PEMBUATAN GAME RPG PARA PENCARI KEBENARAN DENGAN MENGUNAKAN METODE FINITE STATE MACHINE

SKRIPSI

Disusun Oleh:
Muhammad Imam Setiawan
09.18.020

PROGRAM STUDI TEKNIK INFORMATIKA S-1
FAKULTAS TEKNIK INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
2014

LEMBAR PERSETUJUAN

PEMBUATAN GAME RPG MAKER PARA PENCARI KEBENARAN DENGAN METODE FINITE STATE MACHINE

SKRIPSI

Disusun dan Diajukan Sebagai Salah Satu Syarat Untuk Memperoleh Gelar Sarjana Teknik Informatika Strata Satu (S-1)

Disusun Oleh :
Muhammad Imam Setiawan
09.18.020

Diperiksa dan disetujui oleh

Dosen Pembimbing I	Dosen Pembimbing II
Ir. Yusuf Ismail Nahkoda, MT	**Febriana Santi W. S.Kom, M.Kom**
NIP. Y. 1018800189	**NIP: 1031000425**

Mengetahui,
Ketua Jurusan Teknik Informatika S-1

Joseph Dedy Irawan, ST, MT
NIP. 197404162005021002

JURUSAN TEKNIK INFORMATIKA S-1
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
2014

PROGRAM STUDI TEKNIK INFORMATIKA S-1
FAKULTAS TEKNOLOGI INDUSTRI
INSITUT TEKNOLOGI NASIONAL
MALANG

FORMULIR PERBAIKAN UJIAN SKRIPSI

NAMA : M Imam Setiawan

NIM : 0918020

JURUSAN : TeknikInformatika

JUDUL : PEMBUATAN GAME RPG MAKER PARA PENCARI KEBENAN DENGAN METODE FINITE STATE MACHINE

Perbaiki Kesimpulan
Perbaiki Penulisan
Lampirkan Beberapa Questionare
Lampirkan Beberapa Script

DosenPenguji I

Ali Mahmudi, Beng, PhD
NIP. 1031000429

DosenPenguji II

Yosep Agus Pranoto.S.T,M.T
NIP. 1031000432

FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG

PEMBUATAN *GAME RPG* PARA PENCARI DENGAN METODE *FINITE STATE MACHINE*

Muhammad Imam Setiawan

Program Studi Teknik Informatika S-1
Fakultas Teknologi Industri
Institut Teknologi Nasional Malang
Jl. Raya Karanglo Km 2 Malang
Email : iwanjaki@yahoo.com

Dosen Pembimbing : 1. Ir. Yusuf Ismail Nakhoda, MT.
2. Febriana Santi W. S.Kom, M.Kom

Abstrak

Perkembangan Game di dunia terjadi dengan sangat pesat, tak terkecuali di Indonesia. Sekarang game di pilih sebagai salah satu hiburan alternative, tidak hanya untuk kalangan anak-anak sampai kalangan orang tua pun suka bermain game. Perkembangan game yang semakin hari semakin maju membuat banyak orang tertarik untuk mengembangkan sebuah game tidak terkecuali penulis.

Role Playing Game (RPG)merupakan permainan game yang dimainkan dengan cara seorang user di haruskan memainkan suatu tokoh tertentu didalam permainan. User dapat melakukan improvisasi di dalam permainan untuk menentukan arah cerita di dalam permainan tersebut Didalam RPG salah satu unsur yang paling penting adalah NPC, dimana perilaku NPC didalam suatu game sangat berpengaruh pada alur cerita game itu sendiri. Oleh karena itu penullis memdesign suatu game dengan mengunakan metode FSM finite state machine sebagai bahan pemetaan perilakunya sehingga NPC yang ada dalam game bisa terlihat lebih interaktif.

Dari hasil pengujian fungsi yang dilakukan kepada 4 sistem operasi yang berbeda.game ini dapat dimainkan pada sistem operasi WindowsXP2,Windows XP 3,Windows 7 dan Windows 8 dengan tingkat keberhasilan 100% tanpa adanya gangguan

Kata kunci : RPG (role playing game), NPC (non playable character), FSM (finite state machine).

PROGRAM STUDI TEKNIK INFORMATIKA S-1

FAKULTAS TEKNOLOGI INDUSTRI

INSTITUT TEKNOLOGI NASIONAL

MALANG

PERNYATAAN KEASLIAN SKRIPSI

Saya yang bertanda tangan di bawah ini:

Nama	:	M Imam Setiawan
Nim	:	09.18.020
Program Studi	:	Teknik Informatika S-1
Fakultas	:	Teknologi Industri

Menyatakan dengan sesungguhnya bahwa Skripsi saya yang berjudul:

"PEMBUATAN GAME RPG PARA PENCARI KEBENARAN DENGAN METODE FINITE STATE MACHINE"

Adalah bukan duplikat atau mengutip dan tidak memuat seluruhnya karya orang lain kecuali dari sumber aslinya.

Malang, 22 Agustus 2014

Yang membuat pernyataan

METERAI TEMPEL

TGL 20

AC0CCADF289303139

6000

ENAM RIBU RUPIAH

M Imam Setiawan

KATA PENGANTAR

Dengan memanjatkan Puji syukur kehadirat Tuhan Yang Maha Esa, karena atas rahmat dan hidayahnya sehingga penyusun dapat menyelesaikan skripsi dengan judul **PEMBUATAN GAME RPG PARA PENCARI KEBENARAN DENGAN MENGUNAKAN METODE *FINITE STATE MACHINE.***

Penyusunan skripsi ini merupakan salah satu persyaratan dalam menyelesaikan program pendidikan **Strata Satu (S-1) Teknik Informatika**, Fakultas Teknologi Industri di **Institut Teknologi Nasional Malang.**

Pada penyusunan skripsi ini penulis mengucapkan banyak terima kasih sebesar-besarnya kepada :

1. Ibu Fatmawati, yang merupakan orang tua yang menjadi pendukung utama dari segi moril maupun materil.
2. M. Yasin Fansuri, ST selaku alumnus Institut Teknologi Malang Dari FTI_Teknik Elektro S1 yang selalu membantu urusan penulisan skripsi sebagai saudara yang paling tua, Dewi Muharyati, ST yang merupakan saudara perempuan dari penulis yang telah memberikan motivasi dan semangat dalam mengerjakan skripsi tugas akhir ini.
3. Ir.Soeparno Djiwo, MT, selaku Rektor Institut Teknologi Nasional Malang.
4. Ir. Anang Subardi, MT, selaku Dekan Fakultas Teknologi Industri, Institut Teknologi Nasional Malang.
5. Joseph Dedy Irawan, ST, MT, selaku Ketua Program Studi Teknik Informatika, Institut Teknologi Nasional Malang.
6. Sonny Prasetio, ST, MT, selaku Sekertaris Program Studi Teknik Informatika, Institut Teknologi Nasional Malang.
7. Bapak Ir. Yusuf Ismail Nahkoda, MT, selaku Dosen Pembimbing I, yang selalu memberikan masukan.

8. Febriana Santi W. S.Kom, M.Kom, selaku Dosen Pembimbing II yang telah memberikan dukungan dan masukan.
9. Semua dosen Program Studi Teknik Informatika yang telah membantu dalam penulisan dan masukan.
10. Aski, Dida, Ken dan semua teman-teman seperjuangan yang telah membantu sehingga terselesaikannya skripsi ini.
11. Serta semua pihak yang telah membantu baik secara langsung maupun tidak langsung sehingga penulis dapat menyelesaikan laporan ini.

Penyusun menyadari bahwa skripsi dipersembahkan adalah masih jauh dari sempurna, oleh karena itu penyusun mengharapkan kritik dan saran dari pembaca, Semoga skripsi ini bisa bermanfaat bagi pembaca yang ingin melanjutkan kuliah di Institut Teknologi Nasional Malang yang selalu ku ingat di perjalanan karierku, Amin.

Malang, 1 Agustus 2014

Muhammad Imam Setiyawan

Penyusun

DAFTAR ISI

HALAMAN JUDUL i
LEMBAR PERSETUJUAN ii
LEMBAR KEASLIAN iii
ABSTRAK iv
KATA PENGANTAR v
DAFTAR ISI vii
DAFTAR GAMBAR ix
DAFTAR TABEL x

BAB I PENDAHULUAN 1
1.1 Latar Belakang 1
1.2 Rumusan Masalah 2
1.3 Tujuan 2
1.4 Batasan Masalah 2
1.5 Metode Penelitian 3
1.6 Sistematika Penulisan 3

BAB II LANDASAN TEORI 5
2.1 Game 5
2.2 Sejarah Game 6
2.3 Jenis-Jenis Game 8
2.4 RPG 10
2.4.1 Elemen Pada RPG 10
2.5 NPC 12
2.6 FSM 12
2.7 RPG Maker 13
2.7.1 Komponen Utama RPG MAKER VX Ace 14

BAB III ANALISIS DAN PERANCANGAN 17
3.1 Analisis Sistem 17
3.1.1 Pengenalan Game Para Pencari Kebenaran 17
3.1.2 Storyline 17
3.1.3 Analisis Kebutuhan Nonfungsional 17
3.1.3.1 Analisis Kebutuhan Perangkat Lunak 18
3.1.3.2 Analisis Kebutuhan Perangkat Keras 18
3.1.3.3 Analisis Penguna 18
3.1.4 Analisis Kebutuhan fungsional 18
3.2 Perancangan Sistem 19
3.2.1 Perancangan Struktur Game 19
3.2.2 Flowchart Aplikasi 20

3.3 Perancangan Metode FSM pada NPC..... 23
3.4 Perancangan Layout 25
3.4.1 Perancangan Layout Menu Utama 25
3.4.2 Perancangan Menu Lanjutkan..... 26
3.4.3 Perancangan Menu Papan Nilai..... 27
3.4.4 Perancangan Menu Desain Sprite Karakter..... 28

BAB IV IMPLEMENTASI DAN PENGUJIAN 30
4.1 Implementasi 30
4.1.1 Implementasi Menu Utama..... 30
4.1.2 Implementasi Tampilan Menu Lanjutkan 31
4.1.3 Implementasi Menu Papan Nilai..... 31
4.1.4 Implementasi Metode Finite State Machine Pada NPC..... 32
4.2 Pengujian Sistem..... 37
4.3 Pengujian Resolusi Game..... 41

BAB V PENUTUP..... 42
5.1 Kesimpulan..... 42
5.2 Saran..... 42

DAFTAR PUSTAKA..... 43
LAMPIRAN..... 44

DAFTAR GAMBAR

Gambar 2.1 Contoh Diagram State Sederhana 12
Gambar 2.2 Database RPG Maker VX Ace 14
Gambar 2.3 Map Editor..... 16
Gambar 3.1 Struktur Game..... 19
Gambar 3.2 Flowchart Aplikasi..... 21
Gambar 3.3 Diagram State NPC Musuh..... 23
Gambar 3.4 Diagram State NPC Quest..... 24
Gambar 3.5 Diagram State NPC Pendukung..... 24
Gambar 3.6 Rancangan Layout Title Menu..... 25
Gambar 3.7 Layout Menu Lanjutan..... 26
Gambar 3.8 Perancangan Layout Menu Papan Nilai..... 27
Gambar 3.9 Sprite Aditya..... 28
Gambar 3.10 Sprite Gori 29
Gambar 3.11 Sprite Diman 29
Gambar 3.12 Sprite Memey..... 29
Gambar 3.13 Sprite Duan..... 29
Gambar 4.1 Tampilan Menu Utama..... 30
Gambar 4.2 Tampilan Menu Lanjut 31
Gambar 4.3 Tampilan Papan Nilai Tertinggi 32
Gambar 4.4 State NPC Musuh Patrol 33
Gambar 4.5 State NPC Musuh Battle Mode..... 34
Gambar 4.6 State NPC Musuh Dead..... 34
Gambar 4.7 State Awal NPC Quest..... 35
Gambar 4.8 State Checking NPC Quest 36
Gambar 4.9 State NPC Quest..... 36
Gambar 4.10 State NPC Pendukung..... 37

DAFTAR TABEL

Tabel 4.1 Pengujian System Game..... 37
Tabel 4.2 Pengujian FSM Musuh..... 38
Tabel 4.3 Pengujian FSM Quest..... 39
Tabel 4.4 Pengujian FSM Pendukung..... 39
Tabel 4.5 Pengujian Terhadap 10 Responden..... 39
Tabel 4.6 Pengujian Resolusi Screen Game..... 41

BAB I
PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi sekarang ini sudah semakin maju, sehingga banyak membantu masyarakat dunia untuk menikmati berbagai kemudahan yang di hasilkan teknologi tersebut. Salah satunya ada teknologi *game*, seiring dengan perkembanganan *gadget*, *game* banyak diminati oleh kalangan anak-anak hingga dewasa. pada tahun 2000-an dikenal *game* yang disebut dengan *game playstation* (PS) yang pada saat awal kemunculannya pada saat itu langsung menjamur di Indonesia. Banyak rental-rental penyewaan PS di buka saat itu karena banyak peminat *game playstation*. *Game* PS ini merupakan sebuah permainan yang mengunakan TV sebagai tampilan dan hanya bisa di mainkan oleh satu atau dua orang saja melalui sebuah *console*. Di *Game playstation* bisa di mainkan berberapa genre seperti *action, racing, fighting, strategy, RPG*, dan masih banyak yang lainnya.

Salah satu genre *game* adalah *Role Playing Game*. *Role Playing Game* atau yang biasa disebut dengan RPG merupakan permainan dimana *player* memainkan sebuah tokoh tertentu dan berkolaborasi dengan tokoh lain untuk melakukan suatu alur cerita. Didalam *game* pemain dapat melakukan improvisasi dalam menentukan hasil akhir dari pemainan. Biasanya *user* harus melakukan suatu perjalan dan melawan *monster-monster* untuk memperkuat karakter melakukan *quest-quest* yang terdapat di sepanjang permainan supaya bisa menyelesaikan permainan bergenre seperti ini

Di *game* yang bergenre RPG salah satu *factor* yang sangat penting adalah *factor* NPC. *Non playable character* atau NPC adalah suatu karakter pada game atau simulasi yang perilakunya tidak dapat dikendalikan oleh *user*. Contoh yang paling banyak ditemui dalam suatu *game* adalah musuh yang biasa kita lawan ketika bermain *game*. Misalkan suatu NPC dalam *game* memiliki perilaku yang cenderung berulang-ulang maka kerap kali *user* akan cepet merasa bosan. Oleh karena itu di

dalam penelitian ini saya akan mencoba membuat suatu *game* RPG dengan mengunakan metode *finite state machine* dalam pemberian AI pada NPC tersebut supaya *game* yang di hasilkan lebih terlihat dinamis karena dengan metode *finite state machine* kita dapat memberikan NPC lebih banyak pola perilaku supaya, dapat mengurangi unsur pengulangan dalam perilaku NPC.

1.2 Rumusan Masalah

Adapun rumusan masalah yang akan dibahas adalah bagaimana membuat sebuah *game* RPG dengan mengunakan *metode Finite State Machine* sehingga *game* yang akan dibuat akan terlihat lebih dinamis.

1.3 Tujuan

Tujuan dari penulisan skripsi ini adalah untuk membangun sebuah *game* RPG dengan mengunakan metode *Finite State machine* untuk memberikan kecerdasan buatan pada NPC.

1.4 Batasan Masalah

Batasan masalah disediakan untuk menjaga pembahasan yang di ambil agar sesuai dengan tujuan dan tidak terjadi penyimpangan maksud dan tujuan utama, maka di tentukan ruang lingkup pembahasan sebagai berikut :

1. *Game* yang akan dibuat adalah *game* RPG dan hanya dimainkan oleh satu pemain
2. Game ini mengunakan kecerdasaan buatan (AI) dengan metode *finite state machine* (FSM) sebagai pemetaan perilaku NPC
3. *Game* ini di buat denggan mengunakan *rubby game scripting system* RGSS dengan *game engine RPG MAKER*

1.5 Metode Penelitian

Adapun meteode penelitian yang digunakan sebagai berikut :

1. Studi Literatur

   Pengumpulan dilakukan dengan cara mencari bahan–bahan refrensi dari berbagai sumber sebagai landasan teori yang berhubungan dengan permasalahan yang di jadikan objek penelitian

2. Analisa Kebutuhan

   Data dan informasi yang di peroleh akan di analisa agar mendapat suatu kerangka yang dapat di gunakan untuk perancangan *game*.

3. Perancangan Aplikasi

   Setelah pengumpulan data selesai selanjutnya yaitu ketahap perancangan *game* RPG

4. Pengujian

   Jika *game* telah selesai selanjutnya ketahap pengujian,hal ini dilakukan untuk mengetahui apakah *game* yang dirancang sudah benar

1.6 Sistematika Penulisan

Sistematika penulisan ditujukan untuk memberikan gambaran dan uraian dari skripsi secara garis besar yang meliputi bab-bab sebagai berikut :

BAB I : PENDAHULUAN

Pada Bab ini membahas tentang Latar Belakang, Rumusan Masalah, Batasan Masalah, Maksud dan Tujuan Penelitian, Manfaat Penelitian, Metode Penelitian dan Sistematika Penulisan Laporan Penelitian.

BAB II : LANDASAN TEORI

Pada Bab ini membahas tentang Landasan Teori yang merupakan tinjauan pustaka, menguraikan teori-teori yang mendukung judul dan pembahasan secara detail. Landasan teori dapat berupa definisi-definisi atau model yang langsung berkaitan dengan ilmu atau masalah yang diteliti. Pada bab ini juga dituliskan tentang

software (komponen) yang digunakan dalam pembuatan *Program* atau keperluan saat penelitian.

BAB III : ANALISIS DAN PERANCANGAN

Bab ini berisi antara lain: Dalam bab ini berisi mengenai perancangan aplikasi dimulai dari desain antar muka *(interface)*, pemrograman dan pengujian.

BAB IV : IMPLEMENTASI DAN PENGUJIAN

Pada Bab ini akan membahas paparan implementasi dan analisis hasil uji coba *program*, serta memaparkan hasil-hasil dari tahapan penelitian, dari tahap analisis, desain, implementasi desain, hasil *testing* dan implementasinya, berupa penjelasan teoritik, baik secara kualitatif, kuantitatif, atau secara statistic, serta membandingkannya dengan hasil penelitian yang masih manual.

BAB V : PENUTUP

Pada Bab ini berisi kesimpulan dan saran. Kesimpulan didapat dari ulasan data-data penelitian, menyimpulkan bukti-bukti yang diperoleh dan akhirnya menarik intisari apakah hasil yang didapat (dikerjakan) adalah layak untuk digunakan (di implementasikan).

BAB II
LANDASAN TEORI

2.1 *Game*

Menurut Jack Febrian dalam bukunya "Pengetahuan Komputer dan Teknologi Informasi" terbitan INFORMATIKA bandung, *game* permainan, di dalam dunia teknologi informasi istilah ini digunakan untuk sarana hiburan yang mengunakan perangkat elektronik, Sedangkan pengertian *video game* adalah permainan yang menggunakan interaksi dengan *interface* pengguna melalui gambar yang dihasilkan oleh piranti *video*. Permainan *video* umumnya menyediakan sistem penghargaan misalnya *score* yang dihitung berdasarkan tingkat keberhasilan yang dicapai dalam menyelesaikan tugas-tugas yang ada di dalam permainan

Game merupakan permainan yang merujuk pada kelincahan intelektual atau *intelellectual capability*. kelincahan intelektual pada tingkatan tertentu merupakan ukuran sejauh mana *game* tersebut menarik untuk dimainkan. Tidak hanya pada kelincahan intelektual saja, *game* juga mampu melatih kemampuan seseorang untuk berfikir dan bertindak dalam memecahkan suatu masalah dengan tepat dan cepat. Karena didalam sebuah *game* terdapat berbagai atau masalah yang menuntut sang pemain dapat menyelesaikan dengan hasil maksimal.

Finite state machine merupakan metodologi kecerdasan buatan yang memperlihatkan perilaku sistem berdasarkan tiga hal, yaitu *state (*keadaan*), event* (kejadian) dan *action* (aksi). Pada suatu saat sistem akan berada pada salah satu *state* yang aktif. Kemudian sistem akan beralih atau bertransisi menuju *state* yang lain jika mendapatkan suatu *event* tertentu. Transisi keadaan ini umumnya juga di sertai oleh aksi yang di lakukan oleh sistem ketika menanggapi masukan yang terjadi.

2.2 Sejarah *Game*

Dalam perjalanannya *game* juga memiliki sejarah yang cukup panjang. Berikut adalah sejarah *game* mulai dari generasi pertama hingga sekarang :

1. *Game* Generasi Pertama

Generasi pertama *console* permainan *video* berlangsung dari tahun 1972, saat itu banyak orang yang masih belum mengenal *console game* dan *computer game* dengan rilis dari Magnavox Odyssey. Meluncurkan *video game* bernama *Odyssey. Odyssey* adalah *console game* pertama di dunia yang mengoperasikan ***PONG*** hingga 1977, ketika ***"PONG-STYLE"*** mengalami kebangkrutan maka sang produsen *console* meninggalkan pasar secara massal karena pengenalan dan keberhasilan mikroprosesor berbasis *console* .

2. *Game* Generasi Kedua

Pada tahun 1976 , *Fairchild* mencoba menghidupkan kembali dunia *video game* dengan menciptakan *VES (Video Entertainment System). VES* adalah mesin pertama yang disebut *console. Console* ini mengunakan kaset magnetik yang disebut *cartridge.* Konsep ini kemudian diikuti oleh beberapa produsen lain termasuk *Atari, Magnavox, dan RCA* dengan cara merilis *console game* serupa.

Pada tahun 1977, dunia *game* menjadi ambruk, karena *game* yang tersedia di pasaran tidak ada yang berhasil menarik minat pembeli. *Fairchild* dan *RCA* pun mengalami kebangkrutan. Beberapa saat kemudian *Atari* meluncurkan *console* yang cukup melegenda yaitu *Atari* 2600, yang terkenal dengan *game space invaders*.

Tahun 1983 dunia *game* kembali mengalami penurunan. Kurangnya *game* yang kreatif mengakibatkan *console game* kembali di tinggalkan. Beberapa contoh *console game* pada generasi kedua Dari **KI-KA** adalah *Fairchild Channel F,Atari* 2600, *Magnavox Odyssey ver.* 2 , *Atari* 5200.

3. *Game* Generasi Ketiga

Generasi ketiga dimulai pada tahun 1983 dengan dipasarkannya *Jepang Family Computer* atau lebih dikenal dengan nama *FAMICOM* (kemudian dikenal sebagai *Nintendo Entertainment System* di seluruh dunia). Walaupun *console* generasi sebelumnya juga menggunakan 8-bit *processor*, pada akhir generasi inilah *console* rumah yang pertama kali diberi label oleh mereka dengan merek "bit". Ini juga masuk ke mode sebagai sistem 16-bit seperti *Mega Drive / Genesis* dipasarkan untuk membedakan antara generasi *console*. Di Amerika Serikat, generasi *game* ini terutama didominasi oleh *NES / Famicom*. Di era ini pulalah terjadi perang *console game* yang pertama antara perusahaan *console NINTENDO* dengan *SEGA*.

4. *Game* Generasi Keempat

Generasi keempat atau biasa disebut dengan era 16 bit,pada generasi ini *NES* mendapat sambutan hangat di seluruh dunia, dan sebuah perusahaan bernama *SEGA* mencoba menyaingi *Nintendo*. Pada tahun 1988, *SEGA* merilis *console next-generation* mereka yaitu *Sega Mega Drive* (yang juga dikenal dengan SEGA GENESIS). *Console* ini menyajikan gambar yang lebih tajam dan animasi yang lebih halus dibanding *NES*. *Console* ini cukup berhasil memberi tekanan, tetapi *NES* tetap bertahan dengan angka penjualan yang tinggi. Dua tahun berselang, pada 1990, *Nintendo* kembali menggebrak dengan *console next-gen* mereka, *SNES* (*Super Nintendo Entertainment System*). Selama 4 tahun, *Nintendo* dan *Sega* menjadi bebuyutan, meskipun ada beberapa produsen seperti *SNK* dengan *NeoGeo*-nya, *NEC* dengan *Turbo Grafx-16* dan *Phillips CD-i*, tapi kedua *console* mereka begitu handal dan populer.

5. *Game* Generasi Kelima

Generasi kelima atau disebut juga dengan era *console* 32 bit. Pada tahun 1990-1994, SEGA dan NINTENDO tetap bersaing. Berbagai *game* fenomenal diliris. SNES menyertakan *chip* super fx pada *cartridge*, dan SEGA mengunakan *Sega*

Virtual Processor. Keduanya bertujuan untuk meningkatkan kualitas grafis dari *game.*

6. *Game* Generasi Keenam

Generasi keenam ini ditandai dengan munculnya *console-console game next generation* dari masing-masing perusahaan seperti SONY, SEGA, NINTENDO serta munculnya satu lagi *console game* baru yang diluncurkan oleh MICROSOFT yang diberi nama XBOX. Perang *console game* ini akhirnya mengakibatkan jatuhnya perusahaan *console* SEGA yang tidak dapat lagi meneruskan *console next generation (Dreamcast)* mereka dan lebih memilih untuk berkonsentrasi dibidang pembuatan *game console.*

7. *Game* Generasi Ketujuh

Dikarenakan semakin canggihnya teknologi di bidang komputerisasi maka kemudian 3 perusahaan *console* terbesar (SONY, NINTENDO dan MICROSOFT) mengeluarkan kembali *console next-generation* yang terbaru. SONY mengeluarkan *console next-gen* yang diberi nama PS3 *(Playstation3)*, lalu NINTENDO dengan NINTENDO WII, kemudian MICROSOFT dengan XBOX 360. Pada generasi ini semakin dikembangkan sistem permainan *online* atau permainan yang melibatkan banyak pemain yang terhubung dengan console mereka dan semakin ditinggalkannya permainan *single player.*

2.3 Jenis-Jenis *Game*

Berikut ini adalah beberapa jenis *game* yang ada hingga saat ini :

1. Action games

Permainan aksi (*Action games*) memerlukan pemain untuk menggunakan refleks, akurasi, dan waktu yang tepat untuk menyelesaikan sebuah tantangan. Ini mungkin merupakan genre dasar dari sebuah permainan, dan yang memiliki permainan terbanyak. Dalam permainan aksi, biasanya terdapat pertempuran.

2. *Fighting games*

Permainan pertarungan (*Fighting games*) adalah ragam permainan dimana terjadi pertarungan satu lawan satu antara dua karakter, salah satunya dapat merupakan komputer. Ragam permainan ini biasanya dimainkan dengan menghubungkan kombinasi-kombinasi tombol untuk bertarung. Rata-rata gerakan yang dihasilkan oleh karakter dramatis dan tidak mungkin dilakukan di kehidupan nyata.

3. *Shooter games*

Permainan tembak-menembak (*Shooter games*),, adalah sub-genre dari permainan aksi, meskipun banyak pemain yang menganggap bahwa ragam ini merupakan ragam yang berdiri sendiri. Ragam ini menguji kecepatan dan reaksi dari pemainnya. Tembak-menembak memiliki sub-genre tersendiri, yang semuanya terfokus pada "mengontrol seseorang yang menggunakan sebuah senjata, biasanya pistol, senapan, atau senjata jarak jauh lainnya". Hal yang biasanya ada pada permainan ragam ini adalah amunisi. Biasanya tujuan dari permainan ini adalah menembak lawan dan melanjutkan misi tanpa karakter yang dipakai mati.

4. *Racing games*

Permainan balapan *(Racing games)* juga termasuk dalam kategori ini,dimana ragam permainan *video* yang menuntut keterampilan pemain untuk mengemudi dalam sebuah kompetisi balap-membalap.

5. *RPG games*

Permainan *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 tokok-tokoh mereka berdasarkan karakteristik tokoh tersebut, dan keberhasilan aksi mereka tergantung dari sistem peraturan permainan yang telah ditentukan. Asal tetap mengikuti peraturan yang ditetapkan, para pemain bisa berimprovisasi membentuk arah dan hasil akhir permainan ini.

2.4 RPG

RPG (role-playing game) merupakan *game* yang menekankan pada perkembangan karakter. Pengembangan karakter pada *game RPG* umumnya melalui pertarungan. Unsur lain yang selalu ada pada *game RPG* adalah jalan cerita yang melibatkan tokoh utama dan tokoh lain. Tokoh-tokoh ini ada yang dapat dimainkan dan ada yang tidak dapat dimainkan. Tokoh yang tidak dapat dimainkan disebut *NPC (non playable character).*

2.4.1 Elemen Pada RPG

1. Story line

Cerita dalam *RPG* adalah salah satu unsur yang paling utama, tanpa adanya cerita yang menarik ,kreatif, *RPG* akan sangat membosankan. Cerita yang disajikan dalam *game RPG* (*Role Playing Game*) harus terkesan kreatif, inovatif, dan dapat menceritakan perkembangan karakter utama serta pemain pendukung dalam *game* yang berperan membantu karakter utama serta inti cerita dari suatu *game* tersebut. Pada elemen *Storyline* ini. Dibagian *storyline* ini biasanya akan muncul suatu permasalahan yang harus di selesaikan oleh *player* dan bagaimana cara *player* dapat menyelesaikan *game* tersebut.

2. Battle system

Ada beberapa jenis *battle system* yang sering digunakan pada *game* bergenre RPG ini salah satu contohnya adalah:

a. Turn Based Battle System (Default RPG Maker)

System ini adalah *system default* yang dipakai oleh *RPG Maker*. Dimana *actor* dan *enemy* punya giliran satu sama lain. *Actor* mengambil aksi melalui menu, dan dieksekusi setelahnya.

b. Action Battle System (ABS)

System ini adalah *battle system* langsung di tempat. *Actor* langsung melancarkan serangan kepada musuh dan musuh juga langsung melancarkan serangan kepada *actor*.

c. *Tactical BattleSystem*

System ini adalah *battle system* tipe taktik, dimana *player* harus menempatkan posisi *actor* pada sebuah *grid map* lalu memposisikannya pada posisi tertentu sehingga menentukan efektifitas *battle* tersebut

3. *Menu System*

Menu System berisi petunjuk dan manual yang ada dalam *RPG* tersebut. Menu ini sangat diperlukan oleh pemain dalam memahami dan mensetting karakternya sehingga dapat mendapatkan karakter yang sesuai dengan keinginan sang pemain.

4. *Status Efect*

Berikut adalah status-status yang biasanya muncul pada *game* RPG yaitu :

a. *Confuse*

Status ini cukup merugikan karakter. Karakter (*Player* atau musuh) tidak akan bisa menyerang lawan dengan tepat atau serangan akan mengenai rekan sendiri dikarenakan karakter tersebut pusing dan hilang kesadaran sejenak. Status efek ini dapat dihilangkan dengan memberikan obat penawar, atau tetap membiarkannya berlangsung karena status ini akan hilang dengan sendirinya setelah 2 atau 3 kali putaran giliran permainan.

b. *Paralysis*

Status ini menyebabkan karakter yang bersangkutan tidak bisa bergerak maupun mengeluarkan kemampuannya sampai efek dari status ini menghilang. Menghilangkan efek dari status ini adalah dengan memberikan obat penawar pada karakter (*Player* atau musuh).

c. *Sleep*

Salah satu status yang juga tidak menguntungkan bagi karakter (*Player* atau musuh), karakter yang terkena *effect* ini akan tertidur untuk beberapa saat dan akan terbangun kembali setelah beberapa giliran permainan atau bisa juga disadarkan dengan menggunakan obat penawar yang sudah dimiliki.

2.5 NPC

Non playabe character (*NPC*) adalah suatu karakter pada *game* atau simulasi yang perilakunya tidak dapat dikendalikan oleh *user*. Contoh yang paling banyakditemui dalam suatu *game* adalah musuh yang dihadapi ketika bermain *game*. Keberadaan *NPC* seringkali menjadi faktor yang menyebabkan *user* memainkan *game* tersebut terus – menerus.

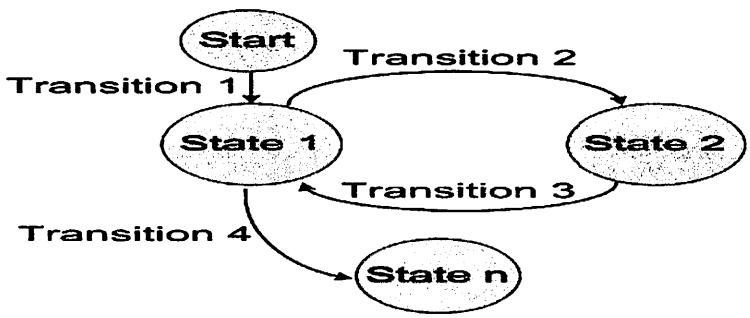
Banyak teknik yang digunakan untuk membuat *NPC* yang dapat berperilaku realistis dan bervariasi. Salah satu teknik yang paling banyak digunakan adalah *Finite State Machine (FSM)*. Kelebihan utama dari *FSM* adalah implementasi yang mudah dan hasil yang cukup memuaskan pada tingkat tertentu.

2.6 FSM

Finite State Machine (FSM) merupakan pemodelan dari perilaku *(behavior)* sebuah sistem atau obyek yang kompleks dengan beberapa kondisi atau mode yang terdefinisikan dimana mode transisi berubah sesuai dengan keadaan.

FSM terdiri dari empat elemen utama :

1. *State* yang mendefinisikan kelakuan dan mungkin menghasilkan aksi.
2. Transisi *state* dimana merupakan perpindahan dari satu *state* ke *state* lain.
3. Aturan atau kondisi yang harus dipenuhi supaya ada transisi *state* kejadian *(events)*.
4. *Input* yang terjadi baik *internal* maupun *external*, yang memungkinkan trigger aturan dan mengacu ke transisi *state*.



Gambar 2.1 Contoh *Diagram State* Sederhana

Pada gambar 2.1 diagram tersebut memperlihatkan contoh *state digram fsm* sederhana.

2.7 RPG Maker

RPG maker adalah sebuah program untuk membuat *game RPG*. Proses pembuatan dibantu *tools* yang sudah tersedia didalam program. Program ini dibuat oleh grup Jepang bernama ASCII, kemudian diteruskan oleh ENTERBRAIN. *RPG maker* telah di luncurkan di jepang dan negara luar Jepang seperti kawasan Asia Timur, Amerika Utara, Eropa, dan Australia.

RPG Maker VX Ace adalah *software game engine* seri kelima dalam seri PC *RPG Maker*, dimulai dari seri *RPG Maker 2000*, *RPG Maker 2003*, *RPG Maker XP*, *RPG Maker VX*, dan *RPG Maker VX Ace* . Seperti *RPG Maker* sebelumnya, *RPG Maker VX Ace* memiliki antarmuka grafis *user-friendly* (sangat bersahabat) yang sudah ditingkatkan set fiturnya sehingga sangat mengagumkan dalam pengolahan *maping* (perancangan peta dalam *game*), pembuatan karakter dengan bantuan *tools Character Generator*, dan penentuan atribut-atribut dari suatu karakter yang akan dibuat nantinya. *RPG Maker VX Ace* menggunakan *Database* yang terintegrasi didalam *engine software* dalam pengelolaan segala atribut dari karakter, *item*, jurus yang akan dipakai dan bahan-bahan sistem yang semakin dilengkapi, memungkinkan bahkan bagi pemula untuk menciptakan *role-playing game* nya sendiri tanpa perlu memiliki pengetahuan pemrograman.

RPG Maker VX Ace memiliki fitur yang semakin diperbarui, selain fiturnya yang menarik, keefektivannya yang dapat membuat siapa saja cepat memahami dan mendesain *game* sesuai keinginan mereka. *RPG Maker VX Ace* dilengkapi dengan *tools tileset* yang sudah tersedia seperti *pintu*, *losmen*, *gunung*, *laut*, *gua*, *karakter*, *player*, *item* dan sebagainya. *Tools* pembuatan karakter dengan *Character Generator*, *tools* pengelola *database*, *tools eventing* (adalah *tools* dimana kita dapat mengolah *game* buatan kita semenarik mungkin) dan lain-lain.

2.7.1. Komponen Utama RPG Maker VX Ace

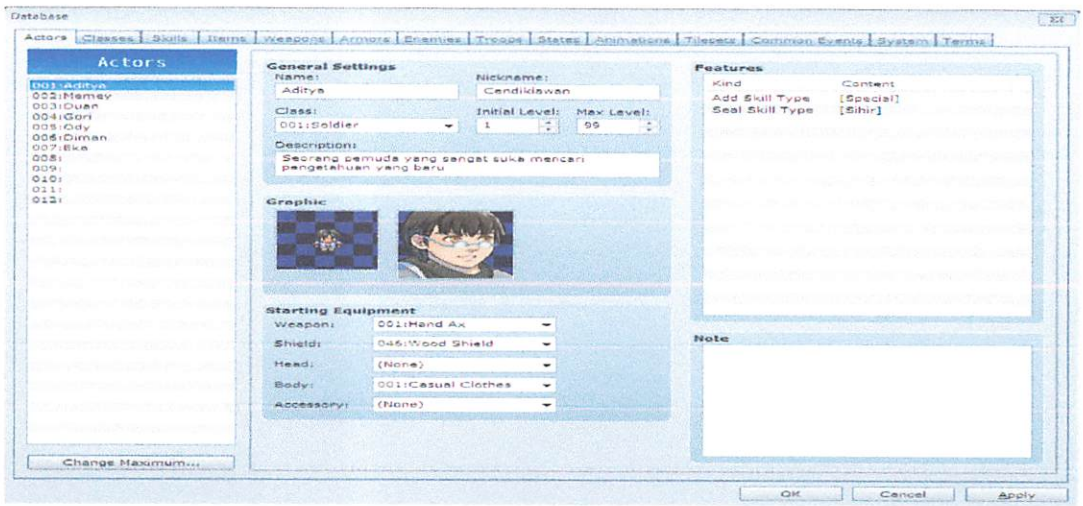
Pada *RPG maker VX Ace* terdapat beberapa komponen utama yang biasanya di gunakan dalam pembutan suatu game. Berikut komponen-komponen utama yang terdapat pada *RPG Maker VX Ace*:

1. RGSS3

RGSS3 (Ruby Game Scripting System), merupakan struktur dasar dari *RPG Maker VX Ace* yang harus di akses oleh penguna. *RGSS3* merupakan pengembangan dari bahasa pemprograman yang di kembangankan dari versi sebelumnya sehingga lebih mudah dan stabil digunakan. *RGSS3* mengunakan skrip bahasa pemprograman Ruby, yang menginterprestasikan segala komponen dalam *game*. Skrip ini mengatur mulai dari sistem pertarungan sampai pembuatan peta. *RGSS3* merupakan bahasa pemprograman yang kuat. oleh karena itu, memungkinkan kita untuk mengatur berbagai aspek dalam *game* yang dibuat.

2. Database

Database merupakan tempat untuk membuat sebagian besar komponen *game*. *Database* dibedakan dalam bentuk *tab*. Setiap *tab* akan berisi sebuah kategori seperti gambar dibawah ini :



Gambar 2.2 *Database RPG Maker VX Ace*

Pada *database* terdiri dari beberapa *tab*, yaitu:

a) *Actors*. *Tab* ini berisi data untuk membuat karakter utama. Pada bagian ini Anda dapat menentukan dimana karakter memulai perannya di dalam game.

b) *Classes*. *Tab* ini digunakan untuk mengatur kelas yang akan di pakai karakter. Perbedaan kelas, membuat perkembangan dari tiap karakter dan perlengkapan dari karakter tersebut.

c) *Skills dan Animation*. *Tab* ini mengatur kemampuan pergerakan karakter.

d) *Items,Weapon* dan *Armor*. *Tab* ini mengatur benda,senjata, dan perlengkapan karakter.

e) *Enemies* dan *Troops*. *Tab* ini berisi data untuk membuat musuh.

f) *States*. *Tab* ini mengatur perkembangan karakter dalam game.

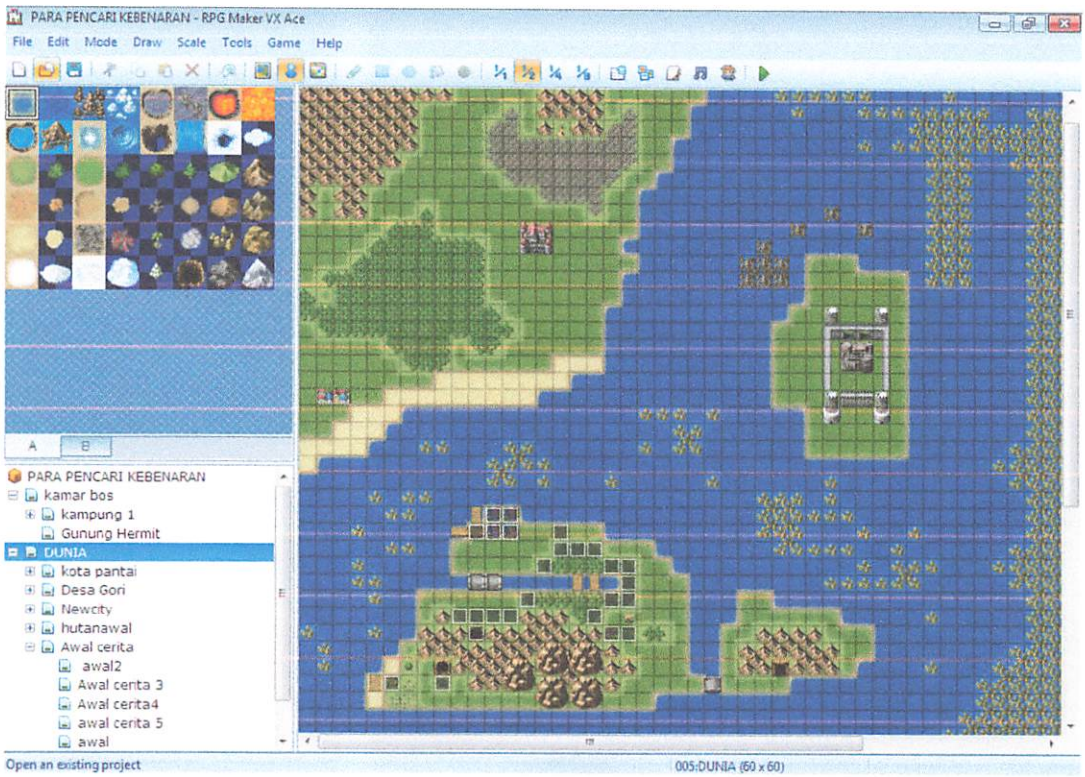
g) *System* dan *Term*. *Tab* ini berisi standar pengaturan, seperti sound effect dan menu pada game.

h) *Common Events* dan *Tileset*. *Tab* ini untuk mengatur event yang terjadi di dalam game dan aksesoris pendukung pembuatan game.

3. Map Editor

Map Editor merupakan tempat untuk membuat peta yang akan dijelajahi oleh karakter. Peta dibuat untuk dengan memilih yang ada di kiri map editor. Kita dapat mengatur Kondisi peta seperti hutan, gunung, padang pasir, padang rumput, dan lain sebagainya. Untuk peta kota anda dapat membuat rumah, taman, jalan, dan lain sebagiannya. Pada setiap kota akan terdapat bangunan, oleh karena itu disediakan juga aksesoris untuk mengisi bangunan seperti meja, tempat tidur, hiasan ruangan, dan lain sebagiannya.

Pada *map editor* kita dapat menentukan daerah mana saja yang dapat di jelajahi dan daerah mana saja yang tidak dapat dijelajahi. Kita dapat membagi wilayah-wilayah pada *game*. Pembagian berguna untuk membagi alur cerita di dalam game menjadi beberapa tempat.



Gambar 2.3 *Map Editor*

4. *Events*

Pada *RPG Maker, events* memiliki peran yang cukup penting. Pertama, untuk memberikan karakter untuk berinteraksi dengan peta yang ada di dalam *game*. Hal ini termasuk berinteraksi dengan *NPC*, peti harta, kendaraan dan lain sebagiannya. Kedua, untuk mengatur aliran adegan yang ada didalam *game*. Contoh pada peran ini adalah perubahan percakapan oleh *NPC* dan lain sebagainnya.

BAB III
ANALISIS DAN PERANCANGAN

3.1. Analisis Sistem

Analisis merupakan tahapan awal didalam membanguna sebuah *game*. analisa adalah tahapan sebelum kepada perancangan. Dimana pada tahap ini, penulis menguraikan pokok permasalahan dan mengumpulkan informasi yang dibutuhkan unuk ke tahap perancangan.

3.1.1 Pengenalan Game Para Pencari Kebenaran

Game yang akan dibanguan adalah *game* **"Para Pencari Kebenaran"** dengan genre *role playing game*. Fitur-fitur yang digunakan pada *game* ini adalah :

1. Grafik yang digunakan adalah 2 dimensi.
2. *Game* bergenre RPG *game*.
3 Kecerdasan buatan mengunakan *Finite State Machine*.

3.1.2 Storyline

Newcity adalah suatu daerah kepulauan yang mana penghasilan penduduknya berasal dari perdagangan dan nelayan. Suatu ketika kota *newcity* yang biasanya hidup dengan damai mendadak sering mendapatkan musibah. Dari rumah penduduk yang sering terbakar. Kapal-kapal pedagang yang menuju ke *newcity* sering tengelam dan kejadian-kejadian aneh yang lainnya. Hingga suatu saat para penduduk pun mulai meninggalkan kota *newcity*. Kota yang dulunya ramai sekarang telah menjadi kota tua yang tertinggalkan.

Tujuan utama dari *game* ini adalah memecahkan masalah yang di hadapi *newcity* dan membuat *newcity* kembali menjadi seperti dulu lagi.

3.1.3 Analisis Kebutuhan Non-Fungsional

Kebutuhan nonfungsional adalah tipe kebutuhan yang berisi properti perilaku yang dimiliki oleh sistem. Spesifikasi *hardware* dan *software* yang dibutuhkan untuk membuat *game* **"Para Pencari Kebenaran"** adalah sebagai berikut :

3.1.3.1 Analisis Kebutuhan Perangkat lunak

Perangkat lunak yang digunakan untuk membangun *game RPG* **"Para Pencari Kebenaran"** ini adalah sebagai berikut :

1. *Windows 7 Ultimate 32 Bit* sebagai sistem operasi.
2. *RPG Maker VX Ace*, perangkat lunak yang digunakan untuk membuat *game*.
3. *Adobe Photoshop CS3*, perangkat lunak yang digunakan untuk membuat *graphic, face sprite* tampilan awal *game RPG* **"Para Pencari Kebenaran**, dan lain - lain untuk desain perangkat lunak.

3.1.3.2 Analisis Kebutuhan Perangkat Keras

Dalam pembuatan *game* **"Para Pencari Kebenaran"** ini memerlukan spesifikasi perangkat keras yang digunakan. Adapun perangkat keras yang digunakan dalam pembuatan *game* **"Para Pencari Kebenaran"** ini adalah sebagai berikut :

1. *RAM 3 GB DDR 2*
2. *Harddisk 300 GB*
3. *14" WXGA LED, Max. Resolution 1280 x 768, Clear SuperView LED*
4. *Prosessor Intel (R) Core(TM) 2 Duo CPU*
5. *Standart VGA Graphic*

3.1.3.3 Analisis Penguna

Selain dibutuhkannya perangkat lunak dan perangkat keras, *user* juga sangat dibutuhkan dalam penggunaan *game* ini, Adapun spesifikasi *user* yang dibutuhkan antara lain :

1. Prioritas *game* ini dibuat untuk anak-anak dan remaja dari umur 10-20 tahun.
2. Pengguna umum yang mengerti dalam penggunaan *computer*.

3.1.4 Kebutuhan Fungsional

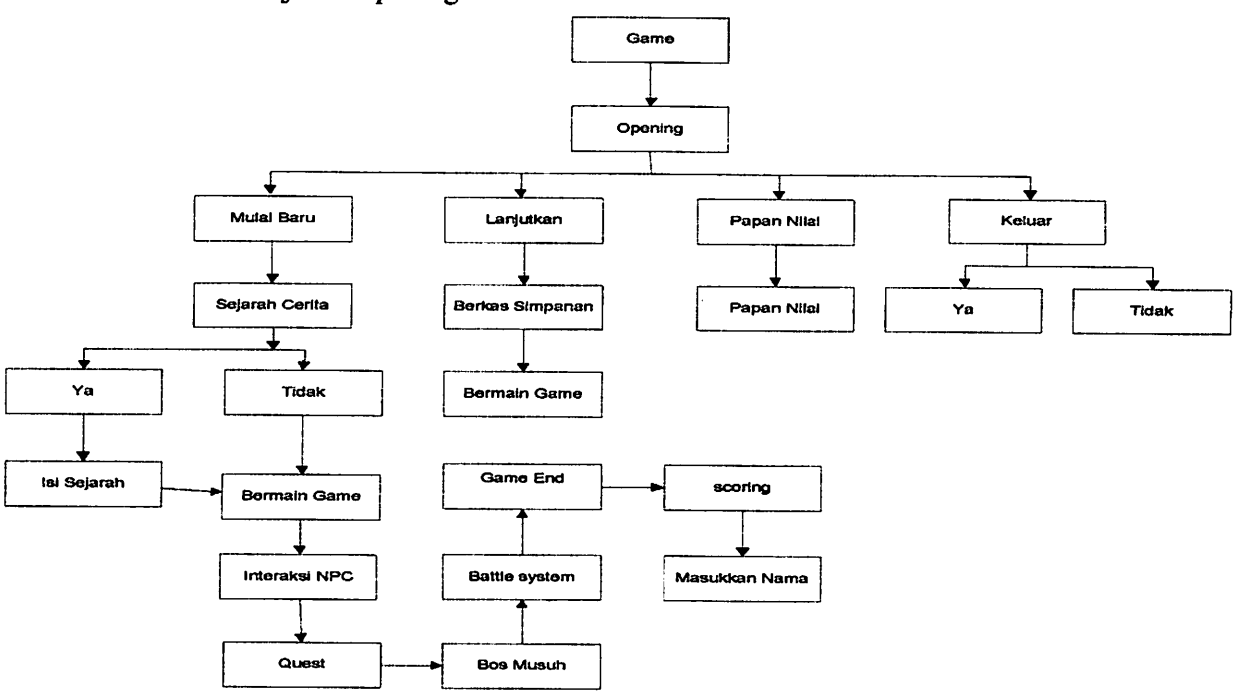
Jenis kebutuhan yang berisi proses2 yang nantinya dilakukan oleh sistem. Kebutuhan fungsional juga berisi informasi-informasi apa saja yang harus ada dan dihasilkan *system*.

3.2 Perancangan Sistem

Perancangan sistem adalah suatu bagian dari metodologi pengembangan pembangunan suatu perangkat lunak yang dilakukan untuk memberikan gambaran secara terperinci. Berdasarkan uraian diatas perancangan sistem merupakan tahapan dari siklus pengembangan sistem yang didefinisikan dari kebutuhan-kebutuhan fungsional dan persiapan untuk rancang banguan implementasi yang mengambar kan bagaimana suatu sistem dibentuk, yang dapat berupa penggambaran, perancangan, dan pembuatan sketsa.

3.2.1 Perancangan Struktur *Game*

Pada perancangan struktur *game* ini terdapat beberapa tahapan proses yang dapat dijalankan saat pertama kali *Game* sudah dimainkan. *Game* ini mempunyai beberapa menu dan fitur yang sudah disediakan bagi *player* untuk mengakses data berupa Menu mulai baru, lanjutkan, nilai tertinggi, dan keluar. Gambar perancangan struktur ditunjukkan pada gambar 3.1.



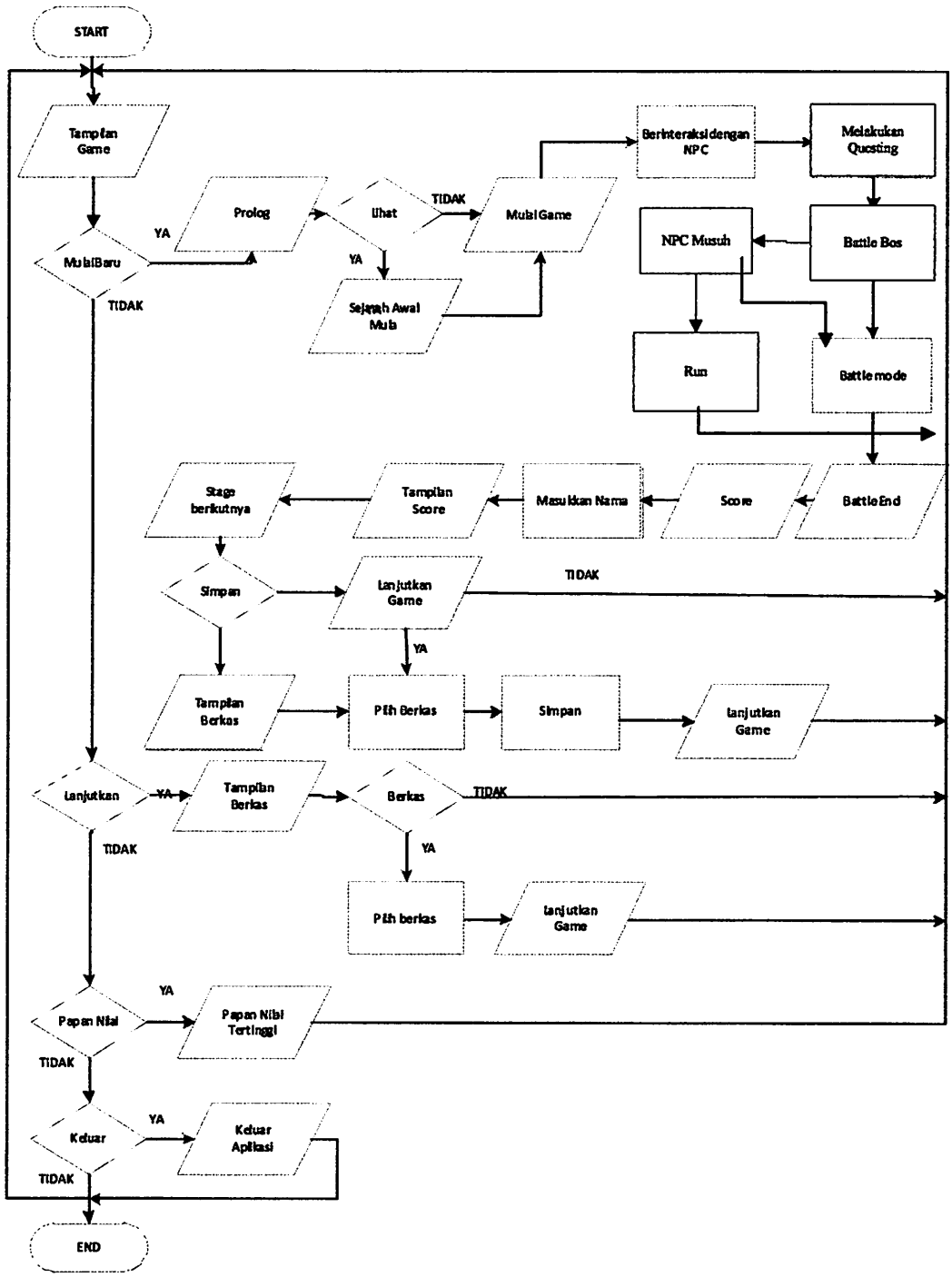
Gambar 3.1 Struktur *Game*

Pada Perancangan Struktur *game* dalam gambar 3.1 dapat dijelaskan sebagai berikut :

1. Proses *Opening* adalah proses pertama kali yang akan ditemui oleh *player* sebelum *player* dapat memainkan sebuah *game*. Pada tahap *Tampilan Awal Game* ini, *player* akan disuguhkan *wallpaper* dan *background* utama sebagai pendukung kelengkapan dalam *game* serta tombol navigasi yang memiliki beberapa menu fungsi.
2. Proses *Mulai Baru* adalah langkah awal seorang *player* untuk masuk kedalam *game* dan memainkannya.
3. Proses *Lanjutkan* adalah proses untuk melanjutkan dari save point yang sudah di simpan sebelumnya
4. Proses *papan nilai* adalah proses highscore yang berhasil player dapatkan.
5. Proses *Interaksi NPC* adalah proses dimana tampilan dari penerapan metode *finite state machine* ditunjukkan dalam sebuah pertarungan.
6. Proses *Battle System* adalah proses dimana terjadinya pertarungan antara *player* dan *NPC* musuh
7. Proses *Game End* adalah proses dimana player berhasil menyelesaikan *game* ini ketika pemain selesai, *player* dapat menginput nama kedalam papan nilai.

3.2.2 *Flowchart* Aplikasi

Flowchart perancangan *Game RPG* **"Para Pencari Kebenaran"** dimulai dari tampilan *Home*, tampilan menu-menu pendukung, proses *battle system*, hingga berakhirnya sebuah *game*, ditunjukkan pada gambar 3.2.



Gambar 3.2 *Flowchart* aplikasi

Pada *Flowchart* Perancangan *Game* dalam gambar 3.2 dapat dijelaskan sebagai berikut :

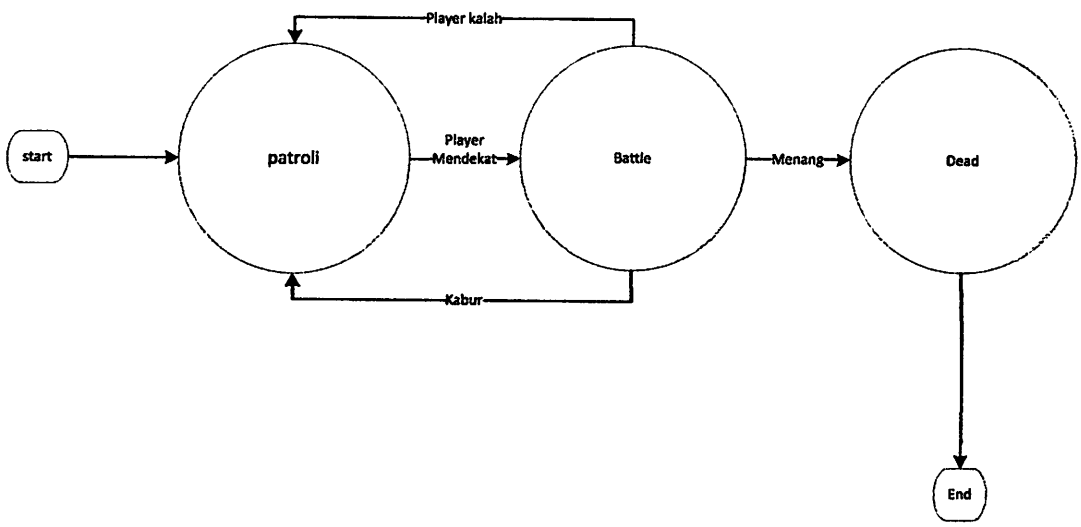
1. Dimulai dari tampilan *game* yang memiliki banyak menu.
2. Jika memilih menu *Mulai Baru*, maka *player* akan masuk pada tampilan *Prolog game*. Pada Menu *prolog game player* dapat melihat awal sejarah dari *game RPG* **"Para Pencari Kebenaran"**.
3. Di dalam sebuah permainan *game RPG (Role Playing Game), player* akan menghadapi pertarungan melawan musuh dalam perjalanannya. Pertarungan tersebut terjadi antar *karakter* dan masuk dalam *Battle Sistem*. Metode pertarungan antar *karakter* tersebut dengan prinsip *wait-and-see battle* (pertarungan bergiliran menyerang satu-satu) .
4. Metode *Finite state machine* akan digunakan untuk pemetaan perilaku *NPC* yang terdapat pada *game* ini.
5. Jika memilih menu *Lanjutkan*, maka *player* akan memilih berkas hasil simpanan dari permainan yang sudah dimainkan sebelumnya. Apabila *player* tidak menyimpan permainan sebelumnya yang sudah dimainkan, tombol menu *Lanjutkan* tidak dapat diakses. Hal itu menerangkan bahwa tidak ada simpanan permainan dalam berkas.
6. Jika memilih menu *Papan Nilai,* maka *player* akan dihadapkan pada tampilan Nilai tertinggi berdasarkan poin yang sudah dimainkan. Nilai berdasarkan pada kemenangan menyelesaikan *game*. Jika menang melawan bos akhir, maka nilai akan muncul dan *player* dapat menyimpan angka kemenangan yang didapat.
7. Jika memilih menu *Keluar*, maka *player* akan keluar dari permainan.

3.3 Perancangan Metode FSM Pada NPC

Dalam game ini terdapat beberapa kecerdasan buatan yang diberikan kepada *NPC* musuh, *NPC quest* dan *NPC* pendukung sebagai metode pemetaan perilakunya.

1. NPC Musuh

NPC musuh memiliki *state* awal berupa patroli di map ketika pemain tidak ada di sekitar *NPC*. Pada saat *player* mendekat *NPC* kemudian akan mengejar *player* saat *player* dan *NPC* musuh bersentuhan maka akan masuk ke *battle mode*. Kalau *player* menang maka *NPC* musuh akan masuk ke *state Dead* dan *player* mendapatkan hadiah berupa *xp* dan *gold*. *Diagram state* dari *NPC* musuh bisa dilihat di gambar 3.3 dibawah ini :

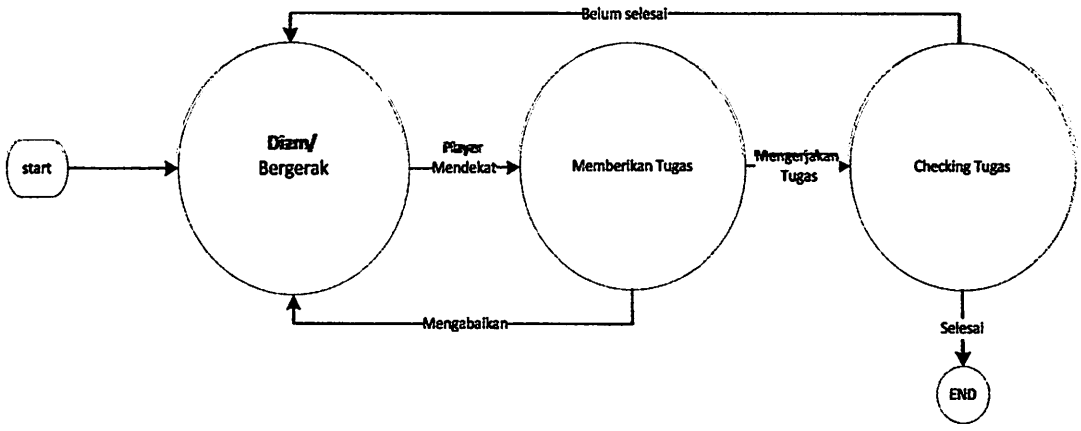


Gambar 3.3 *Diagram State NPC* Musuh

2. *NPC Quest*

NPC quest adalah *NPC* yang bertugas untuk memberikan *quest-quest* pada *game* sehingga *player* dapat meneruskan alur cerita yang ada. Pada *NPC quest state* awal yang dimilikinya adalah diam atau sedang berjalan, ketika *player* melakukan interaksi *NPC* akan menanyakan apakah dia mau menerima tugasnya, jika iya maka

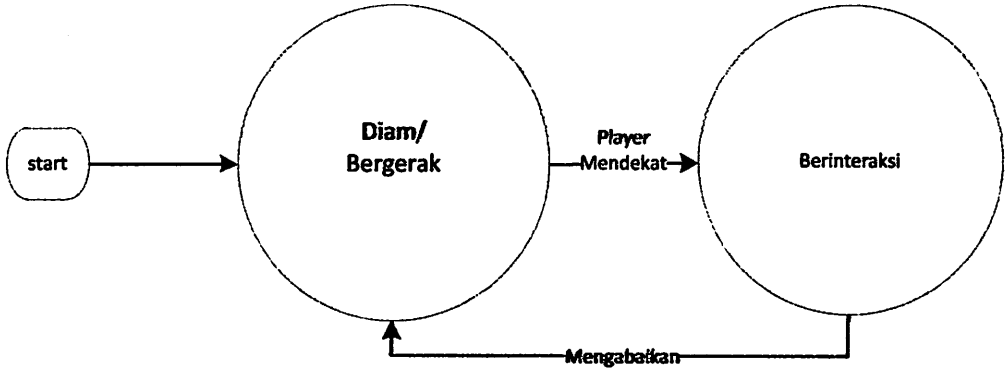
NPC akan menunggu *player* melakukan tugasnya. Ketika selesai *NPC quest* akan memberikan hadiah dan petunjuk untuk melanjutkan permainan. *Diagram state NPC Quest* bisa di lihat pada gambar 3.4 dibawah ini :



Gambar 3.4 *Diagram State NPC Quest*

3. NPC Pendukung

NPC pendukung adalah suatu *NPC* yang biasanya hadir dalam permainan sebagai *NPC* yang akan sering berinteraksi dengan *player*. Perancangannya pun cukup sederhana seperti gambar 3.5 dibawah ini :

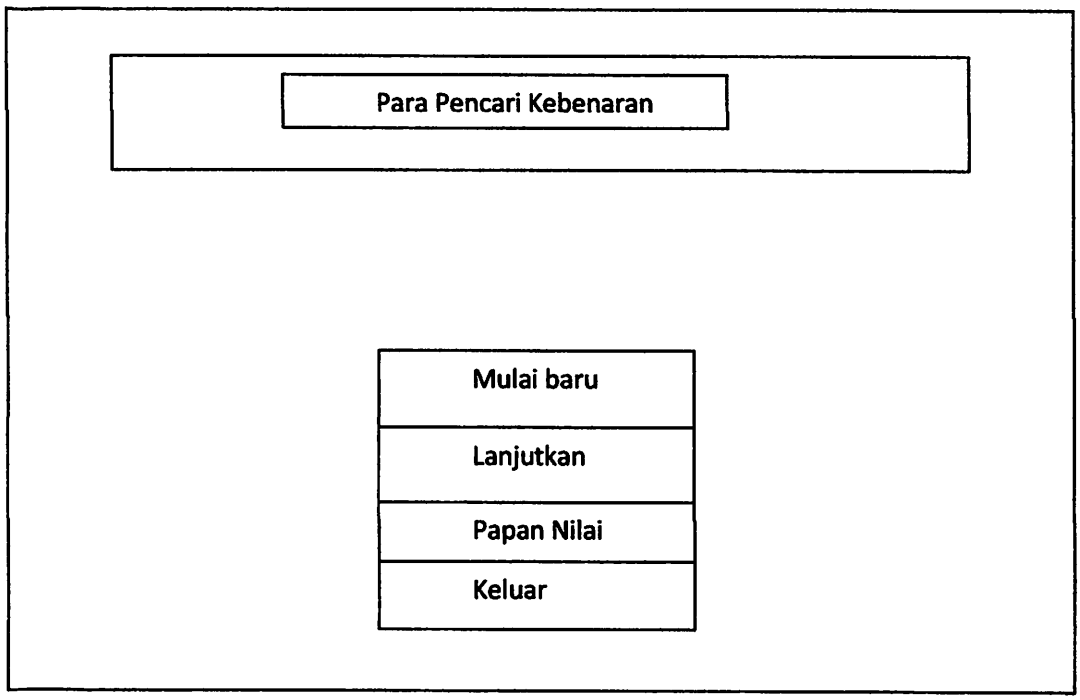


Gambar 3.5 *Diagram State NPC* Pendukung

3.4 Perancangan *Layout*

Perancangan *layout* bertujuan untuk memberikan gambaran tentang aplikasi *game* yang akan di bangun.

3.4.1 Perancangan *Layout* Menu Utama



Gambar 3.6 Rancangan *Layout Title* Menu

Pada perancangan tampilan *game RPG* **"Para Pencari Kebenaran"** yang ditunjukkan dalam gambar 3.6 dapat dijelaskan sebagai berikut :

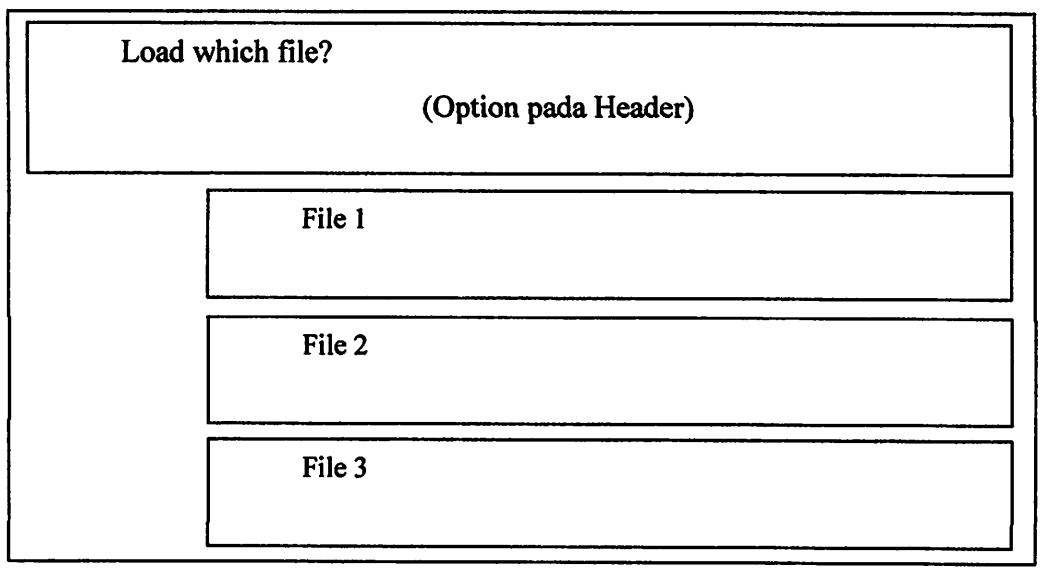
1. Kolom paling atas adalah nama dari sebuah *game* dengan judul **"Para Pencari Kebenaran".**
2. Menu *Mulai Baru* adalah pilihan dari menu yang pertama kali dipilih oleh *player* untuk memainkan *game* tersebut.
3. Menu *Lanjutkan* adalah pilihan dari menu yang berisi berkas data *game* yang sudah dimainkan sebelumnya oleh *player* yang telah disimpan. Jika *player*

tidak menyimpan permainan, maka tidak akan ada berkas dalam *file Lanjutkan*, dan menu ini tidak akan dapat diakses.

4. Menu *Papan Nilai* adalah pilihan dari menu yang menampilkan *nilai* dari permainan yang telah dimainkan. *Nilai* akan masuk jika *player* berhasil menyelesaikan permainan.
5. Menu *Keluar* adalah pilihan dari menu yang berfungsi untuk menutup jalannya aplikasi.

3.4.2 Perancangan Menu Lanjutkan

Menu *Lanjutkan* berisi tampilan dimana *berkas/data* simpanan *game* yang telah dimainkan *player* berada dan dapat diakses kembali untuk dimainkan. Perancangan tampilan menu *Lanjutkan* ditunjukkan pada gambar 3.7.



Gambar 3.7 *Layout* Menu lanjutkan

Perancangan *Layout* menu Lanjutkan yang ditunjukkan pada gambar 3.7 dapat dijelaskan sebagai berikut:

1. Jika *player* menyimpan data yang sudah dimainkan sebelumnya, maka *player* akan diberi pilihan untuk menyimpan data tersebut pada *file* yang sudah tersedia.

2. Data yang sudah tersimpan, akan ditaruh didalam *file*. Jika *player* menginginkan datanya kembali untuk dimainkan, *player* tinggal mengakses menu *Lanjutkan* dan memilih *file* tempat *player* menyimpan data tersebut.
3. Jumlah maksimal *file* yang dapat dipakai adalah *16 file*. Maksud dari hal ini adalah *player* dapat menyimpan data sebanyak 16 kali data simpanan permainan yang berbeda.
4. *Player* juga diperbolehkan menumpuk *file* baru kedalam *file* lama yang sebelumnya pernah dimainkan.

3.4.3 Perancangan Menu Papan Nilai

Menu *Papan Nilai* berisi tampilan dimana *nilai* yang didapat dari permainan setelah mengalahkan bos musuh ditampilkan. Perancangan tampilan menu *Lanjutkan* ditunjukkan pada gambar 3.8.

Papan Nilai Tertinggi		
No.	Nama	Nilai
1.		
2.		
3.		
4.		
Untuk menghapus nilai tekan tombol SHIFT|| Tekan X untuk keluar		

Gambar 3.8 Perancangan *Layout* Menu Papan Nilai.

Perancangan *Layout* menu *Papan Nilai* yang ditunjukkan pada gambar 3.7 dapat dijelaskan sebagai berikut:

1. Menu *Papan Nilai* menampilkan nilai (*score*) yang sudah dimainkan oleh *player* setelah mengalahkan tiap-tiap bos musuh sebelumnya.
2. Jika berhasil mengalahkan bos musuh, nilai akan keluar dan *player* diminta menyebutkan nama.
3. Hasil nilai akan ditampilkan pada menu *Papan Nilai*.
4. *Player* juga diperbolehkan untuk menghapus nilai (*score*) dengan cara menekan tombol *SHIFT* pada *keyboard* komputer.

3.4.4. Perancangan Desain *Sprite* Karakter

Sprite adalah representasi grafik 2 Dimensi dari karakter yang nantinya akan dimainkan oleh *player*. *Sprite* bisa disebut sebagai objek. Berikut adalah desain *sprite* dari karakter utama :

Gambar 3.9

Sprite Aditya

Selain Karakter utama didalam *game*, karakter pendamping yang selalu membimbing *player* juga sangat dibutuhkan. Oleh karena itu dibuatlah karakter pendamping. Berikut adalah desain *sprite* dari karakter pendamping :

Gambar 3.10

Sprite Gori

Gambar 3.11

Sprite Diman

Gambar 3.12

Sprite Memey

Gambar 3.13

Sprite Duan

BAB IV

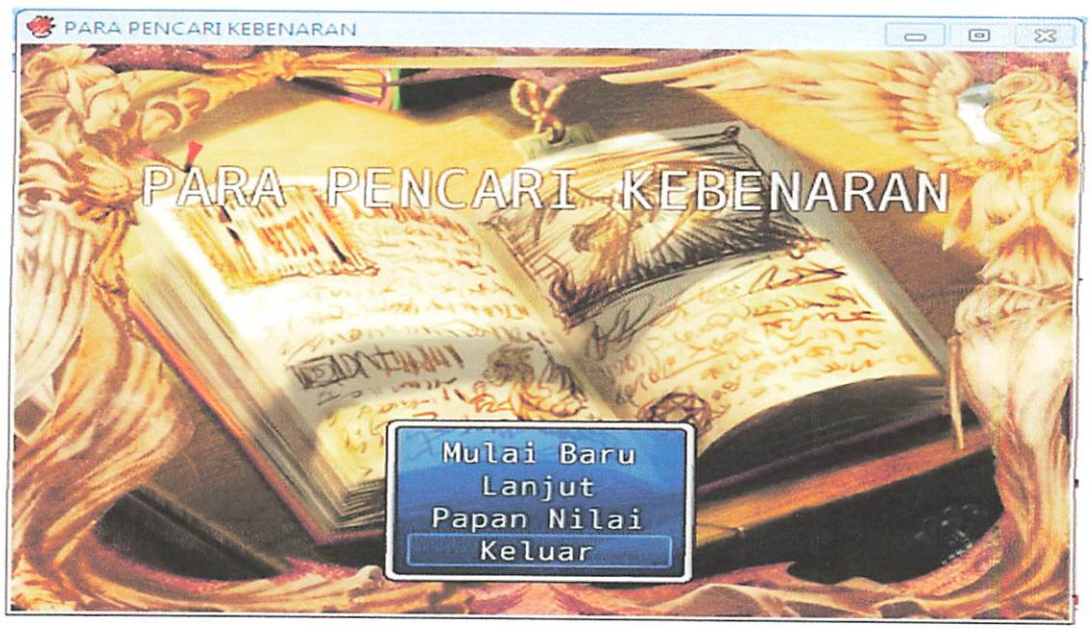
IMPLEMENTASI DAN PENGUJIAN

4.1 Implementasi

Impelementasi adalah tahapan dalam membangun dan menjalankan *game RPG* **"Para Pencari Kebenaran"** dengan menerapakan *Finite State Machine* sebagai kecerdasan buatan pada *NPC*.

4.1.1 Implementasi Menu Utama

Pada tampilan menu utama terdapat 4 menu utama yaitu Mulai Baru,Lanjut, Papan Nilai, dan Keluar. Untuk menu mulai baru akan membawa *player* ke prolog *game*. Menu lanjutkan adalah menu untuk melanjutkan dari *file* yang sudah tersimpan ketika kita bermain sebelumnya. Menu Papan Nilai ada menu untuk mengetahui nilai tertinggi yang pernah di dapatkan *player*. Sedangkan Menu keluar adalah menu untuk menutup program. Tampilan menu utama dapat dilihat pada gambar 4.1.



Gambar 4.1 Tampilan Menu Utama

4.1.2 Impelementasi Tampilan Menu lanjutkan

pada menu lanjutan ini akan di tampilan *file-file* Menu Lanjutkan adalah pilihan menu yang dapat dipilih oleh *player* jika *player* ingin memainkan kembali *game* yang sudah dimainkan sebelumnya, dan data simpanan *game* sudah tersimpan didalam berkas simpanan. Tampilan Menu Lanjutkan ditunjukkan pada gambar 4.2.

Gambar 4.2 Tampilan Menu Lanjut

4.1.3 Impelementasi Menu Papan Nilai

Menu Papan Nilai adalah menu yang dapat diakses oleh *player* untuk melihat perolehan *score* (nilai) tertinggi yang pernah di peroleh oleh *player*. Didalam menu Papan Nilai, selain *player* dapat melihat perolehan *score* (nilai) yang didapat, *player*

juga dapat me-*reset* nilai yang didapat. Cara mereset *score* (nilai) adalah dengan menekan tombol *SHIFT* pada keyboard. Fungsi *reset* nilai adalah untuk mengosongkan kembali *score* (nilai) yang didapat menjadi 0 atau tidak terisi. Tampilan Papan Nilai ditunjukkan pada gambar 4.3.

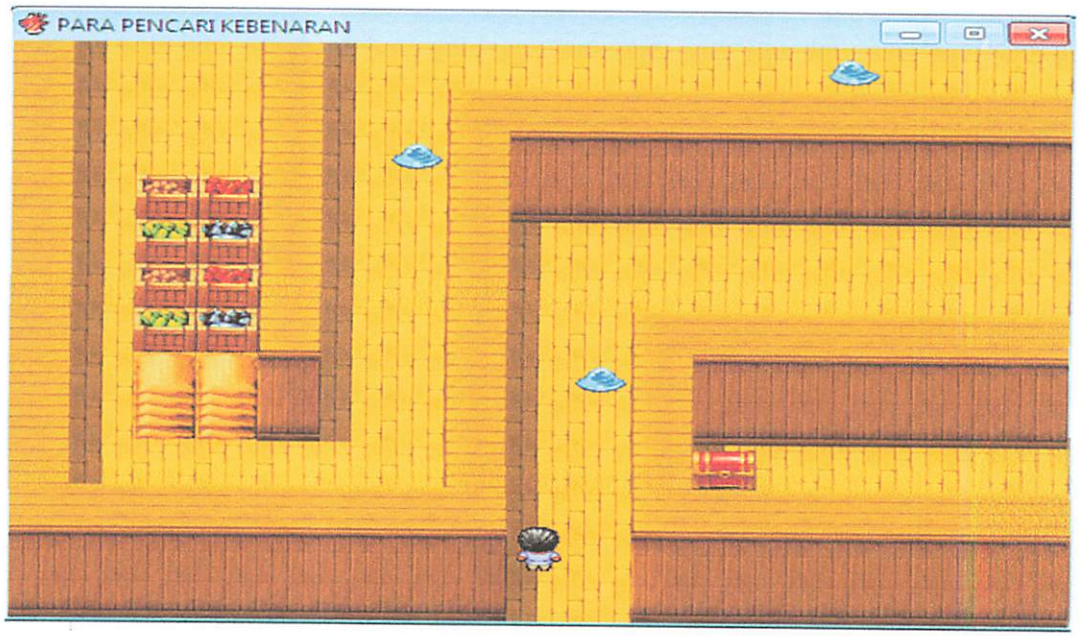
Gambar 4.3 Menu Papan Nilai Tertinggi

4.1.4 Implementasi Metode *Finite State Machine* Pada *NPC*

Metode *Finite State Machine* digunakan untuk pemetaan perilaku *NPC* yang terdapat pada *game*. Metode *Finite State Machine* akan di terapkan pada *NPC* musuh, *NPC quest*, *NPC* pendukung. Implementasi yang sudah di lakukan pada *Game* ini adalah :

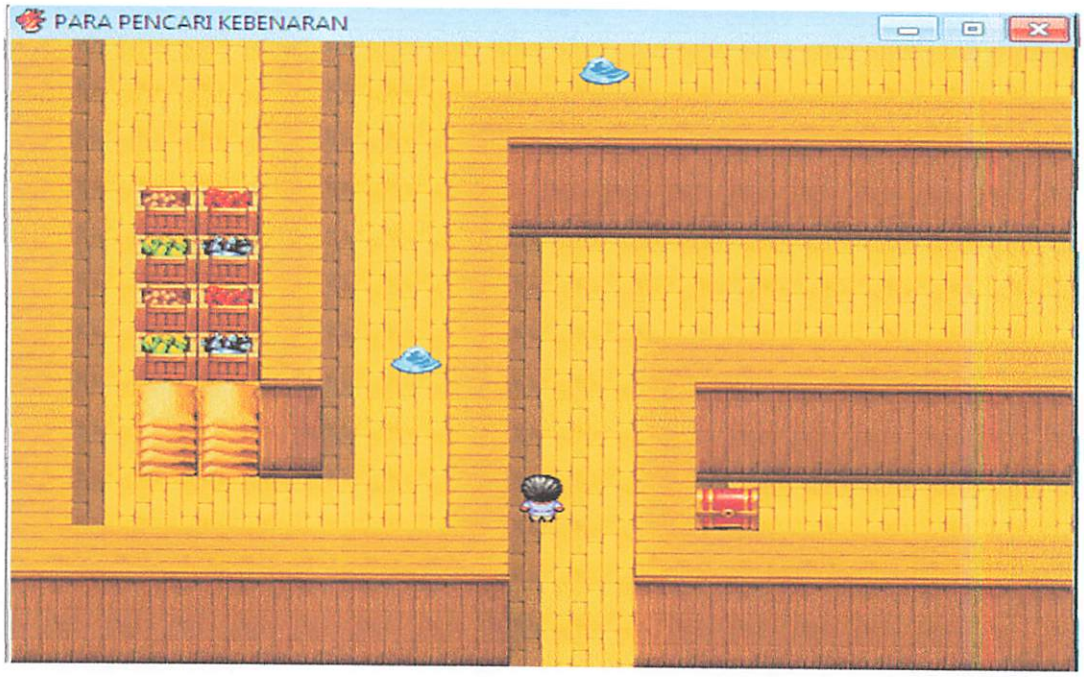
1. Implementasi pada *NPC* musuh.

Finite state machine yang di terapkan pada *NPC* musuh terdapat beberapa *state*. *State* awal *patrol*, *state* kedua bertarung, *state* ketiga adalah *dead*. Pada *state* patrol *NPC* musuh akan melakukan patrol di daerah sekelilingnya, ketika *player* mendekat dan menyentuh *NPC* maka player dan *NPC* akan masuk ke mode battle, ketika player menang maka *NPC* akan masuk ke mode *dead*.



Gambar 4.4 *State NPC* musuh *patrol*

Gambar 4.5 *State NPC* musuh *battle mode*



Gambar 4.6 *State NPC* musuh *dead*

2. Implementasi pada *NPC Quest*

Pada *NPC quest* terdapat 3 *state, state* pertama adalah *state* bergerak atau diam, *state* kedua adalah memberikan *quest*, *state* ketiga adalah melakukan pengecekan *quest*. Pada *state* pertama *NPC* akan berada pada posisinya , ketika *player* mendekat, *NPC* akan memberikan *Quest*, ketika *player* mengerjakan *quest* *NPC* akan kembali ke *state* awalnya. Ketika *Player* selesai mengerjakan *quest* maka *NPC* akan Melakukan pengecekan. *State* pada *NPC Quest* bisa dilihat pada gambar 4.7, gambar 4.8 & gambar 4.9.

Gambar 4.7 *State* awal *NPC quest*

Gambar 4.8 ***State Checking NPC quest***

Gambar 4.9 *State NPC quest*

3. Implementasi pada *NPC* Pendukung

NPC pendukung adalah *NPC* yang bertugas untuk memberikan informasi dan berinteraksi dengan pemain sehingga *game* terlihat lebih hidup. *State NPC* pendukung bisa dilihat pada gambar 4.10.

Gambar 4.10 *State NPC* Pendukung

4.2 Pengujian *System*

Pengujian *system game* merupakan proses menampilkan *system game* dengan maksud untuk menemukan adanya kesalahan atau tidak pada *system*, sebelum *system* tersebut diberikan kepada *user / player*.

Tabel 4.1 Pengujian *System Game*

No	Menu	Windows Xp Sp 2	Windows Xp Sp 3	Windows 7 (32-bit/64bit)	Windows 8 (32-bit/64bit)
1	Menu main baru	Ya	Ya	Ya	Ya
2	Menu Lanjut	Ya	Ya	Ya	Ya
3	Papan Nilai	Ya	Ya	Ya	Ya

4	Menu Keluar	Ya	Ya	Ya	Ya
5	Prolog Game	Ya	Ya	Ya	Ya
6	Simpan Data	Ya	Ya	Ya	Ya
7	Battle Sistem	Ya	Ya	Ya	Ya
8	Time	Ya	Ya	Ya	Ya
Daftar Tingkat Keberhasilan (%)		100 %	100 %	100 %	100 %

Dari **Tabel 4.1** di atas dapat dilihat bahwa daftar tingkat keberhasilan dalam proses pengujian *system game* adalah 100 % (Seratus Persen) ketika di ujikan pada sistem operasi Windows Xp Sp 2, Windows Xp Sp 3, Windows 7 (32-bit/64bit) & Windows 8 (32-bit/64bit). Daftar keterangan untuk **Tabel 4.1** Pengujian *System Game* adalah sebagai berikut :

1. Ya = Sukses.
2. Tidak = Gagal.

Berdasarkan hasil pengujian dari **Tabel 4.1** dapat dijelaskan bahwa *Game* berdasarkan jenis sistem operasinya, *game RPG* **"Para Pencari Kebenaran"** dapat berjalan baik di beberapa sistem operasi computer.

Tabel 4.2 Pengujian *FSM* Musuh

No	State	Hasil Pengujian
1	Patrol	Sukses
2	Battle	Sukses
3	Dead	Sukses

Dari **Tabel 4.2** dapat dilihat bawa ketika dilakukan pengujian terhadap *NPC* musuh semua *state* berjalan dengan baik.

Tabel 4.3 Pengujian *FSM Quest*

No	State	Hasil Pengujian
1	Bergerak diam/random	Sukses
2	Memberikan Quest	Sukses
3	Checking Quest	Sukses

Dari **Tabel 4.3** dapat dilihat bahwa *state-state* yang di berikan pada *NPC quest* berjalan dengan baik.

Tabel 4.4 Pengujian *FSM* pendukung

No	State	Hasil Pengujian
1	Move random /diam	Sukses
2	Interaksi	Sukses

Pada **Tabel 4.4** dapat disimpulkan bahwa *state* yang terdapat pada *NPC* pendukung ini berjalan dengan baik.

Selain pengujian terhadap sistem, pengujian juga dilakukan terhadap responden dengan cara menyebarkan kuisioner. Hasil tersebut dapat dilihat pada **Tabel 4.5.**

Tabel 4.5 Pengujian terhadap 10 responden

Pertanyaan	B	C	K
Apakah game mudah di mainkan?	6	4	-
Bagaimana tampilan battle di game ?	6	4	-
Apakah anda puas dengan tampilan game ?	7	3	-
Apakah anda puas dengan game ?	8	2	-

Daftar keterangan untuk **Tabel 4.5** Pengujian terhadap 10 responden adalah sebagai berikut :

1. B = Baik.
2. C = Cukup.
3. K = Kurang.

Berdasarkan pada **Tabel 4.5** maka kesimpulan yang dapat diambil dari 10 orang responden adalah sebagai berikut:

1. Tingkat kemudahan dalam *game*.
   a. B : 6 responden 60 %
   b. C : 4 responden 40 %
   c. K : 0 responden 0 %
2. Tampilan pertarungan dalam *game*.
   a. B : 6 responden 60 %
   b. C : 4 responden 40 %
   c. K : 0 responden 0 %
3. Tampilan *game*.
   a. B : 7 responden 70 %
   b. C : 3 responden 30 %
   c. K : 0 responden 0 %
4. Kepuasan terhadap *game* keseluruhan.
   a. B : 8 responden 80 %
   b. C : 2 responden 20 %
   c. K : 0 responden 0 %

Dari daftar data kuisioner tentang pengujian pada 10 responden diatas dapat disimpulkan bahwa :

1. 60% dari jumlah responden berpendapat bahwa kemudahan dalam *game* ini baik, dan 40% responden berpendapat cukup.

2. 60% dari jumlah responden berpendapat bahwa *battle* mode di *game* ini baik, dan 40% responden berpendapat cukup.
3. 70% dari jumlah responden berpendapat bahwa tampilan pada menu *game* ini baik, dan 30% respoden berpendapat cukup.
4. 80% dari jumlah responden berpendapat kepuasan dalam *game* ini baik, dan 20% berpendapat cukup.

4.3 Pengujian Resolusi *Screen Game*

Pengujian resolusi *screen game* pada tingkat resolusi komputer diperlukan untuk membuktikan apakah *game* dapat berjalan atau tidak pada *screen* resolusi tinggi maupun rendah. Hasil pengujian terhadap resolusi *screen game* ditunjukkan pada **Tabel 4.6.**

Tabel 4.6*Tabel Pengujian Game pada Screen Resolusi yang digunakan.*

Resolusi Sistem Operasi	Resolusi 800 x 600	Resolusi 1024 x 728	Resolusi 1200 x 768
Windows XP	Berjalan	Berjalan	Berjalan Baik
Windows 7	Berjalan	Berjalan	Berjalan Baik
Windows 8	Berjalan	Berjalan	Berjalan Baik
Mac OS	Berjalan	Berjalan	Berjalan Baik

Pada **Tabel 4.6** ditunjukkan hasil pengujian *game* dari berbagai ukuran *screen resolution*. Pada *screen* resolusi ukuran 800 x 600 dan 1024 x 728 *game* dapat berjalan, namun ukuran piksel dari *game* buram dan tidak jelas.

Sedangkan pada ukuran normal screen resolution 1200 x 768 game dapat berjalan dengan baik. Game tidak pecah dan tampilan tidak buram.

BAB V

PENUTUP

5.1 Kesimpulan

Dari hasil pengujian *game RPG* **"Para Pencari Kebenaran"** maka dapat disimpulkan bahwa :

1. *Game* ini dapat di terima dengan baik oleh *user*, Sebagai salah satu media hiburan alternatif.
2. *Game* ini dapat berkerja dengan baik dengan komputer atau *laptop* mengunakan sistem operasi yang berbeda yaitu *windows xp, windows 7, windows 8.*
3. Dengan Metode *Finite State Machine* dapat disimpulkan bahwa sebagian besar *user* merasa puas pada *game RPG (Role Playing Game)* **"Para Pencari Kebenaran"** secara keseluruhannya. Hal tersebut dibuktikan dengan hasil pengujian pada sebagian besar responden.
4. Ukuran *file game RPG* **"Para Pencari Kebenaran"** ini adalah 40 Mb.

5.2 Saran

Dalam pembuatan *game* **"Para Pencari Kebenaran"** ini tentu masih jauh dari sempurna, karena masih banyaknya kekurangan - kekurangan yang perlu untuk ditambahkan antara lain :

1. *Game* **"Para Pencari Kebenaran"** ini masih perlu ditambahkan sistem petanya.
2. *Game* **"Para Pencari Kebenaran"** ini dapat dikembangkan dengan cara memperbaharui *battle system* sehingga pertarungan bisa lebih animatif lagi.
3. Dapat ditambahkan map baru pada *game*.
4. Dapat menambahkan **"Bos Baru"**.
5. Alur cerita dapat di kembangkan lagi.

DAFTAR PUSTAKA

Febrian, Jack. 2004. *Pengetahuan Komputer dan Teknologi Informasi.* Bandung: Penerbit INFORMATIKA.

Alfian, Theolius. 2012. *24 Jam Pintar Desain RPG Maker.*Jakarta: Penerbit GRAMEDIA.

Yami Engine Ace. 2012. *Battle System Symphony.* Jakarta: Penerbit CV ANDY OFFSET.

Permana, Rizal. 2012. *Dasar-dasar Pemrograman Bahasa Ruby dalam RPG Maker.* Semarang: Penerbit GRAMEDIA.

Membuat Game RPG dengan RPG Maker. 2013. Jakarta: Penerbit CV ANDY OFFSET

LAMPIRAN

LAMPIRAN *SCRIPT* PAPAN NILAI PADA *GAME RPG* "PARA PENCARI KEBENARAN" DENGAN METODE *FINITE STATE MACHINE*

```
#
==========================================================================
# Konfigurasi :
#
==========================================================================
module Theo
module HScore

# ----------------------------------------------------------------
# *) Setting umum
# ----------------------------------------------------------------
 Variable_ID  = 1
# Tentukan variable ID yang akan jadi acuan untuk penilaian skor

 Reset_Key    = :SHIFT
# Key buat ngereset score kalo kamu lagi di menu highscore

# ----------------------------------------------------------------
# *) Kosakata umum
# ----------------------------------------------------------------
 ScoreTitle   = "Papan Nilai Tertinggi"  # Title untuk menu highscore
 ScoreTable   = "No.  Nama"        # Text untuk tabel kiri
 ScoreVocab   = "Nilai"            # Sebutan untuk skor
 ScoreCommand = "Papan Nilai"      # Command yang ada di title screen
 YourScore    = "Nilai anda :"     # Text saat kamu memasukkan nama
```

```
# ------------------------------------------------------------
# *) Untuk perihal konfirmasi ngereset skor
# ------------------------------------------------------------
  ScoreFooter  = "Hapus tekan SHIFT || Keluar tekan X"
# Kata bantuan untuk ngereset
  Confirmation  = "Anda yakin ingin me-reset Nilai Teringgi?"
# Konfirmasi saat player mo ngereset skor

  YesCommand   = "Ya"   # Kata untuk iya
  NoCommand    = "Tidak"   # Kata untuk tidak

 end
end
#
================================================================
================
# Akhir dari konfigurasi
#
================================================================
================
class << DataManager

 alias theo_hscore_load_db load_database
 def load_database
  theo_hscore_load_db
  load_scoreboard
 end
```

```
def load_scoreboard
  open_scoreboard
  $scoreboard = load_data(score_path)
end

def open_scoreboard(reset = false)
  if !FileTest.exist?(score_path) || reset
    score = []
    File.open(score_path, "w") do |file|
      Marshal.dump(score, file)
    end
  end
end

def reset_score
  open_scoreboard(true)
  $scoreboard = load_data(score_path)
end

def update_score
  save_data($scoreboard, score_path)
end

def score_path
  return "Data/Score.rvdata2"
end

def score
```

```
    $game_variables[Theo::HScore::Variable_ID]
  end

end

class ScoreBoard
  attr_accessor :name
  attr_accessor :score

  def initialize(name, score)
    @name = name
    @score = score
  end
end

class Window_HScoreTitle < Window_Base

  def initialize
    super(0,0,Graphics.width,fitting_height(2))
    draw_title
  end

  def draw_title
    draw_text(contents.rect, Theo::HScore::ScoreTitle,1)
  end

end
```

```
class Window_HScoreBoard < Window_Base

 def initialize(y)
  super(0,y,Graphics.width, Graphics.height - y - line_height*2)
  refresh
 end

 def refresh
  contents.clear
  draw_contents
 end

 def draw_contents
  scorelist = $scoreboard.sort {|a,b| b.score <=> a.score }
  draw_header
  ((contents.height / line_height) - 1).times do |line|
   score = scorelist[line]
   line += 1
   rect = Rect.new(0,line*line_height,contents.width,line_height)
   if score
    txt = sprintf("%d.   %s",line, score.name)
    draw_text(rect,txt)
    draw_text(rect, score.score,2)
   else
    txt = sprintf("%d.   ",line)
    draw_text(rect,txt)
   end
  end
```

```
  end

  def draw_header
    rect = Rect.new(0,0,contents.width, line_height)
    change_color(system_color)
    draw_text(rect, Theo::HScore::ScoreTable)
    draw_text(rect, Theo::HScore::ScoreVocab, 2)
    change_color(normal_color)
    color = Color.new(255,255,255,128)
    contents.fill_rect(0,23,contents.width,2,color)
  end

end

class Window_HScoreFooter < Window_Base

  def initialize
    super(0,0,Graphics.width, fitting_height(1))
    self.y = Graphics.height - height
    draw_text(contents.rect, Theo::HScore::ScoreFooter,1)
  end

end

class Window_HScoreCommand < Window_Command

  def initialize
    super(0,0)
```

```
    self.openness = 0
    select(0)
    deactivate
  end

  def make_command_list
    add_command(Theo::HScore::YesCommand, :yes)
    add_command(Theo::HScore::NoCommand, :cancel)
  end

end

class Window_HScoreConfirm < Window_Base

  def initialize
    super(0,0,window_width,fitting_height(1))
    self.openness = 0
    update_placement
    draw_text(contents.rect, Theo::HScore::Confirmation, 1)
  end

  def update_placement
    self.x = (Graphics.width - width) / 2
    self.y = ((Graphics.height - height) / 2) - 25
  end

  def window_width
    return 400
```

```
  end

end

class Scene_ScoreBoard < Scene_Base

  def start
    super
    @title = Window_HScoreTitle.new
    @score = Window_HScoreBoard.new(@title.height)
    @footer = Window_HScoreFooter.new
    @confirm = Window_HScoreConfirm.new
    create_confirm
  end

  def create_confirm
    @command = Window_HScoreCommand.new
    @command.x = (Graphics.width - @command.width) / 2
    @command.y = @confirm.y + @confirm.height
    @command.set_handler(:yes, method(:reset_score))
    @command.set_handler(:cancel, method(:close_confirm))
  end

  def update
    super
    if Input.trigger?(:B) && !@command.active
      Input.update
      Sound.play_cancel
```

```
      return_scene
    elsif Input.trigger?(Theo::HScore::Reset_Key) && !@command_active
      open_confirmation
    end
  end

  def open_confirmation
    @command.activate
    @command.open
    @confirm.open
  end

  def reset_score
    DataManager.reset_score
    @score.refresh
    close_confirm
  end

  def close_confirm
    @command.deactivate
    @command.close
    @confirm.close
  end

end

class Window_ScoreName < Window_NameEdit
  class PseudoChar
```

```
    attr_accessor :name

    def initialize
      @name = ""
    end
  end
  attr_reader :actor

  def initialize(max_char)
    super(PseudoChar.new, max_char)
  end

  def draw_actor_face(actor, x, y)
    rect = Rect.new(0,0,100,line_height)
    change_color(system_color)
    draw_text(rect, Theo::HScore::YourScore)
    rect.y += line_height
    change_color(normal_color)
    draw_text(rect, DataManager.score)
  end
end

class Scene_ScoreName < Scene_MenuBase

  def prepare(max_char)
    @max_char = max_char
  end
```

```
    def start
      super
      @edit_window = Window_ScoreName.new(@max_char)
      @input_window = Window_NameInput.new(@edit_window)
      @input_window.set_handler(:ok, method(:on_input_ok))
    end

    def on_input_ok
      $scoreboard.push(ScoreBoard.new(@edit_window.name,
DataManager.score))
      DataManager.update_score
      return_scene
    end

    def return_scene
      SceneManager.goto(Scene_ScoreBoard)
    end

  end

  class Window_TitleCommand < Window_Command

    alias theo_hscore_make_command make_command_list
    def make_command_list
      theo_hscore_make_command
      hash = {
        :name=> Theo::HScore::ScoreCommand,
        :symbol=> :hscore,
```

```
      :enabled=> true,
      :ext=> nil
    }
    @list.insert(2, hash)
  end

end

class Scene_Title < Scene_Base

  alias theo_hscore_start start
  def start
    theo_hscore_start
    @command_window.set_handler(:hscore,method(:goto_hscore))          if
@command_window
  end

  def goto_hscore
    SceneManager.goto_scoreboard
  end

end

class << SceneManager

  def goto_scoreboard
    call(Scene_ScoreBoard)
  end
```

```
  def input_scorename(max_char = 8)
    call(Scene_ScoreName)
    scene.prepare(max_char)
  end

end
```

Kuisioner

Nama : Muhammad Yasin Fansuri ST

Pekerjaan : Wira Swasta.

Pertanyaan	B	C	K
Apakah game ini mudah untuk dimainkan?	✓		
Bagaimana tampilan pertarungan di game ini?		✓	
Apakah anda puas dengan tampilan yang ada di game ini?	✓		
Apakah anda puas dengan game ini?	✓		

Saran : Kurang memperhatikan Script dalam bahasa Pemrograman Komputer.

TTD

M. Yasin f

M. Yasin Fansuri ST.

Kuisioner

Nama : Anto

Pekerjaan : Mahasiswa

Pertanyaan	B	C	K
Apakah game ini mudah untuk dimainkan?		✓	
Bagaimana tampilan pertarungan di game ini?	✓		
Apakah anda puas dengan tampilan yang ada di game ini?	✓		
Apakah anda puas dengan game ini?	✓		

Saran :

TTD

Kuisioner

Nama : Aldo Y. N.

Pekerjaan : Mahasiswa

Pertanyaan	B	C	K
Apakah game ini mudah untuk dimainkan?	✓		
Bagaimana tampilan pertarungan di game ini?		✓	
Apakah anda puas dengan tampilan yang ada di game ini?		✓	
Apakah anda puas dengan game ini?	✓		

Saran :

TTD

Kuisioner

Nama : Gandha D

Pekerjaan : Mahasiswa

Pertanyaan	B	C	K
Apakah game ini mudah untuk dimainkan?	✓		
Bagaimana tampilan pertarungan di game ini?		✓	
Apakah anda puas dengan tampilan yang ada di game ini?	✓		
Apakah anda puas dengan game ini?	✓		

Saran :

TTD

Kuisioner

Nama : Muhamad Adam

Pekerjaan : wira usaha

Pertanyaan	B	C	K
Apakah game ini mudah untuk dimainkan?		✓	
Bagaimana tampilan pertarungan di game ini?	✓		
Apakah anda puas dengan tampilan yang ada di game ini?	✓		
Apakah anda puas dengan game ini?	✓		

Saran :

TTD

Kuisioner

Nama : Bomal Adhyan

Pekerjaan : Mahasiswa

Pertanyaan	B	C	K
Apakah game ini mudah untuk dimainkan?	✓		
Bagaimana tampilan pertarungan di game ini?	✓		
Apakah anda puas dengan tampilan yang ada di game ini?		✓	
Apakah anda puas dengan game ini?	✓		

Saran :

TTD

Kuisioner

Nama : Sigit Pangesto

Pekerjaan : Mahasiswa

Pertanyaan	B	C	K
Apakah game ini mudah untuk dimainkan?	✓		
Bagaimana tampilan pertarungan di game ini?	✓		
Apakah anda puas dengan tampilan yang ada di game ini?	✓		
Apakah anda puas dengan game ini?	✓		

Saran :

TTD

Kuisioner

Nama : DWI PRASETIYO

Pekerjaan : MAHASISWA

Pertanyaan	B	C	K
Apakah game ini mudah untuk dimainkan?	✓		
Bagaimana tampilan pertarungan di game ini?	✓		
Apakah anda puas dengan tampilan yang ada di game ini?		✓	
Apakah anda puas dengan game ini?		✓	

Saran :

TTD

Kuisioner

Nama : Bagu Kurniawan

Pekerjaan : Mahasiswa

Pertanyaan	B	C	K
Apakah game ini mudah untuk dimainkan?		✓	
Bagaimana tampilan pertarungan di game ini?	✓		
Apakah anda puas dengan tampilan yang ada di game ini?	✓		
Apakah anda puas dengan game ini?	✓		

Saran : Perbaikilah tampilan menu di dalam game

TTD

Kuisioner

Nama : Made Shri Wijaya Shnu

Pekerjaan : Mahasiswa

Pertanyaan	B	C	K
Apakah game ini mudah untuk dimainkan?		✓	
Bagaimana tampilan pertarungan di game ini?		✓	
Apakah anda puas dengan tampilan yang ada di game ini?	✓		
Apakah anda puas dengan game ini?		✓	

Saran : Keep cool

TTD

NSTITUT TEKNOLOGI NASIONAL
Jln. Bendungan Sigura-gura No. 2
Jln. Raya Karanglo Km2
MALANG

PERNYATAAN KESEDIAAN DALAM PEMBIMBINGAN SKRIPSI

Sesuai permohonan dari mahasiswa/i :

Nama : MUHAMMAD IMAM SETIYAWAN

Nim : 0918020

Program Studi : Teknik Informatika

Dengan ini menyatakan bersedia / ~~tidak bersedia~~ *) membimbing skripsi dari mahasiswa tersebut dengan judul :

Penerapan Metode Finite States Machine Pada Game RPG "Petualangan Si Pencari Kebenaran"

Demikian Surat Pernyataan ini kami buat agar dipergunakan seperlunya.

Malang, 19 - 6 - 2014

Hormat Kami,

Ir. Yusuf Ismail Nakhoda, MT
NIP.Y.1018800189

Catatan :
Setelah disetujui agar formulir ini diserahkan mahasiswa/i yg bersangkutan kepada Jurusan untuk diproses lebih lanjut
*) coret yang tidak perlu

Form S-3b

NSTITUT TEKNOLOGI NASIONAL
Jln. Bendungan Sigura-gura No. 2
Jln. Raya Karanglo Km2
M A L A N G

PERNYATAAN KESEDIAAN DALAM PEMBIMBINGAN SKRIPSI

Sesuai permohonan dari mahasiswa/i :

Nama : MUHAMMAD IMAM SETIYAWAN

Nim : 0918020

Program Studi : Teknik Informatika

Dengan ini menyatakan bersedia / tidak bersedia *) membimbing skripsi dari mahasiswa tersebut dengan judul :

Penerapan Metode Finite States Machine Pada Game RPG "Petualangan Si Pencari Kebenaran"

Demikian Surat Pernyataan ini kami buat agar dipergunakan seperlunya.

Malang, 19-6-2014

Hormat Kami,

Febriana S.W.S.Kom.M.Kom
NIP.P.1031000425

Catatan :
Setelah disetujui agar formulir ini diserahkan mahasiswa/i yang bersangkutan kepada Jurusan untuk diproses lebih lanjut
*) coret yang tidak perlu

Form S-3b

INSTITUT TEKNOLOGI NASIONAL
Jl. Bendungan Sigura-Gura No. 2
MALANG

Malang, 21 April 2014

Lampiran : 1(Satu) berkas

Perihal : Kesediaan sebagai Pembimbing Skripsi

Kepada : Yth. Bpk/Ibu **Ir. Yusuf Ismail Nakhoda, MT**
Dosen Pembina Prodi Teknik Informatika S-1
Institut Teknologi Nasional
MALANG

Yang bertanda tangan dibawah ini:

Nama : MUHAMMAD IMAM SETIYAWAN
Nim : 0918020
Prodi : Teknik Informatika S-1

Dengan ini mengajukan permohonan, kiranya Bapak/Ibu bersedia menjadi Dosen Pembimbing Utama / ~~Pendamping~~ *), untuk penyusunan Skripsi dengan judul (Proposal Terlampir) :

Penerapan Metode Finite States Machine Pada Game RPG "Petualangan Si Pencari Kebenaran"

Adapun tugas tersebut sebagai salah satu syarat untuk menempuh Ujian Akhir Sarjana Teknik. Demikian permohonan kami dan atas kesediaan Bapak/Ibu kami sampaikan terima kasih.

Prodi T. Informatika S-1
Ketua,

Joseph Dedy Irawan,ST., MT.
NIP : 197404152005021002

Hormat Kami,

MUHAMMAD IMAM SETIYAWAN

Form S-3a

INSTITUT TEKNOLOGI NASIONAL
Jl. Bendungan Sigura-Gura No. 2
MALANG

Malang, 21 April 2014

Lampiran : 1(Satu) berkas

Perihal : Kesediaan sebagai Pembimbing Skripsi

Kepada : Yth. Bpk/Ibu **Febriana S.W.S.Kom.M.Kom**
Dosen Pembina Prodi Teknik Informatika S-1
Institut Teknologi Nasional
MALANG

Yang bertanda tangan dibawah ini:

Nama : MUHAMMAD IMAM SETIYAWAN
Nim : 0918020
Prodi : Teknik Informatika S-1

Dengan ini mengajukan permohonan, kiranya Bapak/Ibu bersedia menjadi Dosen Pembimbing ~~Utama~~ / Pendamping *), untuk penyusunan Skripsi dengan judul (Proposal Terlampir) :

Penerapan Metode Finite States Machine Pada Game RPG "Petualangan Si Pencari Kebenaran"

Adapun tugas tersebut sebagai salah satu syarat untuk menempuh Ujian Akhir Sarjana Teknik. Demikian permohonan kami dan atas kesediaan Bapak/Ibu kami sampaikan terima kasih.

Prodi T. Informatika S-1
Ketua,

INSTITUT TEKNOLOGI NASIONAL MALANG FAKULTAS TEKNOLOGI INDUSTRI PROGRAM STUDI TEKNIK INFORMATIKA

Joseph Dedy Irawan,ST., MT.
NIP : 197404162005021002

Hormat Kami,

MUHAMMAD IMAM SETIYAWAN

Form S-3a