

**YAYASAN LEMBAGA PENDIDIKAN ISLAM DAERAH RIAU
UNIVERSITAS ISLAM RIAU
FAKULTAS TEKNIK**

**PERANCANGAN *GAME* RPG (*ROLE PLAYING GAME*) “SIGNA”
BERBASIS ANDROID**

SKRIPSI

Diajukan Untuk Memenuhi Salah Satu Syarat
Penyusunan Skripsi Pada Fakultas Teknik
Universitas Islam Riau Pekanbaru



DONI RIADI
153510704

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS ISLAM RIAU
PEKANBARU
2022**

Dokumen ini adalah Arsip Miik :
Perpustakaan Universitas Islam Riau

LEMBAR PENGESAHAN PEMBIMBING SKRIPSI

Nama : Doni Riadi
NPM : 153510704
Jurusan : Teknik
Program Studi : Teknik Informatika
Jenjang Pendidikan : Strata Satu (S1)
Judul Skripsi : Perancangan Game RPG (Role Playing Game) "SIGNA" Berbasis Android

Format sistematika dan pembahasan materi pada masing-masing bab dan sub bab dalam skripsi ini telah dipelajari dan dinilai relatif telah memenuhi ketentuan-ketentuan dan kriteria-kriteria dalam metode penelitian ilmiah. Oleh karena itu skripsi ini dinilai layak dapat disetujui untuk disidangkan dalam **Ujian Seminar Komprehensif**.

Pekanbaru, 19 Agustus 2022

Disahkan Oleh :
Ketua Jurusan Teknik Informatika

Disetujui Oleh :
Dosen Pembimbing



Dr. Apri Siswanto, S.Kom., M.Kom



Panji Rachmat Setiawan, S.Kom., MMSI



**LEMBAR PENGESAHAN TIM
PENGUJI UJIAN SKRIPSI**

Nama : Doni Riadi
NPM : 153510704
Fakultas : Teknik
Program Studi : Teknik Informatika
Jenjang Pendidikan : Strata Satu(S1)
Judul Skripsi : PERANCANGAN *GAME* RPG (*ROLE PLAYING GAME*) "SIGNA" BERBASIS ANDROID

Skripsi ini secara keseluruhan dinilai telah memenuhi ketentuan-ketentuan dan kaidah-kaidah dalam penulisan penelitian ilmiah serta telah diuji dan dapat dipertahankan dihadapan tim penguji. Oleh karena itu, Tim Penguji Ujian Skripsi Fakultas Teknik Universitas Islam Riau menyatakan bahwa mahasiswa yang bersangkutan dinyatakan **Telah Lulus Mengikuti Ujian Komprehensif Pada Tanggal 18 Agustus 2022** dan disetujui serta diterima untuk memenuhi salah satu syarat guna memperoleh gelar Sarjana Strata Satu Bidang Ilmu **Teknik Informatika**.

Pekanbaru, 19 Agustus 2022

Tim Penguji

1. Dr. Arbi Haza Nasution, M.IT

Sebagai Tim Penguji I

2. Ana Yulianti, ST, M.Kom

Sebagai Tim Penguji II

Disahkan Oleh

Ketua Prodi Teknik Informatika

Dosen Pembimbing

Dr. Apri Siswanto, S.Kom., M.Kom.

Panji Rachmat Setiawan, S.Kom., MMSI

LEMBAR PERNYATAAN BEBAS PLAGIARISME

Saya yang bertanda tangan dibawah ini:

Nama : Doni Riadi

Tempat/Tgl Lahir : Rengat, 15 Juni 1997

Alamat : Jl. Sultan No.36. Rt: 006, Rw : 001, Kel : Kampung Dagang ,
Kec : Rengat, Kab : Indragiri Hulu

Adalah mahasiswa Universitas Islam Riau yang terdaftar pada:

Fakultas : Teknik

Jurusan : Teknik Informatika

Program Studi : Teknik Informatika

Jenjang Pendidikan : Strata-1 (S1)

Dengan ini menyatakan dengan sesungguhnya bahwa skripsi yang saya tulis adalah benar dan asli hasil dari penelitian yang telah saya lakukan dengan judul "PERANCANGAN GAME RPG (*ROLE PLAYING GAME*) "SIGNA" BERBASIS ANDROID". Apabila dikemudian hari ada yang merasa dirugikan dan atau menuntut karena penelitian ini menggunakan hasil tulisan atau karya orang lain tanpa mencantumkan nama penulis yang bersangkutan, atau terbukti karya ilmiah ini bukan karya saya sendiri atau **plagiat** hasil karya orang lain, maka saya bersedia menerima sanksi sesuai dengan peraturan perundangan yang berlaku.

Demikian surat pernyataan ini saya buat dengan sesungguhnya untuk dapat digunakan sebagaimana mestinya.

Pekanbaru, 19 Agustus 2022
Yang membuat pernyataan,



DONI RIADI

LEMBAR IDENTITAS PENULIS

Nama : DONI RIADI
NPM : 153510704
Tempat/Tanggal Lahir : Rengat, 15 Juni 1997
Alamat Orang Tua : Kampung Dagang, Rengat
Nama Orang Tua :
Nama Ayah : Syamsir Alam
Nama Ibu : Nursayulis
No.HP/Telp : 082281850201
Fakultas : Teknik
Program Studi : Teknik Informatika
Masuk Th.Ajaran : 2015
Keluar Th. Ajaran : 2022
Judul Penelitian : Perancangan *Game Rpg (Role Playing Game)*
"Signa" Berbasis Android

Pekanbaru, 19 Agustus 2022



DONI RIADI

KATA PENGANTAR

Segala puji bagi Allah SWT yang selalu memberikan rahmat dan hidayahnya serta nikmat yang tak terhingga, sehingga Penulis dapat menyelesaikan laporan skripsi ini dengan judul “Perancangan *Game* RPG (*Role Playing Game*) “SIGNA” Berbasis Android” laporan ini telah Penulis susun dengan maksimal dan mendapatkan bantuan dari berbagai pihak sehingga dapat memperlancar pembuatan laporan skripsi ini.

Dalam penyusunan proposal ini, penulis sadar bahwa tanpa bantuan dan bimbingan berbagai pihak lain maka proposal ini sulit untuk terwujud. Untuk itu penulis ingin menyampaikan ucapan terima kasih kepada kedua orang tua dan keluarga, yang selalu mendo’akan, dan dukungan dan seluruh dosen Prodi Teknik Informatika yang mendidik serta memberi arahan.

Terlepas dari semua itu, Penulis menyadari sepenuhnya bahwa masih ada kekurangan baik dari segi susunan kalimat maupun tata bahasanya. Oleh karena itu dengan tangan terbuka penulis menerima segala saran dan kritik dari pembimbing dan penguji agar penulis dapat memperbaiki laporan skripsi ini.

Akhir kata Penulis berharap semoga laporan skripsi ini dapat memberikan manfaat dan dapat dipergunakan terhadap pembimbing ataupun instansi terkait.

Pekanbaru, 22 Juni 2022

Doni Riadi

PERANCANGAN GAME RPG (*ROLE PLAYING GAME*) “SIGNA” BERBASIS ANDROID

DONI RIADI

Fakultas Teknik

Teknik Informatika

Universitas Islam Riau

Email: doniriadi4@gmail.com

ABSTRAK

Sejak tanggal 11 Maret 2020 WHO (World Health Organization) menetapkan Corona Virus Disease 2019 (COVID-19) sebagai wabah atau Kedaruratan Kesehatan Masyarakat yang Meresahkan Dunia (KKMMD). Sehingga muncul masalah banyak orang yang membatasi aktifitas dan lebih banyak berdiam diri dirumah dan cepat merasa bosan dan mulai mencari cara untuk menghilangkan rasa bosan, salah satunya dengan cara memainkan game di perangkat khusus, komputer, maupun ponsel mereka. Pada zaman sekarang video game hampir dimainkan oleh semua orang, game dapat dijadikan alat untuk menghilangkan rasa bosan dan penghilang suntuk dalam kegiatan sehari-hari. Salah satu jenis game yang beredar dipasaran adalah jenis Role Playing Game (RPG), Role Playing Game adalah sebuah genre yang memiliki unsur peran yang membuat seseorang merasa seperti menjadi seorang tokoh utama pada game tersebut, game biasanya berlangsung dalam satu set dunia dalam waktu tertentu. Pembuatan game ini menggunakan Unity3D sebagai game engine. Berdasarkan pengujian yang dilakukan dengan memberikan kuisioner kepada 50 orang. Dari kategori penilaian 0 sampai 3, nilai penilaian memiliki nilai 2.55 dapat disimpulkan bahwa penilaian pada variabel tampilan termasuk kategori sangat baik.

Kata Kunci: Corona Virus Disease 2019, COVID-19, Role Playing Game, Unity3D.

DEVELOPMENT OF ANDROID-BASED RPG (ROLE PLAYING GAME) GAME "SIGNA"

DONI RIADI

Faculty Of Engineering

Informatics Engineering

Islamic University Of Riau

Email: doniriadi4@gmail.com

ABSTRACT

Since March 11, 2020 WHO (World Health Organization) has designated Corona Virus Disease 2019 (COVID-19) as an outbreak or a public health emergency that is troubling the world (KKMMD). The problem arises many people who limit activities and more stay at home and quickly feel bored and start looking for ways to eliminate boredom, one of them by playing games on special devices, computers, and their phones. In today's video games are almost played by everyone, Games can be used as a tool to relieve boredom and busting in daily activities. One type of game on the market is a type of Role-Playing Game (RPG), Role Playing Game is a genre that has role elements that make a person feel like a main character in the game, the game usually takes place in a set of worlds in a certain time. The creation of this game uses Unity3D as the game engine. Based on testing conducted by giving questionnaires to 50 people. From the assessment category 0 to 3, the value of the assessment has a value of 2.55 it can be concluded that the assessment on the display variable is very good category.

Keywords: Corona Virus Disease 2019, COVID-19, Role Playing Game, Unity3D.

DAFTAR ISI

	Hal
KATA PENGANTAR	i
ABSTRAK	ii
DAFTAR ISI	iv
DAFTAR TABEL	ix
DAFTAR GAMBAR	x
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Identifikasi Masalah	2
1.3 Batasan Masalah	3
1.4 Rumusan Masalah	3
1.5 Tujuan Penelitian	4
1.6 Manfaat Penelitian	4
BAB II LANDASAN TEORI	5
2.1 Tinjauan Pustaka	5
2.2 Dasar Teori	7
2.2.1 <i>Game</i>	7

2.2.2	<i>Genre Game</i>	9
2.2.3	Virus Corona	14
2.2.4	Algoritma <i>Forward Chaining</i>	16
2.2.5	<i>Finite State Machine</i>	17
2.2.6	Unity 3D	18
2.2.7	C#	22
2.2.8	Core RP Library	23
2.2.9	Android	23
2.2.10	Data Flow Diagram	24
2.2.11	Flowchart	25
2.2.12	Hipotesis	27
BAB III METODE PENELITIAN		28
3.1	Perancangan Sistem	28
3.1.1	Perangkat Yang Digunakan	28
3.1.1.1	Spesifikasi Perangkat Keras (<i>Hardware</i>)	28
3.1.1.2	Spesifikasi Perangkat Lunak (<i>Software</i>)	28
3.1.2	Pengumpulan Data	29
3.1.3	Metode Extreme Programming (XP)	29

3.2	Perancangan Aplikasi	31
3.2.1	Deskripsi <i>Game</i>	31
3.2.2	Use Case Diagram	32
3.2.3	Activity Diagram	33
3.2.4	Storyboard	34
3.2.5	Rancangan Grafik	37
3.2.5.1	Grafik Karakter Utama	37
3.2.5.2	Grafik NPC (<i>Non Playable Character</i>) Pemandu	37
3.2.5.3	Grafik NPC (<i>Non Playable Character</i>) Musuh	38
3.2.5.4	Grafik Jebakan	39
3.2.5.5	Grafik Penambah HP	40
3.2.5.6	Grafik Koin	40
3.2.5.7	Grafik Papan Petunjuk	41
3.2.5.8	Grafik Trampolin	41
3.2.5.9	Grafik Peti Harta	42
3.3	<i>Finite State Machine</i>	42
3.4	Metode <i>Forward Chaining</i>	45
3.5	Proses Representasi Cara Kerja <i>Game</i>	46

3.6	Desain Logika Program	46
BAB IV HASIL DAN PEMBAHASAN		54
4.1	Pengujian Masalah Yang Sedang Berjalan	54
4.1.1	Tampilan Awal Aplikasi	54
4.1.2	Tampilan Menu Judul	55
4.1.3	Tampilan Menu Utama	57
4.1.4	Tampilan Menu Permainan	59
4.2	Pengujian <i>Black Box</i>	60
4.2.1	Pengujian <i>Black Box</i> Pada Menu Judul	61
4.2.2	Pengujian <i>Black Box</i> Pada Menu Utama	62
4.2.3	Pengujian <i>Black Box</i> Pada Menu Permainan	64
4.3	Implementasi Sistem.....	66
4.4	Uji Kelayakan Sistem	67
4.4.1	Angket.....	68
4.4.2	Tabulasi Data.....	68

BAB V KESIMPULAN DAN SARAN	71
5.1 Kesimpulan.....	71
5.2 Saran	71

DAFTAR PUSTAKA



Dokumen ini adalah Arsip Miik :
Perpustakaan Universitas Islam Riau

DAFTAR TABEL

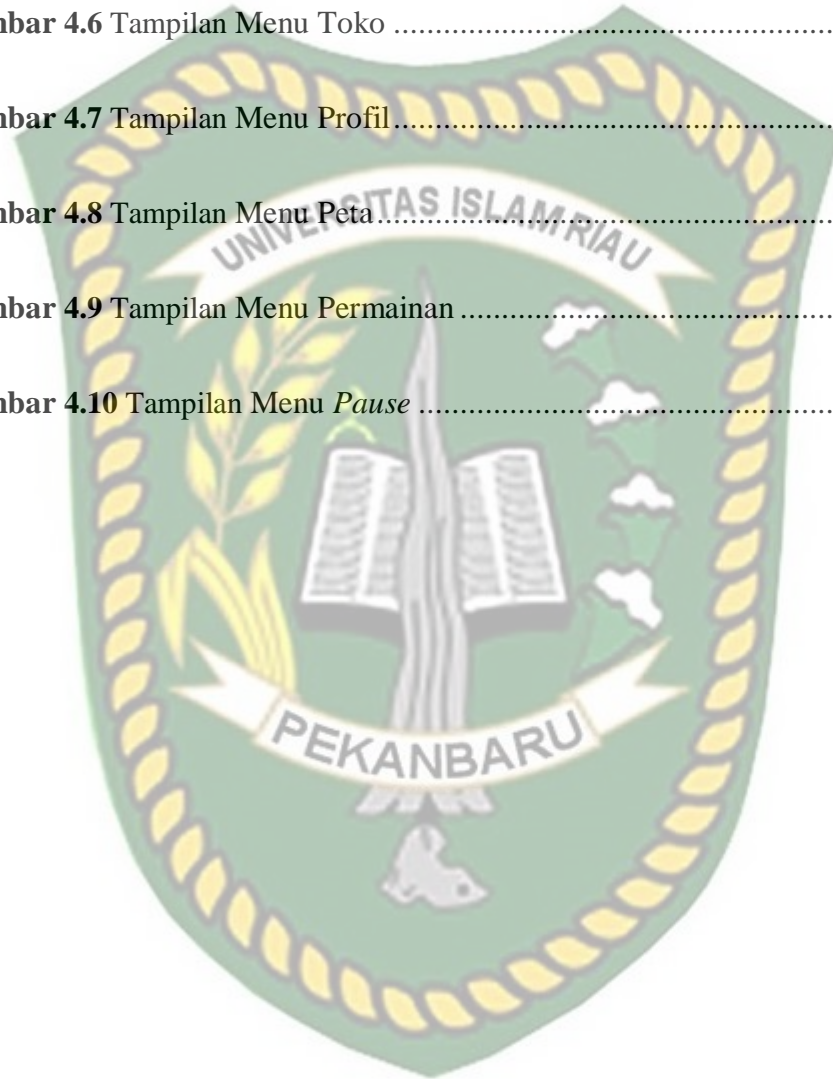
	Hal
Tabel 2.1 Simbol <i>Simbol Data Flow Diagram</i>	24
Tabel 2.2 Simbol <i>flowchart</i>	26
Tabel 3.1 <i>Storyboard</i>	34
Tabel 3.2 <i>Variable Rule</i>	46
Tabel 3.3 Variable Hasil	46
Tabel 3.4 <i>Variable Stage</i>	46
Tabel 3.5 Representasi <i>Rule</i>	48
Tabel 3.6 Representasi Hasil	48
Tabel 3.7 Representasi <i>Stage</i>	49
Tabel 3.8 Aturan Berdasarkan Kenaikan <i>Stage</i>	49
Tabel 3.9 Aturan Berdasarkan <i>Review</i>	49
Tabel 4.1 Hasil Pengujian <i>Blackbox</i> Menu Judul.....	61
Tabel 4.2 Hasil Pengujian <i>Blackbox</i> Menu Utama	62
Tabel 4.3 Hasil Pengujian <i>Blackbox</i> Menu Permainan.....	64
Tabel 4.4 Hasil Implementasi Sistem	66
Tabel 4.5 Kategori Penilaian Skala Likert	68
Tabel 4.6 Hasil Penilaian Variabel	69

DAFTAR GAMBAR

	Hal
Gambar 2.1 <i>game</i> Shadow Fight 2.....	9
Gambar 2.2 <i>game</i> Limbo.....	10
Gambar 2.3 <i>game</i> Monument Valley.....	10
Gambar 2.4 <i>game</i> Final Fantasy VII.....	11
Gambar 2.5 <i>game</i> SimCity BuildIt	12
Gambar 2.6 <i>game</i> Clash Royale	12
Gambar 2.7 <i>game</i> Geometry Dash.....	13
Gambar 2.8 <i>game</i> English for Kids. Fixies. Learning the language	14
Gambar 2.9 gambaran kerja <i>forward chaining</i>	16
Gambar 2.10 Diagram <i>state</i> sederhana	18
Gambar 2.11 Logo Unity 3D.....	18
Gambar 3.1 Metode Extreme Programing	29
Gambar 3.2 Use Case Diagram	32
Gambar 3.3 Activity Diagram	33
Gambar 3.4 Karakter utama dengan ragam kostum.....	37
Gambar 3.5 NPC Pemandu.....	38

Gambar 3.6 NPC Musuh Virus.....	39
Gambar 3.7 Tampilan Jebakan	39
Gambar 3.8 <i>Item</i> Pemulihan HP	40
Gambar 3.9 <i>Item</i> Koin.....	40
Gambar 3.10 Papan Petunjuk	41
Gambar 3.11 Trampoline	41
Gambar 3.12 Peti harta.....	42
Gambar 3.13 <i>Finite State Machine</i> Musuh	43
Gambar 3.14 <i>Finite State Machine</i> Musuh Terbang.....	43
Gambar 3.15 <i>Finite State Machine</i> Musuh Boss	44
Gambar 3.16 Kriteria Kenaikan <i>Stage</i>	45
Gambar 3.17 Proses Representasi Cara Kerja <i>Game</i>	47
Gambar 3.18 <i>Flowchart Finite State Machine</i> Musuh Virus Biasa	51
Gambar 3.19 <i>Flowchart Finite State Machine</i> Musuh Boss	52
Gambar 3.20 <i>Flowchart Metode Forward Chaining</i>	53
Gambar 4.1 Tampilan Awal Aplikasi	54
Gambar 4.2 Tampilan Menu Judul	55
Gambar 4.3 Tampilan Menu Tentang	56

Gambar 4.4 Tampilan Menu Keluar	56
Gambar 4.5 Tampilan Menu Utama	57
Gambar 4.6 Tampilan Menu Toko	57
Gambar 4.7 Tampilan Menu Profil.....	58
Gambar 4.8 Tampilan Menu Peta.....	58
Gambar 4.9 Tampilan Menu Permainan	59
Gambar 4.10 Tampilan Menu <i>Pause</i>	60



BAB I

PENDAHULUAN

1.1 Latar Belakang

Hampir semua manusia pada zaman sekarang tahu yang namanya *game*, mulai dari *game* yang tersedia di komputer maupun *handphone*, karena itu perkembangan *game* saat sekarang ini sangatlah cepat baik itu dalam perangkat khusus, komputer maupun *handphone*. *Game* dapat dijadikan alat untuk menghilangkan rasa bosan dan penghilang suntuk dalam kegiatan sehari-hari. Namun kenyataannya banyak dari mereka yang memainkan *game* yang tidak bermanfaat dan malah ketagihan sehingga lupa waktu.

Apalagi pada masa pandemi, sejak tanggal 11 Maret 2020 WHO (*World Health Organization*) menetapkan *Corona Virus Disease* 2019 (COVID-19) yang disebabkan oleh virus SAR-COV-2. COVID-19 sebagai wabah karena COVID-19 telah menjadi *Public Health Emergency of International Concern* (PHEIC) atau Kedaruratan Kesehatan Masyarakat yang Meresahkan Dunia (KKMMD).

Sehingga muncul masalah banyak orang yang membatasi aktifitas dan lebih banyak berdiam diri dirumah. Dampaknya orang-orang akan cepat merasa bosan dan mulai mencari cara untuk menghilangkan rasa bosan, salah satunya dengan cara memainkan *game* di perangkat khusus, komputer, maupun ponsel mereka.

Membahas tentang teknologi *mobile* sekarang ini tentu sangat menarik karena perkembangan *device mobile* sangat cepat dan luas salah satunya android. Perkembangan android yang sangat cepat ini selain karena android merupakan

sistem operasi mobile yang *open source* dan juga inovasi dari para pengembang aplikasi yang beragam. Android adalah sistem operasi linux untuk perangkat seperti telepon pintar, TV pintar, tablet. Sebagai sistem operasi yang terbuka, android adalah platform terbuka bagi pengembang untuk menciptakan aplikasi atau *game* mereka sendiri yang bisa digunakan oleh pengguna lainnya. Banyak sekali *game* yang beredar pada perangkat *mobile*, salah satu jenis *game* yang beredar dipasaran adalah jenis *Role Playing Game* (RPG).

Role Playing Game (RPG) adalah sebuah *genre* yang memiliki unsur peran yang membuat seseorang merasa seperti menjadi seorang tokoh utama pada *game* tersebut. *Game* berdasarkan *genre* RPG biasanya berlangsung dalam satu set dunia dalam waktu tertentu. *Game* ini hanya berjalan dengan menggunakan satu karakter utama yang akan di jalankan pemain untuk bermain, biasanya memainkan karakter yang diwakili oleh avatar dan memecahkan berbagai tantangan sepanjang permainan.

Maka akan diusulkan untuk membangun aplikasi dengan judul Perancangan Game RPG (*Role Playing Game*) “SIGNA” Berbasis Android berdasarkan latar masalah tersebut, dengan harapan dapat sebagai media informasi dan hiburan yang menyenangkan.

1.2 Identifikasi Masalah

Adapun identifikasi masalah yang didapat dari latar belakang adalah:

1. Banyaknya *game* yang tersedia pada ponsel android kurang bermanfaat dan hanya membuang waktu.

2. Pandemi membuat orang-orang lebih banyak berdiam diri dirumah dan cepat merasa bosan oleh karena itu membutuhkan hiburan.
3. Sedikitnya media hiburan sekaligus mengedukasi tentang *Corona Virus Disease* yang tersedia di android.

1.3 Batasan Masalah

Penelitian ini perlu adanya batasan masalah agar penelitian dapat terarah dan tidak menyimpang dari tujuan yang ditetapkan. Batasan tersebut adalah:

1. *Game* ini hanya tersedia sepuluh tantangan level.
2. Pembuatan *game* ini menggunakan Unity3D sebagai *game engine*.
3. Pembuatan *game* menggunakan ini metode *forward chaining* dan *finite state machine*.
4. Data-data mengenai informasi Virus Corona di dapatkan dari materi tanya jawab promosi kesehatan kementerian kesehatan.
5. *Game* ini *single player* dan berbasis android.

1.4 Rumusan Masalah

Adapun rumusan masalah yang dapat diambil dari latar belakang adalah:

1. Bagaimana menjadikan *game* ini tidak hanya sebagai media hiburan tetapi juga mengedukasi tentang *Corona Virus Disease* yang tersedia di android.
2. Bagaimana menerapkan metode *forward chaining* untuk mengatur penentuan kenaikan *stage level* pemain.
3. Bagaimanakah menerapkan metode *finite state machine* sebagai pengatur perilaku karakter utama maupun musuh pada *game*.

1.5 Tujuan Penelitian

Adapun tujuan dari pembuatan *game* ini adalah:

1. Membangun dan merancang sebuah *game* android dengan mengangkat tema *Corona Virus Disease*.
2. Membangun sebuah *game* yang dapat dimainkan dimana saja dan kapan saja.
3. Mengimplementasikan algoritma *forward chaining* yang digunakan untuk menentukan kenaikan *stage* dan *finite state machine* untuk menentukan perilaku karakter.

1.6 Manfaat Penelitian

Adapun beberapa manfaat yang ingin dicapai dari penelitian ini adalah:

1. Pemain diharapkan dapat bermain *game* yang mengedukasi mengenai pandemi virus corona.
2. Diharapkan menjadikan *game* ini sebagai media untuk meningkatkan rasa peduli terhadap pentingnya edukasi tentang virus corona.
3. Dapat membantu pemain dalam menghilangkan rasa bosan dan jenuh.

BAB II

LANDASAN TEORI

2.1 Tinjauan Pustaka

Dalam menyusun dan pembuatan penelitian ini, peneliti mengambil referensi beberapa penelitian sebelumnya yang ada. Hal ini berguna sebagai perbandingan bahan referensi bagi penulis untuk menyelesaikan penelitian ini. Jurnal-jurnal yang dipilih tentunya berkaitan serta akan digunakan sebagai perbandingan dengan penelitian yang peneliti lakukan.

Studi literatur dilakukan dengan cara mencari, mengumpulkan, membaca dan mencari lebih dalam data dan fakta dan informasi yang dikumpulkan melalui penelitian yang berhubungan dengan buku, jurnal, dan literatur lainnya. Cara ini dilakukan sebagai referensi, acuan, dan pendukung dalam mengaplikasikan metode yang diterapkan. Terdapat beberapa penelitian sebelumnya yang terkait dengan judul penelitian yang berhubungan dengan penelitian ini.

Pada Tahun 2018, Risti Nurarifah, (Nurafifah, 2018) membuat *game* “Mr Garbage” berbasis desktop dengan metode *Fisher Yates Shuffle* dan *Forward chaining*. Untuk menentukan kenaikan level tantangan pemain berdasarkan kriteria yang telah ditetapkan metode *forward chaining* digunakan. *Game* yang dibuat ini diharapkan dapat memberikan pengetahuan dan menyadarkan kepada anak-anak akan pentingnya menjaga lingkungan dari sampah. Pola hidup cinta kepada lingkungan ini diawali dari

hal kecil dalam kehidupan sehari-hari yaitu membuang sampah ke tempatnya. Perangkat lunak yang digunakan dalam perancangan dan pembuatan aplikasi ini yaitu Unity 3D. Berdasarkan hasil pengujian proses *review* dan perpindahan sebanyak 3 *rules level*, *rules* yang digunakan pada metode *Forward Chaining*.

Pada tahun 2018, Aulia Fitrul Hadi, (Hadi, 2018) merancang *game* Edukasi bernama “ANI!”. *Game* yang berbasis android ini dimainkan dengan cara mengingat kemudian menjawab pertanyaan yang diberikan NPC (*Non Player Character*) yang nanti akan digunakan seiring berjalannya *game*. Metode *forward chaining* digunakan sebagai metode kecerdasan buatan di *game* untuk mengambil keputusan yang cerdas. dan *Game* yang terdiri dari 3 level ini dirancang menggunakan software Microsoft Visual Studio 2010. Perancangan dan pembuatan aplikasi ini menggunakan RPG Maker MV.

Pada tahun 2019, Guido Mau, (Mau, 2019) membuat *game* edukasi “2D SHOOTER PLATFORMER” merupakan permainan di mana pemain akan menjalankan karakter *shooter platformer* yang menyelamatkan bumi dari kehancuran yang ingin di kuasai oleh musuh dan pengikutnya. *Game* ini menggunakan *game engine* Unity 3D, dan menerapkan metode FSM (*Finite State Machine*). Metode *Finite State Machine* di gunakan pada karakter NPC (*Non playable Character*) yaitu karakter yang digerakkan oleh kecerdasan buatan yang di gunakan untuk mendukung permainan didalam *game* tersebut.

Pada tahun 2018, Yanto, (Yanto, 2018) membuat aplikasi “Implementasi *Forward Chaining* Pada Game Interaktif “Bersihkan Kotaku” Dengan Pendekatan

Gamification Berbasis Android” merupakan aplikasi edukasi yang memiliki nilai moral dalam menjaga kebersihan kota dengan media game interaktif android. Aplikasi menerapkan metode *forward chaining* dan pendekatan *Gamification*, menggunakan Unity 3D sebagai *game engine*, permainan memiliki 5 tingkatan level dan metode *forward chaining* digunakan untuk membuka tingkatan level selanjutnya dengan syarat yang telah ditentukan.

Pada tahun 2020, Jodi Zakharia, Indah Fenriana, dan Ellysha Dwiyanthi Kusuma (Zakharia et al., 2020) membuat aplikasi “Aplikasi Pengenalan Suara Hewan Pada TK Pelangi Dengan Metode *Forward Chaining*” Aplikasi berbasis dekstop ini dibuat sesuai dengan permintaan untuk membantu dalam pembelajaran mengenal suara hewan. Aplikasi mengkonversi data yang telah diperoleh ke dalam aplikasi pengambilan keputusan menggunakan algoritma *forward chaining*.

Dari penelitian-penelitian tersebut, penulis mengambil referensi bagaimana menerapkan metode *forward chaining* yang digunakan dalam menentukan kenaikan *level stage* berdasarkan kriteria yang ditetapkan dan metode FSM (*Finite State Machine*) untuk menentukan keadaan, kejadian atau aksi pada karakter utama maupun musuh-musuh pada *game*.

2.2 Dasar Teori

2.2.1 Game

Teori permainan (*games*) pertama kali ditemukan pada tahun 1944 oleh sekelompok ahli matematika. Teori ini dikemukakan oleh John Von Neumann dan Oskar Morgenstern, dan berisi “Permainan berisi seperangkat aturan, yang menetapkan

dari dua hingga beberapa situasi kompetitif individu atau beberapa orang kelompok dengan menentukan strategi untuk memaksimalkan kemenangan atau meminimalkan kemenangan lawan pada *game*. Aturan menentukan kemungkinan setiap pemain mengambil tindakan, dan jumlah informasi yang diperoleh setiap pemain dalam permainan dan jumlah kemenangan atau kekalahan dalam berbagai situasi”.

Sedangkan *Video Game* atau disebut juga permainan video adalah permainan yang menggunakan interaksi pemain dengan antarmuka pengguna melalui gambar yang dihasilkan oleh peranti video dan dimana pemain akan terlibat konflik buatan, yang didefinisikan oleh aturan. Menurut Lenhart et al, permainan video adalah suatu perangkat lunak (*software*) yang dapat memberikan hiburan interaktif kepada pemain. *Video game* adalah *software* populer di kalangan remaja dan anak muda.

Sedangkan menurut Pong Story, *video game* adalah *game* yang menggunakan layar antar muka pengguna yang memungkinkan pengguna berinteraksi melalui gambar atau tampilan yang dihasilkan oleh perangkat. Biasanya permainan ini memberikan penghargaan kepada pemainnya, contohnya seperti hadiah, level, ataupun skor.

Menurut Semiawan (2007:19-20) *game* (permainan) dapat dibuat secara sengaja (*intentionally*) dengan tujuan agar pemain untuk meningkatkan kemampuan tertentu berdasarkan pengalaman bermain. Melalui aktifitas bermain, berbagai tindakan dapat terwujud. Bermain adalah aktifitas yang dipilih sendiri oleh pemain karena menyenangkan.

2.2.2 Genre Game

Jenis-jenis *game* dikenal dengan istilah genre game. *Genre* sebuah *game* bisa murni ataupun campuran dari beberapa *genre* lain. Berikut beberapa genre game (Henry Samuel, 2005:44):

A. Action Game

Game bergenre laga (*action*) memberikan tantangan secara fisik yang membutuhkan koordinasi tangan, mata dan waktu reaksi, serta dengan kemampuan motorik yang baik pada pemain. Contoh seperti pada **Gambar 2.1** dari *game* Shadow Fight 2.



Gambar 2.1 *game* Shadow Fight 2

B. Adventure Game

Game petualangan (*adventure*) mulai ada pada tahun 1970, yaitu sebuah *video game* bernama “Colossal Cave Adventure”. Permainan petualangan memberikan sedikit tekanan pada pemainnya untuk menyelesaikan misi, seperti memasuki gua, melompati bebatuan, bergelayutan dari pohon ke pohon, atau

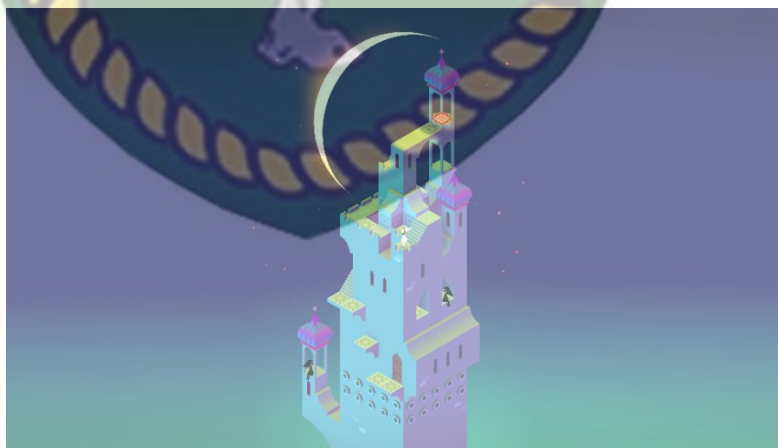
sekedar mencari kunci untuk lanjut ke tantangan berikutnya. Contoh dari *game* ini adalah Limbo seperti pada **Gambar 2.2**.



Gambar 2.2 *game* Limbo

C. Puzzle Game

Sesuai namanya, *game* ini berisikan mengenai pemecahan teka-teki. Seperti menyusun balok, menyamakan kata, gambar, atau warna, bisa juga memecahkan perhitungan matematika. Contoh dari *game* ini adalah Monument Valley seperti pada **Gambar 2.3**.



Gambar 2.3 *game* Monument Valley

D. Role Playing Game (RPG)

Game ini kalau diterjemahkan menjadi, Bermain Peran. Genre ini menekankan pada tokoh utama pada permainan yang biasanya dimainkan oleh pemain. Pemain dapat mengembangkan tokoh menjadi yang di inginkan seperti, menaikkan level karakter, mengubah tampilan karakter, atau mengubah status tokoh. Contoh dari *game* ini adalah Final Fantasy VII seperti pada Gambar 2.4.

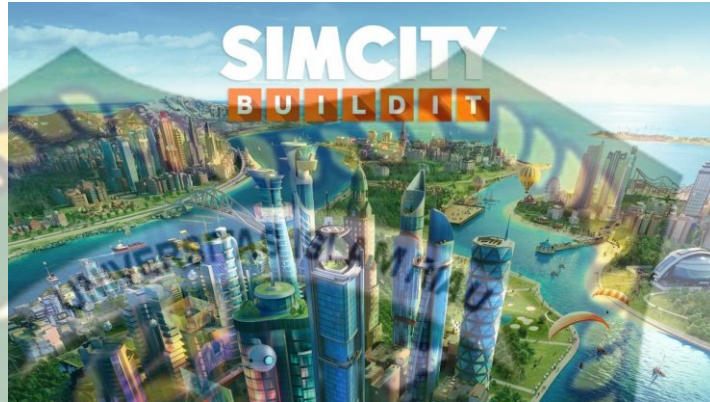


Gambar 2.4 *game* Final Fantasy VII

E. Simulation Game

Game ini menggambarkan sebisa mungkin dengan keadaan dunia nyata dan memperhatikan detail dari faktor tersebut. Permainan ini biasanya melibatkan banyak penelitian dalam pembuatannya karena simulasi biasanya harus bisa semirip mungkin dengan keadaan di dunia nyata. *Game* jenis simulasi membuat pemain mendirikan, membangun, dan mengatasi masalah

dengan menggunakan sumber daya yang dimiliki. Contoh dari *game* ini adalah SimCity BuildIt seperti pada **Gambar 2.5**.



Gambar 2.5 game SimCity BuildIt

F. Strategy Game

Permainan video dengan genre strategi fokus pada cara bermain dan membutuhkan kemampuan berpikir dan perencanaan matang untuk mencapai suatu kemenangan. Seperti catur, pemain membutuhkan lebih banyak keterampilan untuk berpikir dengan hati-hati dan memutuskan setiap gerakan. Contoh dari *game* ini adalah Clash Royale seperti pada **Gambar 2.6**.



Gambar 2.6 game Clash Royale

G. Arcade Game

Permainan *arcade* adalah permainan berupa hiburan yang biasanya memiliki pola yang berulang-ulang. *Video game* bergenre *arcade* biasanya dimainkan dalam jangka waktu yang relatif pendek. Permainan tipe *endless run* termasuk permainan video genre ini. *Endless run* adalah suatu permainan tanpa akhir, tanpa tingkatan, maupun level pemain, biasanya permainan dimulai dengan tempo yang lambat dan semakin meningkat seiring bertambahnya skor atau waktu. Contoh dari *game* bisa dilihat pada **Gambar 2.7** *game* Geometry Dash.



Gambar 2.7 *game* Geometry Dash

H. Educational Game

Game edukasi menciptakan lingkungan *game* yang membantu pemain dalam mendidik, memotivasi atau menambah pengetahuan pemain itu sendiri. Menurut Handriyantini (2009), permainan edukatif adalah permainan yang dirancang atau diproduksi untuk merangsang berpikir, termasuk meningkatkan

konsentrasi dan keterampilan memecahkan masalah. Contoh dari *game* ini adalah English for Kids. Fixies. Learning the language seperti pada **Gambar 2.8** berikut.



Gambar 2.8 *game* English for Kids. Fixies. Learning the language

2.2.3 Virus Corona

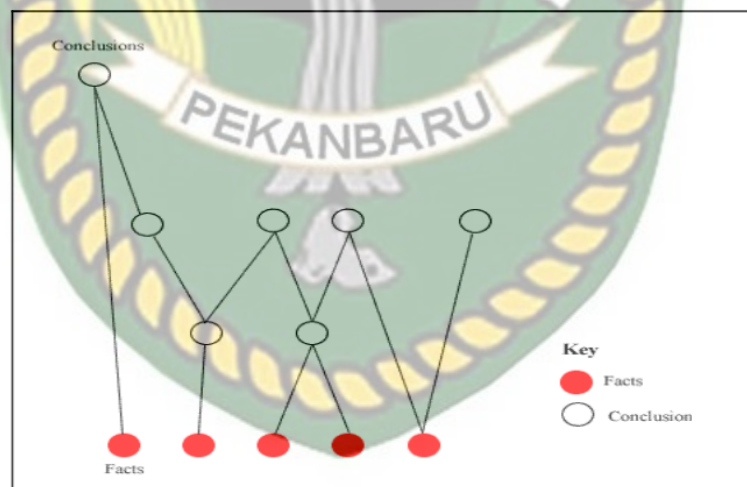
Virus Corona adalah nama virus dari sekelompok besar virus yang dapat menyerang manusia dan hewan. Virus Corona pada manusia umumnya menyerang organ pernapasan. Manusia tidak bisa menularkan virus ini kepada hewan peliharaannya. Setidaknya sudah ada lima jenis virus Corona yang sudah teridentifikasi pada manusia. Tiga diantaranya yang paling terkenal adalah SARS, MERS, dan virus Corona. Dalam Bahasa Latin Virus Corona memiliki arti mahkota. Karena bila dilihat menggunakan mikroskop, virus ini seperti terlihat dilingkari oleh mahkota. Nama Corona Virus Disease-19 (Covid-19) adalah penyakit yang diakibatkan oleh jenis virus Corona yang baru (Kemenkes, 2020).

Gejala dari Covid-19 antara lain: batuk kering, demam, sesak napas, nyeri tenggorok atau nyeri saat menelan, dan rasa pegal-pegal atau merasa kelelahan. Manusia bisa terinfeksi Virus Corona salah satunya yaitu menyebar melalui percikan air (droplet) dari hidung atau mulut pada saat batuk, bersin atau berbicara. Selanjutnya percikan air (droplet) tersebut jatuh pada benda di sekitar. Kemudian jika ada orang lain menyentuh benda yang terkontaminasi dengan droplet tersebut, lalu orang itu menyentuh segitiga wajah (mata, hidung atau mulut), maka orang itu berkemungkinan besar dapat terinfeksi Virus Corona juga, atau seseorang juga dapat terinfeksi virus Corona ketika tidak sengaja menghirup droplet dari penderita.

Virus Covid-19 menyebabkan penyakit yang menyerang pernapasan dan dapat ditularkan dari orang ke orang lain, biasanya setelah kontak dengan seorang yang telah terinfeksi misalnya di tempat kerja, rumah, atau fasilitas pelayanan lainnya. Seperti penyakit pernapasan lainnya, virus Corona dapat menyebabkan gejala-gejala ringan seperti demam, pilek, sakit tenggorokan, dan batuk. Sekitar 80% kasus dapat sembuh tanpa perawatan khusus. Sekitar 1 dari 6 orang dapat mengalami penyakit yang serius, seperti pneumonia atau kesulitan bernapas. Kondisi ini biasanya terjadi secara bertahap. Walaupun angka kematian dari penyakit ini tergolong rendah (sekitar 3%), namun bagi orang yang sudah berusia lanjut (lansia), dan orang-orang dengan kondisi medis sebelumnya (seperti penyakit jantung, tekanan darah tinggi, atau diabetes), merekalah yang biasanya lebih rentan untuk menjadi sakit parah.

2.2.4 Algoritma Forward Chaining

Metode *Forward Chaining* adalah suatu rantai yang dilewati atau dicari dari suatu permasalahan untuk memperoleh solusinya. Cara lain menggambarkan metode *forward chaining* adalah dengan penalaran dari fakta menuju konklusi yang terdapat dari fakta. Didalam mengambil keputusan dari fakta-fakta yang telah diketahui dengan menggunakan inferensi atau metode pencarian atau penarikan kesimpulan yang berdasarkan pada data atau fakta yang ada menuju ke kesimpulan, Pencarian dimulai dengan fakta yang ada dan kemudian bergerak maju ke premis untuk mencapai kesimpulan. (Kadek Tutik A. et al., 2011), *Forward chaining* melakukan pencarian dari suatu masalah kepada solusinya. Jika premisnya cocok dengan situasi, prosesnya akan memberikan kesimpulan. seperti diilustrasikan pada **Gambar 2.9** berikut.



Gambar 2.9 gambaran kerja *forward chaining*

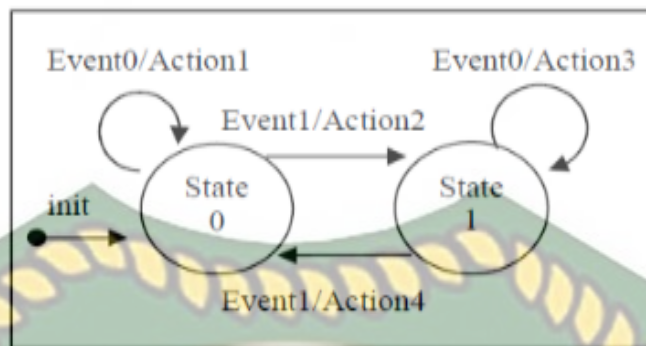
Keuntungan dari metode *forward chaining* adalah data baru dapat dimasukkan ke dalam *table database* inferensi, kemudian menjalankan inferensi *rules*, sehingga

dalam penerapannya metode *forward chaining* sangat membantu pengembang dalam membangun sistem. Jika pengembang ingin menambahkan beberapa ketentuan dan aturan, pengembang tidak perlu membongkar lagi kode program dari awal. Pendekatan ini juga dapat memberikan banyak informasi dari hanya sejumlah kecil data. Dan Keuntungan utama dari *forward chaining* adalah bekerja dengan baik ketika masalah dimulai, dari mengumpulkan atau menyatukan informasi lalu kemudian mencari kesimpulan apa yang diambil dari informasi tersebut.

Metode *forward chaining* disebut juga *data driven* karena mesin inferensi menggunakan informasi yang ditentukan pengguna untuk memindahkan keseluruhan jaringan dari logika “AND” dan “OR” hingga sebuah terminal ditentukan sebagai objek. (Nurafifah, 2018).

2.2.5 Finite State Machine

Finite State Machine adalah sebuah metode perancangan sistem kendali yang menggambarkan tingkah laku atau prinsip suatu kerja sistem dengan menggunakan tiga hal berikut: *state* (keadaan), *event* (kejadian) dan *action* (aksi). Pada satu saat dalam periode waktu tertentu, sistem akan berada pada salah satu keadaan yang aktif. Sistem dapat beralih atau transisi menuju keadaan lain jika mendapatkan masukan atau kejadian tertentu, baik yang berasal dari perangkat eksternal atau bagian dari dalam sistemnya itu sendiri. Transisi keadaan ini biasanya disertai oleh aksi yang dilakukan oleh sistem ketika menanggapi suatu masukan yang terjadi. (Mau, 2019) Aksi tersebut dapat berupa aksi yang sederhana atau aksi yang melibatkan serangkaian proses yang rumit. seperti pada **Gambar 2.10** berikut.



Gambar 2.10 Diagram *state* sederhana

2.2.6 Unity 3D

Unity adalah salah satu *game engine* paling luar biasa saat ini. Unity Technologies didirikan pada tahun 2004 oleh David Helgason, Nicholas Francis dan Joachim Ante. Unity 3D merupakan *software* pengembang terintegrasi yang digunakan untuk membuat video game atau konten lain bernuansa 2D dan 3D, seperti visualisasi arsitektural atau animasi *real-time*. Unity dapat digunakan pada sistem operasi Microsoft Windows dan Mac Os. Permainan yang dihasilkan dapat dijalankan secara multiplatform.



Gambar 2.11 Logo Unity 3D

Unity 3D pada sejarah dan perkembangannya dalam beberapa tahun perkembangannya sebelum dirilis, Unity 3D telah diluncurkan pertama kali sebagai

versi pra-rilis dengan gooball sebuah video *game* yang didesain khusus untuk Apple Macintosh. Gooball dengan versi pra-rilis Unity dirilis atau diumumkan pada Maret 2005. Bersamaan dengan itu, dua bulan kemudian, pada Juni 2005 Unity resmi dirilis sebagai aplikasi komersial. Satu tahun kemudian di tahun 2006, aplikasi pengembang game ini telah masuk menjadi nominasi untuk Apple Design Awards dalam kategori “Best Os X Graphics”. Unity3D juga dikenal sebagai aplikasi pengembang multiplatform, yang berarti Unity3D mendukung pengembangan aplikasi *game* dan aplikasi lain untuk berbagai *platform* (seperti konsol *game*, *platform* seluler, Windows, dan macOS). Semenjak Unity resmi merilis Unity versi 1.0.1 banyak pembaruan (*update*), pembaruan-pembaruan dan fitur yang telah ditambahkan selama tahun perilisannya tersebut dan Unity terus berkembang secara berkala. Sekarang ini Unity sudah ada pada versi 2021.3.6. adapun fitur-fitur utama dari Unity 3D adalah:

1. Rendering

Mesin grafis yang didukung oleh Unity 3D adalah Direct3D (Windows, Xbox 360), OpenGL (Mac, Windows, Linux, PS3), OpenGL ES (Android, iOS) dan Proprietary APIs (Wii). Unity juga memiliki kemampuan untuk bump mapping, reflection mapping, parallax mapping, screen space ambient occlusion (SSAO), dynamic shadows using shadow maps, render-to-texture and full-screen post-processing effects.

ShaderLab adalah bahasa untuk shader yang dapat menyediakan pemrograman deklaratif untuk pipeline fungsi tetap dan program shader yang ditulis dalam GLSL atau Cg. Shader dapat menyertakan beberapa varian dan

spesifikasi deklaratif fallback, yang memungkinkan Unity mendeteksi rentang kartu video terbaik yang tersedia saat ini. Jika tidak kompatibel, shader alternatif dapat digunakan untuk proyeksi, yang dapat mengurangi fungsionalitas dan performa.

2. Asset Tracking

Unity juga menyertakan server untuk Unity Asset, sebuah solusi terkontrol untuk developer game, untuk *asset* dan *script* mereka. Menggunakan postgresql sebagai backend, sistem audio dibuat menggunakan postgresql sebagai *backend*, menggunakan fmod library (dengan kemampuan untuk memutar audio terkompresi Ogg Vorbis), pemutar video atau video playback menggunakan theora codec, engine daratan dan vegetasi (mendukung tree billboard, occlusion culling dengan umbra), lightmapping built-in dan global illumination dengan beast, jaringan multipemain menggunakan raknet dan navigasi grid pathfinder built-in.

3. Asset Store

Diluncurkan pada November 2010, Unity Asset Store adalah sebuah fitur untuk mencari *resource* yang hadir di Unity 3D editor. Asset store terdiri dari koleksi asset packages, 3D models, textures dan materials, sistem particle, musik dan efek suara, tutorial dan project, scripting package, editor extensions dan service online. Asset Store merupakan tempat untuk developer mendapatkan asset yang digunakan dalam menunjang pembuatan *game*.

4. Physics

Sejak Unity 3.0, Unity memiliki suport built-in untuk PhysX physics engine dari Nvidia (sebelumnya Ageia) dengan penambahan kemampuan untuk simulasi real-time cloth pada arbitrary dan skinned meshes, thick ray cast, dan collision layers.

5. Platforms

Untuk mengembangkan aplikasi perangkat mobile, web browser, desktop, dan game konsol. Unity mendukung pengembangan ke berbagai platform, Unity juga mengizinkan spesifikasi kompresi *texture* dan pengaturan resolusi di setiap platform yang didukung.

Untuk saat ini beberapa platform yang didukung oleh unity adalah BlackBerry 10, Windows 8, Windows Phone 8, Windows, Mac, Linux, Android, iOS, Unity Web Player, Adobe Flash, PlayStation 3 dan 4, Xbox 360, Wii U, Wii, dan PlayStation Vita. Meskipun tidak semua terkonfirmasi secara resmi.

6. Scripting

Bahasa pemrograman hal yang umum dalam pembuatan suatu game. Pada *game engine* anda tidak perlu membuat *game* dari awal. Alasannya karena *game engine* sudah menyediakan bahan-bahan dasar pembuatan game seperti, camera, light, tileset, dan lain sebagainya.

Scripting pada *game engine* Unity dibangun menggunakan Mono 2.6, yang merupakan implementasi *open source* dari .net Framework. Bahasa

pemrograman yang didukung oleh Unity 3D adalah, Javascript, c#, dan boo. Namun, untuk Unity versi terbaru Javascript dan Boo sudah ditinggalkan, karena dirasa telah *obsolete* (usang), sehingga pengguna dapat menggunakan C# sebagai bahasa pemrograman (scripting) utama pada unity dalam proses pengembangan *game*.

2.2.7 C#

C Sharp (C#) adalah bahasa pemrograman yang berorientasi objek (*object oriented programming*) yang dikembangkan oleh Microsoft sebagai bagian dari kerangka .NET Framework. Bahasa C++ sebagai basis pembuatan bahasa pemrograman ini yang telah dipengaruhi oleh aspek-aspek maupun fitur bahasa pemrograman lainnya seperti Java, Delphi, Visual Basic, dan bahasa pemrograman lainnya dengan beberapa penyederhanaan. C# juga dapat di jalankan ke komputer dan dapat di proses dalam mode *offline*.

C# adalah bahasa pemrograman yang bisa digunakan untuk pengembangan *game* dan bisa juga digunakan dalam unity untuk pengembangan game 2D maupun 3D oleh karena itu C# terintegrasi dengan unity untuk membuat game arsitektur bangunan dan simulasi yang di rancang unity untuk membangun game arsitektur dan simulasi yang dirancang untuk pemodelan dan rendering dalam aplikasi unity.

Unity C# adalah fungsi *scripting* dan mudah digunakan untuk memutar dan menskalakan objek hanya dengan sebaris kode. Begitu pula dengan menduplikasi, menghapus dan mengubah properti. C# juga mudah digunakan untuk variabel properti

visual yang ditentukan dengan skrip yang ditampilkan di editor yang dapat dijalankan di aplikasi unity.

2.2.8 Core RP Library

Core RP Library mempermudah dalam membuat atau menyesuaikan *Scriptable Render Pipeline* (SRP). SRP berisi kode yang dapat digunakan kembali, termasuk kode boilerplate untuk bekerja dengan API grafis khusus platform, fungsi utilitas untuk rendering umum dan shader libraries kode di SRP digunakan dalam *High Definition Render Pipeline* (HDRP) dan *Universal Render Pipeline* (URP). Jika membuat SRP dari awal atau menyesuaikan SRP, menggunakan SRP Core akan menghemat waktu dalam pembuatan aplikasi.

2.2.9 Android

Menurut Safaat (2012), Android adalah perangkat lunak (*software*) sistem operasi yang memakai basis kode komputer yang dapat didistribusikan secara terbuka (*open source*) sehingga pengguna bisa membuat aplikasi baru untuk perangkat android. Sistem operasi berbasis Linux untuk perangkat seluler, termasuk sistem operasi, middleware, dan aplikasi. Android menyediakan *platform* terbuka bagi pengembang untuk membuat aplikasi mereka sendiri untuk digunakan oleh berbagai perangkat android lainnya. Awalnya, Android Inc di akuisisi oleh Google Inc, Android yang merupakan pendatang baru yang membuat perangkat lunak untuk *smartphone*. Kemudian untuk mengembangkan Android, dibentuklah OHA (Open Handset Alliance), konsorsium dari 34 perusahaan peranti keras, peranti lunak, dan


telekomunikasi termasuk Googl, HTC, Intel, Motorola, Qualcomm, T-Mobile dan Nvidia.

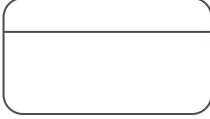


Android pertama kali dirilis pada 5 November 2007. Dengan berdirinya OHA (Open Handset Alliance), OHA mengumumkan produk Android pertama mereka, yang merupakan perangkat seluler dengan kernel Linux 2.6 yang ditingkatkan.

2.2.10 Data Flow Diagram

Data Flow Diagram (DFD) merupakan alat pemodelan yang memungkinkan profesional sistem untuk menggambarkan sistem sebagai jaringan proses fungsional yang dihubungkan satu sama lain dengan alur data secara manual maupun komputerisasi. DFD semacam ini sering disebut juga sebagai *Bubble chart*, *Bubble diagram*, model proses, diagram alur kerja, atau model fungsi. DFD ini merupakan salah satu alat pembuatan model yang sering digunakan terutama ketika fungsi sistem merupakan bagian yang lebih penting dan kompleks dari pada data yang dioperasikan oleh sistem. Dengan kata lain, DFD adalah alat pemodelan yang hanya menekankan fungsionalitas sistem.

Tabel 2.1 Simbol *Data Flow Diagram* (Rosa Salahuddin, 2014)

Simbol	Nama	Fungsi
	Simbol entitas eksternal	Digunakan untuk merepresentasikan asal data atau sumber data.

	Simbol proses	Digunakan untuk menunjukkan tugas atau proses yang dilakukan secara manual atau otomatis.
	Simbol penyimpanan data	Digunakan untuk menampilkan informasi atau gudang data.
	Simbol arus data	Digunakan untuk menunjukkan arus dari proses.

2.2.11 Flowchart

Flowchart adalah representasi *grafis* dan langkah-langkah yang harus diikuti dalam memecahkan masalah yang terdiri dari serangkaian simbol, yang masing-masing mewakili aktivitas tertentu. *Flowchart* membantu analis dan pengembang dalam memecah masalah menjadi bagian-bagian yang lebih kecil dan membantu menganalisis alternatif dalam operasi.







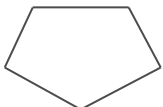
Flowchart diawali dengan penerimaan *input* dan diakhiri dengan penampilan *output*. *Flowchart* adalah suatu gambaran yang menjelaskan urutan:

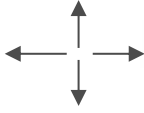




1. Pembacaan data.
2. Pemrosesan data.
3. Pengambilan keputusan terhadap data.

4. Penyajian hasil pemrosesan data.

Simbol *flowchart* yang bisa dipakai adalah simbol-simbol dari *flowchart standart* yang dikeluarkan oleh *ANSI* dan *ISO*. Berikut ini pembahasan dari simbol-simbol yang digunakan untuk menyusun *flowchart* adalah:

Table 2.2 Simbol *flowchart*

No.	Simbol	Fungsi
1		Terminal, berfungsi memulai dan mengakhiri suatu proses.
2		Proses, simbol yang menunjukkan setiap pengolahan yang dilakukan oleh komputer.
3		<i>Input-output</i> , simbol untuk memasukkan data atau menunjukkan hasil dari suatu proses.
4		<i>Decision</i> , suatu kondisi yang akan menimbulkan beberapa kemungkinan jawaban atau pilihan.
5		<i>Predefined</i> proses, simbol yang menyediakan tempat untuk pemrosesan data dalam <i>storage</i> .
6		<i>Connector</i> , simbol suatu prosedur masuk atau keluar melalui ini dalam lembar yang sama.
7		<i>Off-line Connector</i> , simbol masuk atau keluarnya suatu prosedur pada lembar kertas atau lainnya.

8		<i>Flow</i> , prosedur alur yang dilakukan dari atas kebawah, dari bawah keatas, dari kiri kekanan, dan dari kanan kekiri.
9		<i>Docuement</i> , merupakan simbol untuk data dalam bentuk kertas ataupun untuk informasi.
10		Untuk menyatakan sekumpulan langkah proses yang ditulis sebagai prosedur.
11		Simbol untuk <i>output</i> , ditunjukkan ke suatu <i>device</i> , seperti printer, <i>plotters</i> dan lain-lain sebagainya.
12		Simbol untuk menyimpan data

2.2.12 Hipotesis

Penelitian ini bertujuan untuk menerapkan metode *Forward Chaining* untuk menentukan kenaikan tantangan *stage* pada *game* agar pemain menyelesaikan target pada *stage* sebelumnya untuk membuka *stage* berikutnya dan metode *finite state machine* untuk penentuan keadaan pada karakter utama maupun musuh pada *game*, sehingga *game* menjadi menyenangkan, menantang, serta menambah kepedulian kita terhadap pandemi virus corona.

BAB III

METODE PENELITIAN

3.1 Perancangan Sistem

Adapun alat, bahan dan metode yang digunakan dalam pengembangan yang digunakan dalam perancangan sistem ini adalah sebagai berikut:

3.2.1 Perangkat Yang Digunakan

Perangkat dalam pembuatan *game* SIGNA ini terdiri dari dua perangkat, yaitu perangkat keras atau *hardware* dan perangkat lunak atau *software*.

3.1.1.1 Spesifikasi Perangkat Keras (*Hardware*)

Untuk menjalankan aplikasi dengan baik, pastinya perangkat keras (*hardware*) harus memenuhi spesifikasi untuk kebutuhan aplikasi yang dikembangkan, Perangkat keras (*hardware*) yang digunakan dalam pembuatan *game* SIGNA berbasis Android ini adalah sebagai berikut:

1. *Processor* : Intel Core i5-1035G1
2. *Ram* : 8,00 GB
3. *Hardisk* : 500 GB
4. *Syitem Type* : 64-bit *Operating Syatem*

3.1.1.2 Spesifikasi Perangkat Lunak (*Software*)

Perangkat lunak (*software*) yang digunakan dalam pembuatan *game* SIGNA berbasis Android ini adalah sebagai berikut:

1. *Sistem Operasi* : Microsoft Windows 10
2. *Game Engine* : Unity3D 2020.3.13F1

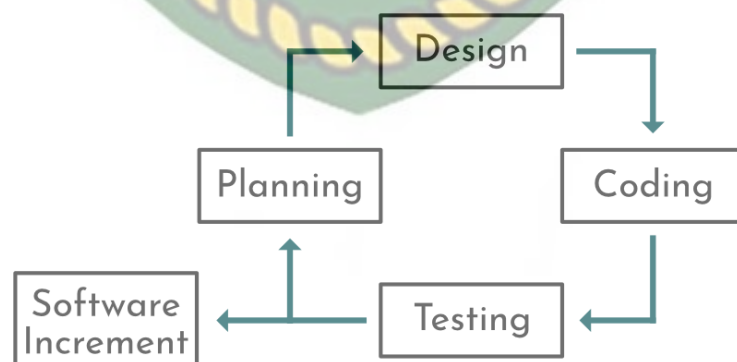
3. Design Konsep dan *Assets* : Affinity Designer 1.10
4. *Sound effects* : SFXR
5. Bahasa Pemograman : C#
6. *Script Writer* : Visual Studio 2019

3.2.2 Pengumpulan Data

Pada metode ini penulis pengumpulan data dalam pembuatan *game* ini melalui tulisan-tulisan ataupun buku mulai dari soal pernyataan-pernyataan sampai pada pencarian hasil yang akan menjadi *output* pada aplikasi. Penulis melakukan pengamatan pada objek atau bidang yang sedang diteliti dan pengumpulan data atau informasi dengan cara mencari data-data atau informasi-informasi yang berhubungan dengan materi yang dibutuhkan untuk menunjang penelitian.

3.2.3 Metode Extreme Programming (XP)

Dalam membuat *game Role Play Game (RPG)* “SIGNA”, penulis menggunakan metode Extreme Programming (Pressman 2012). Terdapat empat proses dalam metode Extreme Programming:



Gambar 3.1 Metode Extreme Programing

1. Perencanaan (*Planning*)

Perencanaan biasanya dimulai dengan mengumpulkan persyaratan yang memungkinkan pengembang memahami konteks utama dari perangkat lunak yang akan dikembangkan dan merasakan kebutuhan akan keluaran, fitur utama, dan fungsionalitas.

2. Perancangan (*Design*)

Perancangan metode Extreme Programming secara ketat mengikuti prinsip-prinsip "tetap sederhana". Desain yang sederhana akan selalu mengungguli desain yang lebih kompleks dan rumit. Selain itu, desain metode Extreme Programming akan memberikan pedoman implementasi untuk cerita yang ditulis, tidak lebih, tidak kurang. Desain dan fitur diluar dari itu atau yang lainnya sangat tidak disarankan.

3. Pengkodean (*Coding*)

Setelah sebuah cerita dikembangkan dan karya rancangan awal dilakukan, pengembang tidak langsung beralih ke kode program, melainkan melakukan serangkaian tes unit dikembangkan dan dijalankan untuk setiap cerita yang akan disertakan dalam versi yang ada. Setelah tes unit dibuat, pengembang perangkat lunak dapat lebih fokus pada apa yang harus diimplementasikan untuk lulus tes unit pada pengujian tersebut. Setelah kode program selesai ditulis, kode program tersebut dapat diuji secara langsung dengan pengujian unit yang telah dirancang sebelumnya, sehingga dapat memberikan umpan balik langsung kepada pengembang.

4. Pengujian (*Testing*)

Uji kelayakan XP, juga dikenal sebagai pengujian pelanggan, dirinci oleh pelanggan dan pada dasarnya berfokus pada fitur dan fungsionalitas sistem atau perangkat lunak secara keseluruhan yang dapat dilihat dan ditinjau oleh pelanggan. Uji kelayakan berasal dari cerita pengguna yang diimplementasikan sebagai bagian dari rilis perangkat lunak.

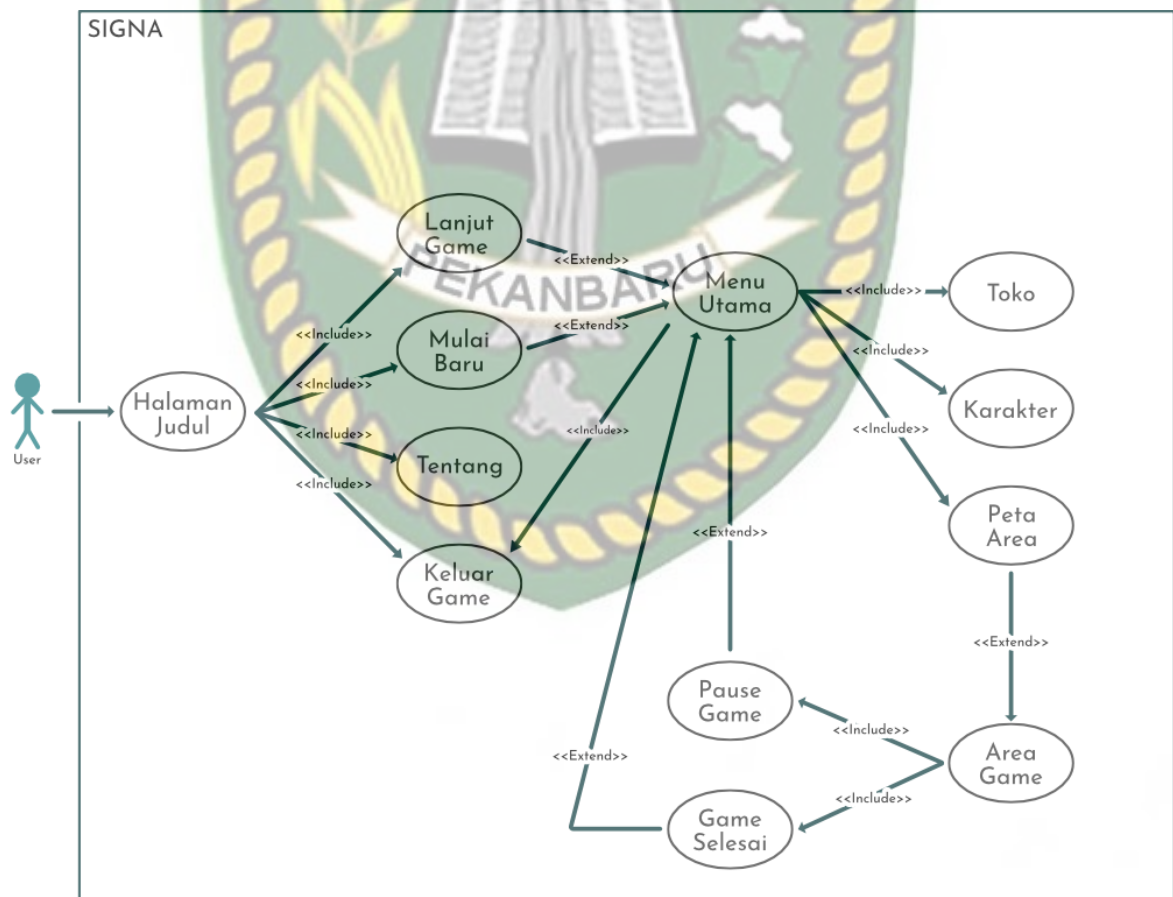
3.2 Perancangan Aplikasi

3.2.1 Deskripsi *Game*

Game yang bergenre RPG dan dimainkan secara *single player* dan *game* ini dapat dimainkan untuk semua umur. Pada *game* ini terdapat karakter utama yang akan dijalankan oleh pengguna. Karakter utama yang baru bergabung dalam sebuah organisasi bernama Signa diharuskan untuk menjelajahi area tantangan membasmi virus yang berkeliaran, membuka peti, menjawab pertanyaan seputar Virus Corona untuk meningkatkan kemampuan karakter utama, mengumpulkan koin dan menyelesaikan area tantangan tersebut, nanti pemain atau karakter utama akan mendapatkan upah dari membasmi virus, membuka peti dan mengumpulkan koin. Untuk membuka area tantangan selanjutnya, pemain harus membasmi virus dan mengumpulkan koin sebanyak target atau melebihi pada masing-masing area tantangan. Di dalam permainan ini terdapat 10 area tantangan dan pada area 10 pemain akan menghadapi musuh utama dari permainan ini. Untuk mendapatkan petunjuk dari pertanyaan dari NPC pada permainan ini pemain dapat melihat papan yang tersebar di seluruh area tantangan.

3.2.2 Use Case Diagram

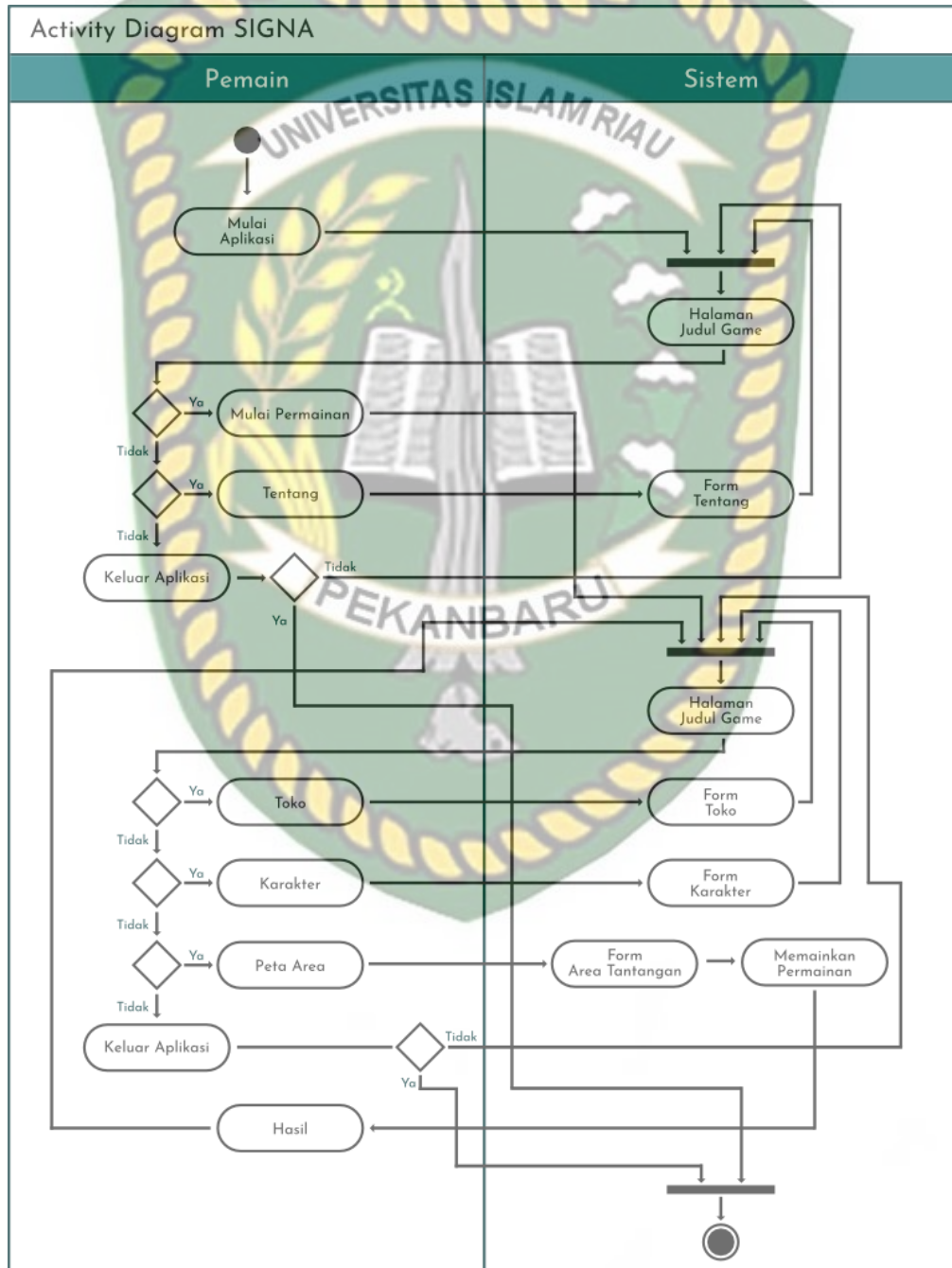
Use case diagram adalah diagram yang digunakan untuk menggambarkan secara ringkas siapa yang menggunakan sistem dan apa saja yang bisa dilakukannya. Use case class digunakan untuk memodelkan dan menyatakan unit fungsi atau layanan yang disediakan oleh sistem atau bagian sistem seperti subsistem atau class ke pemakai. Diagram use case tidak menjelaskan secara detail tentang penggunaan usecase, namun hanya memberi gambaran singkat hubungan antara usecase, aktor, dan sistem. Use case diagram pada aplikasi ini dapat dilihat pada gambar 3.2.



Gambar 3.2 Use Case Diagram

3.2.3 Activity Diagram

Activity diagram merupakan rancangan aliran aktivitas atau aliran kerja dalam sebuah sistem yang akan dijalankan. Activity diagram pada aplikasi ini dapat dilihat pada **gambar 3.3**.



Gambar 3.3 Activity Diagram

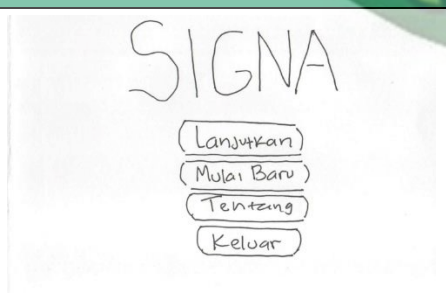
Activity diagram juga digunakan untuk mendefinisikan atau mengelompokkan aluran tampilan dari sistem tersebut. Activity diagram memiliki komponen dengan bentuk tertentu yang dihubungkan dengan tanda panah. Panah tersebut mengarah ke-urutan aktivitas yang terjadi dari awal hingga akhir.


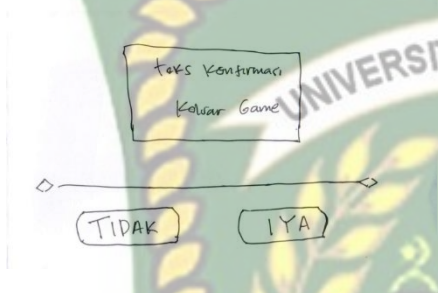


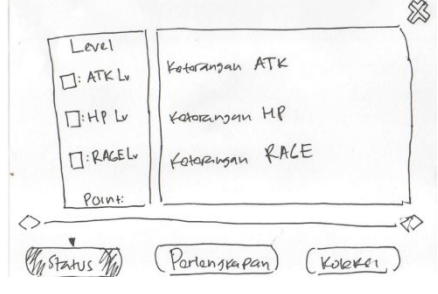
Bisa dilihat fungsi dari activity diagram memperlihatkan urutan aktifitas proses pada sistem, Membantu memahami proses secara keseluruhan, dan menggambarkan proses urutan aktivitas dalam sebuah proses.



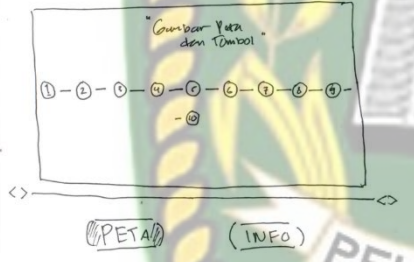
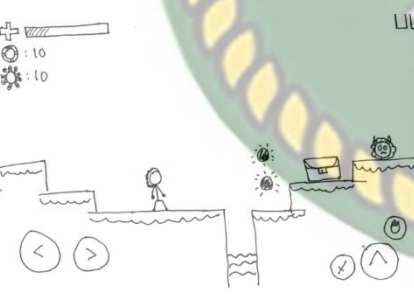

3.2.4 Storyboard

Storyboard adalah sebuah desain sketsa gambar yang disusun sesuai dengan naskah cerita yang telah dibuat. Dengan pembuatan *storyboard* sendiri, maka pembuat cerita atau naskah dapat menyampaikan pesan atau ide dengan lebih mudah kepada orang lain. *Storyboard* pada permainan ini bisa dilihat pada **tabel 3.1**.

Tabel 3.1 *Storyboard*

NO	Rancangan <i>Storyboard</i>	Audio	Keterangan
1		<i>Backsound</i> musik “To Zanarkand“.	Pada tampilan awal <i>game</i> akan menampilkan judul, tombol mulai, tentang, dan tombol keluar <i>game</i> .

2		<p><i>Backsound musik</i> "To Zanarkand".</p>	<p>Jika pemain memilih menu tentang maka akan menampilkan informasi tentang permainan ini.</p>
3		<p><i>Backsound musik</i> "To Zanarkand".</p>	<p>Jika pemain memilih keluar, akan menampilkan menu konfirmasi apakah pemain ingin keluar dari game.</p>
4		<p><i>Backsound musik</i> "To Zanarkand".</p>	<p>Memulai permainan, akan menampilkan menu utama dari permainan. Yang terdapat menu toko, karakter dan peta.</p>
5		<p><i>Backsound musik</i> "To Zanarkand".</p>	<p>Menu toko akan menampilkan senjata atau kostum, harga dan info yang akan dibeli.</p>
6		<p><i>Backsound musik</i> "To Zanarkand".</p>	<p>Pada menu status pemain dapat meningkatkan status karakter jika memiliki point status.</p>

7		<p>Backsound musik "To Zanarkand".</p>	<p>Pemain dapat mengubah perlengkapan karakter pada menu ini.</p>
8		<p>Backsound musik "To Zanarkand".</p>	<p>Pemain dapat melihat koleksi-koleksi apa saja yang telah pemain dapatkan.</p>
9		<p>Backsound musik "To Zanarkand".</p>	<p>Sebelum memasuki area, pemain dapat memilih area tantangan yang tersedia.</p>
10		<p>Backsound musik "04forest6".</p>	<p>Pada area tantangan pemain dapat memainkan karakter utama dengan menekan tombol pada saat memasuki area tantangan.</p>
11		<p>Backsound musik "14levelup2NL".</p>	<p>Akhir tantangan, pemain dapat melanjutkan ke tantangan berikutnya atau kembali ke menu utama.</p>

3.2.5 Rancangan Grafik

Terdapat Beberapa rancangan grafik pada *game* yang akan dibangun yaitu:

3.2.5.1 Grafik Karakter Utama

Karakter ini memerankan sebagai tokoh utama dalam permainan yang dapat pemain kendalikan, kenakan senjata, kostum, dan meningkatkan kemampuan. Untuk menyelesaikan misi, pemain harus membasmi musuh mengumpulkan koin yang sesuai dengan target pada masing-masing area tantangan dan menjawab pertanyaan dengan benar untuk mendapatkan *point skill*. Grafik karakter utama merupakan desain penulis sendiri dan dapat dilihat pada **Gambar 3.4**.

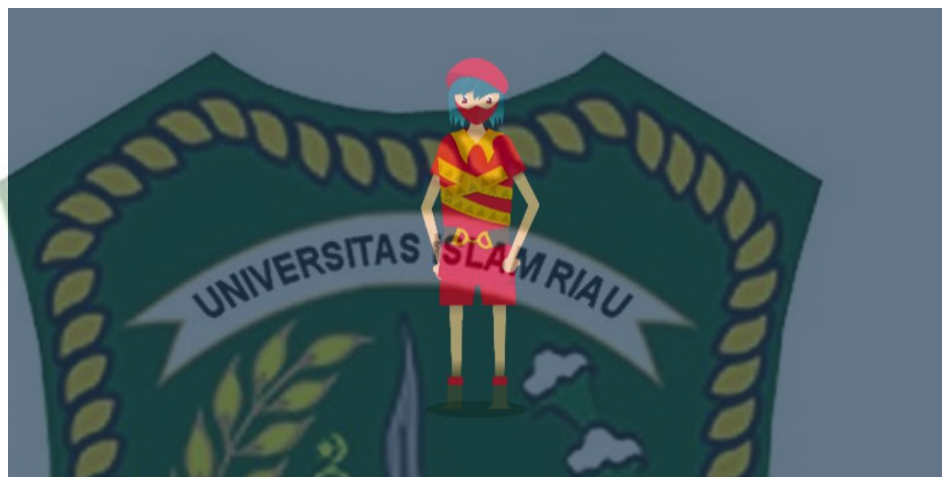


Gambar 3.4 Karakter utama dengan ragam kostum

3.2.5.2 Grafik NPC (*Non Playable Character*) Pemandu

Karakter ini akan menjelaskan kepada pemain saat pertama kali bermain dan karakter ini juga dapat anda temui di area tantangan, pemain dapat berinteraksi dengan NPC (*Non Playable Character*) ini dan menjawab pertanyaan seputar virus Corona, jika pemain menjawab dengan benar maka pemain akan mendapatkan *skill point* yang dapat digunakan untuk meningkatkan kemampuan karakter utama.

Grafik NPC Pemandu merupakan desain penulis sendiri dan dapat dilihat pada **Gambar 3.5**.



Gambar 3.5 NPC Pemandu

3.2.5.3 Grafik NPC (*Non Playable Character*) Musuh Virus

Karakter ini berkeliaran di sekitar area tantangan, jika karakter ini mendeteksi player karakter ini akan mendekat dengan cepat ke arah karakter pemain dan jika tersentuh akan mengakibatkan pengurangan HP pada pemain dan yang memiliki sayap akan menembakan serangan kebawah dan juga mengakibatkan pengurangan HP pemain. NPC musuh virus ini dibagi menjadi 3 variasi dari yang berwarna kuning yang memiliki paling sedikit HP sampai ke warna ungu yang memiliki HP tinggi dan serangan yang besar. Pemain dapat membasmi dengan cara menyerangnya hingga musnah. Grafik karakter ini merupakan desain penulis sendiri dan dapat dilihat pada **Gambar 3.6**.



Gambar 3.6 NPC Musuh Virus

3.2.5.4 Grafik Jebakan

Jebakan juga tersebar di seluruh area tantangan, jebakan juga dapat mengakibatkan pengurangan HP pemain tetapi jebakan tidak dapat mendeteksi pemain dan jebakan juga tidak dapat di hancurkan. Terdapat 2 jenis jebakan pada permainan ini. Grafik tampilan dari jebakan merupakan desain penulis sendiri dan dapat dilihat pada **Gambar 3.7**.



Gambar 3.7 Tampilan Jebakan

3.2.5.5 Grafik Penambah HP

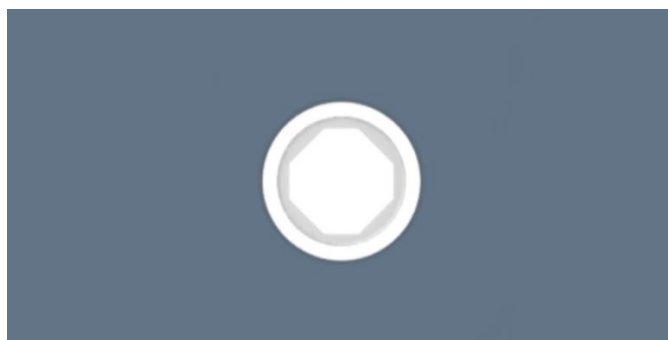
Barang (*item*) ini juga tersebar di area tantangan, jika pemain mengambil barang ini pemain akan mendapatkan pemulihan HP sebesar 5 *point*. Grafik tampilan dari pemulihan HP merupakan desain penulis sendiri dan dapat dilihat pada **Gambar 3.8**.



Gambar 3.8 *Item Pemulihan HP*

3.2.5.6 Grafik Koin

Koin yang banyak tersebar di area tantangan, kumpulkan koin untuk dapat ditukarkan dengan uang pada akhir area tantangan dan koin juga merupakan salah satu syarat target untuk membuka area tantangan selanjutnya yang belum terbuka. Grafik tampilan dari koin merupakan desain penulis sendiri dan dapat dilihat pada **Gambar 3.9**.



Gambar 3.9 *Item Koin*

3.2.5.7 Grafik Papan Petunjuk

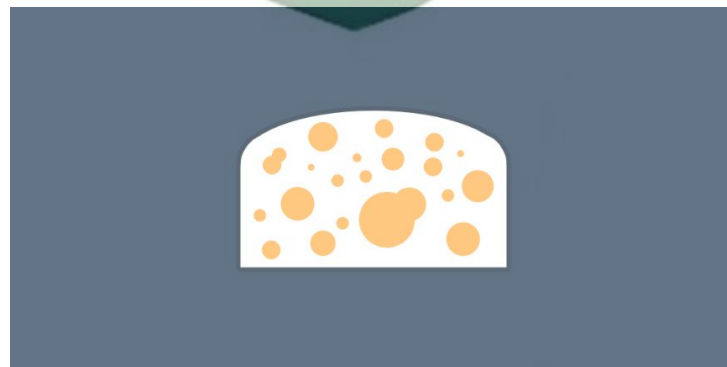
Player dapat berinteraksi dengan papan petunjuk dan akan menampilkan informasi seputar virus Corona yang akan memberikan petunjuk dari pertanyaan NPC Pemandu. Grafik tampilan papan petunjuk merupakan desain penulis sendiri dan dapat dilihat pada **Gambar 3.10**.



Gambar 3.10 Papan Petunjuk

3.2.5.8 Grafik Trampolin

Player dapat menggunakan trampolin untuk mencapai area yang tinggi yang tidak bisa dicapai dengan lompatan biasa, trampolin akan memantulkan player ke atas lebih tinggi dari lompatan biasa player. Grafik tampilan trampolin merupakan desain penulis sendiri dan dapat dilihat pada **Gambar 3.11**.



Gambar 3.11 Trampoline

3.2.5.9 Grafik Peti Harta

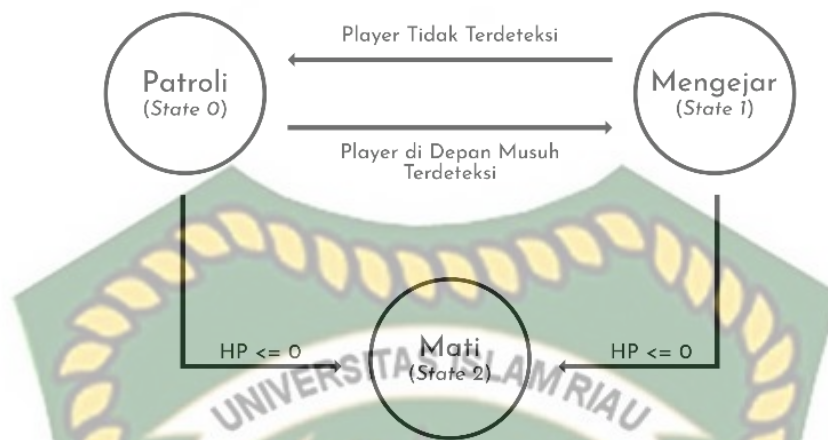
Peti harta dapat ditemukan tersebar di sekitar area tantangan player dapat membuka peti dengan serangan biasa, pemain bisa mendapatkan uang dengan jumlah acak sesuai dengan jenis peti dari peti perunggu hingga peti emas yang lebih banyak memberikan uang dan juga peti emas bisa berisikan benda koleksi. Grafik tampilan peti harta merupakan desain dari penulis sendiri dan dapat dilihat pada Gambar 3.12.



Gambar 3.12 Peti harta

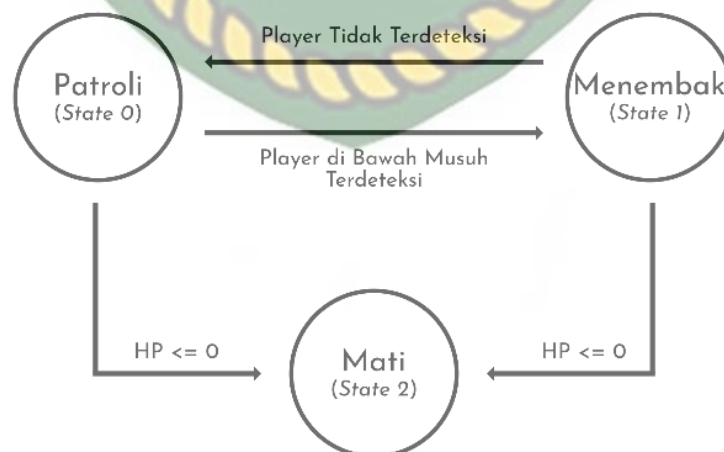
3.3 *Finite State Machine*

Implementasi *Finite State Machine* di *game* ini adalah untuk mengatur perilaku karakter NPC (*Non Playable Character*). Sedangkan perilaku karakter utama mengikuti perintah orang yang bermain *game*. Pada permainan ini terdapat tiga karakter NPC, yaitu NPC Musuh, Musuh Terbang dan NPC Karakter BOSS. Pergerakan NPC otomatis sesuai dengan yang diprogramkan, FSM berikut ini menggambarkan perilaku NPC Musuh Virus, Musuh Virus Terbang dan NPC BOSS.



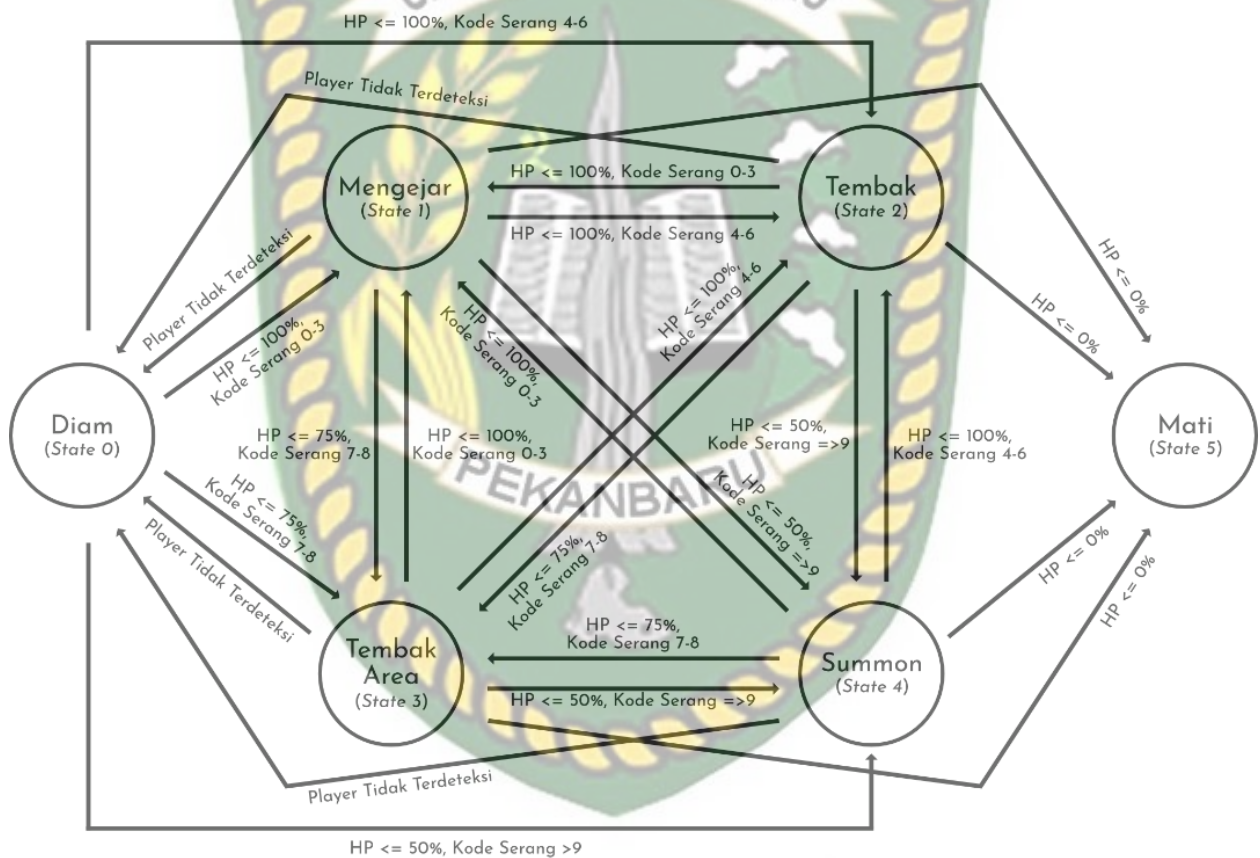
Gambar 3.13 *Finite State Machine* Musuh

Seperti terlihat pada Gambar 3.12, ketika sistem pertama kali dijalankan maka sistem FSM pada musuh virus berada pada *state 0* (berpatroli), jika musuh mendeteksi keberadaan pemain di depannya musuh akan memasuki *state 1* (mengejar) dan mulai mendekati player secara cepat, jika player menjahui dan keluar dari arah pandang musuh, maka musuh akan kembali ke *state 0* atau kembali berpatroli dan ketika HP (*health point*) musuh kecil atau sama dengan 0 maka *state 2* (mati) akan dijalankan.



Gambar 3.14 *Finite State Machine* Musuh Terbang

Seperti terlihat pada Gambar 3.13, musuh terbang tidak jauh berbeda dengan musuh biasa hanya saja musuh terbang mendeteksi karakter pemain yang berada dibawahnya, yang awalnya *state 0* musuh terbang juga berpatroli akan mulai menembakkan sebuah serangan ke arah bawah *state 1* (menembak) dan ketika HP (*health point*) musuh terbang kecil atau sama dengan nol maka musuh akan memasuki *state 2* (mati).



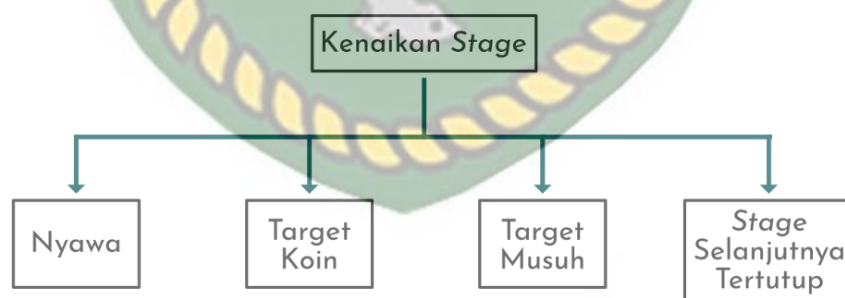
Gambar 3.15 Finite State Machine Musuh Boss

Pergerakan Boss diatur menggunakan *random number generator* dari 0 sampai 10 dan jumlah HP pada Boss jika sistem pertama kali dijalankan dan boss tidak mendeteksi player maka FSM berada pada *state 0* (diam) dan jika Boss mendeteksi pemain maka akan mengacak nomor dan juga tergantung HP Boss. Jika

HP boss dibawah atau sama dengan 100% keatas dan pengacakan nomor angka 0 sampai 3 maka Boss FSM berada pada *state* 1 (mengejar dan serangan jarak dekat), jika HP Boss dibawah atau sama dengan 100% dan pengacakan nomor 4 sampai 6 maka FSM Boss berjalan pada *state* 2 (menembak atau serangan jarak jauh), jika HP Boss dibawah atau sama dengan 75% dan pengacakan nomor menunjukkan angka 7 sampai 8 maka FSM Boss berada pada *state* 3 (serangan menembak area), jika HP Boss dibawah atau sama dengan 50% dan pengacakan menunjukkan angka 9 atau lebih maka FSM berada pada *state* 4 (serangan *summon*), dan jika HP Boss lebih kecil atau sama dengan 0% maka FSM berada pada *state* 5 (mati).

3.4 Metode *Forward Chaining*

Pada *game* ini metode *Forward Chaining* digunakan untuk menentukan kondisi untuk membuka *stage* berikutnya. Ada beberapa komponen yang mempengaruhinya berdasarkan Algoritma *Forward Chaining*, yaitu nyawa, target koin, dan target musuh, *stage* selanjutnya tertutup.



Gambar 3.16 Kriteria Kenaikan *Stage*

Berikut adalah variabel yang digunakan untuk sebagai parameter dalam *forward chaining* untuk pengembangan *game* SIGNA:

Tabel 3.2 Variable Rule

No	Kode Rule	Kondisi	Keterangan
1.	R01	[≤ 0 , > 0]	Nyawa
2.	RXX

Tabel 3.3 Variable Hasil

No	Kode Rule	Review
1.	H01	Tantangan Berhasil, Mendapat Uang, Buka Stage Selanjutnya
2.	HXX	...

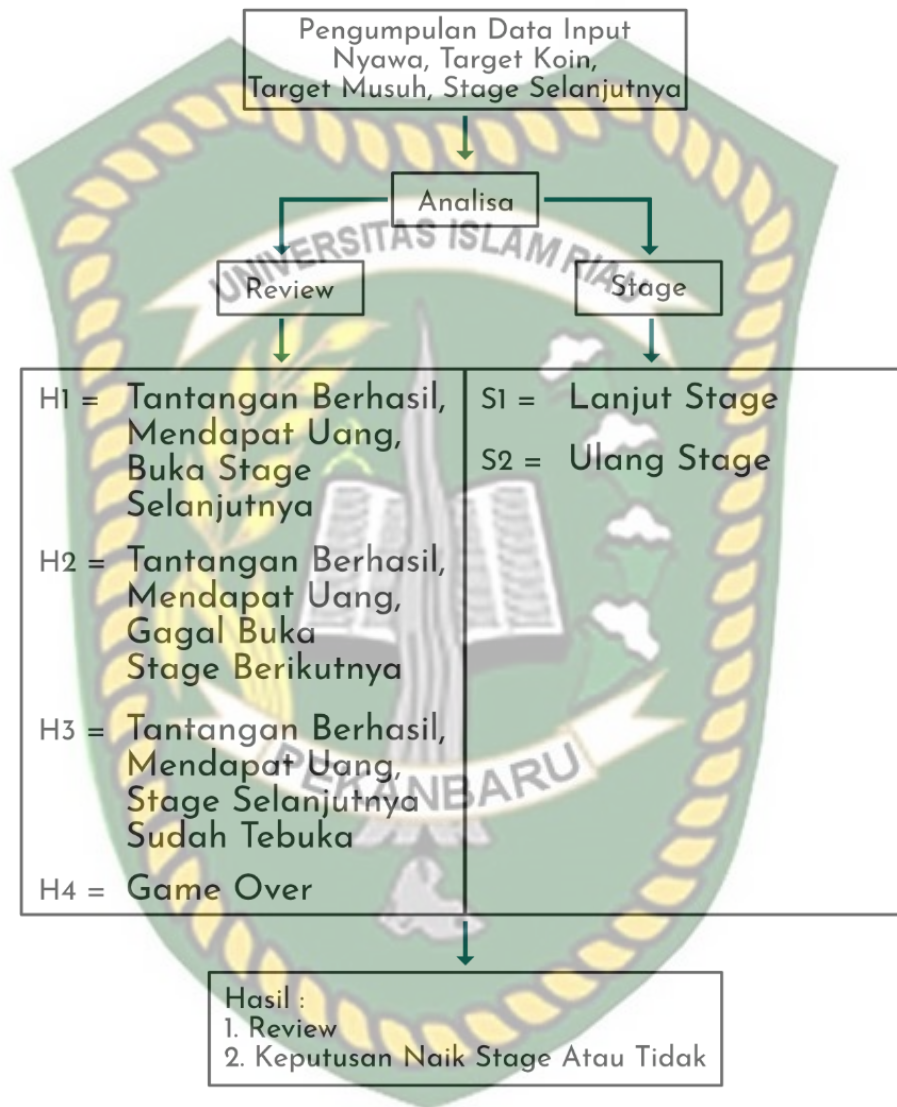
Tabel 3.4 Variable Stage

No	Kode Rule	Keterangan
1.	S01	Lanjut Stage
2.	SXX	...

3.5 Proses Representasi Cara Kerja Game

Proses representasi pengetahuan dilakukan dengan cara mengumpulkan kondisi untuk memperoleh hasil. Berdasarkan data input yang diperoleh, dilakukan

analisis untuk menentukan *review* apa yang didapat dalam setiap levelnya serta layak atau tidaknya *player* untuk naik ke *stage* berikutnya dalam *game*.



Gambar 3.17 Proses Representasi Cara Kerja *Game*

Hasil representasi pengetahuan *rules* kondisi disajikan pada Tabel 3.2, representasi pengetahuan *review* disajikan pada Tabel 3.3, sedangkan representasi pengetahuan kenaikan *stage* disajikan pada Tabel 3.4.

Tabel 3.5 Representasi *Rule*

No	Kode Rule	Kondisi	Keterangan
1.	R01	[$\leq 0, > 0$]	Nyawa
2.	R02	[$< \text{Target}, \geq \text{Target}$]	Target Koin
3.	R03	[$< \text{Target}, \geq \text{Target}$]	Target Musuh
4.	R04	[TB, TT]	Stage Selanjutnya

Tabel 3.6 Representasi Hasil

No	Kode Rule	Review
1.	H01	Tantangan Berhasil, Mendapat Uang, Buka Stage Selanjutnya
2.	H02	Tantangan Berhasil, Mendapat Uang, Gagal Buka Stage Selanjutnya
3.	H03	Tantangan Berhasil, Mendapat Uang, Stage Selanjutnya Sudah Terbuka Sebelumnya
4.	H04	Game Over

Tabel 3.7 Representasi *Stage*

No	Kode Rule	Keterangan
1.	S01	Lanjut <i>Stage</i>
2.	S02	Ulang <i>Stage</i>

Berikut adalah hasil representasi kedalam tabel aturan yang akan digunakan untuk proses analisa pada *game* SIGNA:

Tabel 3.8 Aturan Berdasarkan Kenaikan *Stage*

No	Kode <i>Stage</i>	Keterangan
1.	H01	Lanjut <i>Stage</i>
2.	H02, H03, H04	Ulang <i>Stage</i>

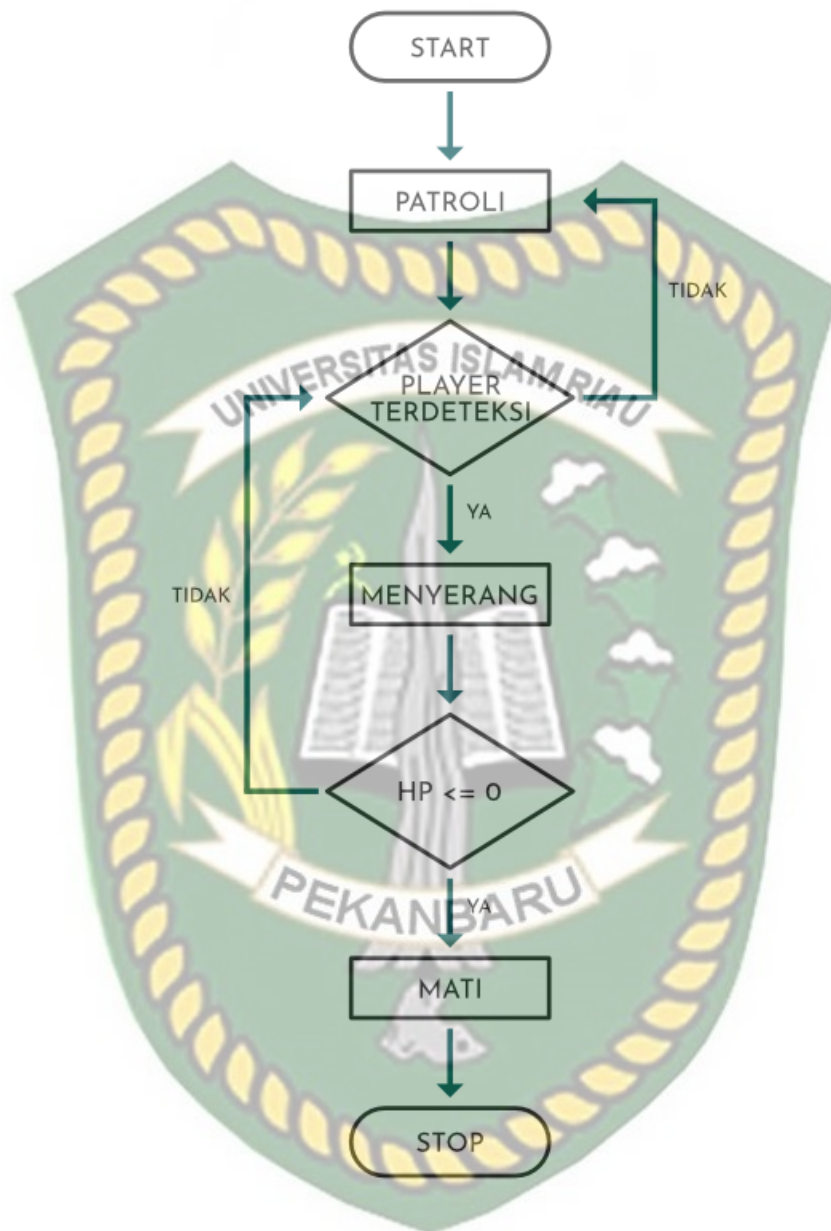
Tabel 3.9 Aturan Berdasarkan *Review*

No	<i>IF</i>	<i>THEN</i>
1.	R01 [>0] AND R02 [\geq Target] AND R03 [\geq Target] AND R04 [TT] Nyawa diatas 0, koin lebih atau sama dengan target, musuh lebih atau sama dengan target, <i>stage</i> selanjutnya masih tertutup	H01 Tantangan Berhasil, Mendapat Uang, Buka <i>Stage</i> Selanjutnya

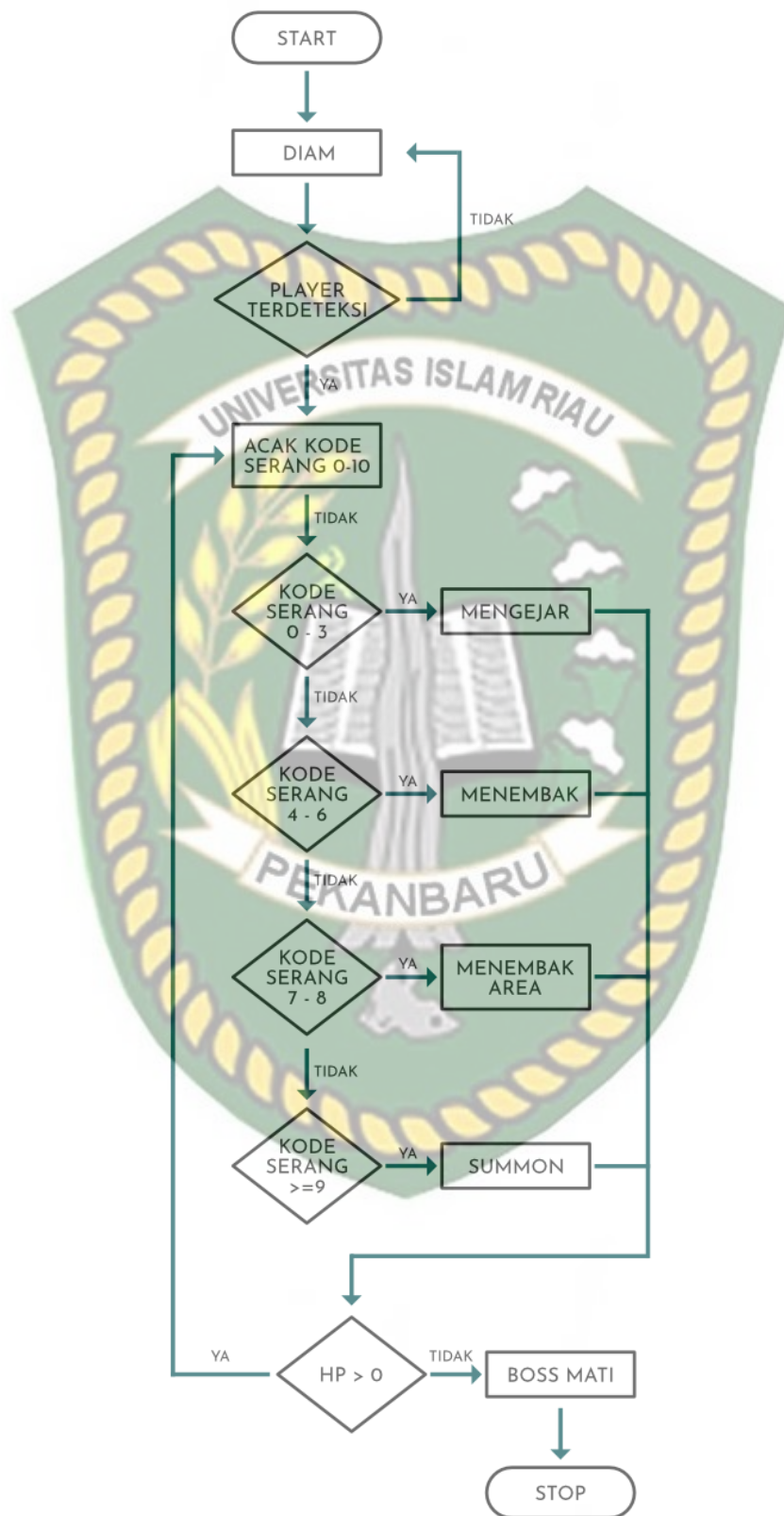
2.	<p>R01 [>0] AND R02 [$<Target$] OR R03 [$<Target$] AND R04 [TT]</p> <p>Nyawa diatas 0, koin kurang dari target atau musuh kurang dari target, <i>stage</i> selanjutnya masih tertutup</p>	<p>H02</p> <p>Tantangan Berhasil, Mendapat Uang, Gagal Buka Stage Selanjutnya</p>
3.	<p>R01 [>0] AND R02 [$\geq Target$] OR R02 [$<Target$] OR R03 [$\geq Target$] OR R03 [$<Target$] AND R04 [TB]</p> <p>Nyawa diatas 0, koin lebih atau sama dengan atau kurang dari target, musuh lebih atau sama dengan atau kurang dari target, <i>stage</i> selanjutnya telah terbuka</p>	<p>H03</p> <p>Tantangan Berhasil, Mendapat Uang, Stage Selanjutnya Sudah Terbuka Sebelumnya</p>
4.	<p>R01 [≤ 0] AND R02 [$\geq Target$] OR R02 [$<Target$] OR R03 [$\geq Target$] OR R03 [$<Target$] AND R04 [TT] OR R04 [TB]</p> <p>Nyawa kurang atau sama dengan 0, koin lebih atau sama dengan atau kurang dari target, musuh lebih atau sama dengan atau kurang dari target, <i>stage</i> selanjutnya masih tertutup atau telah terbuka</p>	<p>H04</p> <p>Game Over</p>

3.6 Desain Logika Program

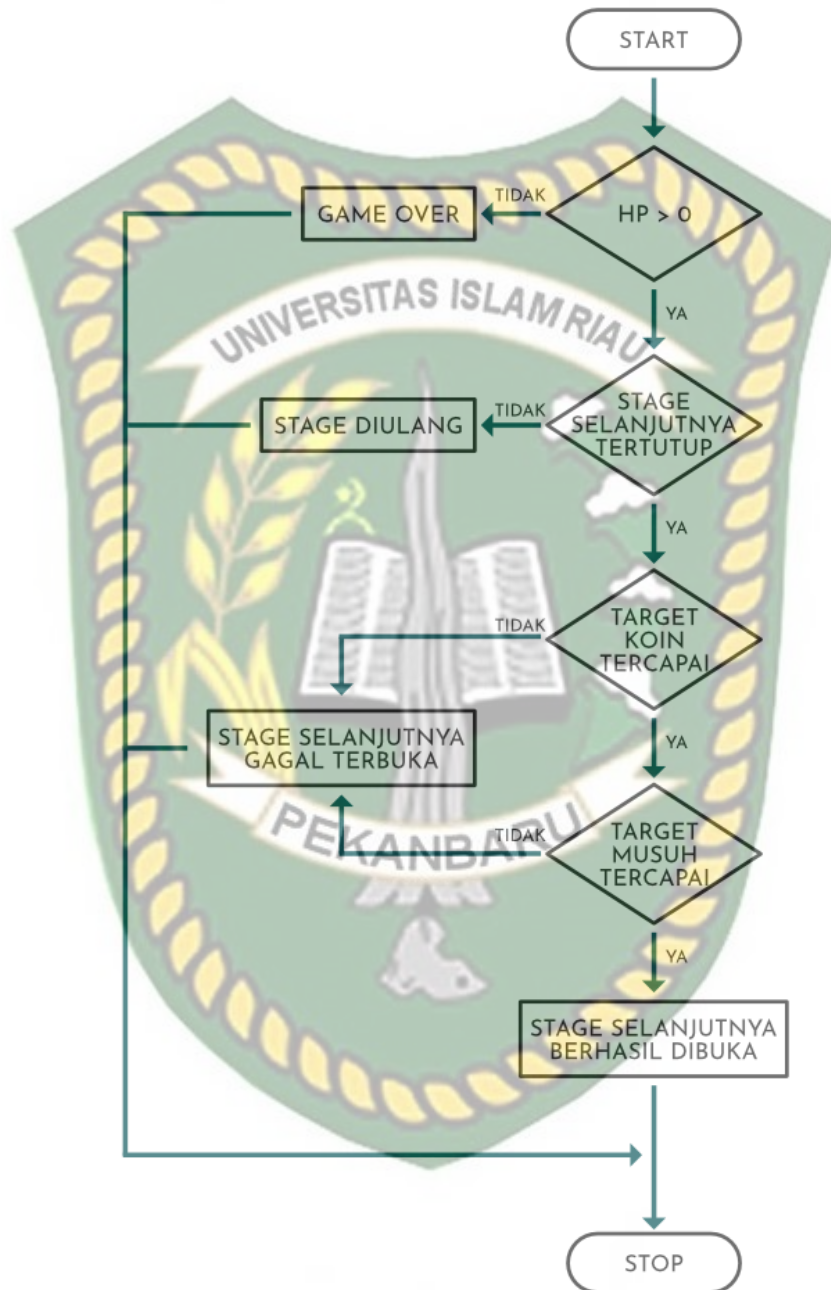
Dalam merancang sebuah sistem, pengembangan alur adalah hal yang sangat penting dalam memahami proses dari sebuah sistem. Pada tahap ini akan digambarkan alur proses dalam bentuk *Flowchart*.



Gambar 3.18 *Flowchart Finite State Machine Musuh Virus Biasa*



Gambar 3.19 Flowchart Finite State Machine Musuh Boss



Gambar 3.20 Flowchart Metode Forward Chaining

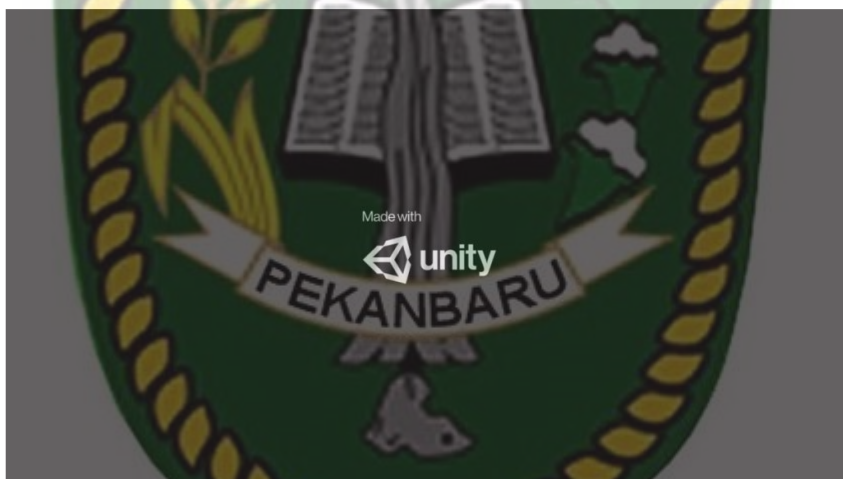
BAB IV

HASIL DAN PEMBAHASAN

4.1 Pengujian Masalah Yang Sedang Berjalan

Sebelum sistem digunakan oleh *user* maka sistem harus dilakukan pengujian terlebih dahulu. Pengujian sistem dilakukan untuk mengetahui hasil akhir atau output yang berupa informasi. Untuk mengetahui hasil output dari sistem dan *interface* dari keseluruhan *game*.

4.1.1 Tampilan Awal Aplikasi



Gambar 4.1 Tampilan Awal Aplikasi

Gambar 4.1 adalah tampilan halaman awal saat aplikasi dijalankan. Tampilan awal akan menampilkan *splash screen* dari *Unity Engine*, tampilan tersebut dibuat otomatis dari *Unity Engine* sebagai *game engine* yang digunakan. selanjutnya akan muncul dapat dilihat gambar 4.2 menu judul.

4.1.2 Tampilan Menu Judul

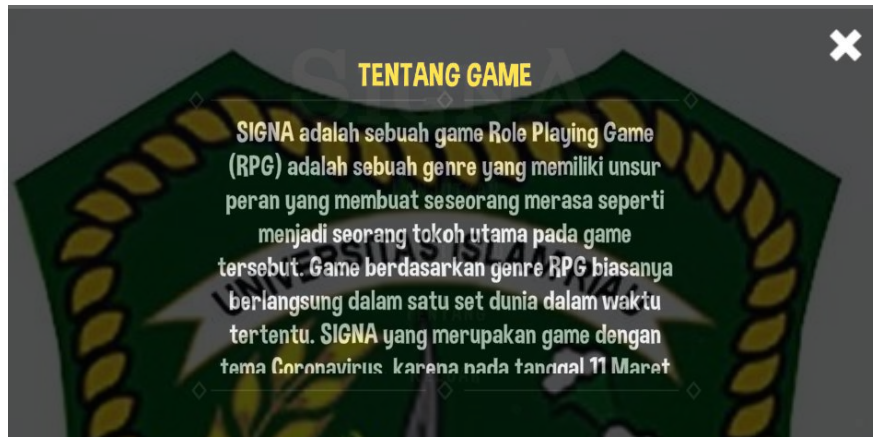


Gambar 4.2 Tampilan Menu Judul

Pada gambar 4.2 menu judul aplikasi juga memiliki beberapa *Button*, ada *Button* Lanjutkan, Mulai Baru, Tentang, Keluar. Berikut fungsi dan kegunaan dari *button* pada menu judul yaitu :

1. *Button* Lanjutkan digunakan untuk masuk kedalam tampilan menu utama jika pemain telah bermain sebelumnya dan *button* akan tidak aktif atau tidak dapat digunakan jika pemain pertama kali bermain.
2. *Button* Mulai Baru digunakan untuk masuk kedalam tampilan menu utama sebagai pemain baru atau mengulang permainan.

3. *Button* Tentang untuk menampilkan keterangan singkat dari *game* ini bisa dilihat pada gambar 4.3 dan profil personal dari pembuat *game*.



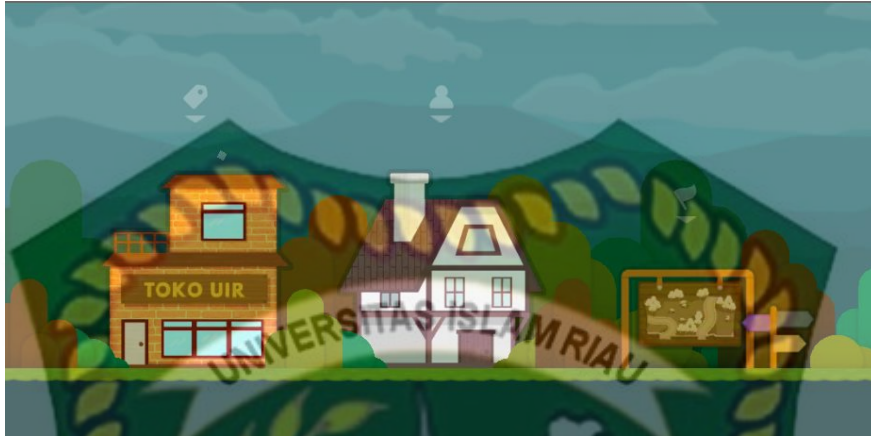
Gambar 4.3 Tampilan Menu Tentang

4. *Button* Keluar untuk menampilkan menu pilihan menutup atau keluar dari *game* dapat dilihat pada gambar 4.4.



Gambar 4.4 Tampilan Menu Keluar

4.1.3 Tampilan Menu Utama



Gambar 4.5 Tampilan Menu Utama

Pada gambar 4.5 menu utama aplikasi juga memiliki beberapa *Button*, ada *Button* Toko, Profil, dan Peta. Berikut ini fungsi dan kegunaan dari *button* pada *scene* menu utama yaitu :

1. *Button* Toko digunakan untuk masuk kedalam tampilan menu toko dapat dilihat pada gambar 4.6 pemain dapat bisa membeli perlengkapan untuk karakternya pada menu ini.



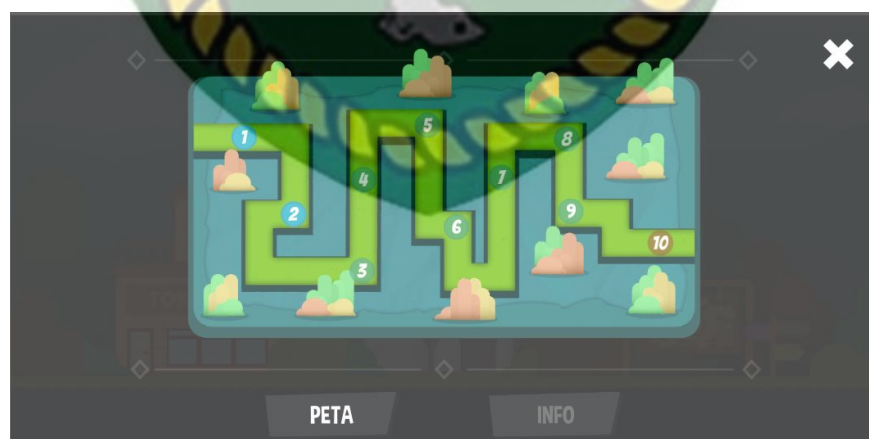
Gambar 4.6 Tampilan Menu Toko

2. *Button* Profil digunakan untuk masuk kedalam tampilan menu profil dapat dilihat pada gambar 4.7 pada menu ini pemain dapat meningkatkan status karakter mereka, mengganti perlengkapan, dan melihat koleksi yang pemain telah dapatkan.



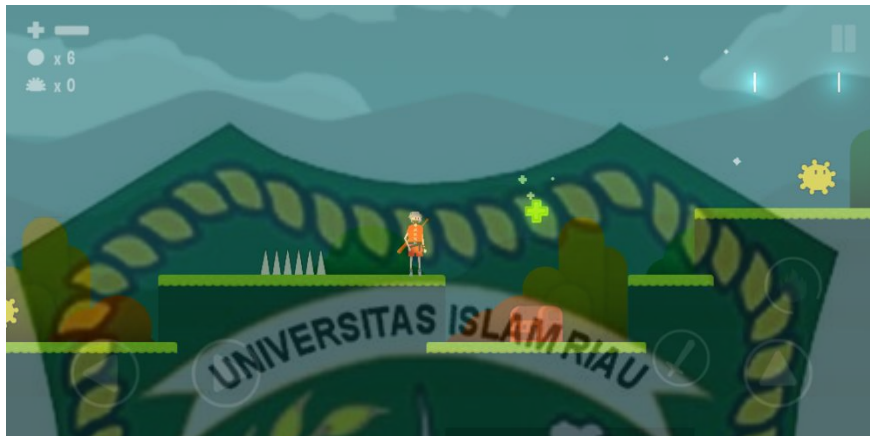
Gambar 4.7 Tampilan Menu Profil

3. *Button* Peta digunakan untuk menampilkan menu pilihan *stage* dan info sebelum masuk ke tantangan bisa dilihat pada gambar 4.8.



Gambar 4.8 Tampilan Menu Peta

4.1.4 Tampilan Menu Permainan

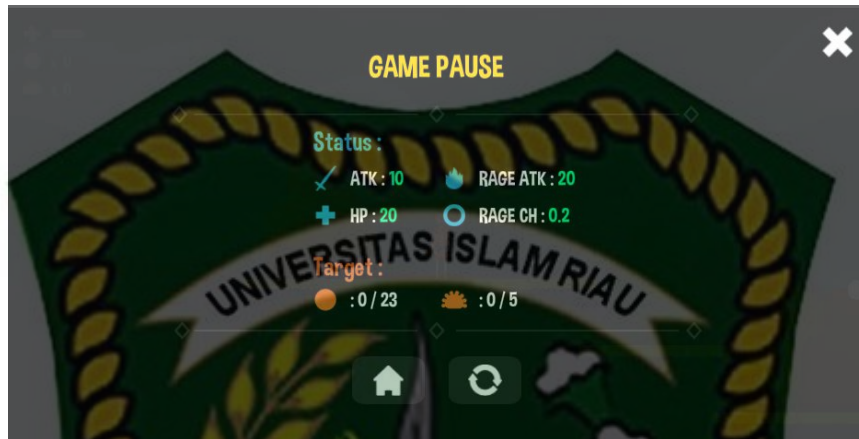


Gambar 4.9 Tampilan Menu Permainan

Pada gambar 4.9 menu permainan memiliki beberapa *Button*, ada *Button* Arah Kiri, Arah Kanan, Lompat, Serang, *Rage Mode*, dan *Pause*. Berikut fungsi dan kegunaan dari *button* pada menu permainan yaitu :

1. *Button* Arah Kiri digunakan untuk mengerakkan karakter pemain ke arah kiri.
2. *Button* Arah Kanan digunakan untuk mengerakkan karakter pemain ke arah kanan.
3. *Button* Lompat digunakan untuk mengerakkan karakter ke arah atas atau melompat.
4. *Button* Serang digunakan untuk menyerang musuh atau membuka peti.
5. *Button* *Rage Mode* dapat digunakan agar karakter masuk ke *rage mode* serangan karakter menjadi berlipat ganda.

6. *Button Pause* digunakan untuk menghentikan permainan sementara pada area tantangan dan akan menampilkan menu yang bisa dilihat pada gambar 4.10.



Gambar 4.10 Tampilan Menu *Pause*

4.2 Pengujian Black Box

Pengujian *black box* adalah metode pengujian dimana data tes berasal dari persyaratan fungsionalitas yang ditentukan tanpa memperhatikan struktur program akhir (Komarudin, 2016).

Pada perancangan sistem pendukung keputusan ini, pengujian merujuk pada fungsi-fungsi yang dimiliki sistem, kemudian membandingkan hasil program dengan hasil yang diharapkan. Bila hasil sesuai dengan yang diharapkan dari hasil pengujian, berarti hal ini perangkat lunak telah sesuai dengan desain yang telah diharapkan sebelumnya. Bila belum sesuai yang diharapkan maka perlu dilakukan pengecekan lebih lanjut dan perbaikan. Berikut ini pengujian *black box* terhadap perancangan *game* SIGNA berbasis android.

4.2.1 Pengujian *Black Box* Pada Menu Judul

Pada menu judul terdapat *button* atau tombol pilihan yang akan menampilkan setiap halaman atau fungsi dari *button* yang dipilih. Halaman judul merupakan *scene* dari aplikasi yang akan menampilkan *scene* sesuai pilihan *button* yang telah ditekan. Berikut hasil pengujian *button* dan menu yang dapat dilihat pada tabel 4.1.

Tabel 4.1 Hasil Pengujian *Blackbox* Menu Judul

Skenario	Tindakan	Fungsi Sistem	Yang Diharapkan	Hasil	
				Gagal	Berhasil
Tombol Lanjutkan	Tekan tombol Lanjutkan	Untuk masuk ke menu utama pada <i>game</i> , menjalankan animasi transisi antar <i>scene</i>	Menampilakan <i>scene</i> menu utama		✓
Tombol Mulai Baru	Tekan tombol Mulai Baru	Untuk masuk ke menu utama pada <i>game</i> , menjalankan animasi transisi antar <i>scene</i>	Menampilakan <i>scene</i> menu utama dan <i>intro game</i>		✓
Tombol Tentang	Tekan tombol Tentang	Untuk menampilkan menu tentang <i>game</i>	Menampilakan informasi tentang <i>game</i>		✓

Tombol Keluar	Tekan tombol keluar	Untuk menampilkan menu konfirmasi ingin keluar dari <i>game</i>	Menampilaka n menu konfirmasi keluar <i>game</i>		✓
------------------	---------------------------	--	---	--	---

4.2.2 Pengujian *Black Box* Pada Menu Utama

Pada menu utama terdapat *button* atau tombol pilihan yang akan menampilkan setiap halaman atau fungsi dari *button* yang dipilih. Halaman menu utama merupakan *scene* dari aplikasi yang akan menampilkan *scene* sesuai pilihan *button* yang telah ditekan. Berikut hasil pengujian *button* dan menu yang dapat dilihat pada tabel 4.2.

Tabel 4.2 Hasil Pengujian *Blackbox* Menu Utama

Skenario	Tindakan	Fungsi Sistem	Yang Diharapkan	Hasil	
				Gagal	Berhasil
Tombol Toko	Tekan tombol Toko	Untuk menampilkan menu toko	Menampilakan menu toko dan jumlah uang		✓
Tombol Beli	Tekan tombol Beli	Untuk menampilkan menu konfirmasi membeli perlengkapan	Menampilkan menu konfirmasi beli atau uang tidak cukup		✓

Tombol Profil	Tekan tombol Mulai Baru	Untuk menampilkan menu profil	Menampilkan menu profil berisi menu status, perlengkapan, dan koleksi	✓
Tombol Status	Tekan tombol Status	Untuk menampilkan menu status karakter	Menampilkan <i>level</i> status dan informasi status karakter	✓
Tombol Perlengkapan	Tekan tombol Perlengkapan	Untuk menampilkan menu Perlengkapan karakter	Menampilkan karakter dan perlengkapan yang digunakan	✓
Tombol Koleksi	Tekan tombol Koleksi	Untuk menampilkan menu koleksi karakter	Menampilkan nama koleksi dan deskripsi dari koleksi	✓
Tombol Peta	Tekan tombol peta	Untuk menampilkan menu peta pilih area	Menampilkan pilihan area permainan yang dapat dimasuki	✓

Tombol Info	Tekan tombol info	Untuk menampilkan menu info tentang area tantangan	Menampilkan informasi tentang koin, peti, musuh, dan lainnya	✓
-------------	-------------------	--	--	---

4.2.3 Pengujian *Black Box* Pada Menu Permainan

Pada menu permainan terdapat beberapa *button* atau tombol yang akan menampilkan setiap halaman atau fungsi dari *button* yang dipilih. Halaman menu permainan merupakan *scene* dari aplikasi yang akan menampilkan *scene* sesuai pilihan *button* yang telah ditekan. Berikut hasil pengujian *button* dan menu yang dapat dilihat pada tabel 4.3.

Tabel 4.3 Hasil Pengujian *Blackbox* Menu Permainan

Skenario	Tindakan	Fungsi Sistem	Yang Diharapkan	Hasil	
				Gagal	Berhasil
Tombol Arah Kiri	Tekan tombol Arah Kiri	Untuk mengerakkan karakter ke arah kiri	Ketika ditekan karakter bergerak ke arah kiri dan berhenti ketika tombol dilepas		✓

Tombol Arah Kanan	Tekan tombol Arah Kanan	Untuk mengerakkan karakter ke arah kanan	Ketika ditekan karakter bergerak ke arah kanan dan berhenti ketika tombol dilepas	✓
Tombol Lompat	Tekan tombol lompat	Untuk mengerakkan karakter ke arah atas atau melompat	Ketika ditekan dan karakter berdiri di atas <i>platform</i> , karakter melompat	✓
Tombol Serang	Tekan tombol serang	Untuk mengeluarka n serangan di depan karakter	Ketika ditekan karakter memainkan animasi serang untuk menyerang musuh atau membuka peti	✓
Tombol <i>Rage Mode</i>	Tekan tombol <i>rage mode</i>	Untuk melipat gandakan status serangan	Ketika ditekan karakter akan memasuki <i>rage mode</i>	✓

		karakter			
Tombol <i>Pause</i>	Tekan tombol <i>pause</i>	Untuk menghentikan permainan pada area tantangan sementara	Menampilkan menu <i>pause</i> , tombol kembali menu utama, dan ulang area tantangan		✓

4.3 Implementasi Sistem

Implementasi sistem dilakukan dengan memberikan kuisioner kepada 50 orang dengan tujuan untuk mengetahui tanggapan dari pengguna tentang Aplikasi *game* RPG SIGNA berbasis android. Hasil implementasi dengan memberikan kuisioner kepada 50 orang dapat dilihat pada tabel 4.5.

Tabel 4.4 Hasil Implementasi Sistem

NO	Pertanyaan	Jumlah Persentase Responden		
		Sangat Baik	Baik	Kurang Baik
1	Kesesuaian dan kualitas desain pada game	27	21	2

2	Ketepatan fungsi tombol pada game dengan tujuan yang ingin ditampilkan	28	20	2
3	Kualitas tampilan animasi	27	17	6
4	Kesesuaian keterangan atau informasi penjelasan di dalam game	32	16	2
5	Kesesuaian audio dalam game	32	16	2
6	Manfaat game sebagai media hiburan sekaligus mengedukasi tentang Corona Virus.	34	15	1
Total		180	105	15

4.4 Uji Kelayakan Sistem

Pengujian kelayakan sistem dirancang untuk mendapatkan penilaian langsung terhadap respon sistem. Tahapan uji kelayakan ini adalah angket dan tabulasi data.

4.4.1 Angket

Angket Berisi pertanyaan-pertanyaan atau kuisoner yang berkaitan dengan sistem yang dibangun. Kuesioner Uji Kelayakan Sistem menggunakan skala Likert yang dimodifikasi dimana responden dapat memilih dari tiga jawaban yang tersedia, yaitu:

SB (Sangat Baik) : 3 (tiga)

B (Baik) : 2 (dua)

KB (Kurang Baik) : 1 (satu)

4.4.2 Tabulasi Data

Kuesioner yang telah disiapkan kemudian dibagikan kepada responden. Teknik pemilihan responden (sampel) dilakukan melalui metode *simple random sampling*, yaitu pemilihan sampel secara acak atau *random*. Dalam menentukan kategori penilaian untuk menentukan tingkat kelayakan sistem ini bisa dilihat tabel 4.5.

Tabel 4.5 Kategori Penilaian Skala Likert

Skor Interval	Kategori
2,1 - 3	Sangat Baik
1,1 - 2	Baik
0,1 - 1	Kurang Baik

Setelah memperoleh data angket, kemudian dianalisis menggunakan skala *likert* dengan menghitung jumlah pada setiap variabel. Berikut hasil penilaian dari pengujian terhadap pengguna pada tabel 4.6.

Tabel 4.6 Hasil Penilaian Variabel

NO	Variabel	Jumlah Persentase Responden			Rata-rata Kategori
		Sangat Baik	Baik	Kurang Baik	
1	Kesesuaian dan kualitas desain pada game	27	21	2	2.5
2	Ketepatan fungsi tombol pada game dengan tujuan yang ingin ditampilkan	28	20	2	2.52
3	Kualitas tampilan animasi	27	17	6	2.42
4	Kesesuaian keterangan atau informasi penjelasan di dalam game	32	16	2	2.6
5	Kesesuaian audio dalam game	32	16	2	2.6

6	Manfaat game sebagai media hiburan sekaligus mengedukasi tentang Corona Virus.	34	15	1	2.66
Total		180	105	15	
Presentasi Rata-rata		60%	35%	5%	
Total Rata-rata Kategori					2.55
Kategori					Sangat Baik

Terlihat pada tabel 4.6 bahwa penilaian terhadap variabel tampilan mendapatkan nilai rata-rata sebesar 2,55. Berdasarkan kategori penilaian pada tabel 4.5 nilai rata-rata sebesar 2,55 berada dalam rentang nilai 2,1 - 3. jadi bisa disimpulkan bahwa penilaian variabel pada sistem termasuk dalam kategori “Sangat Baik”.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Penelitian dan pembuatan aplikasi game RPG (*Role Playing Game*) SIGNA berbasis android telah dilaksanakan dan telah dilakukan serangkaian pengujian dari aplikasi tersebut dan didapatkan hasil sebagai berikut:

1. *Game* SIGNA sebagai media hiburan dan juga mengedukasi tentang *Corona Virus Disease* yang tersedia di android.
2. Aplikasi *game* menggunakan metode *forward chaining* sebagai penentuan pemain dapat membuka tantangan selanjutnya.
3. Aplikasi *game* juga menggunakan metode *finite state machine* sebagai pengatur perilaku karakter musuh dalam *game*.

5.2 Saran

Aplikasi game RPG (*Role Playing Game*) SIGNA berbasis android masih memerlukan pengembangan yang lebih baik, maka untuk pengembangan selanjutnya dapat menambahkan beberapa pengembangan sebagai berikut:

1. Menambahkan lebih banyak tantangan level pada *game*.
2. Menambahkan lebih banyak variasi perlengkapan karakter, musuh, dan boss di dalam *game*.
3. Mengembangkan *game* ini ke *platform* yang berbeda, contohnya seperti iOS maupun desktop.

DAFTAR PUSTAKA

- Adiwikarta, Rendy, and Harya Bima Dirgantara. 2017. "Pengembangan Permainan Video Endless Running Berbasis Android Menggunakan Framework Game Development Life Cycle." *Pengembangan Permainan Video Endless Running Berbasis Android Menggunakan Framework Game Development Life Cycle* 4.
- Dewi, Ghea Putri Fatma. 2012. "Pengembangan Game Edukasi Pengenalan Nama Hewan Dalam Bahasa Inggris Sebagai Media Pembelajaran Siswa Sd Berbasis Macromedia Flash."
- Firdaus, Sidik. 2014. "Implementasi Metode Forward Chaining Sebagai Sidik Firdaus Jurnal Ilmiah Komputer Dan Informatika (KOMPUTA)." *Jurnal Ilmiah Komputer Dan Informatika (KOMPUTA)*.
- Fitrul Hadi, Aulia. 2018. "Analisa Ketangkapan Daya Ingat Anak Dengan Menggunakan Game Edukasi Non Player Character 'Ani!' Berbasis Android."
- Ghozali, Imam, and Alexius Endy Budianto. 2014. "Game Edukasi Bahasa Jawa Kerajaan Islam Jawa." *Jurnal Mahasiswa Fakultas Sains Dan Teknologi*.
- Kemenkes. 2020. "Tanya Jawab Seputar Virus Corona (Covid-19)."
- Mau, Guido. 2019. "Rancang Bangun Game 2D Shooter Platformer Menggunakan Metode Finite State Machine." *JATI (Jurnal Mahasiswa Teknik Informatika)*.
- Mongi, Laurent S., Arie S.M. Lumenta, and Alwin M. Sambul. 2018. "Rancang Bangun Game Adventure of Unsrat Menggunakan Game Engine Unity." *Jurnal Teknik Informatika* 13 (1).

- Muhamad Rumakey, Ary, Joseph Dedi Irawan, and Abdul Wahid. 2020. "Pembuatan Game 2D 'Escape Plan' Dengan Metode Finite State Machine." *JATI (Jurnal Mahasiswa Teknik Informatika)*.
- Nurafifah, Risti. 2018. "Penerapan Algoritma Fisher Yates Shuffle Dan Algoritma Forward Chaining Pada Game Edukasi Mr Garbage." *Penerapan Algoritma Fisher Yates Shuffle Dan Algoritma Forward Chaining Pada Game Edukasi Mr Garbage*.
- Program Studi Teknik Informatika. 2014. "Buku Pedoman Penulisan Skripsi".
- Putra, Yogie Susdyastama, M Aziz Muslim, and Agus Naba. 2013. "Game Chicken Roll Dengan Menggunakan Metode Forward Chaining." *Jurnal EECCIS 7*.
- Rahadian, Miftah Fauzan, Addy Suyatno, and Septya Maharani. 2016. "Penerapan Metode Finite State Machine Pada Game 'The Relationship.'" *Informatika Mulawarman : Jurnal Ilmiah Ilmu Komputer*.
- Rambe, Muhammad Yusuf, Mhd Rusdi Tanjung, and Alfa Saleh. 2020. "Perancangan Aplikasi Game Cat Volly Berbasis Android." *Jurnal Mahasiswa Fakultas Teknik Dan Ilmu Komputer 2*.
- Rohmawati, Indah, Sudargo, and Ika Menarianti. 2019. "Development of the Archipelago Cultural Education Game 'TANARA' Using Unity 3D Based on Android." *Jurnal Sistem Informasi Dan Teknologi 2*.
- Saputri, Fiqih Hana, and Dian Pratiwi. 2016. "Pembuatan Game RPG 'Roro Jonggrang' Dengan RPG Maker MV." *Seminar Nasional Cendekiawan*.

Sugianto, and veicy Andri Ansyah. 2018. "Implementasi Metode Forward Chaining Untuk Mengelompokkan Objek Berdekatan Pada Same Game." *Jurnal Manajemen Informatika & Komputerisasi Akuntansi*.

Vinet, Luc, and Alexei Zhedanov. 2011. "Perancangan Sistem Informasi Pendukung Keputusan Bantuan Langsung Tunai (Blt) Covid-19 Pada Warga Kelurahan Babelan."

Yudhantara, Dhamar Seto. 2017. "Implementasi Metode Linear Congruent Method (Lcm) Pada Game Cerdas Cermat Untuk Meningkatkan Motivasi Belajar Siswa Sd Kelas Iv Pada Mata Pelajaran Ilmu Pengetahuan Alam (Ipa),"

Zakharia, Jodi, Indah Fenriana, and Ellysha Dwiyanthi Kusuma. 2020. "Aplikasi Pengenalan Suara Hewan Pada TK Pelangi Dengan Metode Forward Chaining."

