

GAME RPG “THE ROYAL SWORD” BERBASIS DESKTOP DENGAN MENGGUNAKAN METODE *FINITE STATE MACHINE* (FSM)

Fadel Marzian, Mukti Qamal

Teknik Informatika Universitas Malikussaleh Lhokseumawe
Jl. Cot Tgk Nie-Reulet, Aceh Utara, 141 Indonesia
email : fadelmarzian024@gmail.com, mukti.qamal@gmail.com

ABSTRAK

Game adalah permainan yang menggunakan media elektronik dan sebagai media hiburan berbentuk multimedia yang di buat semenarik mungkin agar pemain bisa mendapatkan kepuasan. Penelitian ini bertujuan untuk membuat sebuah *game* RPG yang berjudul “*The Royal Sword*” yaitu *game* yang bertemakan kerajaan dimana pada *game* ini terdapat beberapa misi dan tujuan utama untuk mengembalikan pedang kerajaan yang hilang. Adapun metodologi yang digunakan adalah metode *Finite State Machine* (FSM), yaitu sebuah metodologi perancangan sistem kontrol yang menggambarkan tingkah laku atau prinsip kerja sistem dengan menggunakan tiga hal berikut: *State* (Keadaan), *Event* (kejadian) dan *action* (aksi). Sebagai sebuah metodologi perancangan sistem control. penerapan *Finite State Machine* (FSM) sangat cocok digunakan pada *game*. Oleh karena itu penggunaan metode *Finite State Machine* (FSM) pada *game* RPG *The Royal Sword* berfungsi untuk batas perpindahan misi atau *level* agar *game* menjadi lebih teratur dan tertata dengan penyelesaian secara bertahap dan berurutan.

Kata kunci : *Game, RPG, The Royal Sword, Finite State Machine.*

1.Pendahuluan

Perkembangan teknologi belakangan ini begitu pesat, terutama dalam bidang komputer. Hampir semua bidang kehidupan bisa di simulasikan dalam bentuk visual dengan komputer. Kehidupan manusia, sejarah, budaya bahkan imajinasi pun bisa di simulasikan dalam bentuk *game*.

Berbeda dengan zaman dahulu. Sekarang perkembangan dari *game* sangatlah pesat, dengan adanya dukungan dari berbagai *platform* dalam memberikan pengembang *game* suatu media untuk menjual *game*, sehingga

saat ini *game-game* yang ada di pasar lokal maupun global sangat banyak, dengan kata lain persaingan dalam industri pengembang *game* masih terbuka luas.

Game merupakan salah satu industri yang besar saat ini, perkembangan *game* dapat menjadi alternatif hiburan untuk mengisi liburan maupun waktu luang dan menjadi kegiatan yang sangat menyenangkan bagi anak-anak maupun orang dewasa. Berbagai macam *game* yang berkembang pada saat ini berbasis *Desktop/PC* (*windows, Linux, Mac, dll*), *Mobile* (*Symbian, Java, Android, iOS, BlackBerryOS*), dan berbasis *Web* (*HTML5, Flash*). Dari sekian banyaknya *genre game* yang berkembang, salah satu jenis *genre game* yang banyak diminati ialah *game RPG* (*Role-Playing Game*). Jenis *genre* ini telah tumbuh dan berkembang menjadi jenis *genre* yang di perhitungkan. Tidak seperti *game* pada zaman sekarang yang mengandalkan Grafik, RPG lebih condong terhadap teka-teki, alur dan plot cerita.

RPG merupakan jenis *game* pilihan karena berisi unsur-unsur cerita yang komplek serta seni peran yang membuat *user* merasa seperti menjadi tokoh yang di perankannya dalam *game* tersebut. Oleh karena itu, *game* RPG memunyai daya tarik tersendiri. Hal ini dapat dibuktikan dari penjualan perangkat lunak RPG yang menembus angka jutaan dollar, seperti *Final Fantasy series, digimon world series*, dan lain-lain.

Berdasarkan uraian permasalahan di atas, maka dari itu penulis tertarik mengambil judul "***Game RPG "The Royal Sword" Berbasis Dekstop Dengan Menggunakan Metode Finite State Machine (FSM)***".

2.TINJAUAN PUSTAKA

2.1 Definisi GAME

Game merupakan permainan komputer yang dibuat dengan teknik dan metode animasi. Jika mendalami penggunaan animasi haruslah memahami pembuatan *game*. Jika ingin membuat *game*, maka haruslah memahami teknik dan metode animasi, sebab keduanya saling berkaitan. (Jeffrey & Johannes, 2014).

2.1.1 Pengertian Game

Dalam bahasa Indonesia *game* diartikan sebagai permainan. Permainan adalah kegiatan kompleks yang didalamnya terdapat peraturan, *Play* dan budaya. Sebuah permainan adalah sistem dimana pemain terlibat konflik buatan, disini pemain berinteraksi dengan sistem dan konflik dalam permainan merupakan rekayasa atau buatan (Lestari D, 2012).

Berdasarkan representasi visualnya, *game* dapat dibedakan menjadi 2 jenis, yaitu *game* 2 dimensi (2D) dan 3 dimensi (3D). *Game* 2D adalah *game*, yang secara matematis, hanya melibatkan 2 elemen koordinat kartesius, yaitu x dan y , sehingga konsep kamera pada *game* 2D hanya menentukan "gambar" mana pada *game* yang dapat dilihat oleh pemain. Sedangkan *game* 3D adalah *game* yang selain melibatkan elemen x dan y , juga melibatkan elemen z pada perhitungannya, sehingga konsep kamera pada *game* 3D benar-benar menyerupai konsep kamera pada kehidupan nyata, yaitu selain digeser (seperti pada *game* 2D), juga dapat diputar dengan sumbu tertentu.

2.1.2 Jenis-Jenis Game

Berikut ini akan dijabarkan beberapa jenis *game* berdasarkan cara pembuatannya, cara pemasarannya, mesin yang menjalankannya. Jenis *game* diantaranya adalah :

1. Game PC

Game yang dimainkan pada PC (*Personal Computer*) yang memiliki kelebihan yaitu memiliki tampilan antarmuka yang baik untuk *input* maupun *output*, *output* visual kualitas tinggi karena layar komputer biasanya memiliki resolusi yang jauh lebih tinggi dibandingkan dengan layar televisi biasa. Kekurangannya adalah spesifikasi komputer yang sangat bervariasi antara satu komputer dengan komputer lainnya menyebabkan beberapa *game* dapat ditampilkan dengan baik pada satu komputer, tetapi tidak berjalan dengan baik pada komputer yang lainnya.

2. Game Console

Game yang dijalankan pada suatu mesin spesifik yang biasanya tersedia di rumah pribadi, seperti *Microsoft Xbox 360*, *Nintendo Wii* dan sebagainya.

3. Game Arcade

Game yang dijalankan pada mesin dengan *input* dan *output* audio visual yang telah terintegrasi dan tersedia ditempat-tempat umum, seperti mal, bandara dan sebagainya.

4. *Game Online*

Game jenis ini hanya dapat dimainkan secara *online* melalui kabel LAN atau Menggunakan jaringan *internet* (Baihaqi Reza, 2015).

2.1.3 Berdasarkan *Genre* Permainannya

Game atau permainan adalah sesuatu yang dapat dimainkan dengan aturan tertentu sehingga ada yang menang dan ada yang kalah, biasanya dalam konteks tidak serius dengan tujuan *refreshing*. Macam-macam *game*, antara lain:

1. Aksi

Genre ini merupakan macam *game* yang paling populer. *Game* jenis ini membutuhkan kemampuan reflex pemain. Salah satu *subgenre* *action* yang populer adalah *First Person Shooter* (FPS). Pada *game* FPS diperlukan kecepatan berfikir. *Game* ini dibuat seolah-olah pemain yang berada dalam suasana tersebut.

2. Aksi Petualangan

Genre ini memadukan *game play* aksi dan petualangan. Contohnya pemain diajak untuk menelusuri gua bawah tanah sambil mengalahkan musuh, dan mencari artefak kuno, atau menyeberangi sungai.

3. Simulasi, Konstruksi dan Manajemen

Pemain dalam *game* ini diberi keleluasaan untuk membangun dan suatu proyek tertentu dengan bahan baku yang terbatas.

4. *Role Playing Games* (RPG)

Dalam RPG pemain dapat memilih satu karakter untuk dimainkan. Seiring dengan naiknya *level game*, karakter tersebut dapat berubah, bertambah kemampuannya, bertambah senjatanya, atau bertambah hewan peliharaannya.

5. Strategi

Genre strategi menitikberatkan pada kemampuan berpikir dan organisasi. *Game* strategi dibedakan menjadi dua, yaitu *Turn Based Strategy* dan *Real Time Strategy*. Jika *real time* strategi mengharuskan pemain membuat keputusan dan secara bersamaan pihak lawan juga beraksi hingga menimbulkan serangkaian kejadian dalam waktu yang sebenarnya,

sedangkan *turn based* strategi pemain bergantian menjalankan taktiknya. Saat pemain mengambil langkah, pihak lawan menunggu. Demikian juga sebaliknya.

6. Balapan

Pemain dapat memilih kendaraan, lalu melaju di arena balap. Tujuannya yaitu mencapai garis *finish* tercepat.

7. Olahraga

Genre ini membawa olahraga ke dalam sebuah komputer atau konsol. Biasanya *gameplay* dibuat semirip mungkin dengan kondisi olahraga yang sebenarnya.

8. Puzzle

Genre puzzle menyajikan teka-teki, menyamakan warna bola, perhitungan matematika, menyusun balok, atau mengenal huruf dan gambar.

9. Permainan Kata

Word game sering dirancang untuk menguji kemampuan dengan bahasa atau untuk mengeksplorasi sifat-sifatnya. *Word Game* umumnya digunakan sebagai sumber hiburan, tetapi telah dibuktikan untuk melayani suatu tujuan pendidikan juga. Berdasarkan uraian di atas dapat disimpulkan *game* adalah suatu hasil dari proses multimedia berupa alat untuk bersenang-senang dan dapat digunakan sebagai media untuk pembelajaran. orang dalam usaha mendewasakan manusia melalui upaya pengajaran dan pelatihan (Ghea Putri, 2012).

2.2 ROLE-PLAYING GAME (RPG)

2.2.1 Definisi RPG

Menurut Demon Solvatore, (2010). *Role Playing Game* (RPG) adalah sebuah permainan yang para pemainnya memainkan peran tokoh-tokoh khayalan dan berkolaborasi untuk merajut sebuah cerita bersama. Para pemain memilih aksi tokoh-tokoh mereka berdasarkan karakteristik tokoh tersebut, dan keberhasilan aksi mereka tergantung dari sistem peraturan

Permainan yang ditetapkan, para pemain biasa berimprovisasi membentuk arah dan hasil akhir permainan ini. Dalam sebuah permainan RPG, jarang ada yang “kalah” atau “Menang”. Ini membuat pemain RPG berbeda dari jenis permainan papan lainya seperti Monopoli atau Ular

Tangga, permainan kartu, olahraga, dan permainan lainnya. Seperti sebuah novel atau *film*, permainan RPG mempunyai daya tarik karena permainan-permainan ini mengajak para pemain untuk menggunakan imajinasi mereka. RPG biasa lebih mengarah ke kolaborasi sosial daripada kompetisi. Pada umumnya dalam RPG, para pemain tergabung dalam satu kelompok.

Permainan RPG rata-rata dimainkan seperti sebuah drama radio: ketika seorang pemain “berbicara”, dia berbicara sebagai tokohnya dan ketika si pemain ingin tokohnya melakukan sesuatu yang fisik (seperti menyerang sebuah monster atau membuka sebuah gembok) dia harus meng gambarkannya secara lisan.

Ada pula sejenis permainan RPG di mana para pemain bisa melakukan gerakan fisik tokohnya oleh si pemain sendiri. Ini disebut *Live-Action Role Playing* atau LARP. Dalam permainan LARP, biasanya para pemain memakai kostum dan menggunakan alat-alat yang sesuai dengan tokoh, dunia dan cerita yang dia mainkan. Permainan PC yang menggunakan unsur-unsur dan mekanisme permainan RPG disebut sebagai *Computer Role Playyng Game* atau CRPG. Selain di PC, RPG juga banya diadaptasikan ke mesin-mesin permainan.

2.2.2 Elemen Perancangan *Game* RPG

1. *Map*

Map adalah suatu elemen yang penting di dalam suatu *Game* RPG. Dalam pembuatannya pun tidak bisa sembarangan karena menimbulkan ketidaksesuaian dan ketidaknyamanan bagi *player*. *Map* terdiri dari beberapa *tool* yaitu :

- a. *Map* adalah tempat atau lokasi karakter bergerak dalam *game*. Seperti hutan pantai, desa. Sebuah *Game* bisa saja memiliki banyak *map* yang berbeda-beda yang menunjukkan tempat kejadian di *game* yang berbeda-beda pula.
- b. *Tiles* adalah potongan-potongan kecil gambar yang mewakili suatu objek dari suatu tempat. Seperti batang pohon, sungai, batu. Sekilas *map* terlihat seperti sebuah gambar. Tetapi nyatanya, *map* terdiri dari

potongan-potongan kecil gambar yang disusun sedemikian rupa sehingga berwujud sebuah tempat.

2. *Event*

Event merupakan tempat mengatur kejadian-kejadian yang terjadi di dalam *game*, seorang *Non Playable Character* (NPC) yang memberikan informasi, *quest* yang harus diselesaikan, peti harta karun yang harus dibuka, barang yang dapat diambil. Semuanya merupakan contoh penggunaan *event* di dalam *game*.

3. *Database*

Database adalah kumpulan dari bermacam-macam data yang digunakan dalam *game* ini. Seperti *Actor*, *Animation*, *Question*, semua itu dapat diubah, ditambahkan dan mengurangi data yang akan digunakan dalam *game*.

4. *Cut Scene*

Cut Scene adalah gerakan-gerakan otomatis pada *player* atau NPC dalam *Game*. *Cut Scene* banyak dipakai dalam *intro* sebuah *game* yang menjelaskan tentang cerita awal mula petualangan *player* dan di pakai dalam *event* yang membutuhkan peragaan awal.

5. *Weather*

Weather atau cuaca, merupakan elemen yang disediakan dan dapat diakses dengan menggunakan *event*. *RPG Maker* menyediakan beberapa *weather* yang bisa digunakan antara lain cuaca cerah, hujan, badai, maupun bersalju (Baihaqi Reza, 2015).

2.3 KECERDASAN BUATAN

Kecerdasan buatan termasuk bidang ilmu yang relatif muda. Pada tahun 1950-an para ilmuwan dan peneliti mulai memikirkan bagaimana caranya agar mesin dapat melakukan pekerjaannya seperti yang bisa dikerjakan oleh manusia. Alan Turing, seorang matematikawan dari Inggris pertama kali mengusulkan adanya pengujian untuk melihat bisa tidaknya sebuah mesin dikatakan cerdas. Hasil pengujian tersebut kemudian dikenal dengan *Turing Test*, di mana mesin tersebut menyamar seolah-olah sebagai seseorang di dalam suatu permainan yang mampu memberikan respon terhadap serangkaian pertanyaan yang diajukan. Turing beranggapan bahwa, jika mesin dapat membuat seseorang percaya bahwa dirinya mampu berkomunikasi dengan orang lain, maka dapat

dikatakan bahwa mesin tersebut cerdas (seperti layaknya manusia) (Sri Kusumadewi, 2003).

Kecerdasan buatan atau *Artificial Intelligence* (AI) itu sendiri dimunculkan oleh seorang professor dari *Massachusetts Institute of Technology* yang bernama John McCarthy pada tahun 1956 pada *Dartmouth Conference* yang dihadiri oleh para peneliti *Artificial Intelligence*. Pada konferensi tersebut juga didefinisikan tujuan utama dari kecerdasan buatan, yaitu mengetahui dan memodelkan proses-proses berpikir manusia dan mendesain mesin agar dapat menirukan kelakuan manusia tersebut.

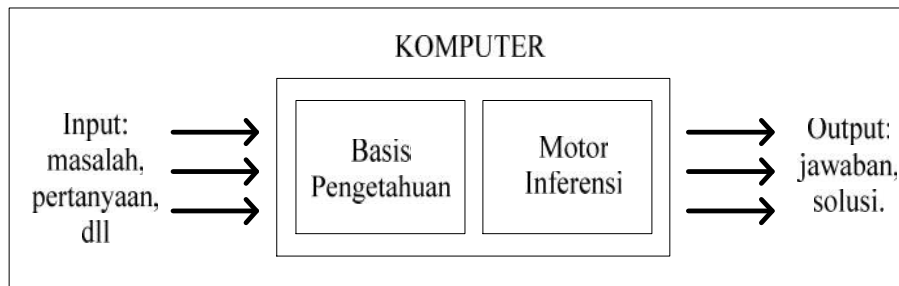
Artificial Intelligence didefinisikan sebagai kecerdasan yang ditunjukkan oleh suatu entitas buatan. Sistem seperti ini umumnya dianggap komputer. Kecerdasan diciptakan dan dimasukkan ke dalam suatu mesin ([komputer](#)) agar dapat melakukan pekerjaan seperti yang dapat dilakukan [manusia](#). Beberapa macam bidang yang menggunakan kecerdasan buatan antara lain sistem pakar, [permainan komputer](#) (*games*), [logika fuzzy](#), [jaringan syaraf tiruan](#) dan [robotika](#).

Untuk menciptakan aplikasi kecerdasan buatan ada 3 bagian utama yang sangat dibutuhkan, yaitu:

1. Basis Pengetahuan (*Knowledge Base*), berisi fakta-fakta, teori, pemikiran dan hubungan antara satu dengan yang lainnya.
2. Kaidah (*rule*) adalah aturan yang bila diterapkan pada suatu keadaan (*state*) dari fakta akan menghasilkan keadaan (*state*) baru.
3. Motor Inferensi (*Inference Engine*), yaitu kemampuan menarik kesimpulan

berdasarkan penalaran dan untaian dari kaidah untuk bergerak dari keadaan awal (*initial state*) menuju keadaan sasaran (*goal state*).

Berikut ini merupakan gambaran konsep penerapan Kecerdasan komputer di dalam komputer :



Gambar 2.1. Penerapan Konsep Kecerdasan Buatan Di Dalam Komputer

2.4 METODE *FINITE STATE MACHINE*

2.4.1 Pengertian *Finite State Machine* (FSM)

Menurut Setiawan (2006), *Finite State Machine* (FSM) adalah sebuah metodologi perancangan sistem kontrol yang menggambarkan tingkah laku atau prinsip kerja sistem dengan menggunakan tiga hal, yaitu *state* (keadaan), *event* (kejadian), dan *action* (aksi). Pada saat saat dalam periode waktu yang cukup signifikan, sistem akan berada pada salah satu *state* yang aktif. Sistem dapat beralih atau bertransisi menuju *state* lain jika mendapat masukan atau *event* tertentu, baik yang berasal dari perangkat luar atau komponen dalam sistemnya itu sendiri. Transisi keadaan ini umumnya juga disertai oleh aksi yang dilakukan oleh sistem ketika menanggapi masukan yang terjadi. Aksi dilakukan tersebut dapat berupa aksi yang sederhana atau melibatkan rangkaian proses yang relatif kompleks.

Menurut Ian Millington (2006) dalam bukunya yang berjudul *Artificial Intelligence for Games* menyebutkan bahwa *Finite State Machines* (FSM) masuk dalam ranah *Decision Making* (Pembuat Keputusan) pada *Artificial Intelligence*.

2.4.2 Proses *Finite State Machine* (FSM) dalam Sistem

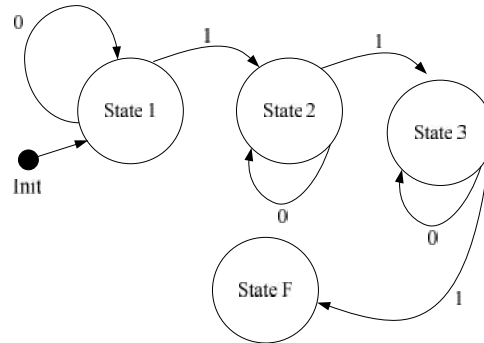
Dalam *Game* ini proses FSM berjalan dengan mendapatkan aturan jika misi terselesaikan atau *quest* terpenuhi, maka akan berpindah *state* ke *level* berikutnya, dan apabila misi tidak terselesaikan atau *quest* tidak terpenuhi maka akan tetap pada *state* awal.

a. Berikut adalah Algoritma *Finite State Machine* dalam sistem :

```
if (keyboard_check_pressed(ok_space)&& global.text=false &&
peraturan.switches["quest1"] = false ){
instance_create(x,y,obj_Player1);
peraturan.switches["quest1"] = true;
```

global.text = true;

2.4.3 Skema Metode Finite State Machine (FSM)



Gambar 2.2 Contoh Diagram State FSM

Keterangan :

Salah : 0 — ~~Kembali~~

Benar : 1 — ~~Next State~~

Penjelasan Diagram di atas :

1. *State 1* merupakan misi ke 1 : jika misi 1 dapat diselesaikan dengan benar, maka masuk *next state* 2, dan jika misi 1 belum diselesaikan maka ia akan tetap berada di *state 1*.
2. *State 2* merupakan misi ke 2 : jika misi 2 dapat diselesaikan dengan benar, maka masuk *next state* 3, dan jika misi 2 belum diselesaikan maka ia akan tetap berada di *state 2*.
3. *State 3* merupakan misi ke 3 : jika misi 3 dapat diselesaikan dengan benar, maka masuk *state final*, dan jika misi 3 belum diselesaikan maka ia akan tetap berada di *state 3*.

2.5 GAME MAKER

Game Maker adalah salah satu *game engine* yang bisa membantu dalam pengembangan *game* komputer tanpa menggunakan *coding* yang banyak. Dengan *Game Maker* kita dapat membuat *game 2D*/jenis *game* yang tidak membebani *system*. Dalam membuat *game* harus memiliki *software* pembuatan *game* yang handal, salah satu *software* komputer untuk membuat *game* yang baik adalah dengan menggunakan *Game Maker*.

Komponen-komponen *Game Maker* Mencakup berbagai *genre*, termasuk aksi, petualangan, permainan *puzzle* dan lainnya lengkap dengan efek suara dan kualitas gambar 2D/jenis *game* yang tidak membebani *system*. Untuk membuat *game* yang bagus dan handal harus mengetahui semua fungsi yang ada di *Game Maker* (Overmars, 2006).

2.6 FLOWCHART

Diagram alir atau *flowchart* merupakan bagan yang memperlihatkan urutan dan hubungan antar proses beserta instruksinya. Fungsinya adalah memberikan gambaran secara garis besar untuk *program* atau aplikasi yang dibuat (Vina Sagita, 2013).

Flowchart view disebut juga diagram tampilan yang memberikan gambaran alir dari satu *scene* (tampilan) ke *scene* lainnya. Sistem *flowchart* menggambarkan tahapan proses dari suatu sistem, termasuk sistem multimedia. Sedangkan *program flowchart* menggambarkan urutan-urutan instruksi dari suatu *program* komputer (Nur Ulfah Sam, 2010).

Flowchart adalah penyajian yang sistematis tentang proses dan logika dari kegiatan penanganan informasi atau penggambaran secara grafik dari langkah-langkah dan urutan-urutan prosedur dari suatu *program*. *Flowchart* menolong analis dan *programmer* untuk memecahkan masalah kedalam segmen-segmen yang lebih kecil dan menolong dalam menganalisis alternatif-alternatif lain dalam pengoperasian. *System flowchart* adalah urutan proses dalam sistem dengan menunjukkan alat media *input*, *output* serta jenis media penyimpanan dalam proses pengolahan data. Program *flowchart* adalah suatu bagan dengan simbol-simbol tertentu yang menggambarkan urutan proses secara mendetil dan hubungan antara suatu proses (instruksi) dengan proses lainnya dalam suatu program (Anharku, 2009).

2.7 UNIFIED MODELLING LANGUGAGUE (UML)

Menurut Kim Hamilton dan Russel Miles, (2006) , UML ialah bahasa pemodelan standar untuk perangkat lunak dan pengembangan sistem. Dalam sebuah desain sistem, model yang anda punya hanya untuk satu alasan yang penting, untuk mengelola kemajemukan. Pemodelan membantumu untuk melihat pohon percabangan, memungkinkan anda untuk memfokuskan dimana yang penting, menangkap, membuat dokumen, dan memberitahukan dimana aspek yang penting dari desain sistem.

Sebuah bahasa pemodelan bisa apa saja yang berisi notasi (cara untuk mengungkapkan model) dan deskripsi dari apa yang notasi maksudkan (meta-model) (Kim Hamilton, 2006).

2.7.2 Model dan Diagram

Banyak pendatang baru di UML fokus pada berbagai jenis diagram digunakan untuk memodelkan sistem mereka. Ini sangat mudah untuk mengasumsikan bahwa himpunan diagram yang telah dibuat sebenarnya model. Ini adalah kesalahan yang kecil untuk membuat karena ketika anda menggunakan UML, anda biasanya akan berinteraksi dengan alat bantu UML dan satu set diagram tertentu. Tapi pemodelan UML bukan hanya tentang diagram, ini tentang menangkap sistem anda sebagai model diagram sebenarnya adalah hanya jendela ke dalam model utama tersebut.

Diagram tertentu akan menunjukkan beberapa bagian dari model Anda tetapi belum tentu semuanya. Ini masuk akal, karena anda tidak ingin diagram yang menunjukkan segala sesuatu dalam model sekaligus anda ingin dapat membagi isi model di beberapa diagram. Namun, tidak semuanya dalam model anda perlu ada pada diagram untuk itu untuk menjadi bagian dari model anda (Kim Hamilton, 2006).

2.7.3 UML dan Proses Pengembangan Perangkat Lunak

Bila anda menggunakan UML untuk memodelkan sistem perangkat lunak, anda menerapkan sebagian dipengaruhi oleh proses pengembangan perangkat lunak yang anda gunakan. Sebuah proses pengembangan perangkat lunak adalah cara yang digunakan untuk membangun perangkat lunak untuk menentukan kemampuan itu, bagaimana dibangun, yang bekerja pada apa, dan kerangka waktu untuk semua kegiatan.

Proses bertujuan untuk membawa disiplin dan prediktabilitas untuk pengembangan perangkat lunak, meningkatkan kesempatan keberhasilan proyek. Karena UML adalah bahasa untuk pemodelan perangkat lunak Anda, ini merupakan bagian penting dari proses pengembangan perangkat lunak (Kim Hamilton, 2006).

Beberapa proses pengembangan perangkat lunak terkenal antara lain:

a. *Waterfall*

Metode *waterfall* mencoba untuk menjabarkan persyaratan awal siklus hidup proyek. Setelah pengumpulan persyaratan, desain perangkat lunak dilakukan secara penuh. Setelah desain adalah lengkap, perangkat

lunak diimplementasikan. Masalah dengan metode ini adalah bahwa jika perubahan persyaratan terjadi, dampaknya bisa menghancurkan keseluruhan sistem.

b. Iteratif

Metode Iteratif berusaha untuk mengatasi kekurangan dari pendekatan *waterfall* dengan menerima perubahan yang akan terjadi dan, pada kenyataannya, merangkul keseluruhan itu. *Unified Process* lebih dikenal sebagai proses berulang. Ini terdiri dari beberapa tahap, setiap tahap mengandung beberapa jumlah kegiatan sebagai berikut: persyaratan, desain, dan implementasi (*coding*). Metode Iteratif mencakup lebih luas dari pendekatan (misalnya, proses berulang tangkas), dan mereka dapat berkisar dari menggunakan UML sebagai sketsa menggunakan UML sebagai *blueprint*.

c. Metode Agile

Metode *Agile* menggunakan iterasi dalam semburan sangat pendek dan upaya untuk meminimalkan risiko dengan selalu memiliki sistem kerja yang memperluas kemampuan. Metodologi dalam kategori ini telah memperkenalkan beberapa praktek pembangunan yang lebih menarik, seperti pasangan pemrograman dan pengembangan *test-driven*. Metode *Agile* menekankan menggunakan UML sebagai sketsa.

3. METODOLOGI PENELITIAN

3.1 LANGKAH-LANGKAH DALAM PENELITIAN

Penelitian adalah suatu penyelidikan atau suatu usaha pengujian yang dilakukan secara teliti dan kritis dalam mencari fakta-fakta atau prinsip-prinsip dengan menggunakan langkah-langkah tertentu. Dalam mencari fakta-fakta ini diperlukan usaha yang sistematis untuk menemukan jawaban yang ilmiah dalam suatu masalah.

Adapun langkah-langkah yang diimplementasikan pada proses penyelesaian tugas akhir ini adalah sebagai berikut:

3.1.1 Pengumpulan Data

Pengumpulan data bertujuan untuk memperoleh informasi yang dibutuhkan dalam rangka mencapai tujuan penelitian. Tujuan yang diungkapkan dalam bentuk hipotesis merupakan jawaban sementara terhadap penelitian. Jawaban itu masih perlu diuji secara empiris, dan untuk maksud inilah dibutuhkan pengumpulan data.

Tahapan yang akan dilakukan setelah merencanakan penelitian adalah mengumpulkan referensi mengenai metode Finite State Machine (FSM),

Teori Bahasa dan Automata, Game dan Multimedia, serta data yang dibutuhkan untuk mendukung proses pembuatan game RPG.

3.1.2 Studi Kepustakaan

Studi kepustakaan bertujuan untuk mengumpulkan data dan informasi ilmiah, berupa teori-teori, jurnal, metode atau pendekatan yang pernah berkembang dan telah didokumentasikan dalam bentuk buku, majalah, naskah, dan lain sebagainya. Melakukan studi kepustakaan terhadap berbagai referensi yang berkaitan dengan penelitian yang dilakukan, seperti membaca dan mempelajari buku-buku yang berkaitan dengan Pembuatan game RPG juga mempelajari mengenai penerapan metode yang digunakan dalam menyelesaikan penelitian ini.

3.1.3 Tempat Penelitian Dan Waktu Penelitian

Penelitian Game RPG “The Royal Sword” dilakukan di Laboratorium Teknik Informatika Universitas Malikussaleh. Lokasi ini diambil karena memiliki segala aspek yang mendukung untuk keperluan game yang akan dibangun agar penelitian berjalan dengan baik. Penelitian dilaksanakan pada bulan April 2017 sampai dengan bulan Mei 2017.

3.2 ANALISA KEBUTUHAN SISTEM

Analisa kebutuhan sistem yaitu memahami dengan sesungguhnya kebutuhan dari sistem yang akan dibangun dan mengembangkan sebuah sistem yang memadahi kebutuhan tersebut atau memutuskan bahwa pengembangan sistem yang baru tidak dibutuhkan.

Tahap metode analisis ini merupakan tahap usaha dalam mengamati secara detil sistem yang akan di bangun. Setelah analisa didapatkan maka langkah selanjutnya adalah membuat sebuah hasil analisa. Hasil analisa tersebut akan menjadi acuan dari perancangan sistem yang dibangun.

3.2.1 Analisis Kebutuhan Perangkat Keras (Hardware)

Analisa kebutuhan perangkat keras (hardware) perlu dilakukan untuk mengetahui perangkat keras (hardware) apa saja yang diperlukan untuk mendukung proses sistem yang akan dibangun.

Perangkat keras merupakan perangkat yang sangat diperlukan didalam sistem komputer, perangkat keras yang digunakan pada pembuatan perancangan sistem ini adalah Laptop Lenovo G40-45 dengan spesifikasi sebagai berikut:

- a. AMD E1-6010 APU 1,35 GHz
- b. DDR3 RAM 4 GB

c. HDD 500 GB

3.2.2 Analisis Kebutuhan Perangkat Lunak (Software)

Analisis kebutuhan merupakan langkah awal untuk menentukan gambaran perangkat yang akan dihasilkan ketika pengembang melaksanakan sebuah pembuatan game yang menggunakan perangkat lunak lainnya. Perangkat lunak yang baik dan sesuai dengan kebutuhan pengguna sangat berpengaruh pada keberhasilan dalam melakukan analisis kebutuhan.

Perangkat lunak juga merupakan salah satu faktor penunjang untuk membuat dan merancang sebuah game. Perangkat lunak berfungsi sebagai pengolah data. Perangkat lunak yang digunakan pada perancangan sistem ini adalah:

- a. Sistem Operasi : Microsoft Windows Ultimate 7 32 Bit
- b. Program Aplikasi : Game Maker Studio 1.4
- c. Aplikasi Tambahan : RPG Maker VX Ace, Adobe Photoshop CS 4

3.3 KEBUTUHAN INPUT

Adapun data yang diinput hanya memainkan permainan dan menjalankan misi yang telah disediakan. Pemain harus mengikuti perintah sesuai misi yang diinstruksikan pada game RPG The Royal Sword.

3.4 KEBUTUHAN PROSES

Pemrosesan data dilakukan pada saat misi terselesaikan semua atau semua quest terpenuhi dengan baik maka akan menampilkan state selanjutnya, sehingga terjadi alur Finite State Machine pada program.

3.5 KEBUTUHAN OUTPUT

Hasil output berupa pemberitahuan kepada user untuk melanjutkan ke level berikutnya hingga level akhir, sehingga selesainya misi sampai finish hingga selesailah permainan game tersebut.

3.6 DIAGRAM ALUR KERJA PENELITIAN

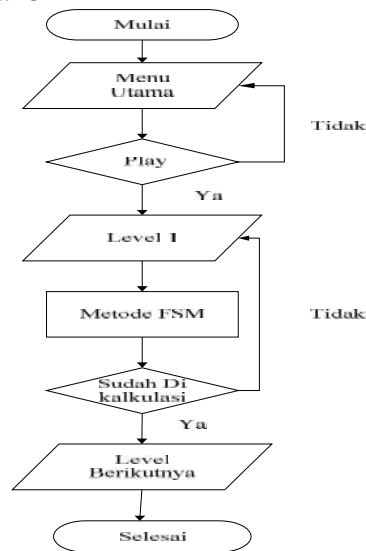
Diagram alur kerja penelitian menjelaskan tentang prosedur yang dilakukan selama penelitian berlangsung. Diagram alur kerja digunakan untuk mendokumentasikan standar proses yang telah ada sehingga menjadi pedoman dalam menjalankan langkah-langkah dalam penelitian. Selain itu untuk melakukan analisis terhadap proses selama penelitian berlangsung sehingga dapat melakukan peningkatan atau perbaikan proses yang berkesinambungan (secara terus menerus).

Diagram alur kerja yang akan dilakukan pada penelitian ini diilustrasikan pada gambar 3.

3.7 SKEMA SISTEM

Istilah skema sebenarnya bukan hal yang baru bagi kita. Kata ini sudah lama milik bahasa Indonesia (merupakan kata serapan yang berasal dari bahasa Inggris 'schema'). Di dalam Kamus Besar Bahasa Indonesia (KBBI), kata skema merupakan peranan dari bagan, rangka-rangka, rancangan. Skema adalah suatu pemberian yang digeneralisasikan, suatu rencana atau struktur (Lilis Siti Sulistyaningsih, 2016).

Skema sistem untuk pengenalan data yang dirancang dalam penelitian ini diilustrasikan pada Gambar 3



Gambar 3.1 Skema Sistem Secara Keseluruhan

4. ANALISA DAN PEMBAHASAN

4.1 ANALISA SISTEM

Dalam merancang suatu sistem yang terkomputerisasi, analisa sistem memiliki peranan yang sangat penting dalam membuat rincian aplikasi yang dikembangkan, yang bertujuan untuk mengidentifikasi dan mengevaluasi permasalahan yang ada dan nantinya diharapkan dapat menciptakan suatu sistem yang lebih baik.

4.2 ANALISA KEBUTUHAN SISTEM

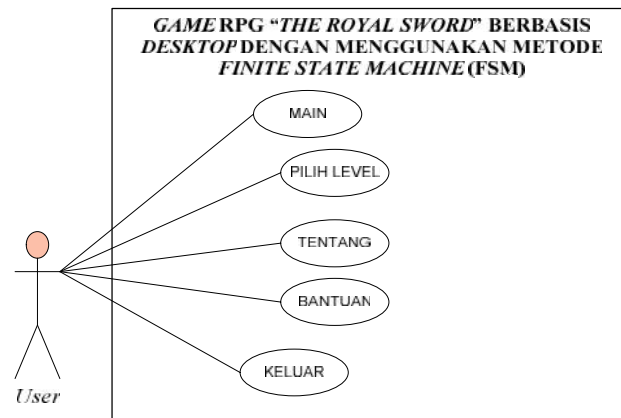
Dalam membangun atau mengembangkan sebuah sistem, analisa kebutuhan sistem memiliki peranan yang sangat penting, karena dapat memberikan gambaran yang jelas terhadap data yang di *input* dan diproses dalam sistem, juga memberikan gambaran bagaimana antarmuka sistem yang akan dibangun nantinya.

4.3 DESAIN SISTEM

Dalam merancang aplikasi *game* RPG *The Royal Sword* berbasis *desktop*, dibutuhkan beberapa bentuk pemodelan atau *activity* yang digunakan untuk perbandingan dengan inputan *user* dalam mengaplikasikan metode *Finite State Machine* (FSM) di dalamnya.

4.3.1 Diagram Use Case

Diagram *use case* merupakan pemodelan untuk menggambarkan kelakuan (*behavior*) atau gambaran fungsionalitas dari suatu sistem, sehingga *user* atau pengguna sistem paham dan mengerti mengenai kegunaan sistem yang akan dibangun. Diagram *Use Case* juga dapat mendeskripsikan sebuah interaksi satu atau lebih aktor yang ada dalam sistem tersebut, Serta untuk mengetahui fungsi apa saja yang ada di dalam sistem dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut.



Gambar 4.1 Use Case Diagram

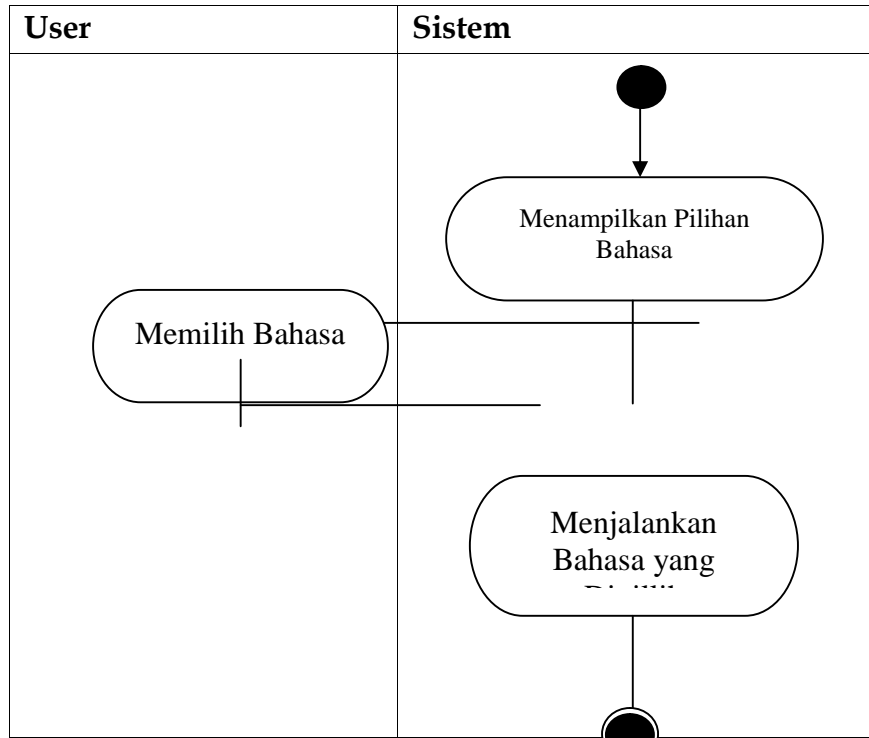
4.3.2 Activity Diagram

Activity diagram memiliki pengertian yaitu lebih fokus kepada menggambarkan urutan aktivitas dalam sebuah proses. Memiliki struktur diagram yang mirip *flowchart* atau data *flow* diagram pada perancangan terstruktur, dan bermanfaat untuk membantu memahami proses secara keseluruhan.

Adapun diagram *activity* pada aplikasi ini adalah sebagai berikut :

- a. *Activity* Diagram Pilih Bahasa

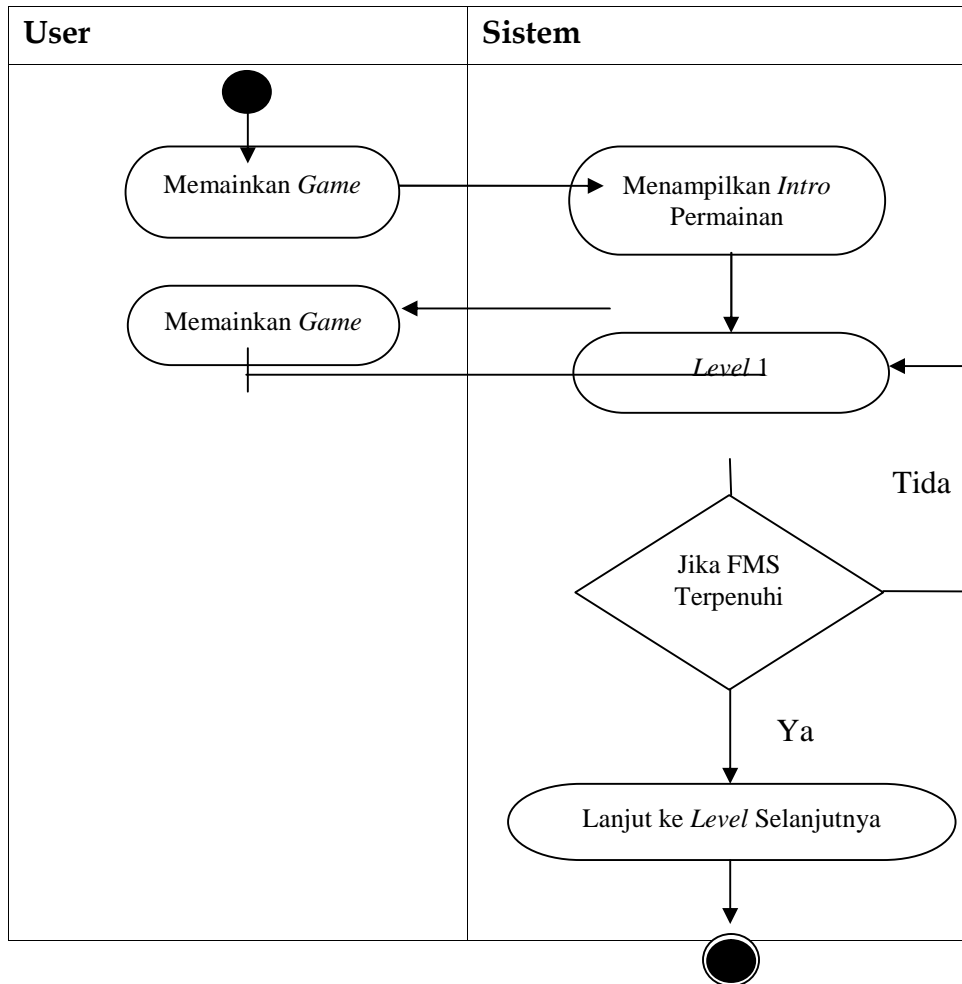
Tabel 4.1 *Activity* Diagram Pilih Bahasa



Pada diagram *activity* ini ditampilkan proses yang terjadi ketika aplikasi dihidupkan maka sistem menampilkan pilihan penggunaan bahasa dalam permainan, kemudian *user* memilih salah satu bahasa dan sistem menjalankan bahasa yang dipilih.

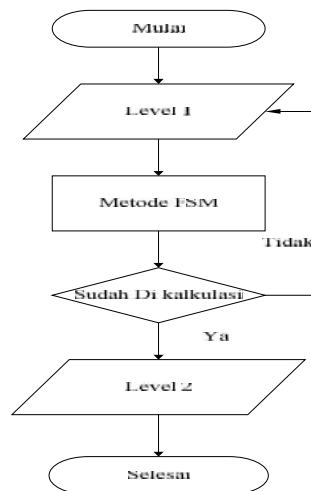
b. Activity Diagram Main

Tabel 4.2 Activity Diagram Main



4.3.3 Diagram Alur

- Diagram Alur *Level 1*

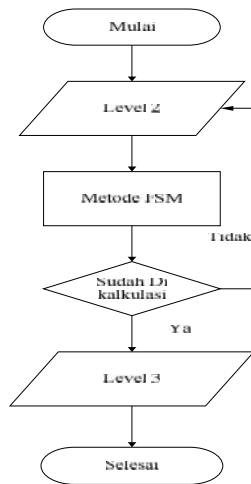


Gambar 4.2 Diagram Alur *Level 1*

Keterangan :

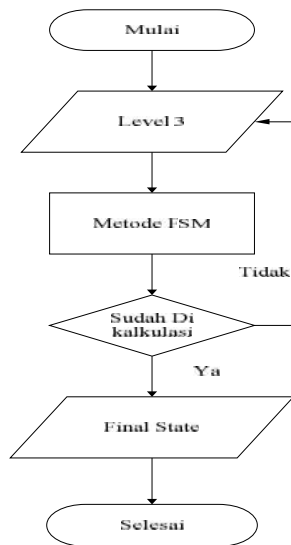
1. Mulai, yaitu proses kita memulai tahap awal pada sebuah aplikasi atau *game*.
2. *level 1*, yaitu kita pemain mengklik tombol *play* maka permainan akan dimulai di *level 1*.
3. Metode FSM, yaitu masuk pada tahap pemrosesan metode *Finite State Machine* (FSM).
4. Proses Kalkulasi Metode *Finite State Machine* (FSM), dimana Jika pemain berhasil menyelesaikan semua misi atau *quest* terpenuhi pada *level 1* maka pemain akan lanjut pada *level 2*. Dan jika tidak terselesaikan misi atau *quest* tidak terpenuhi maka pemain akan tetap beradai di *level 1*.
5. Selesai, maka berakhirilah pemrosesan data pada diagram alur *Level 1*.

- **Diagram Alur *Level 2***



Gambar 4.3 Diagram Alur *Level 2*

- **Diagram Alur *Level 3***



Gambar 4.4 Diagram Alur *Level 3*

4.4 PROSES METODE *FINITE STATE MACHINE* (FMS)

Dibawah ini adalah *Event* peraturan yang dijalankan secara bertahap dengan menerapkan metode FSM pada *Game RPG The Royal Sword* :

```

1 // In-batasnya Metode Finite State Machine
2
3 switches = ds_map_create();
4
5 switches[? "quest1"] = false;
6 switches[? "quest2"] = false;
7 switches[? "quest3"] = false;
8 switches[? "quest4"] = false;
9 switches[? "quest5"] = false;
10 switches[? "quest6"] = false;
11 switches[? "quest7"] = false;
12 switches[? "quest8"] = false;
13 switches[? "quest9"] = false;
14 switches[? "questbeenjata"] = false;
15 switches[? "questbeenjata1"] = false;
16 switches[? "questbeenjata2"] = false;
17 switches[? "quest10"] = false;
18 switches[? "quest11"] = false; //level 3
19 switches[? "questqerbang"] = false;
20 switches[? "quest12"] = true;
21 switches[? "questqerbang2"] = true;
22 switches[? "questqerbang3"] = false;
23 switches[? "questfinal"] = true; //level 4
24 switches[? "questocnd"] = true;
    
```

Gambar 4.5 *Event* Peraturan FMS pada *Game RPG The Royal Sword*

Pada proses metode *Finite State Machine* akan mengambil 3 sampel *quest*/misi pada *event* peraturan yang ada pada *program*.

Berikut *Quest Finite State Machine* (FMS) Pada *Game* :

```

switches[? "quest1"] = false;
switches[? "quest2"] = false;
switches[? "quest3"] = false;

if distance_to_object(obj_hardy) < 4{
if (keyboard_check_pressed(ok_space)&& global.text=false &&
peraturan.switches[? "quest1"] = true ){
instance_create(x,y,obj_hardy)
state = scr_npc_nunggu;
image_speed= 0;
peraturan.switches[? "quest1"] = true;
}
}
    
```

Setelah menyelesaikan *quest* 1 maka *user* bisa melanjutkan atau berpindah *state* pada *quest* 2.

```
if distance_to_object(obj_hardy) < 4{
if (keyboard_check_pressed(vk_space)&& global.text=false &&
peraturan.switches[? "quest1"] = true && peraturan.switches[? "quest2"]
= false){
instance_create(x,y,obj_hardy)
state = scr_npc_nunggu;
image_speed= 0;
peraturan.switches[? "quest2"] = true;
}
}
```

Setelah sistem menyelesaikan *quest* 2 kemudian bisa berpindah *state* ke *quest* berikutnya, misal berpindah ke *quest* 3.

```
if distance_to_object(obj_hardy) < 4{
if (keyboard_check_pressed(vk_space)&& global.text=false &&
peraturan.switches[? "quest2"] = true && peraturan.switches[?
"questLevel2"] = false){
instance_create(x,y,obj_hardy)
state = scr_npc_nunggu;
image_speed= 0;
peraturan.switches[? "quest3"] = true;
}
}
```

4.5 USER INTERFACE

Untuk mengimplementasikan sistem, aplikasi dibuat dengan menggunakan bahasa pemrograman *Game Maker Language* (GML).

4.5.1 Tampilan Halaman Awal

Tampilan awal *program* adalah tampilan dimana aplikasi pertama kali dijalankan atau dibuka. Adapun tampilan sebagai berikut :



Gambar 4.6 Tampilan Awal Game RPG *The Royal Sword*

4.5.2 Tampilan Halaman Pilih Bahasa

Tampilan Halaman Pilih Bahasa adalah sebuah fitur pilihan untuk memilih bahasa pada permainan, Berikut tampilannya:



Gambar 4.7 Tampilan Halaman Menu Pilih Bahasa

4.5.3 Tampilan Halaman Main Menu

Tampilan Halaman *Main Menu* adalah Sebagai berikut:



Gambar 4.8 Tampilan Main Menu Game RPG *The Royal Sword*

4.5.4 Tampilan Halaman *Intro* Permainan Pada *Level 1*

Tampilan halaman *intro* permainan pada *level 1* adalah tampilan dimana misi yang akan di jalankan di *level 1*.



Gambar 4.9 Tampilan *Intro Level 1* Game RPG *The Royal Sword*

4.5.5 Tampilan Halaman Cerita Awal Permainan Pada *Level 1*

Tampilan halaman sinopsis permainan pada *level 1* adalah tampilan dimana cerita awal permainan.



Gambar 4.10 Tampilan Sinopsis *Game RPG The Royal Sword*

4.5.6 Tampilan *Game* Pada *Level 1*

Berikut Tampilan permainan pada *level 1*.



Gambar 4.11 Tampilan *Level 1* *Game RPG The Royal Sword*

4.5.7 Tampilan Halaman *Intro* Permainan Pada *Level 2*

Tampilan halaman *intro* permainan pada *level 2* adalah awalan tampilan dimana ketika permainan dimulai di *level 2*.



Gambar 4.12 Tampilan *Intro Level 2* Game RPG *The Royal Sword*

4.5.8 Tampilan Game Pada *Level 2*

Tampilan halaman permainan pada *level 2*



Gambar 4.13 Tampilan *Level 2* Game RPG *The Royal Sword*

4.5.9 Tampilan Halaman *Intro* Permainan Pada *Level 3*

Tampilan halaman *intro* permainan pada *level 3* adalah awalan tampilan dimana ketika permainan dimulai di *level 3*.



Gambar 4.14 Tampilan Intro Level 3 Game RPG The Royal Sword

4.5.10 Tampilan Game Pada Level 3

Tampilan permainan pada level 3



Gambar 4.15 Tampilan Level 3 Game RPG The Royal Sword

4.5.11 Tampilan Halaman Menu Pause

Berikut Adalah Tampilannya :



Gambar 4.16 Tampilan *Menu Pause* pada *Game RPG The Royal Sword*

4.5.12 Tampilan *Game Pada Menu Ulang Atau Retry*

Tampilan *Menu Ulang* atau *Retry*



Gambar 4.17 Tampilan *Menu Ulang/Retry* pada *Game RPG The Royal Sword*

4.5.13 Tampilan Halaman Pilih *Level*

Tampilan halaman adalah fitur yang dapat memilih langsung *level* yang akan di mainkan. Adapun tampilan sebagai berikut :



Gambar 4.18 Pilih *Play level 1* Game RPG *The Royal Sword*



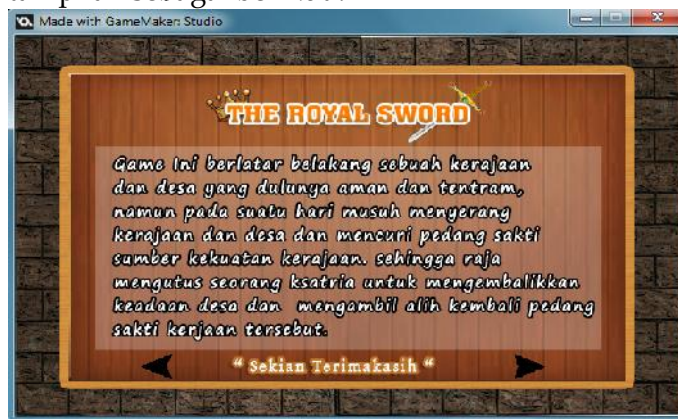
Gambar 4.19 Pilih *Play level 2* Game RPG *The Royal Sword*



Gambar 4.20 Pilih *Play level 3* Game RPG *The Royal Sword*

4.5.14 Tampilan Halaman Tentang

Tampilan halaman tentang adalah tampilan dimana aplikasi menampilkan cerita dan karakter utama dalam permainan. Adapun tampilan sebagai berikut :



Gambar 4.21 Tampilan Halaman Tentang 1



Gambar 4.22 Tampilan Halaman Tentang 2

4.5.15 Tampilan Halaman Bantuan

Tampilan halaman bantuan adalah tampilan dimana aplikasi menampilkan keterangan penggunaan tombol yang akan digunakan dalam permainan. Adapun tampilan sebagai berikut :



Gambar 4.23 Tampilan Halaman Bantuan

KESIMPULAN

Adapun kesimpulan yang diperoleh dari penelitian ini, adalah sebagai berikut :

1. Membangun sebuah game RPG The Royal Sword berbasis desktop yaitu dengan menggunakan aplikasi Game Maker Studio 1.4 adapun bahasa pemrograman yang digunakan ialah Game Maker Language (GML).
2. Mengimplementasikan metode Finite State Machine pada game RPG The Royal Sword dapat membuat permainan menjadi lebih tertata dan teratur dikarenakan penyelesaian perintah atau misi dilakukan secara bertahap.
3. Game genre RPG sangat cocok dimainkan oleh anak-anak, dibandingkan jenis game genre lain, yang mengandung unsur-unsur negatif seperti game perampokan, balap liar yang berkejaran dengan polisi, bahkan adanya tokoh game yang berpakaian sexy, yang perlahan bisa membangun tingkah laku si anak menjadi tidak baik.
4. Fitur gambar, suara, animasi pada aplikasi Game Maker studio 1.4 dirasa cukup kurang karena tidak tersedianya fitur tersebut, disini pembuat game harus memasukkan atau merancang sendiri gambar, suara dan animasi.
5. Game ini memberikan dan menjelaskan suatu pandangan bahwa membuat game itu memiliki kemudahan dan kesulitan sendiri.

SARAN

Beberapa hal yang dapat diperhatikan untuk penelitian ini adalah sebagai berikut :

1. Dari penelitian ini, penulis memberikan saran agar di masa depan nanti peneliti lain dapat melakukan penelitian yang serupa dengan menggunakan metode yang lebih efisien dan efektif, sehingga perkembangan *game genre* RPG dapat menjadi semakin baik dari yang sebelumnya.
2. Diharapkan aplikasi *game* ini dapat ditambahkan konsep *game* yang lebih modern dan cerita permainan yang lebih baik untuk meningkatkan minat bagi pecinta *game* RPG.
3. Penggunaan kosakata dalam cerita masih terlihat kaku dan kurang terlihat hidup untuk *game bergenre* RPG.
4. *Game* ini masih dijalankan menggunakan komputer atau PC sehingga perlu dikembangkan lagi agar dapat dijalankan pada *handphone*.
5. Daerah yang dapat dijelajahi (Peta) dibuat lebih beragam dan menantang lagi agar pemain dapat merasa tertantang dalam memainkan *game* ini.

6. Elemen-elemen permainan seperti peralatan, *skill*, dan musuh dibuat lebih variatif lagi.

DAFTAR PUSTAKA

- Abdullah, Dahlan, Baihaqi Reza, Cut Ita Erliana. 2015. *Game Edukasi Berbasis Role Playing Game Dengan Metode Finite State Machine*.
- Anharku, 2009. *Flowchart*. Jakarta: IlmuKomputer.Org.
- Albab, Ulil, Nur Wakhidah, Susanto. 2013. *Pembuatan Game Puzzle Gambar Untuk Meningkatkan Kemampuan Belajar Menggunakan Game Maker*.
- Dewi, Fatma Putri Ghea. 2012. *Pengembangan Game Edukasi Pengenalan Nama Hewan Dalam Bahasa Inggris Sebagai Media Pembelajaran Siswa SD Berbasis Macromedia Flash*. Universitas Negeri Yogyakarta.
- Harahap, Yudho Murphy. 2015. *Perancangan Game Edukatif Sebagai Media Pembelajaran Pada Anak Usia Dini Dengan Adobe Flash CS 5*.
- Jeffry, Johannes Petrus. 2014. *Rancang Bangun Aplikasi Game Edukasi PuzzlePengenalan Tokoh Sejarah Berbasis Android Dengan Metode Linier Congruential Generator (LCG) Sekolah Tinggi Manajemen Informatika Dan Komputer, Multi Data Palembang*.
- Kim Hamilton, Russel Miles. 2006. *Learning UML*.
- Kusumadewi, Sri. 2003. *Artificial Intelligence (Teknik Dan Aplikasinya)*.
- Millington, Burlington, Morgan Kaufmann. 2009. *Artificial Intelligence for Games*.
- Saputra, Muchlas Iman Teguh. Hanif Al-Fatta. 2015. *Pembuatan Game RPG "Perjalanan Rio" Menggunakan RPG Maker VX Ace*.
- SagitaVina, Irmina Prasetyowati. 2013. *Studi Perbandingan Implementasi Algoritma Boyer-Moore, Turbo Boyer-Moore, Dan Tuned Boyer-Moore Dalam Pencarian String*.

Sulindawati, Muhammad Fathoni. 2010. *Pengantar Analisa Perancangan "Sistem"*

Setiawan, Iwan. 2006. *Perancangan Software Embedded System Berbasis FSM.*

Ulfah Sam Nur. 2010. *Pembuatan Aplikasi Pembelajaran Materi Pengolahan Citra Dengan Menggunakan Teknik Konvolusi Berbasis Multimedia.*