

Application of Finite State Machine in the 3D Game “Virus Hunter”

Fridy Mandita¹, Bramantyo Kusumo Jati²

^{1,2} Program Studi Teknik Informatika, Fakultas Teknik,
Universitas 17 Agustus 1945 Surabaya

Jln. Semolowaru No. 45 Menur Puncungan, Surabaya

¹fridymandita@untag-sby.ac.id, ²bramagreat@gmail.com

Abstrak

Masa pandemi membuat ruang gerak masyarakat menjadi terbatas. Pemerintah menerapkan aturan Work From Home (WFH) yang mengharuskan setiap orang bekerja dengan gadget di rumah masing - masing. Keadaan seperti ini membuat sebagian besar masyarakat menjadi frustrasi dengan keterbatasan yang ada. Game adalah salah satu hiburan yang dapat dilakukan pada masa pandemi. Selain untuk hiburan game juga dapat dijadikan media pembelajaran demi meningkatkan tingkat kewaspadaan terhadap virus atau penyakit. Game Virus Hunter ditujukan agar masyarakat tidak hanya dapat menghibur diri dengan bermain, tetapi juga mendapatkan wawasan mengenai berbagai macam virus. Penelitian ini dilakukan dengan tahapan Research and Development yang kemudian diuji secara fungsional dengan metode white box testing. Pengujian usabilitas dilakukan dengan metode SUS. Dalam perancangan game, pada Non - Player Character (NPC) menggunakan metode Finite State Machine terhadap perilaku terhadap pemain. Hasil uji game Virus Hunter menyatakan bahwa fungsional aplikasi dapat berjalan dengan baik dan mendapatkan skor usabilitas sebanyak 75,87% dengan kategori Acceptable dan Good dalam Adjective Range.

Kata kunci—Game 3D, Third Person Shooter, Finite State Machine, Non-Player Characters, Unity 3D

Abstract

During the pandemic, people's space for movement is limited. The government implements the Work from Home (WFH) rule which requires everyone to work with gadgets in their respective homes. This situation makes most people frustrated with the existing limitations. Games are one of the forms of entertainment that can be done during a pandemic. In addition to entertainment, games can also be used as learning media in order to increase the level of vigilance against viruses or diseases. The Virus Hunter game is intended so that people can entertain themselves by playing and gaining insight into various kinds of viruses. This research was conducted during the stages of research and development which was then tested functionally using white box testing methods. Usability testing was carried out using the SUS method. In game design, the Non-Player Character (NPC) uses the Finite State Machine method to approach player behavior. The virus hunter game test results state that the functional application can run well and get a usability score of 75,87% in the Acceptable and Good categories in the Adjective Range.

Keywords—3D game, Third Person Shooter, Finite State Machine, Non-Player Characters, Unity 3D

1. PENDAHULUAN

Selama pandemi, sebagian besar pekerjaan dilakukan di rumah, terutama mereka yang melakukan *Work From Home (WFH)*, *School From Home (SFH)*, atau aktivitas lain dari rumah. Dilansir dari salah satu media, KOMINFO menyatakan bahwa kenaikan penggunaan *gadget* dan internet mengalami kenaikan sebesar 73,7% selama masa pandemi. Hal ini menunjukkan bahwa aktivitas berbasis digital menjadi target utama selama pandemi. Pertumbuhan pengguna aplikasi *game* mengalami peningkatan sebesar 196% dibandingkan masa sebelum pandemi (Salsabillah, 2021). *Game* adalah salah satu media hiburan karena mampu memberikan rasa kepuasan tersendiri bagi para pemainnya. *Game* juga mampu untuk mengembangkan keahlian seseorang jika mementingkan tingkat kepandaian pemain. *Game* terbagi dalam beberapa jenis yang dikenal dengan sebutan “genre”. *Game* juga tidak lepas dari kecerdasan buatan atau *Artificial Intelligence (AI)* sebagai pendukung dalam perlakuan *Non-Player Characters (NPC)* (Surbakti, 2017).

Artificial Intelligence (AI) atau kecerdasan artifisial didefinisikan sebagai kemampuan komputer untuk berpikir dan bekerja seperti manusia. Penggunaan *AI* dalam *video game* sering dijumpai di berbagai jenis *game* seperti *game* menembak, strategi, dan balapan. Fitur *AI* biasanya digunakan untuk mengontrol karakter non-pemain. Salah satu keahlian kecerdasan buatan yang digunakan yaitu algoritma *Finite-state machine (FSM)* (Yulsilviana & Ekawati, 2019). *Finite-state machine (FSM)* ialah bentuk komputasi yang bermanfaat untuk melakukan simulasi urutan logika atau gambaran urutan eksekusi dari kondisi. Metode ini sangat bagus untuk menerapkan *AI* (kecerdasan buatan) dalam *game* dengan hasil yang memuaskan tanpa kode yang rumit. Karakter dari *Finite-state machine (FSM)* sering dijumpai pada masyarakat dengan urutan tindakan yang ditentukan sebelumnya (Rahadian et al., 2016; Yulsilviana & Ekawati, 2019). Penelitian ini dilakukan untuk membangun *game 3D* dengan menerapkan algoritma *FSM* terhadap perlakuan virus. Implementasi metode *FSM* dalam membangun *game* ini sendiri terletak pada tingkat relasi yang dibangun antara pemain dengan *NPC* utama yang dilakukan dengan interaksi berupa dialog antara *NPC* utama dengan karakter utama. Pemakaian metode ini dalam permainan dapat dimanfaatkan untuk menentukan respon yang diberikan musuh terhadap *player*.

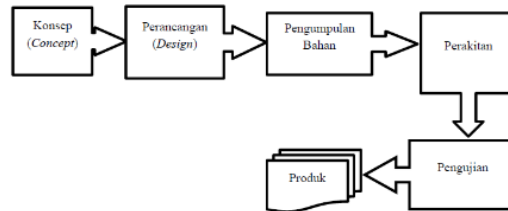
2. METODE PENELITIAN

Dalam penelitian ini menggunakan metode *Research and Development*. Metode penelitian dan pengembangan merupakan metode penelitian yang digunakan untuk menghasilkan produk tertentu (Astuti et al., 2019).

2.1 Penentuan Konsep

Game yang dikembangkan bernama "*Virus Hunter 3D*", yaitu permainan penembakan satu pemain dengan musuh berupa virus. Permainan yang dibangun merupakan *Shooting Game* berjenis *Third Person Shooter (TPS)*. *Game* yang dibuat menggunakan metode *FSM* untuk merancang perilaku karakter non-pemain (*NPC*) menggunakan kecerdasan artifisial. Fokus penelitian ini terletak pada penerapan metode *FSM* terhadap perilaku *NPC* berdasarkan interaksi yang dilakukan pemain. Virus - virus yang telah dihadapi sang pemain akan terdaftar kedalam *archive game*. *Game* akan berakhir pada saat pemain telah mengumpulkan semua barang dan membawanya ke rumah sakit. Konsep yang dirancang pada *game* yang akan dikembangkan sebagai berikut:

- 1) Sistem permainan tunggal atau *single player*
- 2) Mengimplementasikan metode FSM pada perilaku NPC
- 3) Permainan dibuat dengan grafik 3D
- 4) Kontrol pemain menggunakan *mouse* dan keyboard



Gambar 1. Tahapan Penelitian Metode *Research and Development*

2.2 Perancangan

Setelah konsep ditentukan, tahapan yang dilakukan berikutnya adalah perancangan. Tahapan yang dilakukan yaitu pembuatan rancangan atau desain permainan untuk membantu memudahkan dalam pengembangan *game*.

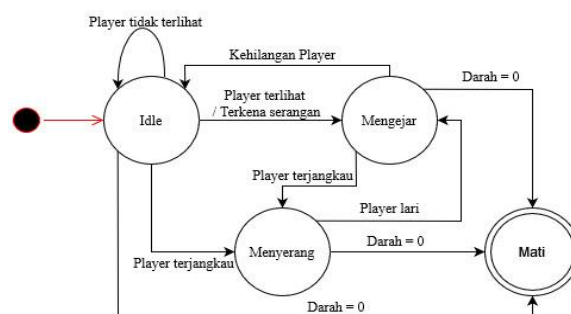
2.2.1 Perancangan Finite State Machine Musuh

Perancangan metode FSM pada musuh dibuat untuk mendefinisikan tiap keadaan pada *NPC*. Perancangan ini bertujuan untuk mengetahui respon yang diberikan musuh terhadap interaksi pemain. Variabel yang mempengaruhi perubahan pada perilaku *NPC* yakni *idle*, mengejar, dan menyerang. Rancangan implementasi perilaku pada *NPC* dijelaskan pada tabel 1.

Tabel 1. Skenario perubahan perilaku pada *NPC* musuh

| Variabel Masukan | Variabel Luaran |
|--|--|
| Tidak ada <i>player</i> | <i>Idle</i> |
| <i>Player</i> memasuki radius virus | Virus mengejar <i>player</i> |
| <i>Player</i> keluar radius virus | Virus kembali ke posisi awal, <i>idle</i> |
| <i>Player</i> diam dan memasuki radius virus | Virus mengejar <i>player</i> , virus menyerang <i>player</i> |
| <i>HP</i> virus habis | Virus mati/hilang |

Rancangan FSM yang diimplementasikan pada *NPC* virus dapat dilihat pada gambar berikut:

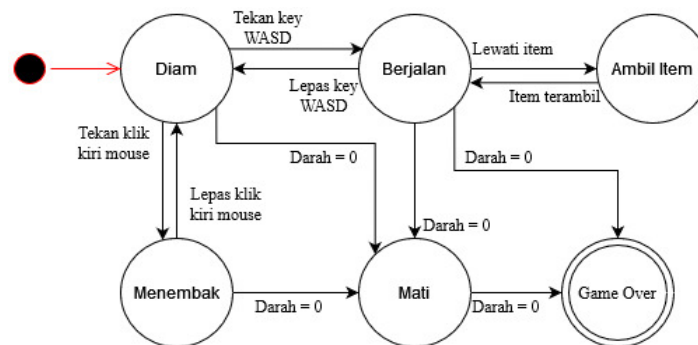


Gambar 2. Rancangan FSM pada *NPC* musuh

Penjelasan perancangan FSM pada *NPC* musuh sebagai berikut:

- 1) *Idle*: Saat permainan dimulai atau jauh dari jangkauan *player* *NPC* berada pada *state idle*.
- 2) *Mengejar*: Saat *player* terlihat dan dalam jangkauan, *NPC* akan beralih ke *state* mengejar *player*. Jika *player* berlari dan keluar dari jangkauan, *NPC* akan kembali ke *state idle*.
- 3) *Menyerang*: Saat *NPC* mengejar *player* dan/atau berada dalam jangkauan menyerang, *NPC* akan beralih ke *state* menyerang *player* yang berada dalam jangkauan serang. Apabila *player* berlari keluar jangkauan serang, *NPC* akan beralih ke *state* mengejar. Apabila *player* tidak terlihat, *NPC* akan kembali pada *state idle*.
- 4) *Mati*: *NPC* akan mati apabila darahnya sudah habis. *NPC* akan mati jika darahnya habis ketika berada di *state Idle*. *NPC* akan mati apabila darahnya habis ketika mengejar pemain. *NPC* akan mati jika darahnya habis ketika sedang menyerang pemain.

2.2.2 Perancangan Finite State Machine Player



Gambar 3. Rancangan FSM pada *player*

Rancangan FSM yang diterapkan pada *player* ditampilkan pada gambar 3. Penjelasan perancangan FSM pada *player* sebagai berikut:

- 1) *Diam*: Posisi awal pemain saat permainan dimulai atau karakter tidak digerakkan berada pada *state* diam. Jika darah *player* ≤ 0 , *player* akan beralih ke *state* mati.
- 2) *Berjalan*: Saat pemain terlihat dan dalam jangkauan, *NPC* akan beralih ke *state* mengejar *player*. Jika *player* berlari dan keluar dari jangkauan, *NPC* akan kembali ke *state* idle. Jika darah *player* ≤ 0 , *player* akan beralih ke *state* mati.
- 3) *Menembak*: Saat *player* menekan klik kiri pada *mouse*, *player* akan beralih ke *state* menembak. Jika *player* melepaskan tekanannya dari klik kiri *mouse*, *player* akan beralih ke *state* diam. Apabila darah *player* ≤ 0 , *player* akan beralih ke *state* mati.
- 4) *Ambil Item*: Saat *player* berada pada *state* berjalan dan melewati *item* yang terdapat dalam permainan, *player* akan beralih ke *state* ambil *item*. Jika *item* telah terambil, *player* akan beralih kembali ke *state* berjalan.
- 5) *Mati*: *Player* akan mati apabila darahnya sudah habis atau ≤ 0 . *Player* akan mati jika darahnya habis ketika berada di *state* diam. *Player* akan mati apabila darahnya habis ketika sedang menembak. *Player* akan mati jika darahnya habis ketika sedang berjalan. Apabila *player* berada pada *state* mati, *player* akan beralih ke *state* Game Over untuk mengakhiri permainan.

2.3 Storyboard

Storyboard adalah sketsa gambar yang disusun secara berurutan sesuai dengan naskah. *Storyboard* yang dibuat merupakan ilustrasi dari game “*Virus Hunter*”. *Storyboard* yang telah dibuat dijelaskan pada gambar 4 berikut.



Gambar 4. *Storyboard*

2.4. Pengumpulan Bahan

Tahapan pengumpulan bahan yang dilakukan adalah mengumpulkan semua bahan yang dibutuhkan untuk membangun game 3D “*Virus Hunter*”. Bahan yang dikumpulkan dengan mengunduh *asset* berupa *audio*, gambar, dan *software Unity*.

2.5. Perakitan

Tahap perakitan yang dilakukan adalah merakit dan menggabungkan semua bahan untuk mengembangkan game yang akan dibuat. Implementasi FSM pada perilaku musuh yang dikembangkan menggunakan bahasa C# atau C Sharp. Perakitan kontrol dan animasi pada setiap objek juga menggunakan bahasa ini.

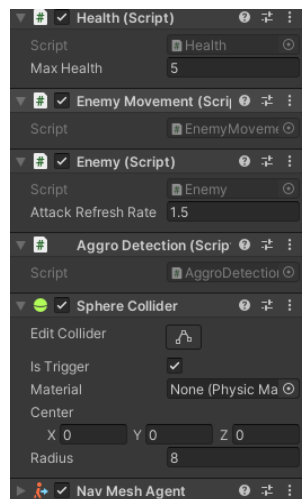
2.6. Pengujian

Pengujian yang dilakukan menggunakan *White Box Testing* dan *System Usability Scale (SUS)* berdasarkan rancangan yang ditetapkan sesuai standar FURPS+.

3. HASIL DAN PEMBAHASAN

3.1. Hasil Implementasi Finite State Machine

Implementasi *Finite State Machine* pada game 3D “*Virus Hunter*” apabila pemain dengan musuh berjarak > 8 , musuh akan berada pada *state Idle*. Ketika rentang jarak musuh dengan pemain < 8 maka musuh akan beralih ke *state* mengejar. Ketika rentang jarak pemain dengan musuh < 1 maka musuh beralih ke *state attack* dan menyerang *player*. Implementasi *Finite State Machine* pada musuh dapat dilihat pada gambar 5.



Gambar 5. Implementasi *Finite State Machine*

3.1.1. Musuh Mengejar

Ketika pemain memasuki radius musuh, maka virus akan mendeteksi keberadaan pemain dan akan beralih dari *state idle* ke *state* mengejar.



Gambar 6. Musuh Mengejar

3.1.2. Musuh Menyerang

Musuh akan mengejar pemain dan menyerang ketika berada dalam radius yang ditentukan. Pemain yang terkena serangan musuh akan kehilangan sebagian darahnya.



Gambar 7. Musuh Menyerang

3.1.3. Efek Serangan Musuh

Musuh yang dihadapi memiliki karakteristik sesuai dengan virus pada *archive*. Salah satunya adalah virus hepatitis. Pada saat virus hepatitis menyerang *player* maka tampilan layar pemain akan menjadi kekuningan akibat dari efek serangan virus tersebut. Efek yang ditimbulkan ini sesuai dengan gejala yang diakibatkan oleh virus hepatitis yaitu mata berwarna kekuningan.



Gambar 8. Efek Serangan Musuh

3.2. Implementasi Sistem

3.2.1. Tampilan Menu Utama

Pada tampilan menu utama terdapat beberapa pilihan tombol yang diberikan kepada pemain. Pada tahap sebelumnya tombol pada menu utama masih belum berfungsi. Pada tahap ini terdapat beberapa tombol yang dapat dipilih pemain. Tombol *Play* akan mengarahkan pemain pada tampilan permainan dan akan memulai permainan. Ketika pemain memilih *Quit* maka permainan akan berhenti dan keluar dari aplikasi.



Gambar 9. Tampilan Menu Utama

3.2.2. Tampilan Permainan

Tampilan permainan yang dibuat dengan kamera sudut pandang orang ketiga (*Third Person*). Pemain dapat melihat wujud karakter yang digunakan. Penggunaan sudut pandang ini memudahkan pemain dalam melihat sekitar peta.



Gambar 10. Tampilan Permainan

3.2.3. Tampilan Archive

Pada tampilan *archive* terdapat dua buah tombol yaitu silang (X) dan tanda panah kanan (>). Tombol silang (X) berfungsi untuk mengembalikan pemain ke menu utama. Tombol panah kanan (>) berfungsi untuk mengarahkan pemain ke halaman berikutnya pada menu *archive*.



Gambar 11. Tampilan *Archive*

3.2.4. Tampilan *Help*

Tampilan ini berisikan cara bermain dan pergerakan dari pemain. Kontrol pemain menggunakan *keyboard* dan *mouse*.

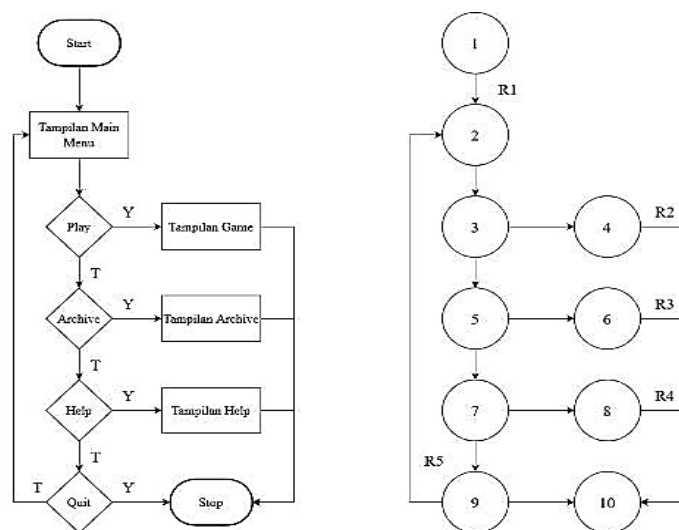


Gambar 12. Tampilan *Help*

3.3. Pengujian *White Box Testing*

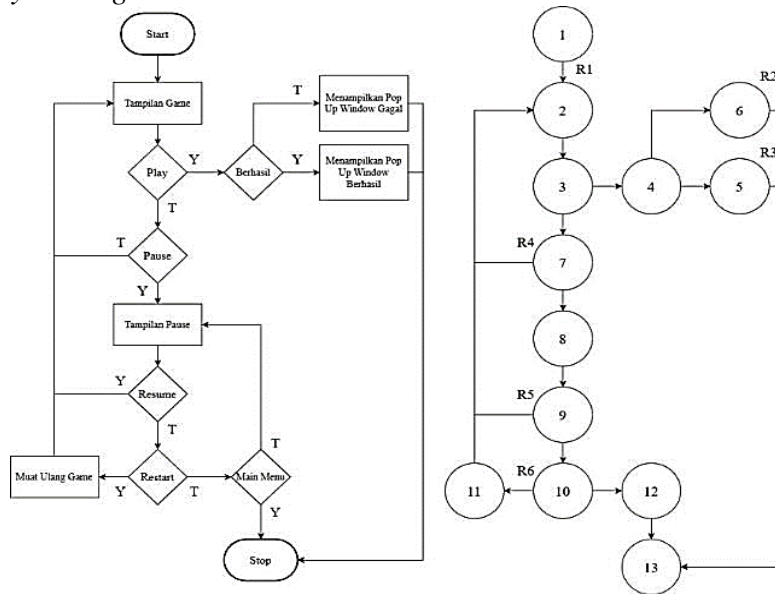
Hasil *White Box Testing* yang diuji dengan melakukan pemetaan *flowchart* ke dalam *flowgraph* dan melalui proses perhitungan kompleksitas siklomatik pada *software* meliputi:

3.3.1. *Main Menu Testing*



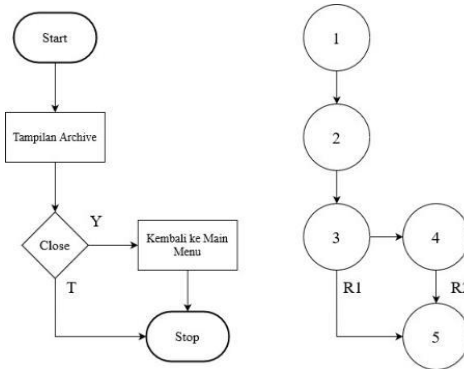
Gambar 13. Pengujian *White Box* Main Menu

3.3.2. Gameplay Testing



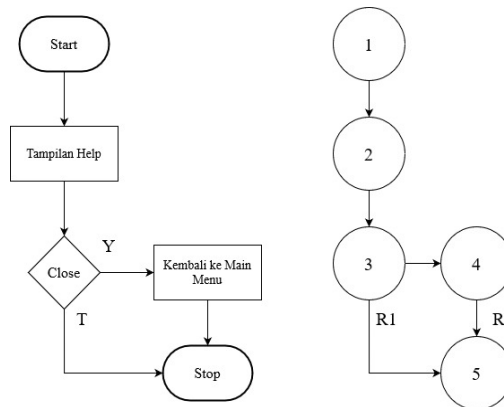
Gambar 14. Pengujian *White Box Gameplay*

3.3.3. Archive Testing



Gambar 15. Pengujian *White Box Archive*

3.3.4. Help Testing



Gambar 16. Pengujian *White Box Help*

3.5. Pengujian System Usability Scale (SUS)

Pengujian usabilitas aplikasi dilakukan menggunakan metode SUS (*System Usability Scale*) melalui 10 pertanyaan yang telah disusun kepada 20 partisipan dengan jenjang usia 7 tahun ke atas. Tercatat 15 responden laki-laki dan 5 responden perempuan yang telah mengisi kuesioner penelitian ini. Hasil rekapitulasi kuesioner yang telah diberikan sebagai pengujian usabilitas aplikasi dijelaskan pada tabel 2.

Tabel 2. Hasil Rekapitulasi Kuesioner

| No | Responden | Jenis Kelamin | Skor | | | | | | | | | |
|----|--------------|---------------|------|----|----|----|----|----|----|----|----|-----|
| | | | Q1 | Q2 | Q3 | Q4 | Q5 | Q6 | Q7 | Q8 | Q9 | Q10 |
| 1 | Responden 1 | Laki - laki | 4 | 1 | 5 | 1 | 4 | 2 | 3 | 2 | 4 | 3 |
| 2 | Responden 2 | Laki - laki | 5 | 4 | 4 | 4 | 4 | 4 | 4 | 3 | 5 | 5 |
| 3 | Responden 3 | Laki - laki | 5 | 3 | 5 | 4 | 4 | 2 | 5 | 2 | 5 | 3 |
| 4 | Responden 4 | Laki - laki | 3 | 2 | 4 | 2 | 4 | 3 | 4 | 2 | 4 | 4 |
| 5 | Responden 5 | Laki - laki | 4 | 1 | 5 | 1 | 5 | 1 | 5 | 1 | 5 | 2 |
| 6 | Responden 6 | Laki - laki | 5 | 1 | 5 | 1 | 5 | 1 | 5 | 1 | 5 | 1 |
| 7 | Responden 7 | Perempuan | 3 | 3 | 3 | 4 | 3 | 3 | 3 | 3 | 3 | 3 |
| 8 | Responden 8 | Laki - laki | 4 | 2 | 4 | 2 | 4 | 2 | 4 | 2 | 4 | 2 |
| 9 | Responden 9 | Perempuan | 3 | 3 | 4 | 2 | 2 | 3 | 4 | 3 | 3 | 4 |
| 10 | Responden 10 | Perempuan | 3 | 4 | 3 | 3 | 3 | 4 | 4 | 4 | 2 | 3 |
| 11 | Responden 11 | Laki - laki | 4 | 3 | 2 | 4 | 4 | 4 | 3 | 2 | 4 | 5 |
| 12 | Responden 12 | Laki - laki | 3 | 2 | 4 | 3 | 2 | 2 | 3 | 3 | 4 | 2 |
| 13 | Responden 13 | Perempuan | 4 | 2 | 4 | 2 | 5 | 3 | 4 | 2 | 3 | 3 |
| 14 | Responden 14 | Laki - laki | 4 | 2 | 4 | 3 | 4 | 2 | 4 | 2 | 3 | 3 |
| 15 | Responden 15 | Perempuan | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| 16 | Responden 16 | Laki - laki | 3 | 2 | 4 | 1 | 4 | 2 | 3 | 2 | 3 | 2 |
| 17 | Responden 17 | Laki - laki | 4 | 2 | 4 | 2 | 4 | 2 | 4 | 2 | 4 | 3 |
| 18 | Responden 18 | Laki - laki | 4 | 3 | 4 | 3 | 4 | 3 | 4 | 3 | 4 | 4 |
| 19 | Responden 19 | Laki - laki | 4 | 2 | 3 | 2 | 4 | 2 | 3 | 2 | 2 | 3 |
| 20 | Responden 20 | Laki - laki | 5 | 1 | 5 | 1 | 5 | 2 | 4 | 2 | 4 | 3 |

Skor yang telah didapat pada tabel 2 akan dikalkulasi dengan rumus yang telah ditentukan oleh SUS. Hasil perhitungan skor dapat dilihat pada tabel 3 dengan rumus sebagai berikut.

$$\text{Pertanyaan nomor ganjil} = (n-1) \quad (1)$$

$$\text{Pertanyaan nomor genap} = (5-n) \quad (2)$$

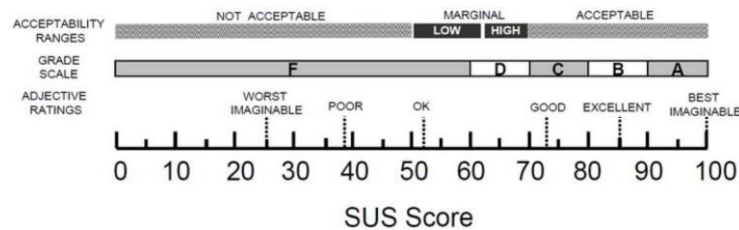
$$\bar{v} = \frac{\sum x}{n} \quad (3)$$

Tabel 3. Hasil Perhitungan Skor

| No | Responden | Skor | | | | | | | | | | Jumlah | Skor (jumlah*2.5) |
|----|--------------|------|----|----|----|----|----|----|----|----|-----|--------|----------------------|
| | | Q1 | Q2 | Q3 | Q4 | Q5 | Q6 | Q7 | Q8 | Q9 | Q10 | | |
| 1 | Responden 1 | 4 | 1 | 5 | 1 | 4 | 2 | 3 | 2 | 4 | 3 | 31 | 77,5 |
| 2 | Responden 2 | 5 | 4 | 4 | 4 | 4 | 4 | 4 | 3 | 5 | 5 | 22 | 55 |
| 3 | Responden 3 | 5 | 3 | 5 | 4 | 4 | 2 | 5 | 2 | 5 | 3 | 30 | 75 |
| 4 | Responden 4 | 3 | 2 | 4 | 2 | 4 | 3 | 4 | 2 | 4 | 4 | 26 | 65 |
| 5 | Responden 5 | 4 | 1 | 5 | 1 | 5 | 1 | 5 | 1 | 5 | 2 | 38 | 95 |
| 6 | Responden 6 | 5 | 1 | 5 | 1 | 5 | 1 | 5 | 1 | 5 | 1 | 40 | 100 |
| 7 | Responden 7 | 3 | 3 | 3 | 4 | 3 | 3 | 3 | 3 | 3 | 3 | 19 | 47,5 |
| 8 | Responden 8 | 4 | 2 | 4 | 2 | 4 | 2 | 4 | 2 | 4 | 2 | 30 | 75 |
| 9 | Responden 9 | 3 | 3 | 4 | 2 | 2 | 3 | 4 | 3 | 3 | 4 | 21 | 52,5 |
| 10 | Responden 10 | 3 | 4 | 3 | 3 | 3 | 4 | 4 | 4 | 2 | 3 | 17 | 42,5 |

| No | Responden | Skor | | | | | | | | | | Jumlah | Skor (jumlah*2.5) |
|------------------------------|--------------|------|----|----|----|----|----|----|----|----|-----|--------|----------------------|
| | | Q1 | Q2 | Q3 | Q4 | Q5 | Q6 | Q7 | Q8 | Q9 | Q10 | | |
| 11 | Responden 11 | 4 | 3 | 2 | 4 | 4 | 4 | 3 | 2 | 4 | 5 | 19 | 47,5 |
| 12 | Responden 12 | 3 | 2 | 4 | 3 | 2 | 2 | 3 | 3 | 4 | 2 | 24 | 60 |
| 13 | Responden 13 | 4 | 2 | 4 | 2 | 5 | 3 | 4 | 2 | 3 | 3 | 28 | 70 |
| 14 | Responden 14 | 4 | 2 | 4 | 3 | 4 | 2 | 4 | 2 | 3 | 3 | 27 | 67,5 |
| 15 | Responden 15 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 20 | 50 |
| 16 | Responden 16 | 3 | 2 | 4 | 1 | 4 | 2 | 3 | 2 | 3 | 2 | 28 | 70 |
| 17 | Responden 17 | 4 | 2 | 4 | 2 | 4 | 2 | 4 | 2 | 4 | 3 | 29 | 72,5 |
| 18 | Responden 18 | 4 | 3 | 4 | 3 | 4 | 3 | 4 | 3 | 4 | 4 | 24 | 60 |
| 19 | Responden 19 | 4 | 2 | 3 | 2 | 4 | 2 | 3 | 2 | 2 | 3 | 25 | 62,5 |
| 20 | Responden 20 | 5 | 1 | 5 | 1 | 5 | 2 | 4 | 2 | 4 | 3 | 34 | 85 |
| Skor Rata-Rata (Hasil Akhir) | | | | | | | | | | | | 75,87 | |

Pengujian SUS dilakukan untuk mengetahui tingkatan penerimaan aplikasi oleh pengguna. Hasil pengujian aplikasi dapat dikatakan *Acceptable* apabila skor rata-rata SUS bernilai lebih dari 70. Skala penilaian dari *System Usability Scale (SUS)* dapat dilihat pada gambar 17 (Lewis & Sauro, 2018).



Gambar 17. Skala Penilaian *System Usability Scale (SUS)*

Setelah perhitungan skor rata-rata dilakukan, didapatkan hasil penerimaan aplikasi oleh user dengan skala sebagai berikut:

- 1) Hasil akhir skor yang didapatkan berdasarkan skala penerimaan, aplikasi yang dibuat termasuk ke dalam kategori *Acceptable*.
- 2) Hasil akhir skor yang didapatkan berdasarkan skala nilai, aplikasi yang dibuat termasuk ke dalam kategori C.
- 3) Hasil akhir skor yang didapatkan berdasarkan skala peringkat, aplikasi yang dibuat termasuk ke dalam kategori *Good*.

4. KESIMPULAN

Berdasarkan pada penelitian yang telah dilakukan yaitu tentang penerapan *Finite State Machine* pada game 3D “Virus Hunter”, terdapat beberapa hasil kesimpulan yang didapat sebagai berikut:

- 1) Hasil implementasi FSM pada NPC musuh telah berhasil dan dapat berjalan sesuai dengan yang diinginkan.
- 2) Pengujian aplikasi pada 20 responden menunjukkan hasil akhir rata - rata 75,87, dimana skor tersebut termasuk dalam kategori yang layak untuk digunakan.
- 3) Fungsionalitas aplikasi *Virus Hunter* berjalan dengan baik. Hal ini dibuktikan dengan pengujian *white box testing* yang diuji pada setiap fungsi pada aplikasi.

5. SARAN

Adapun saran yang diberikan untuk mengembangkan aplikasi yang telah dibuat adalah sebagai berikut:

- 1) Pemberian tingkatan kesulitan agar menjadikan permainan yang menjadi lebih menantang.
- 2) Penambahan level pada permainan untuk memberikan variasi dari permainan.
- 3) Menambahkan ragam jenis NPC musuh yang sesuai dengan karakteristik virus, sehingga permainan menjadi lebih menarik dan edukatif.

UCAPAN TERIMA KASIH

Ucapan terima kasih sebesar-besarnya peneliti sampaikan kepada Tuhan YME, keluarga, dan civitas program studi Teknik Informatika, Fakultas Teknik, Universitas 17 Agustus 1945 Surabaya atas bimbingan, dukungan, petunjuk, serta dorongan, secara moril dan spiritual.

DAFTAR PUSTAKA

- Astuti, Y. W., Yunus, A., & Ahsan, M. (2019). Perilaku Non Player Character (NPC) Pada Game Fps “Zombie Colonial Wars” Menggunakan Finite State Machine (FSM). *Kurawal - Jurnal Teknologi, Informasi Dan Industri*, 2(1), 53–63. <https://doi.org/10.33479/kurawal.2019.2.1.53-63>
- Lewis, J. R., & Sauro, J. (2018). Item Benchmarks for the System Usability Scale. *Journal of Usability Studies*, 13(3), 158–167.
- Rahadian, M. F., Suyatno, A., & Maharani, S. (2016). Penerapan Metode Finite State Machine Pada Game “The Relationship.” *Informatika Mulawarman : Jurnal Ilmiah Ilmu Komputer*, 11(1), 14. <https://doi.org/10.30872/jim.v11i1.198>
- Salsabillah, S. (2021). Membangun Kreativitas Anak di Masa Pandemi Covid-19 Bersama Anak-Anak di Pos Baca Inpres IV Kecamatan Larangan Kota Tangerang. 1–6. <http://jurnal.umj.ac.id/index.php/semnaskat>
- Surbakti, K. (2017). Pengaruh *Game Online* Terhadap Remaja. *Jurnal Curere*, 01(01), 28–38.
- Yulsilviana, E., & Ekawati, H. (2019). Penerapan Metode *Finite State Machine* (FSM) Pada Game Agent Legenda Anak Borneo. *Sebatik*, 23(1), 116–123. <https://doi.org/10.46984/sebatik.v23i1.453>