doctor VS Virus Menggunakan Metode *Finite State Machine*

Virus King Behavior Design In Doctor-VS-Virus Game Using Finite State Machine

1st Alvian Arisandi
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
alvianarisandi@student.telkomuniversity.ac.id

2nd Purba Daru Kusuma
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
purbodaru@telkomuniversity.ac.id

3rd Ashri Dinimaharawati
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
ashridini@telkomuniversity.ac.id

Abstrak—*Game* merupakan suatu bentuk media hiburan yang dapat digunakan juga sebagai media pembelajaran, *game* dapat dimanfaat sebagai media pembelajaran dan juga sebagai media hiburan guna mencegah kebosanan saat belajar, *game* juga dapat digunakan sebagai sarana beriteranksi dan bersosialisasi dengan lingkungan sekitarnya. Salah satu metode pembelajaran adalah berbasis *game* oleh karena itu penulis membuat *game* edukasi terkait konsep diri. *Game "The Doctor vs Virus"* dibuat untuk menjawab persoalan tersebut, *game "The Doctor vs Virus"* ini dibuat dengan menggunakan metode FSM atau *Finite State Machine*. FSM adalah perancangan sistem kontrol yang menggambarkan tingkah laku NPC terhadap lingkungan sekitar dengan mengunakan *state* atau keadaan dan juga transisi sebagai penghubung. Hasil Penelitian ini memberikan ketertarikan kepada pemain, memberikan pemahaan kepada pemain terhadap virus, dan NPC Boss Virus dapat diprogram menggunakan metode *Finite State Machine* berdasarkan. Dengan menggunakan *Finite State Machine* memberikan tingkat kesulitan yang menarik dab Menggunakan Relabilitas hasil kuesioner menunjukan interpretasi sangat tinggi dengan jumlah responden sebanyak tujuh puluh responden.

Kata Kunci— *game, finite state machine, side scrolling game.*

I. PENDAHULUAN

Berkembangnya teknologi membuat manusia semakin mudah untuk mengakses pembelajaran di mana saja, kebutuhan manusia akan Pendidikan juga sangatlah penting untuk mencerdaskan dan mengembangkan potensi individu seperti dapat memiliki pengetahuan yang luas kepribadian yang baik dan menjadi pribadi yang bertanggung jawab dan juga memiliki kreativitas yang tinggi. Salah satu metode pembelajaran adalah berbasis *game*, *game* dapat digunakan sebagai media hiburan dan pembelajaran di era yang serba digital ini. Terdapat beberapa jenis *game* diantaranya adalah *single player* yang berarti hanya dapat dimainkan oleh satu orang saja dan juga *multi player* yang berarti dapat dimainkan oleh beberapa orang. Pada *game single player* biasanya user dihadapkan oleh beberapa *Non-Playable Character*

Non-Playable Character merupakan obyek pada *game* yang dapat berupa karakter seperti hewan, robot, tumbuhan dan lain lain yang tidak dikendalikan oleh *player* [1]. Tingkah laku NPC ditentukan oleh *Artificial Intelegent* yang dapat berupa algoritma seperti Algoritma A-Star, *Intelegent Agent*, *Multi Agent* dan *Finite state machine* yang membuat suasana dalam game lebih menantang dan menjadi lebih seru.

Finite state machine adalah salah satu algoritma atau kecerdasan buatan yang dapat diimplementasikan untuk memodelkan suatu perilaku dari NPC berdasarkan keadaan (*state*) yang dapat menentukan aksi NPC atas setiap respon gerakan yang dilakukan oleh player dalam game [2]. Berdasarkan latar belakang tersebut maka penelitian yang dilakukan adalah Perancangan Perilaku Bos Virus menggunakan metode *Finite State Machine.*

II. KAJIAN TEORI

A. Konsep Diri

Konsep diri adalah pandangan seorang atau individu terhadap dirinya sendiri baik berupa penilaian dan juga hasil interaksi dari lingkungan sekitar tentang diri individu itu sendiri [3]. Pembentukan konsep diri juga dipengaruhi oleh lingkungan sekitar seperti jika dia tinggal di kota cita cita seseorang dapat berbeda seperti menjadi Guru, Dokter, Pilot. Pandangan diri sendiri dapat meliputi aspek fisik maupun psikis contohnya pada fisik yaitu mengenali sejauh mana kemampuan dia berlari dan untuk psikis adalah sadar akan tingkah laku bahwa dia salah [4].

B. Game

Game adalah permainan yang digunakan menggunakan media elektronik seperti *handphone*, ataupun *tablet* yang dibuat semenarik mungkin bagi para pemain untuk mencapai kepuasan tertentu. Bermain *game* merupakan salah satu sarana yang baik untuk belajar. *Game*

sering dimainkan oleh anak-anak, remaja hingga orang dewasa. Jenis permainan juga sangat berpengaruh contohnya anak muda dan remaja lebih senang memaikan game dengan genre *Action* dan anak anak lebih senang game dengan genre *Arcade*. Dalam grafik *game* dapat dibagi menjadi dua kategori yaitu *game* 2D (dua dimensi) dan 3D (tiga dimensi). *Game* tidak hanya sekedar permainan tapi juga dapat digunakan sebagai media untuk menguji ketangkasan, kecepatan reaksi, ataupun kecerdasan berpikir pemainnya [5].

Berikut merupakan beberapa contoh genre pada *game*:

1. *Side-scorlling Games*

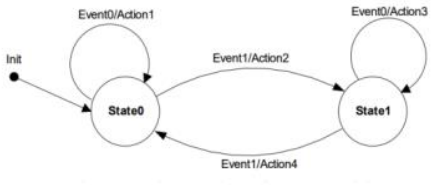
Game side scrolling adalah salah satu genre *game* dengan sudut pandang kamera yang mengikuti karakter atau *player* dan umumnya bergerak dari kiri menuju sisi kanan atau dari bawah menuju atas untuk mencapai goals atau tujuan dari tempat tempat yang sudah ditentukan, [6].

2. *Platformer*

Video *game* platformer biasanya hanya perlu menggerakkan player dari platform satu ke platform lainnya untuk mencapai tujuan dari game tersebu. Game platform biasanya berbentuk visual secara 2D ataupun 3D, platformer juga dikombinasikan dengan genre *side scrolling* seperti game Metal Slug, dan juga Mario Bross [7].

C. Finite State Machine

Finite State Machines (FSM) adalah sebuah algoritma yang menggambarkan tingkah laku atau prinsip kerja sistem yang biasanya menggunakan tiga buah hal yaitu *State* (Keadaan), *Event* (kejadian) dan *action* (aksi) [8]. Sistem beralih menuju state lain jika terjadi *input* atau *event* dari *user* baik yang berasal dari perangkat maupun keadaan. *Finite State Machine* dapat membantu dalam merancang *loop mechanic game* baik itu *game* digital maupun board seperti *game* catur maka dari itu *finite state machine* banyak digunakan pada perangkat lunak khususnya pada *game* [9]. Penerapan FSM dapat dibagi menjadi tiga cara, di mana masing-masing metode memiliki kelebihan dan kekurangannya masing-masing. Ketiga metode tersebut adalah Cara Tradisional, Tabel Pencarian, dan Berorientasi Objek [10]. *state* terhubung ke setidaknya satu *state* bagian lain sehingga dari satu *state* ke *state* lainnya dapat terhubung. untuk berpindah dari satu *state* ke *state* lain membutuhkan jalur dan itu dikenal sebagai transisi atau *event event* dari satu keadaan ke keadaan lain terjadi jika kondisi tertentu terpenuhi seperti diperngaruhi oleh *player* atau pun environment sekitar [2].



GAMBAR 2. 1
FINITE STATE MACHINE

D. Pengukuran R reliabilitas

Metode yang paling umum digunakan untuk mengukur relevansi pertanyaan adalah metode korelasi faktor produk.

$$r = \frac{N\sum XY - \sum X \sum Y}{\sqrt{[N\sum X2 - (\sum X\,2)][N\sum Y\,2 - (\sum Y\,2)]}} \quad (2.1)$$

Di mana:

r : Koefisien Korelasi Pearson
N : Jumlah Responden
X : Skor pertanyaan per responden
Y : Skor total per responden

Dengan melihat perbandingan dari nilai r harus lebih besar daripada r_{tabel} maka akan dinyatakan valid sebaliknya jika nilai r lebih rendah dari r_{tabel} maka akan dinyatakan tidak valid r_{tabel} dapat dilihat pada lampiran F.

Uji reliabilitas adalah uji untuk memastikan apakah kuesioner penelitian yang akan dipergunakan untuk mengumpulkan data variable penelitian reliable atau tidak.

$$r_{11} = \left[\frac{k}{k-1}\right]\left[1 - \frac{\sum si}{st}\right] \quad (2.2)$$

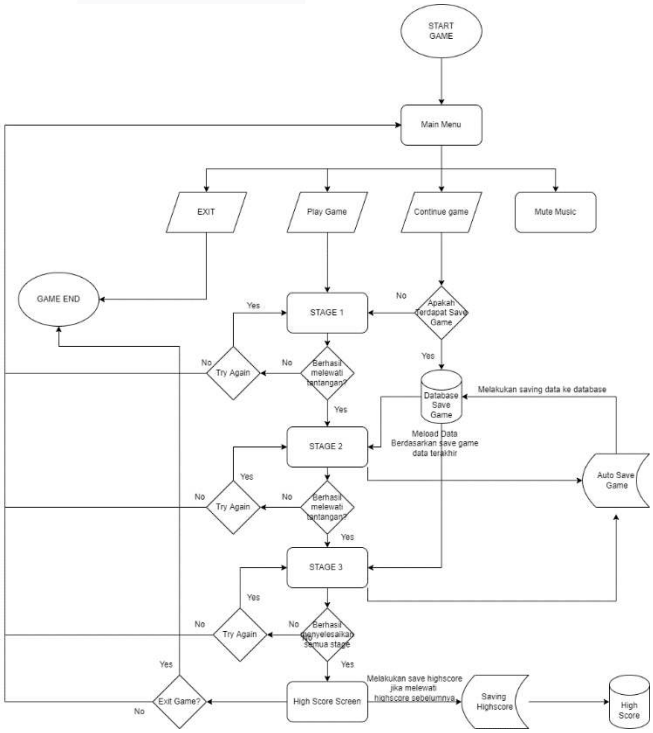
Di mana :

r_{11} : Koefisien realibiltas *alpha*
K : Jumlah pertanyaan
Si : Varian nilai setiap pertanyaan
st : Varian total

Hasil r_{11} atau reabilitas kemudian akan dibandingkan kembali dengan nilai rtabel product moment dengan nilai signifikan 5%. Jika $r_{11} > r_{tabel}$, maka suatu pertanyaan dapat dianggap reliable.

III. METODE

A. Perancangan Sistem

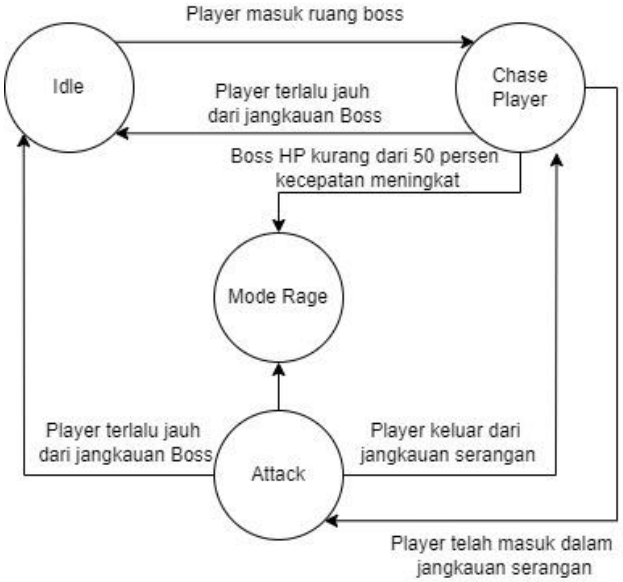


GAMBAR 3. 1
FLOWCHART

1. Pemain menjalankan aplikasi *game* untuk memulai *game*.
2. Setelah berhasil menjalankan game player akan berada pada Main menu, main menu terdiri dari beberapa pilihan yaitu *Play Game, Continue Game, Mute dan Exit*.
3. Menu *Mute* berguna untuk mematikan musik pada seluruh room.
4. Menu *Exit* digunakan sebagai tombol untuk keluar game.
5. Menu *Play game* merupakan menu untuk memulai *game* yang akan dimulai dari *stage* 1 hingga *stage* 3.
6. Pada saat *game* dimulai, *player* dapat menyelesaikan *game* dengan cara melewati tantangan hingga akhir *stage*.
7. Jika *player* mati saat tantangan maka *player* harus memulai kembali sesusai dengan *stage* di mana *player* mati.
8. Jika pemain sudah menyelesaikan seluruh *stage*, maka akan terdapat menu *Highscore* di mana total koins *player* akan digunakan sebagai *highscore*, jika total koins *player* tersebut melebihi *highscore* sebelumnya.
9. *Highscore* tersebut akan disimpan dalam localdisk.
10. Jika *player* keluar ditengah permainan makan sistem akan menyimpan stage di mana pemain tersebut.
11. Pada menu *Continue*, merupakan fungsi di mana *player* akan menuju *stage* terakhir di mana player berada.

B. Perancangan NPC Bos Virus

Boss virus memiliki 4 state yaitu *idle*, *chase player*, *attack*, dan *Rage mode*



GAMBAR 3. 2
FINITE STATE MACHINE BOS VIRUS

Pada saat player memasuki ruang boss, *player* akan *men-trigger* sebuah *event* yang mengakibatkan pintu pada ruang boss tertutup dan me-spawn boss Berikut merupakan penjelasan:

1. *State* awal dari boss virus adalah *idle*
2. Pada saat *player* mendekat ke boss virus tersebut maka boss virus akan mengejar sesuai dengan deteksi range dari boss, boss virus yang awal mulainya diam menjadi mengejar player
3. Saat *player* berada dalam jarak tembak boss virus maka boss virus yang awalnya mengejar *player* berubah *state* menjadi menembak player dan apabila terkena tembakan dari boss virus maka *health* pada player akan berkurang.
4. Pada saat *player* berada diluar jarak tembak boss virus maka boss virus yang awalnya menembak player berubah menjadi mengejar player
5. Pada saat player berada di luar deteksi range dari boss virus maka boss virus tersebut yang awalnya mengejar menjadi idle atau terdiam
6. Pada saat darah boss viruss kurang dari 50 persen maka boss tersebut akan berubah *state* menjadi *rage mode* di mana serangan dan kecepatan boss virus berubah secara signifikan

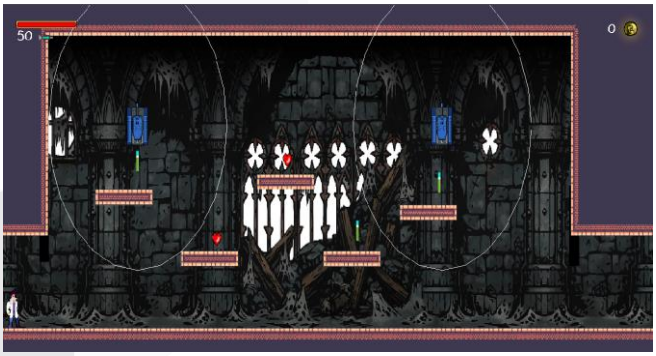
IV. HASIL DAN PEMBAHASAN

A. Implementasi Game

Pada BAB ini dilakukan implementasi game yang sudah dirancang, yang meliputi sebagai berikut:

1. Tampilan *Stage* Bos

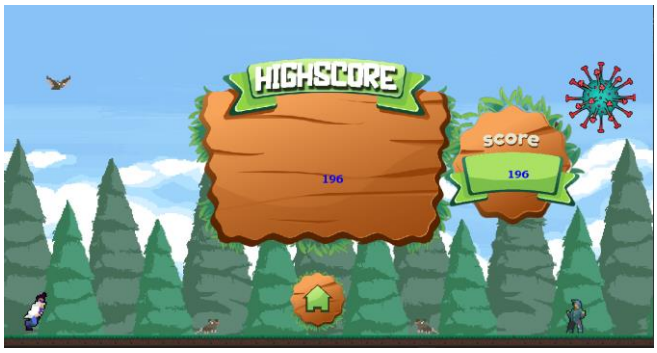
Setelah mencapai stage 3 maka stage terakhir yaitu melawan boss virus di dalam ruangan yang tertutup, player akan melawan bos virus sebanyak 2 buah dan juga turret sebanyak 2 buah jika sudah selesai mengalahkan bos virus dan turret maka pintu akan terbuka untuk menuju menu highscore.



GAMBAR 4. 1
STAGE BOS VIRUS

2. Tampilan *High score*

Menu highscore apabila player dapat menyelesaikan seluruh stage, Highscore diambil dari banyaknya koin dan NPC yang telah dikalahkan oleh player dan terdapat pilihan untuk ke menu utama.



GAMBAR 4. 2
HIGH SCORE

B. Implementasi NPC bos virus

Boss virus memiliki 4 buah *state* yang mengatur kondisi pada bos virus yaitu *idle*, *chase*, *attack, dan rage* mode.

```
enum states {
    idle,
    chase,
    shoot,
    rage
}
```

GAMBAR 4. 3
STATE BOS VIRUS

Boss virus untuk berpindah state dari state satu ke state lainya jika mendapatkan input yang ditentukan. Berikut merupakan input dari bos virus yang telah dirancang sebagai berikut.

```
state = states.idle; // state pertama kali
actionDur = 0; // deklarasi waktu delay antar tembakan
speed = 0.5 // deklarasi speed awal boss virus
distanceTrigger = 200; // jarak attack boss virus
distanceIdle = 500; // jarak chase boss virus
```

GAMBAR 4. 4
INPUT BOS VIRUS

Boss virus terdapat pada ruang boss dan memiliki *trigger* tersendiri yang dapat me-*spawn* boss virus dan juga menutup pintu pada ruang boss.

```
if (triggered == false){
    with(oPintu2) closed = true;
    triggered = true;
```

GAMBAR 4. 5
TRIGGER

Ketika bos virus sudah dikalahkan maka pintu akan terbuka kembali dan *player* dapat menuju *stage* berikutnya.

```
if (remaining[current_wave]<= 0)
{
    if (current_wave == total_waves)
    {
        with(oPintu2) closed = false;
        instance_destroy();
    }
```

GAMBAR 4. 6
BERHASIL MENGALAHKAN BOS VIRUS

Idle adalah ketika player berada di luar jangkauan boss virus speed akan menjadi 0 dan boss virus tidak akan bisa berjalan. Dapat berpindah state menjadi chase player saat bos virus berada pada jarak chase pada player

```
case states.idle:
    speed = 0;
    actionDur = 0;

    if(distance_to_object(oPlayer) < distanceIdle){
        state = states.chase;
    }
```

GAMBAR 4. 7
STATE IDLE DAN *CHASE*

Jika jarak player lebih kecil atau dalam jangkauan boss virus maka akan merubah *state* menjadi *chase*.

```
case states.chase:
    speed = 2;
    direction = point_direction(x+50,y+50,oPlayer.x,oPlayer.y);

    if(distance_to_object(oPlayer) >= distanceIdle){
        state = states.idle;
    }

    if(distance_to_object(oPlayer) <=distanceTrigger){
        actionDur ++;
    if(actionDur >= 10){
        state = states.shoot;
        actionDur = 0;
    }}
```

GAMBAR 4. 8
CHASE

Jika jarak antara distance idle bos virus ke player lebih besar jarak ke *player* maka *state* akan Kembali menuju *state idle*. Untuk menuju state attack jarak antara boss virus *distancetrigger* harus lebih besar dibandingkan jarak ke *player* maka *state* akan berubah menjadi *attack* atau *shoot*

```
case states.shoot:
    if(actionDur == 0){
        var dir = point_direction(x,y,oPlayer.x,oPlayer.y);
```

GAMBAR 4. 9
SHOOT

Terdapat *delay* antara jarak tembakan dan juga ketika berada pada state shoot boss virus akan menembakkan virus sesuai dengan posisi terakhir player berada seperti pada gambar

GAMBAR 4. 10
IMPLEMENTASI *FINITE STATE MACHINE ATTACK*

Pada gambar terlihat lingkaran pada NPC bos virus dengan 2 warna yang berbeda yaitu:

1. Lingkaran merah adalah jarak NPC bos virus melakukan *chase* terhadap *player*, jika diluar garis merah pada NPC bos virus maka bos virus akan menuju *state idle*
2. Lingkaran biru adalah jarak NPC bos virus melakukan attack terhadap *player,* jika diluar garis biru maka NPC bos virus akan melakukan chase terhadap player dan jika diluar dari garis biru dan merah NPC bos virus akan berada pada *state idle*

C. Pengujian Blackbox NPC Bos Virus

TABEL 4. 1
PENGUJIAN NPC BOSS VIRUS

No	Deksripsi	*Input*	*Output* yang diharapkan	Hasil *output*	Keterangan
1	*Idle*	*Player* berada di luar ruang boss	Boss virus belum me-*spawn*	Boss virus belum me-*spawn*	Sesuai
2	*Chase*	Bos virus berada di dalam ruang boss dan berada dalam range 500 px dari player	Boss virus mengejar player yang baru masuk ruang boss	Boss virus mengejar player yang baru masuk ruang boss	Sesuai
3	*Attack*	Bos virus berada dalam *range attack* pada player atau berada dalam 200 px dari player	Boss virus melakukan *attack* pada *player*	Boss virus melakukan *attack* pada *player*	Sesuai
4	*Chase*	*Player* diluar dari *range attack* boss Atau berada diluar 200 px dari bos virus	Boss virus melakukan chase terhadap player	Boss virus melakukan chase terhadap player	Sesuai
5	*Rage mode*	Darah boss virus kurang dari 50 persen	Boss virus menggganti tipe serangan dan menjadi lebih agresif	Boss virus menggganti tipe serangan dan menjadi lebih agresif	Sesuai
6	*Idle*	*Player* keluar dari jangkauan chase virus boss dan juga jangkauan attack	Boss virus berada pada posisi *idle*	Boss virus berada pada posisi *idle*	Sesuai

D. Pengujian Beta

Pengujian Beta dilakukan dengan cara menyebarkan kuesioner kepada responden dari kalangan umum

TABEL 4. 2
METODE SKALA LIKERT

No	Pertanyaan	Persentase
1	Game Doctor vs Virus menarik	88.2
2	Konten yang diberikan *Game* Dokter vs Virus mudah dipahami	89.1
3	Game Dokter vs Virus menghibur	87.7
4	Kontrol Game Dokter vs Virus mudah digunakan	84.1
5	Tingkat Kesulitan Game sesuai	81.4
6	Item pada Game dapat berfungsi dengan baik	87.3
7	NPC Virus Boss dapat memberikan pembelajaran terhadap Kesehatan bahanya virus dan konsep diri	88.6
8	NPC Virus Bos memiliki 2 mode attack yang memberikan tantangan yang baik pada pemain	84.5
9	NPC Virus Bos memiliki tingkat kesulitan yang cukup	78.2
10	NPC Virus Bos dapat mengejar player dengan kecepatan yang cukup	81.4
Rata-rata		85

Uji validitas dilakukan untuk mengukur kelayakan aplikasi dari pertanyaan yang diajukan dalam kuesioner

TABEL 4. 3
PENGUJIAN VALIDITAS

Pertanyaan	r_{xy}	r_{tabel}	Keterangan
1	0.6307	0.2907	Valid
2	0.6352	0.2907	Valid

Pertanyaan	r_{xy}	r_{tabel}	Keterangan
3	0.7923	0.2907	Valid
4	0.6915	0.2907	Valid
5	0.6977	0.2907	Valid
6	0.7082	0.2907	Valid
7	0.6651	0.2907	Valid
8	0.7013	0.2907	Valid
9	0.6264	0.2907	Valid
10	0.7404	0.2907	Valid

Uji reliabilitas adalah uji untuk memastikan apakah kuesioner penelitian yang akan dipergunakan untuk mengumpulkan data variable penelitian reliable atau tidak.

TABEL 4. 4
PENGUJIAN RELIABLITAS

Jumlah Varian item	Jumlah Varian Total	Reliabilitas	rTabel	Keterangan
5.3272	24.8134	0.8726	0.2907	Reliable

V. KESIMPULAN

1. Hasil dari pengujian dan pengimplementasian pada NPC boss virus dengan metode finite state machine untuk pergerakan NPC pada game Doctor vs Virus, NPC dapat melakukan idle, chase, attack dan rage mode yang berfungsi dengan baik sesuai dengan perancangannya.
2. Berdasarkan pengamatan dan pengujian beta dapat diketahui *game* Doctor vs Virus memberikan tingkat kesulitan NPC yang sudah cukup baik dan mendapatkan persentase sebesar 78,2%.
3. Berdasarkan pengamatan dan pengujian beta *game* the Doctor vs Virus mendapatkan persentase sebesar 85 % membuat pengguna lebih tertarik dengan bermain *game* sekaligus belajar.

REFRENSI

[1] Y.-T. Mo, "Comparison between non-NPC interaction and NPC interaction - Taking Fallout 4 and Fallout 76 as examples -," International Journal of Engineering Research and Technology, vol. Volume 12, pp. 1462-1467, 2019.

[2] D. Jagdal, "Finite State Machine in Game Development," International Journal of Advanced Research in Science, Communication and Technology, 10, pp. 384-390, 2021.

[3] S. Hairina Novilita, "KONSEP DIRI ADVERSITY QUOTIENT DAN," JURNAL PSIKOLOGI, vol. 8, pp. 619-632, 2013.

[4] Subaryana, "KONSEP DIRI DAN PRESTASI BELAJAR," JURNAL DINAMIKA PENDIDIKAN DASAR, vol. 7, pp. 21-30, 2015.

[5] I. P. L. H. Indah Rahmawati, "Pengembangan Game Petualang untuk Pembelajaran Berhitung," Jurnal Kajian Teknologi Pendidikan, vol. 5, pp. 11-23, 2020.

[6] N. Ramsari, "PEMBUATAN GAME SIDE SCROLLING 2D THE NAILA'S SURVIVAL BERBASIS ANDROID," Jurnal Teknologi Informasi dan Komunikasi, vol. VIII, pp. 67-80, 2018.

[7] A. Bahtiar, "Penerapan Model Spiral Pada Rancang Bangun Game Platformer," Seminar Nasional Sains dan Teknologi Terapan VII 2019, vol. VII, pp. 601-606, 2019.

[8] M. Firdaus, "PENERAPAN METODE FINITE STATE MACHINE PADA GAME ADVENTURE "TRAPPED MINERS"," JATI Jurnal Mahasiswa Teknik Informatika, vol. III, pp. 158-164, 2019.

[9] M. K. Aulia, "PENERAPAN METODE FINITE STATE MACHINE PADA GAME PANDEMIC NIGHTMARE BERBASIS ANDROID," JATI Jurnal Mahasiswa Teknik Informatika, vol. 5, pp. 291-298, 2021.

[10] R. Andrea, "Finite State Machine Model in Jungle Adventure Game an Introduction to Survival Skills," *Information Engineering and Electronic Business,* vol. 4, pp. 55-61, 2021.