APLIKASI GAME RPG BERBASIS CONTROLLER PSP PADA SISTEM OPERASI ANDROID

TUGAS AKHIR

Diajukan Sebagai Salah Satu Syarat
Untuk Memperoleh Gelar Sarjana
Jurusan Teknik Informatika



Nama: Yogie Aditya Siswandi
No. Mahasiswa: 07523035

JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA
YOGYAKARTA
2011

LEMBAR PENGESAHAN PEMBIMBING

APLIKASI GAME RPG BERBASIS CONTROLLER PSP PADA SISTEM OPERASI ANDROID

TUGAS AKHIR

Oleh :

Nama: Yogie Aditya Siswandi

No. Mahasiswa: 07523035

Yogyakarta, Mei 2011

Pembimbing

Affan Mahtarami, S.Kom., M.Kom

LEMBAR PENGESAHAN PENGUJI

APLIKASI GAME RPG BERBASIS CONTROLLER PSP PADA SISTEM OPERASI ANDROID

TUGAS AKHIR

Oleh :

Nama: Yogie Aditya Siswandi

No. Mahasiswa: 07523035

Telah Dipertahankan di Depan Sidang Penguji Sebagai Salah Satu Syarat

Untuk Memperoleh Gelar Sarjana Jurusan Teknik Informatika

Fakultas Teknologi Industri Universitas Islam Indonesia

Yogyakarta, Mei 2011

Tim Penguji :

Affan Mahtarami, S.Kom., M.T.

Ketua

Beni Suranto, ST.

Anggota 1

Izzati Muhimmah, ST., MSc., PhD.

Anggota 2

Mengetahui,

Ketua Jurusan Teknik Informatika

Universitas Islam Indonesia

Yudi Prayudi, S.Si., M.Kom.

LEMBAR PERNYATAAN KEASLIAN
HASIL TUGAS AKHIR

Saya yang bertanda tangan dibawah ini,

Nama: Yogie Aditya Siswandi
No. Mahasiswa: 07523035

Menyatakan bahwa seluruh komponen dan isi dalam Laporan Tugas Akhir ini adalah hasil karya saya sendiri. Apabila dikemudian dari terbukti bahwa ada beberapa bagian dari karya ini adalah bukan hasil karya saya sendiri, maka saya siap menanggung resiko dan konsekuensi apapun.

Demikian pernyataan ini saya buat, semoga dapat dipergunakan sebagaimana mestinya.

Yogyakarta, Mei 2011

(Yogie Aditya Siswandi)

HALAMAN PERSEMBAHAN

Puji dan syukur kepada Allah SWT yang telah memberi ilmu, kelancaran, hikmah, inspirasi, dan pelajaran yang begitu luas kepada saya, tak henti-hentinya saya bersyukur dan memohon ampun serta berserah diri kepadaNya.

Shalawat dan salam kepada Nabi besar Muhammad SAW yang merupakan teladanku dan panutanku, yang menuntunku kepada iman dan takwa serta ajaranya memberiku cahaya dan hikmah yang tiada terhingga.

Tugas akhir ini kupersembahkan untuk semua yang aku cintai:

Kepada Ayahku Bp. Patri Siswandi yang telah memberikan bimbingan, ajaran dan doa serta dorongan semangat maupun material, beliau adalah seorang Ayah yang mencintai setulus hati. Kepada Ibuku Sulastrie yang telah memberikan bimbingan moral, akhlak dan etika dari saya kecil hingga sekarang, Beliau telah banyak memberikan saya teladan, kasih sayang, dan bimbingan tiada tara. Beliau adalah seorang Ibu yang menyayangi setulus hati.

Kepada adikku Aisha Astricia, ia adalah adik sekaligus sahabat terbaikku. Terimakasih atas cinta kasih, masukan, dan kasih sayangnya selama ini. Ia adalah teman curhatku yang mengasihi setulus hati. Terimakasih atas cinta kasih, dan ketulusannya selama ini, capailah cita-citamu setinggi mungkin. Ia adalah adikku yang sangat pintar.

Kepada kakakku Seifra (Aditya Yogasara), dan teman-teman lainnya kak Richan, kak Raen, kak Endah, kak Haw, kak Shino, dan mami Rui yang telah mendukung dan selalu memberi semangat selama pelaksanaan tugas akhir ini.

Kepada teman-teman terbaikku Gege, Septian Aditya, Satyo Agung W, M Latief Ch, Faisal Aditya S, Iwan Farid, Noppa, Aldila Aprilian Nasikin, Firda, Sistha Widita, Mentari Ambar, Bithana Paragustin yang selama ini sudah mengisi hidupku menjadi sangat berwarna, tertawa dan bersama kalian entah bermain bersama, belajar bersama adalah hal yang sangat berharga buatku. Tanpa kalian aku pasti sangatlah kesepian. Terimakasih untuk segalanya, aku berharap persahabatan kita tidak akan pernah rapuh untuk selamanya.

Untuk teman-teman seluruh Laboratorium Terpadu terutama Lab Komputasi dan Sistem Cerdas semua angkatan, kalian semua adalah motivasi dalam hidupku, kalianlah sumber semangat bagiku. Mengingat kita sudah menempuh kesenangan dan kesusahan sama-sama. Tertawa dan menangis bersama, namun kita tetap sabar dan terus bersatu. Terimakasih karena selama ini kita sudah berjuang bersama-sama hingga akhir, sukses buat kalian semua dan semoga kalian bisa mencapai cita-cita yang kalian harapkan seperti yang sering kita bicarakan. Semoga persahabatan kita tidak akan pernah punah sampai kapanpun.

Untuk teman-temanku KKN Unit SL-43 : Adityo Budi Prakoso (pak ketua), Novita Isnaini (mpok nenii), Akbar Handoko (mas cuci tangan), Permadi Budi Haryadi (mas budi), Sunday Refti Dewi (mbak sande), Maradhiean Syafdiputra (ian/Bernard), Bobby Setiawan (abe), Muh. Andi Aziz Nur A (andi), Retha (mbak prikiyuk), Sugeng Riyadi (sugayang). Terimakasih atas kebersamaannya selama ini dan momen KKN yang takkan terlupakan olehku seumur hidup.

Kepada teman-teman FTI UII angkatan 2005, 2006, 2007, 2008, 2009 dan 2010 yang tak mungkin saya sebutkan satu persatu disini, terimakasih atas kebersamaanya selama ini.

Kepada teman-teman band jepangku Bagea Kriska (Giguk), Faachan, Ditto, Ius, Yoga, Niko, Gilang, Asep, Anugrah, Arkham, Sohji, Arorororon, Hendy, Shinmi,

dan semua teman-teman jepang-jepangan, shimatta, wurz, dan seperjuangan. Terimakasih atas semua dukungannya selama ini. Semoga kita semua juga bisa menjadi band yang lebih baik lagi dan menciptakan seni musik yang bisa menghibur banyak orang.

Kepada teman-temanku baik teman SMA Muhammadiyah 3 maupun teman-temanku di perumahan Ambarketawang Indah Yogyakarta, terimakasih atas dukungannya dan do'anya selama ini.

MOTTO

Sesungguhnya sesudah kesulitan itu ada kemudahan.
(QS Alam nasyrah ayat 6)

Sesungguhnya manusia itu benar-benar berada dalam kerugian, kecuali orang-orang yang beriman dan mengerjakan amal saleh dan nasehat menasehati supaya mentaati kebenaran dan nasehat menasehati supaya menetapi kesabaran.
(QS Al Ashr ayat 2, 3)

Adapun orang yang memberikan dan bertakwa, dan membenarkan adanya pahala yang terbaik, maka kami kelak akan menyiapkan baginya jalan yang mudah.
(QS Al Lail ayat 5, 6, 7)

Bahagialah orang yang diberi Allah dengan kesibukan-kesibukan yang padat, karena keberadaan kesibukan itu membuat hari-hari kita menjadi penuh makna.

Hadapilah kehidupan dengan sabar, shalat, dan rendah hati, jangan takut jatuh atau tersandung, karena orang yang tak pernah jatuh, tak kan pernah bangkit.

Sayangilah kedua Orangtuamu, patuhi perintahnya, dan jalankan amanatnya, karena apapun yang diberikan Orangtuamu kepada anaknya adalah yang terbaik. Ia pernah mengalami muda dan kamu belum pernah mengalami tua, dan Ridho Allah adalah melalui ridho Orangtua.

Sesungguhnya semua ilmu pengetahuan yang bermanfaat itu menarik dan menyenangkan, namun semua itu bergantung bagaimana cara kita mempelajarinya.

Taatlah kepada Allah dan jalankanlah perintahnya dan jauhilah larangannya, sesungguhnya apa yang diperintahkan Allah adalah baik dan akan menuai baik,

dan sesungguhnya apa yang dilarang oleh Allah adalah sesuatu yang dapat merugikan kita dan akan menuai penyesalan.

KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَٰنِ الرَّحِيمِ

Assalamualaikum Wr. Wb.

Puji dan syukur penulis panjatkan kehadirat Allah SWT yang telah memberikan petunjuk, kelancaran dan kemudahan serta ilmu yang sangat tak ternilai harganya, shalawat serta salam kepada Nabi besar Muhammad SAW yang telah mengajarkan jalan yang benar dan cahaya yang terang benderang kepada ummatnya dan teladan yang menghantarkan pada kemuliaan dan kebahagiaan hidup didunia maupun diakhirat kelak.

Setelah melalui proses yang sangat panjang dan kerja keras dan dukungan dari berbagai pihak, maka penulis akhirnya dapat menyelesaikan tugas akhir yang berjudul Aplikasi Game RPG berbasis Controller PSP pada Sistem Operasi Android. Laporan tugas akhir ini adalah sebagai salah satu syarat untuk mendapatkan gelar sarjana S1 pada jurusan Teknik Informatika Fakultas Teknologi Industri Universitas Islam Indonesia.

Dalam pelaksanaan tugas akhir dan pembuatan laporan tugas akhir ini penulis mendapat dukungan dan bantuan dari berbagai pihak, penulis hendak mengucapkan terimakasih sebesar-besarnya kepada:

1. Bapak Gumbolo Hadi Susanto selaku Dekan Fakultas Teknologi Industri Universitas Islam Indonesia.
2. Bapak Yudi Prayudi selaku Ketua jurusan Teknik Informatika Fakultas Teknologi Industri Universitas Islam Indonesia.
3. Bapak Affan Mahtarami selaku Dosen pembimbing yang telah banyak membantu, memberi dukungan, arahan, dan bimbingan selama pelaksanaan tugas akhir.
4. Ayahku Bp. Patri Siswandi yang telah memberi semangat dan dukungan, Ibuku Ibu Sulastrie dan adikku Aisha Astricia yang telah memberi motivasi dan doa.

5. Kakakku Seifra (Aditya Yogasara), dan teman-teman lainnya kak Richan, kak Raen, kak Endah, kak Haw, kak Shino, dan mami Rui yang telah mendukung dan selalu memberi semangat selama pelaksanaan tugas akhir ini.
6. Teman-teman yang mendukung dan menemaniku selama pengerjaan Tugas Akhir, M Latif Ch, Septian Aditya, Satyo Agung W, Noppa, Iwan, Aldila Aprilian Nasikin, Firda, Mentari Ambar, Sistha Widita.
7. Teman-temanku semua baik teman-teman KKN unit 43, teman-teman sesama kuliah, teman-teman SMA Muhammadiyah 3, teman-teman perumahan di Yogyakarta, dan yang lainnya yang tak mungkin saya sebutkan satu persatu disini, terimakasih banyak atas dukungannya selama ini.

Dalam pelaksanaan dan pembuatan program serta laporan tugas akhir ini penulis menyadari bahwa masih banyak kekurangan-kekurangan baik yang disadari maupun yang tidak disadari, oleh karena itu penulis sangat mengharapkan kritik dan saran serta masukan dari para pembaca.

Semoga laporan tugas akhir ini memberikan manfaat bagi pembaca maupun bagi kepustakaan ilmu baik pada Jurusan Teknik Informatika Universitas Islam Indonesia maupun bagi dunia ilmu pengetahuan Teknologi Informasi di Indonesia.

Wassalamualaikum Wr. Wb.

Yogyakarta, Mei 2011

Penulis

SARI

Perkembangan *Game RPG* akhir-akhir ini sangatlah pesat, dari segi sistem permainan, sistem pertarungan, dan sistem *controller* untuk memainkannya. *Controller* yang sering sekali di gunakan adalah *controller* yang menyerupai *controller* pada *Playstation*, berawal dari sebuah *controller* yang terhubung oleh kabel hingga *controller* yang manjadi satu dengan perangkat utamanya seperti pada *Playstation Portable (PSP)*. Perkembangan *game* dilihat dari segi perangkat yang menjalankannya juga sangat pesat, mulai dari sebuah *game* yang berjalan pada PC, hingga pada mobile yang memiliki *sistem operasi* yaitu *smartphone* salah satunya adalah *Android*. *Android* merupakan *sistem operasi* pada mobile yang berkembang sangat pesat, karena masih tergolong *sistem operasi mobile* yang baru. Perkembangan untuk aplikasi yang berjalan pada *Android* sangatlah pesat, namun untuk perkembangan *Game RPG* masih sangat kurang. Sebagian besar *Game RPG* yang terdapat pada *Android* merupakan sebuah *game* yang pada awalnya berasal dari *sistem operasi* lain seperti *IOS* kemudian di terapkan pada *Android*, bukan murni *Game RPG* dari yang dibangun untuk *Android.*

Untuk menyikapi hal ini dibutuhkan sebuah dobrakan kemajuan yang dilakukan agar dapat ikut bersaing dengan kemajuan teknologi yang terjadi. Membangun sebuah *Game RPG* yang berjalan pada mobile ber*sistem operasi Android* menggunakan *controller* virtual yang memiliki fungsi seperti *controller* pada PSP. *Controller* yang digunakan menjadi satu dengan perangkat utamanya, dalam hal ini menggunakan interaksi *Touchscreen*.

Aplikasi *Game RPG* berbasis *Controller PSP* pada *Android* ini merupakan salah satu contoh *Game RPG* yang berjalan pada mobile ber*sistem operasi Android* menggunakan *controller* berbasis pada *PSP*. Pengguna dapat memainkan dan mengeksplorasi *game* ini dan seolah-olah bermain pada perangkat PSP. *Game RPG* yang dibangun memiliki tema yang bersifat hiburan, sehingga pengguna bisa relaks dalam memainkannya. Aplikasi *Game RPG* ini juga di lengkapi dengan animasi dan suara, sehingga juga akan membuat aplikasi ini lebih interaktif.

Kata kunci : *Game RPG, Controller, PSP, Android.*

TAKARIR

Action	:	Aksi.
ActionScript	:	Bahasa pemrograman.
Android	:	Sistem Operasi.
Array	:	Tipe data yang menampung banyak data.
Attack	:	Kekuatan serangan.
Background Music	:	Lagu/musik yang dimainkan saat *game* berlangsung, pergantian lagu biasanya terjadi saat perpindahan kondisi atau tempat dalam suatu game.
Block Inventory	:	Sebuah blok yang setiap bloknya bisa berisikan barang, semisal ada 4 blok artinya kita hanya bisa mengisikan 4 barang dalam *inventory*.
Battle System	:	Sistem Pertarungan.
Touchscreen	:	Layar sentuh.
Controller	:	Sebuah perangkat yang mengendalikan jalannya permainan dalam suatu *game*.
Camera View	:	Sisi kamera.
Defend	:	Pertahanan.
Experience	:	Jumlah point yang digunakan untuk naik ke *level* berikutnya.
Game	:	Permainan.
Game Menu	:	Menu yang terdapat dalam permainan.
Level	:	Tingkatan karakter yang kita mainkan, semakin tinggi level maka semakin kuat karakternya.
HP (Hit Point)	:	Darah dari karakter pada game.
Image Blitting	:	Teknik pemotongan gambar, namun tidak memotong gambar secara langsung. Pengambilan sebagian gambar dari gambar. Misalkan kita hanya ingin mengambil gambar dengan ukuran 20x20 dan gambar berukuran 100x100.
Input	:	Masukkan, biasanya berupa data.
Inventory	:	Tempat untuk menampung barang, dalam *game*

		inventory merupakan tempat diama karakter meletakkan barang-barang yang di milikinya.
Interface	:	Antarmuka
Invisible	:	Tidak terlihat
MovieClip	:	Sebuah objek yang merupakan sebuah pengelompokkan dari banyak objek, dan bisa di manipulasi oleh *ActionScript*.
Map	:	Peta yang digunakan dalam suatu game, yang biasanya menandakan lokasi dari game tersebut.
Monster	:	Musuh yang terdapat dalam game, biasanya bukan berwujud seperti manusia.
NPC (Non-Player Character)	:	Karakter dalam game yang dikendalikan oleh komputer atau karakter dalam *game* yang tidak kendalikan.
Number	:	Tipe data bilangan.
Pixel	:	unsur gambar atau representasi sebuah titik terkecil dalam sebuah gambar grafis yang dihitung per inci.
Pixel Movement	:	Sebuah animasi yang pergerakannya berdasarkan *pixel*.
Prosessor	:	Otak dan pusat pengendali computer yang didukung oleh kompunen lainnya.
PSP (Play Station Portable)	:	Perangkat untuk bermain game, namun bisa dibawa kemana-mana dengan mudah dan bisa dimainkan dimana saja.
RPG	:	Jenis *game*. Merupakan game yang berperan. Semakin lama kita memainkan game berjenis ini, maka karakter yang kita gunakan juga semakin kuat.
Sistem Operasi	:	Perangkat lunak sistem yang bertugas untuk memberikan abstraksi terhadap kontrol dan manajemen perangkat keras serta operasi-operasi dasar sistem, yang digunakan oleh *software* aplikasi seperti program-program pengolah kata dan browser web.

String	:	Tipe data yang merupakan kumpulan dari beberapa karakter.
Sprite	:	Sebuah gambar yang di dalamnya terdapat beberapa gambar yang tujuannya akan digunakan sebagai pembuatan karakter pada suatu *game*.
Shape	:	Bentuk dasar objek
Software	:	Perangkat lunak.
Sound Effect	:	Suara atau lagu yang digunakan saat terjadi sesuatu yang singkat, seperti suara saat menebas pedang, melompat, teriak, dan sebagainya.
TextField	:	Merupakan sebuah objek yang menampung teks.
Tools	:	Alat-alat yang digunakan.
Typing Text	:	Sebuah pesan yang ditampilkan perhuruf, mulai dari huruf pertama hingga huruf terakhir.
User	:	Pengguna atau pemain.

DAFTAR ISI

HALAMAN JUDUL i
LEMBAR PERNYATAAN KEASLIAN HASIL TUGAS AKHIR iv
HALAMAN PERSEMBAHAN v
MOTTO viii
KATA PENGANTAR x
SARI xii
TAKARIR xiii
DAFTAR ISI xvi
DAFTAR GAMBAR xx
BAB I PENDAHULUAN 1
1.1 Latar Belakang 1
1.2 Rumusan Masalah 3
1.3 Batasan Masalah 3
1.4 Tujuan Penelitian 4
1.5 Manfaat Penelitian 4
1.6 Metodologi Penelitian 5
1.6.1 Metode Pengumpulan Data 5
1.6.2 Metode Pembangunan Sistem 5
1.7 Sistematika Penulisan 7
BAB II LANDASAN TEORI 9
2.1 *Game* 9
2.1.1. Konsep Pengembangan *Game* 9
2.1.2 Elemen *Game* 11
2.1.2 *Genre Games* 14
2.1.3 Teori Dalam Pembuatan *Game* 15
2.2 *Game Engine* 17
2.2.1 *Pure Game Engine* 17
2.2.2 *Game Framework* 18
2.2.3 *Game Authoring Tools* 18
2.3 *Game Development* 18
2.4 *Sistem Operasi Android* 18

2.4.1 Perkembangan *Sistem Operasi Android* 19
2.4.1 *Adobe AIR Android* 19
2.5 *Controller Playstation Portable (PSP)* 20
2.5.1 Basis *Controller PSP* 20
2.6 Perangkat Lunak 22
2.7 Bahasa Pemrograman 23
2.7.1 Dasar-Dasar Pemrograman *ActionScript* 3.0 23
2.7.2 *ActionScript 3.0* pada *Sistem Operasi Android* 25
BAB III METODOLOGI 27
3.1 Metodologi Analisis 27
3.2 Analisis Kebutuhan 27
3.2.1 Analisis Kebutuhan Input 27
3.2.2 Analisis Kebutuhan Fungsi dan Kinerja 27
3.2.3 Analisis Kebutuhan Output 28
3.2.4 Analisis Kebutuhan Perangkat Keras 28
3.2.5 Analisis Kebutuhan Perangkat Lunak 29
3.3 Perancangan Perangkat Lunak 30
3.3.1 Metode Perancangan 30
3.3.2 Hasil Perancangan 30
3.4 Perancangan Diagram Alir (Flow Chart) 31
3.4.1 Flow Chart Controller Game Menu 32
3.4.2 Flow Chart Pergerakan Karakter 33
3.4.3 Flow Chart Battle System 35
3.5 Perancangan Antarmuka 36
3.5.1 Antarmuka *Controller* 36
3.5.2 Antarmuka Menu Utama (*Main Menu*) 36
3.5.3 Antarmuka Menu Cara Bermain 38
3.5.4 Antarmuka Menu *About* 38
3.5.5 Antarmuka Menu *Name Input* 39
3.5.6 Antarmuka *Game Play* 40
3.5.7 Antarmuka *Character Message Display* 41
3.5.8 Antarmuka *Game Menu* 42
3.5.9 Antarmuka *Game Menu Status* 43
3.5.10 Antarmuka *Game Menu Exit* 45

3.5.11 Antarmuka *Game Over* 46
BAB IV HASIL DAN PEMBAHASAN 48
4.1 Batasan Implementasi 48
4.2 *Game RPG* Berbasis *Controller PSP* pada *Sistem Operasi Android* 48
4.2.1 Alur Cerita 48
4.3 Implementasi Komponen-komponen *Game RPG* 49
4.3.1 Pergerakan Karakter dan Aksi 49
4.3.2 *NPC (Non-Player Character)* 52
4.3.3 *Typing Character Message Display* 53
4.3.4 *Map* 54
4.3.5 *Collision Detection* 56
4.3.6 *Camera View* 57
4.3.7 Sistem *Inventory* 58
4.3.8 Sistem Pertarungan 59
4.3.9 Kecerdasan Dasar untuk Musuh dalam Bertarung 60
4.4 Implementasi Antarmuka 63
4.4.1 Menu Utama 63
4.4.2 Halaman Cara Bermain 64
4.4.3 Halaman About 65
4.4.4 Halaman Input Nama Karakter 66
4.4.5 *Game Play* 66
4.4.6 Halaman *Game Menu* 68
4.4.7 Halaman *Game Menu Status* 69
4.4.8 Halaman *Game Menu Exit* 69
4.4.9 Halaman *Game Over* 70
4.5 Analisis Kinerja *Game RPG* Berbasis *Controller PSP* pada *Android* 71
4.5.1 Pengujian *Game* pada *Mobile Android* 71
4.5.2 Kelebihan dan Kekurangan Sistem 74
BAB V KESIMPULAN DAN SARAN 76
5.1 Kesimpulan 76
5.2 Saran 76

DAFTAR PUSTAKA xxi

DAFTAR GAMBAR

Gambar 3.1 Proses *Game RPG* secara umum 31
Gambar 3.2 Proses pemilihan menu pada *Game Menu* 32
Gambar 3.3 Proses pergerakkan karakter pada *Game Play* 34
Gambar 3.4 Proses *Battle System* dan manajemen *item* 35
Gambar 3.5 Rancangan Antarmuka *Controller* 36
Gambar 3.6 Rancangan Antarmuka Menu Utama 37
Gambar 3.7 Rancangan Antarmuka Menu Utama dengan Controller 37
Gambar 3.8 Rancangan Antarmuka Cara Bermain 38
Gambar 3.9 Rancangan Antarmuka *About* 38
Gambar 3.10 Rancangan Antarmuka *About* dengan *Controller* 39
Gambar 3.11 Rancangan Antarmuka *Name Input* 39
Gambar 3.12 Rancangan Antarmuka *Game Play* 40
Gambar 3.13 Rancangan Antarmuka *Game Play* dengan *Controller* 41
Gambar 3.14 Rancangan Antarmuka *Character Message Display* 41
Gambar 3.15 Rancangan Antarmuka *Game Menu* 42
Gambar 3.16 Rancangan Antarmuka *Game Menu* dengan *Controller* 43
Gambar 3.17 Rancangan Antarmuka *Game Menu Status* 44
Gambar 3.18 Rancangan Antarmuka *Game Menu Status* dengan *Controller* 45
Gambar 3.19 Rancangan Antarmuka *Game Menu Exit* 45
Gambar 3.20 Rancangan Antarmuka *Game Menu Exit* dengan *Controller* 46
Gambar 3.21 Rancangan Antarmuka *Game Over* 47
Gambar 4.1 Sprite karakter utama kondisi menyerang 50
Gambar 4.2 Sprite animasi naik level 51
Gambar 4.3 Sprite *Non-Player-Character* 52
Gambar 4.4 Sprite wajah karakter 54
Gambar 4.5 Sprite *Background Typing Character Message Display.* 54
Gambar 4.6 Sprite Snow Map 55
Gambar 4.7 Sprite Snow Map layer atas 55
Gambar 4.8 Sprite Snow Map Block 56
Gambar 4.9 Tampilan Sistem *Inventory* pada *Game Play* 58
Gambar 4.10 Sprite musuh (Devil Terbang) kondisi diam 60
Gambar 4.11 Sprite animasi serangan 60

Gambar 4.12 Sprite animasi ledakan 60
Gambar 4.13 Tampilan Halaman Menu Utama 64
Gambar 4.14 Tampilan Halaman Cara Bermain 64
Gambar 4.15 Tampilan Halaman About 65
Gambar 4.16 Tampilan Halaman Input Nama Karakter 66
Gambar 4.17 Tampilan *Game Play* 67
Gambar 4.18 Tampilan Halaman *Game Menu* 68
Gambar 4.19 Tampilan Halaman *Game Menu Status* 69
Gambar 4.20 Tampilan Halaman *Game Menu Exit* 70
Gambar 4.21 Tampilan Halaman *Game Over* 70
Gambar 4.22 Pengujian *Game* pada Samsung Galaxy GT-I5500 71
Gambar 4.23 Pengujian *Game* pada AHA Touch 72
Gambar 4.24 Pengujian *Game* pada Samsung Galaxy Mini 72
Gambar 4.25 Pengujian *Game* pada Samsung Galaxy GT-I5501 73
Gambar 4.26 Pengujian *Game* pada Samsung Galaxy GIO 73

BAB I
PENDAHULUAN

1.1 Latar Belakang

Perkembangan pertumbuhan industri game dari tahun ke tahun amat mencengangkan, berbagai jenis game dengan gaya bermain baru telah bermunculan. Salah satunya adalah *Game RPG*.

Game RPG (Role Playing Game) 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.

Dalam sebuah permainan *RPG*, jarang ada yang "kalah" atau "menang". Ini membuat permain *RPG* berbeda dari jenis permainan papan lainnya seperti Monopoli atau Ular Tangga, permainan kartu, olah raga, 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.

Ada pula sejenis permainan *RPG* dimana 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. Dalam Game *RPG* juga terdapat beberapa rintangan, strategi dan tantangan sehingga bisa meningkatkan kreatifitas seseorang dalam menyelesaikan sebuah permasalahan untuk mencapai suatu tujuan tertentu.

Seiring dengan perkembangan teknologi game, platform dan konsol untuk menjalankan aplikasi gamepun juga berkembang, mulai dari Aplikasi Game yang

berjalan di Playstation, PC, hingga berjalan di platform mobile bahkan sekarang menggunakan tekhnologi *Touchscreen*.

Touchscreen (Layar Sentuh) merupakan sebuah tekhnologi yang sedang marak pada saat ini, yaitu merupakan tekhnologi perkembangan dari perangkat berinteraksi *Keypad* atau memiliki tombol tersendiri untuk melakukan sebuah aksi, dalam hal ini bisa dilihat dengan perkembangan Mobile. Saat ini sudah banyak sekali Mobile yang meminimalisasi tombol-tombol yang berada pada tekhnologi Keypad sebelumnya, namun tombol-tombol tersebut langsung ada pada layar *LCD* Mobile, dan untuk melakukan interaksinya langsung di sentuh pada layar *LCD* tersebut. Sudah banyak sekali sistem operasi Mobile yang sudah support dengan tekhnologi *Touchscreen* terutama untuk sistem operasi *Android*, *iOS* dan *Windows Phone 7*. Dalam hal ini akan digunakan sistem operasi *Android* karena tergolong sistem operasi mobile yang paling baru.

Android merupakan *Operating System* (OS) untuk telepon seluler yang berbasis Linux. *Android* menyediakan platform terbuka bagi para pengembang buat menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam peranti lunak untuk ponsel. Awalnya, Google Inc. membeli *Android* Inc., *Android* dikembangkan oleh Google, dan pertama kali dikenal pada bulan november 2007. Dikarenakan tergolong OS baru, maka aplikasi yang kompatibel untuk OS *Android* belum begitu banyak beredar. Oleh karena itu pada Tugas Akhir kali ini, penulis akan mencoba membuat sebuah game *RPG* yang di tekankan pada sistem permainannya menggunakan *Controller* berbasis *Playstation Portable (PSP)* yang berjalan pada Sistem Operasi *Android.*

Controller PSP merupakan sebuah *controller* yang terhubung langsung pada perangkatnya, beda halnya dengan Controller pada Play Station 1, Play Station 2, Play Station 3, ZNES, Nitendo, dan semacamnya yang Kontrollernya terpisah dengan perangkatnya dan di hubungkan oleh kabel. Basis *controller* PSP merupakan sebuah *controller* yang berbasis pada *controller* pada PSP, yaitu merupakan *controller* yang terhubung langsung dengan perangkatnya, kemudian merupakan *controller* yang multi fungsi dalam berbagai macam *event* dalam setiap permainan, misalkan saja pada Game RPG. Pada Game RPG terdapat

banyak sekali event yang berbeda-beda dan dikendalikan oleh satu *controller*. Pada saat pergerakan aktor utama**,** *controller* berfungsi untuk menggerakkan aktor untuk berjalan bahkan pada Game RPG yang memiliki *Action Battle System,* selain menggerakan karakter juga berfungsi untuk melakukan aksi untuk menyerang maupun bertahan. Pada opsi menu utama maupun menu lainnya, *controller* ini berubah fungsi untuk menggerakkan sebuah *pointer* untuk pemilihan menu.

1.2 Rumusan Masalah

Berdasarkan uraian latar belakang di atas, masalah yang di rumuskan adalah sebagai berikut :

1. Bagaimana cara membangun *Game RPG* beserta komponen-komponennya tanpa menggunakan *Game Engine,* tetapi menggunakan Script murni Action Script 3.0 dan Editor yaitu *Adobe Flash CS5*.
2. Bagaimana cara menerapkan *Game RPG* tersebut pada Sistem Operasi Android dengan menggunakan *Controller* yang berbasis *PSP*.
3. Bagaimana cara mengkombinasikan *Controller* berbasis *PSP* dengan interaksi *Touchscreen* dari segi keseluruhan layar maupun keyboardnya.

1.3 Batasan Masalah

Batasan masalah sangat diperlukan agar penelitian dapat berjalan dengan terarah dan mempermudah persoalan yang dihadapi. Selain itu, batasan masalah juga berfungsi sebagai penegasan agar rumusan masalah yang telah disampaikan dapat terselesaikan. Berikut ini adalah batasan masalah yang dipergunakan dalam penilitan :

1. Aplikasi *Game RPG* ini hanya bisa dijalankan pada Sistem Operasi *Mobile Based Android* versi 2.2 (*Froyo*) dan di atasnya.
2. Aplikasi *Game RPG* ini ditekankan pada pembuatan Sistem Permainan dan komponen-komponen gamenya bukan pada *skenario* dan lama permainannya. Dalam *Game RPG* yang akan dibuat ini terdapat 1 permasalahan/*skenario* yang akan diselesaikan.

3. Interaksi menggunakan *Touchscreen* pada *Controller* dan keseluruhan layar tergantung pada kondisi pada game. Dalam game ini terdapat 6 tombol pada *Controller* yang akan digunakan yaitu tombol atas, kanan, bawah, kiri, A, dan B.

1.4 Tujuan Penelitian

Adapun tujuan penelitian adalah:

1. Membangun sebuah aplikasi *Game RPG* yang menerapkan teknologi baru yang saat ini sedang marak untuk di eksplorasi, yaitu Android. Pada Aplikasi game ini akan di terapkan sebuah permainan berperan (RPG Game) pada sebuah piranti Touchscreen yang menggunakan Controller berbasis PSP, sehingga orang akan merasa seperti bermain PSP dengan kombinasi Touchscreen yang fleksibel.
2. Memanfaatkan dan mempelajari pembuatan Game RPG menggunakan AIR Teknologi dari Adobe, menggunakan pemrograman utama yaitu Action Script 3.0 (AS3).

1.5 Manfaat Penelitian

Berdasarkan penelitian, diharapkan dapat memberikan manfaat sebagai berikut:

a. Dalam Dunia Akademik

   Dapat digunakan sebagai referensi untuk mahasiswa lainnya dalam pembuatan suatu game khususnya *Game* bergenre *RPG* tanpa menggunakan *Game Engine*.

b. Dalam Dunia *Game Development*

   Memberikan motivasi dan inisiatif kepada *Game Developer* dalam pembuatan Game khususnya Game bergenre RPG tanpa menggunakan *Game Engine*, karena dengan menggunakan bahasa pemrograman murni kita bisa mengembangkan Game tersebut sesuka hati kita, tidak terbatas pada *tools-tools* dan fungsi-fungsi yang terbatas yang di sediakan pada *Game Engine* itu. Meskipun bisa mengembangkan Game menggunakan

Game Engine namun kita harus berfikir 2 kali dalam pembuatannya yaitu menganalisa fungsi-fungsi dan *Script* bawaan dari *Game Engine* tersebut terlebih dahulu kemudian mengimplementasikan *algoritma* yang kita inginkan ke dalam *Game Engine* tersebut. Dalam hal ini bukan berarti kita menghindari penggunaan *Game Engine* namun menjadikan *Game Engine* sebagian opsional jika terdapat beberapa permasalahan dalam pembuatan yang tidak bisa di kerjakan, bukan merupakan *Software* utama dalam pembuatan sebuah *Game*.

1.6 Metodologi Penelitian

Metodologi yang digunakan dalam pembuatan *Game* ini meliputi pengumpulan data dan pembangunan sistem meliputi perancangan dan implementasi dari sistem yang akan dibangun.

1.6.1 Metode Pengumpulan Data

Metodologi ini dilakukan dengan cara mengumpulkan data dari beberapa referensi *Game RPG* yang relevan dengan permasalahan yang dihadapi, mempelajari komponen-komponen yang terdapat dalam *Game RPG* tersebut dengan memainkan beberapa *Game RPG* maupun melalui situs Internet.

1.6.2 Metode Pembangunan Sistem

Pada pembangunan ini akan mempergunakan metodologi seperti berikut:

1. Analisis Kebutuhan Sistem *Game RPG*.
   Mengumpulkan Informasi tentang apa saja yang dibutuhkan dalam pembuatan game ini, misalkan salah satunya adalah SDK Android.
2. Perencanaan dan Perancangan *Game RPG*.
   Perencanaan tentang alur dan sistem permainan, variable-variabelnya dan perancangan skenario game.
3. Mendesign interface (antar muka) *Game RPG*.
   Mensketsa design interfacenya setiap bagian kemudian menerapkan pada game yang akan dibuat.

4. *Coding* (Pengkodean).

   Setelah tahap-tahap di atas yang sudah jelas alur permainan dan variable-variabelnya, maka pada tahap ini adalah pembuatan perangkat lunak dari rancangan yang sudah dibuat sebelumnya. Dalam tahap ini di bagi menjadi beberapa step, yaitu :

   a. Pergerakan Karakter dan Aksi

      Membangun Pergerakan dan aksi karakter

   b. *Map*

      Membangun sebuah peta untuk karakter dan penempatan keseluruhan objek berdasarkan dengan file gambar yang digunakan.

   c. *Collision Detection*

      Membangun sebuah Collision (batasan atau tubrukan).

   d. *NPC (Non-Player Character)*

      Membangun sebuah karakter yang tidak dikendalikan oleh player dan bisa berinteraksi dengan player.

   e. *Typing Character Message Display*

      Membangun salah satu komponen utama dalam game RPG, yaitu pembicaraan pada setiap karakter, entah itu katakter utama maupun tidak (NPC).

   f. *Camera View*

      Membagun sebuah kamera sisi untuk menghindari adanya map yang lebih besar dari ukuran layarnya.

   g. Sistem *Inventory*

      Membangun sistem inventory untuk menampung barang-barang yang ditemukan.

   h. Sistem Pertarungan

      Membangun Sistem pertarungan antar karakter dengan musuh.

   i. Kecerdasan Dasar untuk Musuh dalam Bertarung

      Membangun musuh yang bisa berjalan, bergerak dan melakukan aksi sendiri.

5. Pengujian sistem dan editing.

   Pada tahap ini akan dilakukan uji coba terhadap game yang dibuat yang semula menggunakan SDK Android Emulator ke dalam sebuah Mobile yang bersistem Operasi Android yang memiliki spesifikasi sesuai dengan batasan masalah di atas, kemudian dilakukan pengujian juga terhadap game di setiap sisinya, kemudian untuk tahap editing juga merupakan tahap finishing dalam pembuatan game ini.

1.7 Sistematika Penulisan

Dalam penyusunan laporan Tugas Akhir, disusun per bab dan berurutan untuk mempermudah pembahasannya. Secara garis besar, sistematika dari penulisan terdiri atas lima bab, yaitu:

BAB I PENDAHULUAN

Diawali dengan penjelasan mengenai latar belakang masalah, kemudian dilanjutkan dengan rumusan masalah, batasan masalah, maksud dan tujuan penelitian, manfaat penelitian, metodologi penelitian, dan sistematika penulisan.

BAB II LANDASAN TEORI

Membahas lebih dalam mengenai landasan teori yang dapat membantu para pembaca dalam memahami implementasi yang akan dilakukan. Landasan teori tersebut mencakup tentang *game, game engine*, *game development*, Basis *controller PSP*, *Sistem Operasi Android,* perangkat lunak dan bahasa pemrograman yang digunakan.

BAB III METODOLOGI

Bagian ini terdiri dari pembahasan komponen-komponen yang akan dibangun dalam *Game RPG*, analisis kebutuhan dan perancangan sistem yang akan dibuat. Pembahasan sistem *Game RPG* mencakup pembahasan tentang komponen-komponen *Game RPG* yang akan dibuat beserta *Game RPG* yang akan dibuat. Analisis kebutuhan membahas tentang kebutuhan komponen-komponen

Game RPG, Penerapan pada Mobile yang *bersistem Operasi Android*, kebutuhan antarmuka *(interface)* serta kebutuhan software dan hardware yang akan digunakan.

Sedangkan perancangan komponen-komponen *Game RPG* membahas tentang metode perancangan, hasil perancangan, dan perancangan antarmuka *(interface)*.

BAB IV HASIL DAN PEMBAHASAN

Berisi implementasi Komponen-komponen *Game RPG* dan analisis kinerja sistem. Implementasi merupakan tahap dimana sistem siap dioperasikan pada tahap yang sebenarnya, sehingga akan diketahui apakah sistem yang telah dibuat sesuai dengan yang direncanakan. Sedangkan analisis kinerja sistem berisi pengujian program yang telah diimplementasikan ke dalam sebuah *Game RPG* yang berjalan pada *Sistem Operasi Android*.

BAB V KESIMPULAN DAN SARAN

Berisi kesimpulan dari hasil impelementasi dan analisa kinerja yang telah dikembangkan. Serta saran-saran yang perlu diperhatikan berdasarkan keterbatasan yang ditemukan dari sistem yang telah dikembangkan.

BAB II
LANDASAN TEORI

2.1 *Game*

Game adalah sebuah aplikasi permainan komputer yang permainannya dikontrol oleh seorang pemain, yang memiliki peraturan dan misi tertentu. Game itu sendiri mempunyai fungsi dan tujuan sebagai aturan untuk menampilkan atau mengimplementasikan sebuah permainan. (Fazri, 2009)

2.1.1. Konsep Pengembangan *Game*

Secara garis besar, konsep pengembangan *game* terbagi menjadi beberapa bagian, yaitu:

1. Perancangan Konsep *Game*

Penentuan ide dasar *game* secara keseluruhan, pada bagian ini ditentukan juga apakah *game* tersebut ber *genre Puzzle, Action* atau RPG. Bisa juga menentukan konsep dari keseluruhan *game* tersebut, apakah tentang peperangan, persahabatan dan lain sebagainya. Bagian ini juga menentukan tujuan dan sasaran *game*, apakah *game* tersebut ditujukan untuk anak-anak, remaja atau dewasa.

2. Implementasi *Engine Game*

Engine game merupakan inti atau mesin utama dari *game* yang mengendalikan seluruh aktivitas dan interaksi yang terjadi di dalam *game*. *Engine* dapat dibangun dengan bahasa pemrograman seperti C++, Basic, Java atau bahkan Assembly. Saat ini sudah banyak software yang merupakan *engine* sebuah *game*. Software tersebut seringkali disebut dengan *tools game*, contohnya RPG Maker XP. Contoh lain dari *tools game* antara lain Game Maker, 3D Game Studio, Torque, FPS Creator dan masih banyak lagi baik yang bersifat *freeware, shareware* maupun *software* komersial.

Biasanya terdapat empat komponen utama yang membentuk *engine* game. Komponen-komponen tersebut antara lain:

a. *Database* (Basis Data)

Basis data dalam game berguna untuk mendaftarkan dan menyimpan data tentang objek-objek maupun kejadian di dalam *game*. Misalnya daftar karakter, daftar *item*, peta, *level*, *save data* pada *game-game* ber-*genre* RPG, RTS, maupun *Adventure*. Contoh lain adalah *Highest Score* pada *Tetris* atau Daftar Langkah dalam Catur.

b. *Game Objects* (Objek *Game*)

Objek *game* merupakan objek yang akan dikendalikan di dalam *game* tersebut, bagian ini mengimplementasikan objek *sprite game* dan objek lingkungan pada *game*. *Sprite* adalah objek yang dapat dikendalikan oleh pemain atau *AI (Artificial Intelligence)*. Sprite memiliki atribut dan *method* tertentu yang sesuai dengan konsep *gameplay* (contoh: karakter bisa berjalan dan melompat). Objek lingkungan pada *game* yaitu latar belakang dan benda. Latar belakang yaitu wadah atau tempat dimana objek *sprite* akan dikendalikan.

c. *Artificial Intelligence* (Kecerdasan Buatan)

AI (*Artificial Intelligence*) adalah kecerdasan buatan yang diberikan pada suatu objek agar tidak berlaku secara acak. Tujuan pemberian AI adalah agar pemain tidak asal mengambil keputusan dan mengembangkan semangat berkompetisi. Pemberian AI dapat diberikan pada lawan tanding (*opponent*) seperti Catur, *Tic Tac Toe*, *Action*, RPG.

d. *Game Control* (Kendali Game)

Pengendalian *game* merupakan interaksi antara pemain dengan objek *game* sebagai hasil dari keputusan pemain tersebut. Untuk kendali *game* dapat menggunakan *joystick* (pada game konsol), *keyboard*, dan *mouse* (PC Game). Kendali *game* juga membatasi hal-hal yang dapat dilakukan oleh objek *game*.

3. Penanaman Visual dan Audio pada *Game*

Aspek Visual merupakan segala sesuatu yang dapat dilihat oleh pemain, aspek visual dapat berupa model 2 dimensi (2D), model 3 dimensi (3D) dan antarmuka. Model 2 dimensi dibuat dengan garis-garis pada bidang, pewarnaan, *masking*, gradasi dan efek-efek pencampuran warna. Contoh *tools* yang sering dipakai adalah Adobe Photoshop, Corel Draw, GIMP. Model 3 dimensi dibuat dengan rangka *grid*, proporsi, *rendering* dan pencahayaan. Contoh *tools* yang sering dipakai adalah 3D Studio Max, Maya, Blender. Sedangkan antarmuka biasanya berisi informasi yang diperlukan oleh pemain dalam memainkan *game* tersebut, misalnya penjelasan tentang kendali, objek, *gameplay* maupun ikon-ikon *informative*. Representasi aspek visual ini bisa dalam bentuk animasi, grafik objek atau grafik statis.

4. Pengujian

Pengujian dilakukan untuk menjaga mutu *game* yang telah dibuat agar *game* terhindar dari kesalahan kecil atau *bug* yang dapat menurunkan kualitas *game*. Pengujian meliputi pengujian *engine game*, pengujian interaksi *game* dengan pemain melalui pengendali (*controller*), *debugging* dan perbaikan. Bagian ini disebut *Alpha Testing*. Sedangkan *Beta Testing* adalah pengujian pihak ketiga yang merupakan pemain untuk mengetahui apakah di dalam *game* tersebut masih ada bug yang mengganggu atau tidak.

2.1.2 Elemen *Game*

Di dalam sebuah *game*, terdapat elemen-elemen yang akan membentuk *game* itu sendiri, antara lain:

1. *Rules* (Aturan-aturan)

Sebuah *game* mengambil tempat atau *setting* di dunia buatan yang diatur oleh aturan-aturan (*rules*). *Rules* inilah yang menentukan aksi dan gerakan pemain dalam sebuah *game*. Pada *game* komputer, kebanyakan *rules* ini

tersembunyi. Karena pemain berinteraksi dengan *game* hanya melalui suatu alat masukan atau yang biasa disebut *input device*.

2. *Victory Condition* (Kondisi Menang) dan *Lose Condition* (Kondisi Kalah)

Selain mempunyai kondisi kemenangan, game juga memiliki kondisi kekalahan. Kadang kondisi kekalahan ini bersifat implisit, yaitu jika pemain tidak mendapatkan kondisi kemenangan, maka dinyatakan kalah. Sedangkan yang bersifat eksplisit, misalnya tokoh telah mati karena kehabisan nyawa atau darah.

3. *Setting*

Sebuah *game* mangambil tempat (*setting*) pada suatu lokasi tertentu.

4. *Interaction Model* (Model Interaksi)

Interaction Model adalah suatu cara pemain beinteraksi dengan *game* dan melakukan aksi untuk menghadapi tantangan dari *game* tersebut. Model interaksi yang biasa dipakai pada *game* komputer ada dua jenis, yaitu:

a). Jika pemain mengendalikan tokoh tunggal yang merepresentasikan dirinnya dalam *game*, dan tokoh tersebut dapat mempengaruhi dunia sekitarnya, maka tokoh tersebut disebut sebagai *avatar*-nya.

b). Jika pemain memiliki kemampuan untuk melihat berbagai bagian dari *gameworld* (dunia permainan) dan melakukan aksi pada banyak tempat, maka pemain tersebut disebut sebagai *omnipresent*. Namun hal itu hanya dapat diterapkan pada tokoh atau unit yang menjadi miliknya, dengan memberikan perintah kepada mereka satu persatu, contohnya adalah permainan catur.

5. *Perspective* (Sudut Pandang)

Perspective menjelaskan sudut pandang dari suatu *game* pada layar. Adapun jenis-jenisnya adalah:

a). *First person view*, yaitu pemain melihat langsung melalui mata *avatar*-nya, seperti pada *game* Counter Strike.

b). *Third person view*, yaitu pemian mengikuti *avatar*-nya dari belakang dan melihat secara tiga dimensi, seperti pada *game* Blade and Soul, Dragonnest.

c). *Side scrolling view*, yaitu pemain melihat dari sisi samping, seperti pada *game* Mario Bros.

d). *Top-down view*, yaitu pemain melihat langsung dari atas, seperti pada *game* Pac-Man.

e). *Isometric view*, yaitu pemain melihat dari atas dengan sudut berkisar antara 30^o-40^o, seperti pada *game* Age of Empire, Dota AllStars.

f). *Changeable view*, yaitu sudut pandang yang dapat diganti-ganti sesuai keinginan pemain, seperti pada *game* Ragnarok Online.

6. *Role* (Peran)

Role atau peran merupakan tokoh yang dimainkan oleh pemain dalam suatu *game*. Dengan adanya peran ini, maka pemain akan lebih mudah untuk memahami tujuan apa yang sebenarnya ingin dicapai dan aturan apa yang dimainkan. Sebagai contoh, pada seri *game* Dragonica Online, pemain berperan sebagai kesatria yang di tugaskan untuk menyelesaikan beberapa macam jenis permasalahan di setiap kota untuk menemukan sebuah petunjuk dalam mengalahkan Naga, dsb.

7. *Mode*

Beberapa *game*, seperti catur, berlaku dari awal sampai akhir. Pemain selalu ingin menyelesaikan hal yang sama dengan cara yang sama pula. Namun ada juga *game* yang memiliki mode yang nyata, yaitu dimana *gameplay*-nya berubah dari satu mode ke mode yang lainnya. Contohnya seperi pada *game* perang, yaitu sebelum pemain turun ke medan perang, biasanya diberi *briefing* terlebih dahulu, lalu pindah ke mode pemilihan senjata, dan terakhir adalah perang itu sendiri.

8. *Structure*

Hubungan antara mode dan aturan menentukan kapan dan mengapa *game* berubah secara bersama-sama membentuk struktur permainan.

9. *Realism*

Game menggambarkan sebuah dunia, bahkan mungkin dunia khayal. Sebuah *game* yang menerapkan akal sehat dan logika pada aturan permainannya dapat dikatakan sebagai *game* yang realistis. Contohnya adalah *game* Microsoft Flight Simulator, *game* ini mencoba mensimulasikan perilaku secara akurat dan detail dari sebuah pesawat terbang yang sesungguhnya.

10. *Story* (Cerita)

Game komputer merupakan perpaduan antara media pasif, naratif seperti televisi dan film, serta media aktif, non-naratif seperti permainan poker dan domino. Beberapa *game* komputer seperti Tetris, tidaklah mempunyai cerita. Lain halnya dengan seri Final Fantasy. Beberapa *game* memiliki alur cerita yang linear atau hanya satu jalan cerita dan bersifat non-interaktif, namun ada juga yang bersifat interaktif, dimana cerita akan berbeda atau bercabang, tergantung pada pilihan atau tindakan yang diambil pemain. Sehingga cerita akhir (*ending*) yang didapat akan berbeda-beda pula, atau biasa disebut dengan *multiple ending*.

2.1.2 *Genre Games*

Adapun *genre* atau jenis *game* antara lain :

1. *Action Games*

Biasanya meliputi tantangan fisik, teka-teki (*puzzle*), balapan, dan beberapa konflik lainnya. Dapat juga meliputi masalah ekonomi sederhana, seperti mengumpulkan benda-benda.

2. *Strategy Games*

Sebuah *game* yang melibatkan masalah strategi, taktik, dan logika.

3. *Role Playing Games*

Kebanyakan *game* jenis ini melibatkan masalah taktik, logika, dan eksplorasi atau penjelajahan. Kadang juga meliputi teka-teki dan masalah ekonomi, karena pada *game* ini biasanya melibatkan pengumpulan barang-

barang rampasan dan menjualnya untuk mendapatkan senjata yang lebih baik.

4. ***Real World Simulation***

Meliputi permainan olahraga dan simulasi kendaraan, termasuk kendaraan militer. *Game* ini kebanyakan melibatkan masalah fisik dan taktik, tetapi tidak masalah eksplorasi, ekonomi, dan konseptual.

5. ***Construction and Management***

Seperti *game City Ville*. Pada dasarnya adalah ekonomi dan konseptual. *Game* ini jarang yang melibatkan konflik dan eksplorasi, serta hampir tidak pernah meliputi tantangan fisik.

6. ***Adventure Games***

Mengutamakan masalah eksplorasi dan pemecahan masalah teka-teki. Namun terkadang meliputi masalah konseptual dan tantangan fisik, tapi sangat jarang.

7. ***Puzzle Games***

Ditujukkan untuk memecahkan masalah tertentu. Hampir semua tantangan disini menyangkut masalah logika yang biasanya dibatasi oleh waktu.

2.1.3 Teori Dalam Pembuatan *Game*

Urutan pembuatan *game* dan pengembangan program *game* adalah sebagai berikut:

a. Menentukan tipe permainan yang akan dibuat

Penentuan ini sebagai dasar mulai bekerja sampai mendapatkan ide yang bagus untuk dibuat program permainan.

b. Definisikan model permainan dan tujuannya

Pada tahap ini, sebaiknya model permainannya ditulis secara jelas sehingga jika terjadi perubahan, maka permainannya tetap konsisten dan tidak membingungkan.

c. Definisikan secara jelas *Game Worlds*-nya

Game worlds adalah elemen-elemen utama yang terdapat dalam suatu program permainan yang terdiri dari:

1). *Game board*

Merupakan bentuk tampilan, latar belakang, dan lainnya.

2). Instruksi untuk permainan

Instruksi untuk permainan harus jelas supaya tidak membingungkan dan pemain dapat menentukan strategi dari permainannya.

3). Informasi untuk pemain

Informasi ini penting ditampilkan dalam program permainan ketika sedang berjalan.

4). Penghargaan

Digunakan sebagai rangsangan untuk pemain ketika menyelesaikan level tertentu dan mendorong untuk melanjutkan ke level berikutnya.

5). Variasi

Digunakan agar pemain tidak merasa cepat bosan, tetapi variasi tidak boleh berlebihan, karena dapat membuat alur cerita menjadi tidak konsisten dan membingungkan.

6). Tingkat kesulitan

Tingkat kesulitan dari program permainan akan membuat pemain lebih bersemangat jika melewati tingkat kesulitan yang diberikan.

d. Pastikan permainan bisa dimainkan

Permainan yang tidak mungkin diselesaikan akan membuat program permainan tidak dapat dimainkan lagi.

e. Rancang program sebaik mungkin

Gunakan teknik pemrograman yang sesuai dalam membuat *game* dan buat program yang mudah dimodifikasi dan dikembangkan.

f. Pengujian program

Program yang telah selesai harus diuji. Program diuji oleh perancangnya sendiri (*alpha test*) untuk menentukan kesalahan logika yang mungkin

terjadi. Selain itu, program juga diujikan kepada orang lain (*beta test*) untuk mengatahui kesalahan logika yang tidak terlihat oleh perancangnya. Sehingga dapat memberi masukan, saran, atau ide dalam pengembangan program permainan selanjutnya.

2.2 *Game Engine*

Game engine adalah perangkat lunak yang didesain khusus untuk pembuatan dan pengembangan dari *video game.* Saat ini *Game engine* sudah mengalami peningkatan agar dapat memudahkan dalam pengembangan sebuah *game*. *Game engine* dibedakan menjadi tiga kategori yaitu, *pure game engine, game framework,* dan *authoring tools.*

2.2.1 *Pure Game Engine*

Pure Game engine adalah game engine yang dikembangkan dari beberapa engine dan library multimedia yang dibutuhkan dalam pengembangan *game*. Berikut ini beberapa *engine* dan *library* multimedia yang diperlukan dalam proses pengembangan game :

A. *Rendering*

   Engine yang digunakan untuk merender gambar dalam *game* , Contoh : OGRE 3D.

B. *Input*

   Library yang berfungsi untuk mendeteksi *input* dari perangkat yang digunakan *user*. Contoh : OpenCV (input Camera), SDL (Simple Direct Layer)

C. *GUI*

   Library untuk membuat *GUI* (*Graphical User Interfaces*) yang diperlukan di dalam game. Contoh : CEGUI.

D. *Sound*

   Library yang berguna untuk mengeluarkan suara . Contoh : FMOD

2.2.2 *Game Framework*

Framework merupakan pengembangan dari *Game engine* dimana sudah memiliki pattern tersendiri dan memisahkan proses ke dalam 3 lapisan MVC (*Model View Controller*). Penggunaan model framework dapat mengefisiensikan proses pengembangan karena tiap bagian divisi dapat bekerja secara bersama tanpa mengganggu divisi yang lain. Contoh *Framework game* adalah XNA game studio.

2.2.3 *Game Authoring Tools*

Authoring tools adalah perangkat lunak yang di dalamnya terdapat game engine dan menyediakan lingkungan pengembangan *visual* dan *scripting language*. *Authoring tools* memudahkan *developer* indie, karena sudah dilengkapi dengan beberapa editor dan *scripting language* untuk merubah *game system*. Contoh *authoring tools* adalah RPG Maker, Game Maker, dan FPS creator

2.3 *Game Development*

Perkembangan *game development* indonesia yang telah dimulai pada tahun 90-an sampai dengan saat ini sudah mengalami perkembangan yang cukup pesat, hal ini terlihat dari banyaknya studio lokal yang bergerak dibidang *Game Development*. Pendiri dari studio tersebut merupakan anggota dari komunitas *game developer* indonesia (GameDevId).

2.4 *Sistem Operasi Android*

Android adalah sistem operasi untuk telepon seluler yang berbasis Linux. *Android* menyediakan platform terbuka bagi para pengembang buat menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam peranti bergerak. Awalnya, Google Inc. membeli *Android* Inc., pendatang baru yang membuat peranti lunak untuk ponsel. Kemudian untuk mengembangkan *Android*, dibentuklah *Open Handset Alliance*, konsorsium dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia.

Pada saat perilisan perdana Android, 5 November 2007, Android bersama Open Handset Alliance menyatakan mendukung pengembangan standar terbuka pada perangkat seluler. Di lain pihak, Google merilis kode–kode Android dibawah lisensi Apache, sebuah lisensi perangkat lunak dan standar terbuka perangkat seluler.

Di dunia ini terdapat dua jenis distributor sistem operasi Android. Pertama yang mendapat dukungan penuh dari *Google* atau *Google Mail Services* (GMS) dan kedua adalah yang benar–benar bebas distribusinya tanpa dukungan langsung *Google* atau dikenal sebagai *Open Handset Distribution* (OHD).

2.4.1 Perkembangan *Sistem Operasi Android*

Dilihat dari perkembangannya, *Android* sudah sangat pesat berkembang dan sudah memiliki berbagai macam versi *Sistem Operasi* diantaranya:

1. Android 1.1
2. Android 1.5 (Cup Cake)
3. Android 1.6 (Donut)
4. Android 2.0/2.1 (Eclair)
5. Android 2.2 (Froyo/Frozen Yogurt)
6. Android 2.3 (Gingerbread)
7. Android 3.3 (Honeycomb)

Di mulai dari versi 1.1 hingga 2.3 Android di kembangkan sebagai *Sistem Operasi* pada Mobile, hingga pada versi 3.3 Android mulai di kembangkan menjadi Sistem Operasi yang berjalan pada Tablet seperti Motorola Xoom.

2.4.1 *Adobe AIR Android*

Adobe Air berguna untuk menjalankan aplikasi internet secara lokal. Menurut Adobe, aplikasi Adobe Air tidak memerlukan sebuah direktori untuk di-install. Namun demikian, AIR akan dipasang (*pre-installed*) di sebagian besar ponsel Android. Aplikasi Adobe AIR dalam waktu dekat juga akan tersedia di Android Market.Apabila Anda memiliki ponsel Android dan ingin menjajal Air App, maka bisa mengunduhnya di Adobe Labs.

Adobe AIR merupakan komponen utama para *Flash Developer* untuk bisa mengembangkan aplikasi maupun game pada *Sistem Operasi Android.* Namun ada sejumlah persyaratan minimal sebelum meng-install aplikasi Adobe Air, diantaranya sistem operas ponsel haruslah Froyo (Android 2.2) dan memori RAM 256MB serta CPU ARMv7-A.

Perkembangan Adobe AIR yang terbaru saat ini adalah Adobe AIR 2.6, dimana Adobe AIR dapat dijalankan pada *Sistem Operasi Android* 2.2 (*Froyo*), 2.3 (*Gingerbread*), dan 3.0 (*Honeycomb*).

2.5 *Controller Playstation Portable (PSP)*

Controller PSP merupakan sebuah kontroller yang terhubung langsung pada perangkatnya, beda halnya dengan Controller pada Play Station 1, Play Station 2, Play Station 3, ZNES, Nitendo, dan semacamnya yang Kontrollernya terpisah dengan perangkatnya dan di hubungkan oleh kabel.



Gambar 2.1 Playstation Portable. (Damianos, 2011)

2.5.1 Basis *Controller PSP*

Basis *controller* PSP merupakan sebuah *controller* yang berbasis pada *controller* pada PSP, yaitu merupakan *controller* yang terhubung langsung dengan

perangkatnya, kemudian merupakan *controller* yang multi fungsi dalam berbagai macam *event* dalam setiap permainan, misalkan saja pada Game RPG.

Pada Game RPG terdapat banyak sekali event yang berbeda-beda dan dikendalikan oleh satu *controller*. Pada saat pergerakan aktor utama**,** *controller* berfungsi untuk menggerakkan aktor untuk berjalan bahkan pada Game RPG yang memiliki *Action Battle System,* selain menggerakan karakter juga berfungsi untuk melakukan aksi untuk menyerang maupun bertahan. Pada opsi menu utama maupun menu lainnya, *controller* ini berubah fungsi untuk menggerakkan sebuah *pointer* untuk pemilihan menu.



Gambar 2.2 Pergerakan Karakter pada Game Secret of Mana. (SEGA, 2000)

Dilihat pada gambar di atas, *controller* akan berfungsi untuk menggerakan karakter ke atas, bawah, kanan, kiri, maupun melakukan serangan.



Gambar 2.3 Pemilihan Menu *Equipment* pada Grandia 1. (SEGA, 2000)

Dengan *controller* yang sama, event berubah ke dalam tampilan menu peralatan bukan sebagai penggerak karakter.

2.6 Perangkat Lunak

Perangkat lunak (*software*) yang dibutuhkan untuk pengembangan dan implementasi Pembuatan *Game RPG* dengan menggunakan *Action Script 3.0* pada *Sistem Operasi Android* adalah:

1. **Sistem Operasi Windows 7**

   Sebuah Sistem Operasi yang merupakan komponen utama untuk menjalankan beberapa aplikasi yang di butuhkan dan berjalan pada *Sistem Operasi Windows.*

2. **Adobe Flash CS5**

   Merupakan sebuah editor yang pada dasarnya digunakan sebagai editor animasi. Adobe Flash yang pada awalnya di kembangkan oleh *Macromedia. Inc*. Perkembangan Adobe Flash sendiri cukup pesat, dimulai dari Adobe Flash versi CS2, hingga yang terbaru saat ini adalah Adobe Flash CS5. Saat ini Adobe Flash sering pula digunakan untuk membuat sebuah Aplikasi Desktop, maupun Website. Pada Adobe Flash CS5, kita bisa membuat sebuah aplikasi yang berjalan pada *Sistem Operasi Android*, memiliki sebuah fungsi yang bisa menggenerate *ActionScript 3.0 Application* menjadi sebuah aplikasi yang berekstensi (.Apk) sehingga bisa dijalankan pada *Sistem Operasi Android.*

   Adapun kekurangan yang dimilik oleh Adobe Flash adalah File flash relative lebih besar dibandingkan HTML, dan adanya kesulitan untuk meng*update* data-data pada file Flash.

3. **Flash Develop**

   Merupakan sebuah editor untuk mempermudah *scripting ActionScript 3.0*, meskipun juga bisa dilakukan secara langsung pada Adobe Flash CS5, namun Flash Develop memiliki performa yang lebih ringan, karena hanya memiliki fungsi untuk melakukan pengkodean saja yang hanya berisikan text, terutama untuk manajemen file-file *ActionScript 3.0* yang

menggunakan Konsep Pemrograman Beorientasi Object (OOP). File-file *ActionScript 3.0* yang dibangun dengan Flash Develop akan disimpan menjadi sebuah File *ActionScript (.as)* agar bisa digunakan oleh Adobe Flash.

4. **Adobe Photoshop**

   Merupakan sebuah editor gambar maupun foto untuk mempermudah untuk memanipulasi dan membuat sebuah perancangan Interface.

5. **Android SDK 2.2 (Froyo)**

   Android SDK merupakan sebuah aplikasi yang berjalan pada *Sistem Operasi Windows*, yang merupakan sebuah simulator agar Sistem Operasi Android bisa di simulasikan pada *Sistem Operasi Windows*. Karena yang akan digunakan adalah Sistem Operasi Android versi 2.2 (Froyo), maka yang akan digunakan adalah Android SDK 2.2 (Froyo).

2.7 Bahasa Pemrograman

ActionScript merupakan bahasa pemrograman yang bekerja di dalam platform Adobe Flash. Adobe *ActionScript* memang dibangun sebagai cara untuk mengembangkan pemrograman interaktif secara efisien menggunakan platform aplikasi adobe Flash *ActionScript* mulai dari animasi yang sederhana sampai dengan yang kompleks sekalipun, penggunaan data, dan aplikasi interface yang interaktif. Pertama kali diperkenalakn dalam Flash Player 9, *ActionScript* merupakan bahasa pemrograman berorientasi objek didasarkan pada ECMA.

Script-standar yang sama yang menjadi dasar JavaScript-dan memberikan hasil yang luar biasa dalam kinerja dan produktifitas pengembang. *ActionScript* 2.0, versi *ActionScript* yang telah digunakan dalam Flash Player 8 dan sebelumnya, tetap didukung dalam Flash Player 9 dan Flash Player 10. (http://www.adobe.com/devnet/ActionScript/).

2.7.1 Dasar-Dasar Pemrograman *ActionScript* 3.0

1. Penggunaan perintah *Trace*

Perintah Trace digunakan untuk menampilkan informasi pada panel *Output*. Panel *Output* hanya digunakan untuk menguji jalannya program saja, bukan sebagai tampilan utama program. Yang menjaditampilan utama program adalah *Stage* kita. Perintah deklarasi *Trace* mempunyai bentuk umum:

```
trace("Halo Dunia !! ");
```

2. Pendeklarasian *Variabel*

   Variabel-variabel yang akan digunakan dalam Script kita harus dideklarasikan terlebih dahulu. Beberapa tipe variabel yang sering digunakan antara lain: *TextField, Number, String, Array, Sprite, Shape, dan MovieClip*. Kita juga dapat membuat tipe variabel sendiri berupa objek Class untuk aplikasi kita. Perintah deklarasi variabel mempunyai bentuk umum:

   ```
   var namaVariabel:TipeVariabel;
   ```

   atau cara kedua berikut juga memberikan hasil yang sama:

   ```
   var namaVariabel:TipeVariabel=new TipeVariabel();
   ```

   Anda bisa menambahkan program berikut ke dalam kode program anda untuk memunculkan tulisan “Hello World” ke dalam Stage Anda.

   ```
   var textSaya:TextField=new TextField();
   textSaya.text="Hello World";
   stage.addChild(textSaya);
   ```

   Lanjutkan dengan menguji hasil program anda dengan menekan keyboard Ctrl+Enter, maka anda akan melihat tulisan “Hello World” pada *Stage* anda.

3. *Event, Listener, dan Event Handler*

   Semua interaksi user di dalam Flash dikelola oleh "*event-event*" beberapa contoh event adalah: *mouse click, mouse move, input keyboard, loading file,* dan lain-lain. Dalam buku ini akan banyak menggunakan pengaturan event untuk mendisain interaksi user.

   Dalam *ActionScript* , suatu objek bisa mendengarkan (listen) event yang terjadi pada dirinya dengan menggunakan method addEventListener pada objek tersebut. *Method* ini membutuhkan dua argumen, yaitu:

   a. *Event* apa yang akan didengarkan, misalnya *MouseEvent* atau pun *KeyboardEvent*
   b. Fungsi apa yang akan saya kerjakan apabila saya mendegarkan *event*. Dalam contoh kita, baris:

```
tombolMulai.addEventListener(MouseEvent.CLICK,tekanTom
bol);
function tekanTombol(mevt:MouseEvent):void{
// ISI FUNGSI
}
```

      Mempunyai maksud sebagai berikut:

      Tombol dengan nama instance tombolMulai akan mendengarkan event *Mouse Click (MouseEvent.CLICK)*. Apabila tombolMulai di-click dengan mouse, maka fungsi tekanTombol akan dipanggil dan dikerjakan. Fungsi tekanTombol dinamakan Event Handler. Sebuah event handler harus berupa fungsi dengan satu nilai input (misalnya dalam contoh kita adalah *mevt:MouseEvent*) dan tidak mengembalikan nilai apa-apa (yaitu mempunyai tipe keluaran void).

2.7.2 *ActionScript 3.0* pada *Sistem Operasi Android*

Penerapan bahasa pemrograman *ActionScript 3.0* pada *Sistem Operasi Android,* tidak memiliki perbedaan yang terlalu significant. Pada perangkat CPU

interaksi yang digunakan adalah *Mouse*, pada Sistem Operasi Android yang akan digunakan menggunakan interaksi *Touchscreen*. Namun *ActionScript 3.0* membutuhkan tekhnologi Adobe AIR agar bisa berjalan pada *Sistem Operasi Android* seperti yang sudah di uraikan sebelumnya. Untuk memasukkan aplikasi berbasis *ActionScript 3.0* pada *Sistem Operasi Android*, kita membutuhkan sebuah Editor yang dapat menggenerate file *ActionScript 3.0* tersebut menjadi sebuah aplikasi yang berekstensi (.apk) agar bisa dijalankan pada *Sistem Operasi Android*.

BAB III
METODOLOGI

3.1 Metodologi Analisis

Metode Analisis merupakan sebuah metode untuk menguraikan aplikasi *Game RPG* menjadi komponen-komponen untuk di identifikasi dan dievaluasi permasalahannya. Sistem yang di analisis adalah sistem yang berisikan segala sesuatu yang berkaitan dengan aplikasi *game* yang akan dibangun, di mulai dari komponen-komponen *Game RPG,* hingga implementasi *Controller* berbasis *PSP* pada *Game RPG* kemudian di terapkan pada Mobile yang bersistem operasi *Android.* Tahap analisis ini merupakan tahapan yang paling penting dalam program yang dirancang, karena jika terjadi kesalahan dalam tahap ini akan menyebabkan terjadinya kesalahan pada tahap selanjutnya. Karena itu dibutuhkan suatu metode sebagai pedoman dalam mengembangkan sistem yang dibangun.

3.2 Analisis Kebutuhan

Dari metode dan langkah yang dilakukan maka hasil yang diperoleh adalah sebagai berikut.

3.2.1 Analisis Kebutuhan Input

Input adalah suatu bentuk masukan dan berupa data yang telah ada yang dibutuhkan oleh perangkat lunak sehingga dapat mencapai tujuan yang diinginkan masukan dari aplikasi *Game RPG* berbasis *Controller PSP* pada *Android* ini dengan menggunakan layar sentuh (*Touchscreen*).

3.2.2 Analisis Kebutuhan Fungsi dan Kinerja

Fungsi dan kinerja yang dibutuhkan pada aplikasi ini adalah :

1. Membaca input dari layar *Touchscreen*.
2. Pergerakan posisi avatar dan interaksi avatar dengan objek-objek lainnya seperti musuh, *NPC,* dan Mapnya.

3. Pertarungan secara langsung dengan musuh yang akan mempengaruhi beberapa kondisi avatar, seperti naik level, perubahan status avatar, terselesaikannya permasalahan, dan kematian avatar.

3.2.3 Analisis Kebutuhan Output

Output dari aplikasi ini adalah berupa *Game RPG* 2D yang dapat dieksplore langsung sesuai kehendak user. Objek dari aplikasi ini adalah komponen-komponen *Game RPG* yang dibangun dan 1 skenario atau permasalahan yang akan diselesaikan.

3.2.4 Analisis Kebutuhan Perangkat Keras

Perangkat keras digunakan sebagai alat pengolah data yang bekerja secara otomatis mengolah data yang berbentuk teks, gambar, audio dan animasi. Konten yang akan ditampilkan dalam aplikasi ini menggunakan konten 2d, meskipun dibuat dengan mayoritas penggunaan script yang berisikan text dan berkonten 2d, namun untuk mengaplikasikan Adobe Flash CS5 membutuhkan spesifikasi yang cukup tinggi. Spesifikasi komponen yang dibutuhkan akan dibagi menjadi dua bagian yaitu dari sisi pembangunan aplikasi dan dari sisi pengguna.

Adapun spesifikasi komponen perangkat keras yang diperlukan untuk pembangunan aplikasi ini adalah sebagai berikut:

1. Piranti berupa Keyboard dan mouse.
2. Piranti output berupa monitor dengan resolusi minimal 1024x768 dan speaker.
3. Kartu grafis yang mendukung *directX 9*, dengan memori minimal 1 GB.
4. *Processor* minimal *dual core* dan meiliki kecepatan 2 ghz.
5. Memori RAM minimal 1 GB, rekomendasi 2 GB.
6. Harddisk yang memiliki ruang kosong minimal 2 GB.

Adapun spesifikasi komponen perangkat keras yang diperlukan oleh pengguna yang menggunakan aplikasi ini adalah sebagai berikut :

1. Mobile bersistem operasi *Android* dengan versi minimal *Android* 2.2 (*Froyo*)
2. Mobile yang mendukung Adobe AIR 2.5 untuk *Android* 2.2 (*Froyo*), dan Adobe AIR 2.6 untuk *Android* 2.3 (*Gingerbread*).

3. Piranti output berupa *monitor* dengan resolusi minimal 240x320 dengan *TFT capacitive touchscreen*, dan *speaker*.
4. *Procesor* minimal memiliki kecepatan 600 Mhz.
5. *Internal Memory* yang memiliki ruang kosong minimal 20 MB.

3.2.5 Analisis Kebutuhan Perangkat Lunak

Selain perangkat keras, perangkat lunak juga diperlukan dalam pembangunan aplikasi dan dalam menjalankan aplikasi. Perangkat lunak yang dibutuhkan akan dibagi menjadi dua sisi, pada sisi pembangunan aplikasi dan dari sisi pengguna aplikasi.

Perangkat lunak yang dibutuhkan pada pembangunan aplikasi adalah sebagai berikut:

1. *Sistem Operasi*, sistem operasi yang dibutuhkan untuk pembangunan aplikasi adalah antara lain Windows XP, Vista, atau Seven.
2. *Adobe Flash CS5*, Adobe Flash CS5 merupakan aplikasi yang digunakan untuk membangun sebuah animasi, website, maupun game. *Adobe Flash CS5* bukan merupakan Perangkat Lunak *Game Engine* atau bisa di artikan sebagai Editor yang mempermudah dalam pembuatan *Game,* Fitur-fitur yang terdapat dalam *Adobe Flash CS5* ini mayoritas adalah tools-tools standar untuk pembuatan persegi, lingkaran, text, namun di lengkapi dengan tools untuk animasi maupun efek-efeknya. Fungsi utama penggunaan *Adobe Flash CS5* dalam pembuatan *Game RPG* ini adalah untuk meletakkan beberapa file-file yang dibutuhkan seperti gambar dan audio, kemudian juga digunakan untuk mengexport aplikasi flash menjadi aplikasi yang berekstensikan (*.Apk*), sehingga bisa dijalankan pada *Sistem Operasi Android*.
3. *Flash Develop,* Flash Develop merupakan sebuah *Flash Scripting Editor*, biasanya digunakan untuk manajemen *script-script ActionScript 3.0* yang menggunakan metode Pemrograman Berorientasi Object, selain merupakan aplikasi gratis, Flash Develop juga tergolong aplikasi yang ringan untuk dijalankan. File-file *script* yang dibuat menggunakan Flash

Develop ini akan berekstensikan (*.as*) sehingga bisa di panggil dan di akses oleh *Adobe Flash CS5*.

4. *Adobe Photoshop*, digunakan untuk mengedit gambar-gambar yang akan dimasukkan ke dalam aplikasi ini seperti gambar *map, character*, musuh, efek, dan sebagainya.

Sedangkan perangkat lunak yang diperlukan oleh pengguna aplikasi adalah sebagai berikut:

1. *Sistem operasi*, sistem operasi yang dapat digunakan untuk menjalankan aplikasi adalah antara lain *Android 2.2 (Froyo)* yang didukung oleh *Adobe AIR* 2.5, *Android 2.3 (Gingerbread)*, dan *Android 3.0 (honeycomb)* dengan *Adobe AIR* 2.6.

3.3 Perancangan Perangkat Lunak

3.3.1 Metode Perancangan

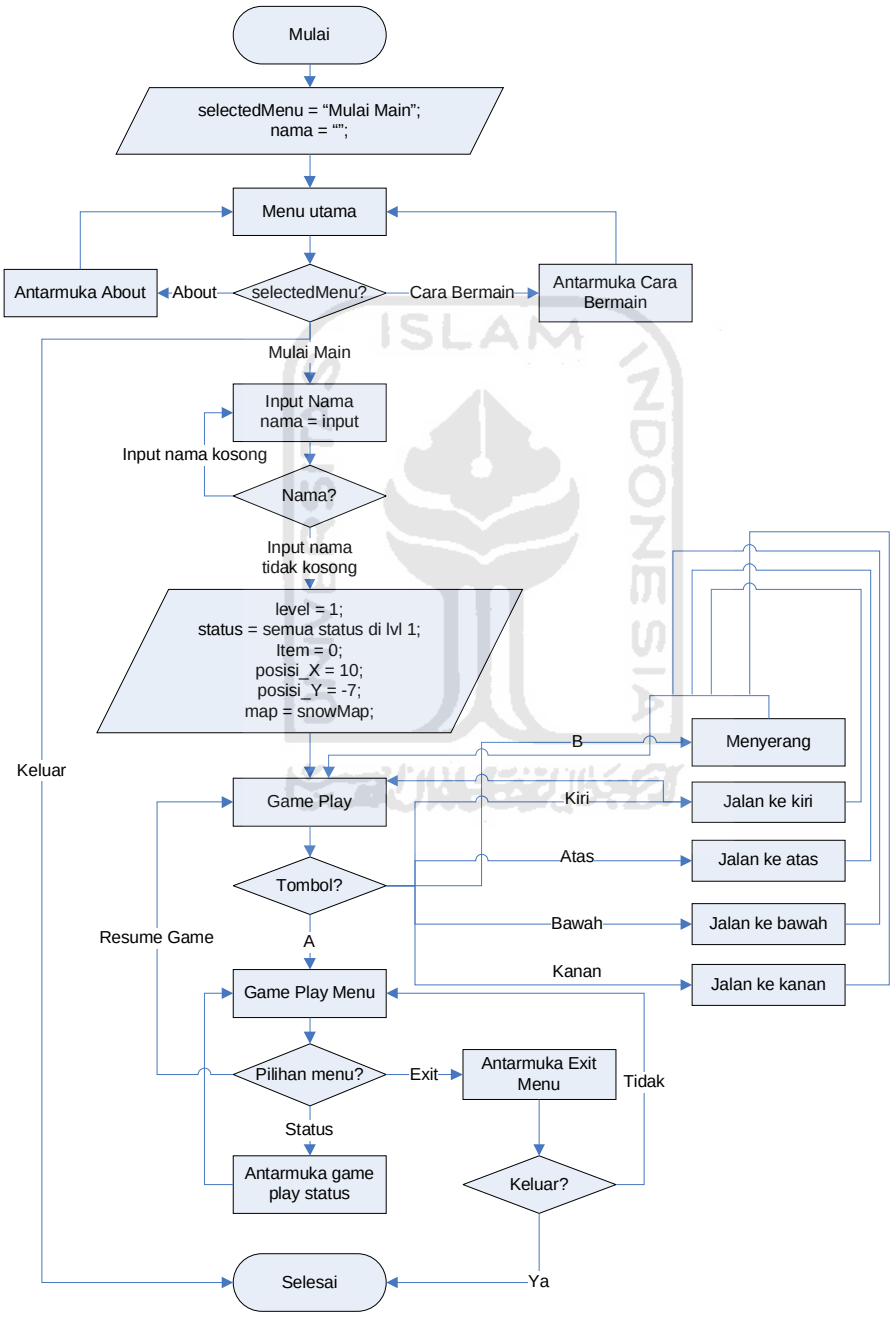
Dalam perancangan sistem ini, proses pembangunan aplikasi ini digambarkan dengan diagram Alir (*Flow Chart*), menunjukan alur yang dilakukan dalam pembangunan program hingga akhirnya menjadi aplikasi ini. Proses jalannya aplikasi ini digambarkan dalam grafik-grafik *flowchart* secara umu dan secara detail untuk memudahkan user memahami penggunaan aplikasi ini.

3.3.2 Hasil Perancangan

Hasil pada tahap perancangan berkaitan erat dengan hasil tahap analisis. Karena pada tahap analisis telah ditemukan metode, fungsi-fungsi yang digunakan, perangkat lunak yang digunakan, serta antarmuka yang diharapkan.

3.4 Perancangan Diagram Alir (Flow Chart)

Diagram alir digunakan untuk menggambarkan langkah-langkah kerja dari sistem yang akan dibuat. Untuk mempermudah, perancangan *flowchart* dibuat berdasarkan masing-masing prosedur.

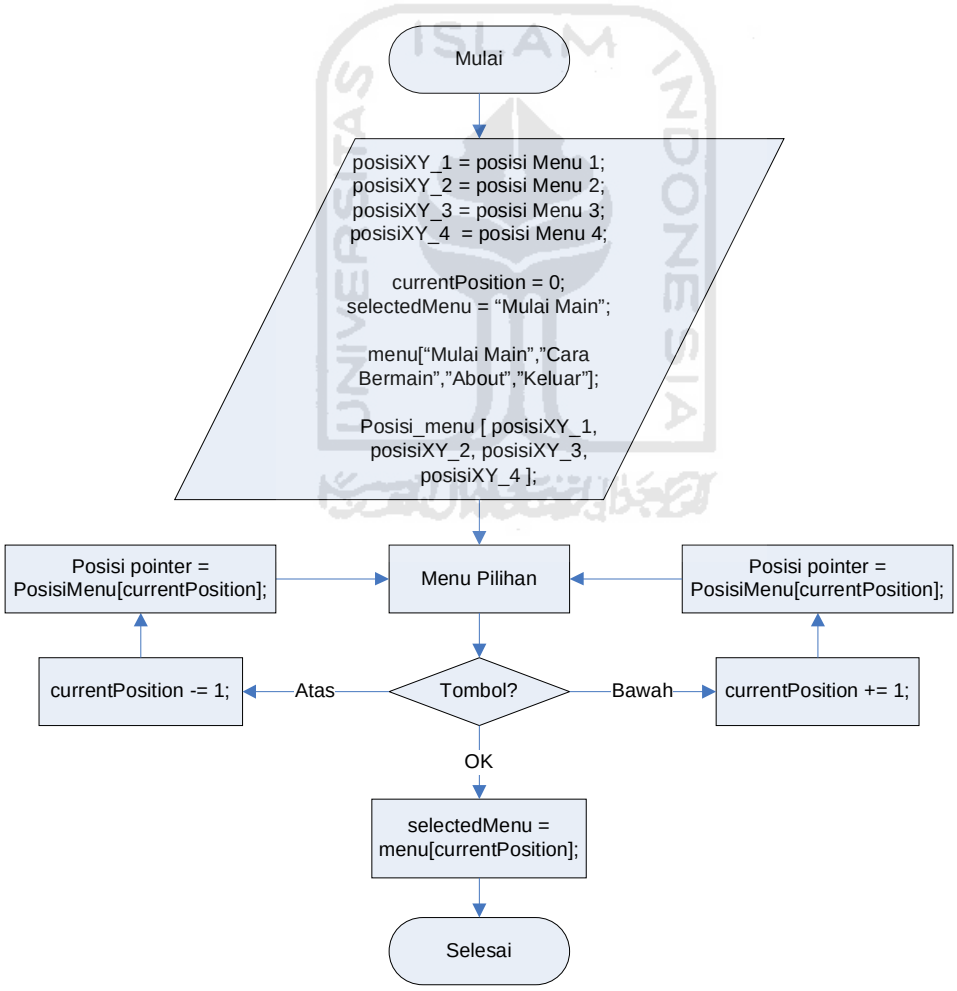


Gambar 3.1 Proses *Game RPG* secara umum

Flowchart di atas menjelaskan tentang proses *Game RPG* yang dibangun secara umum, maksudnya adalah proses yang menunjukkan alur *game* mulai dari awal hingga akhir, namun pada proses tersebut belum terspesifik alur di beberapa fungsinya.

3.4.1 Flow Chart Controller Game Menu

Flowchart ini akan menjabarkan alur dari proses *Game Menu* pada Menu Utama dan *Game Menu* pada *Game Play*. Berikut adalah gambar dari *flowchart*nya :

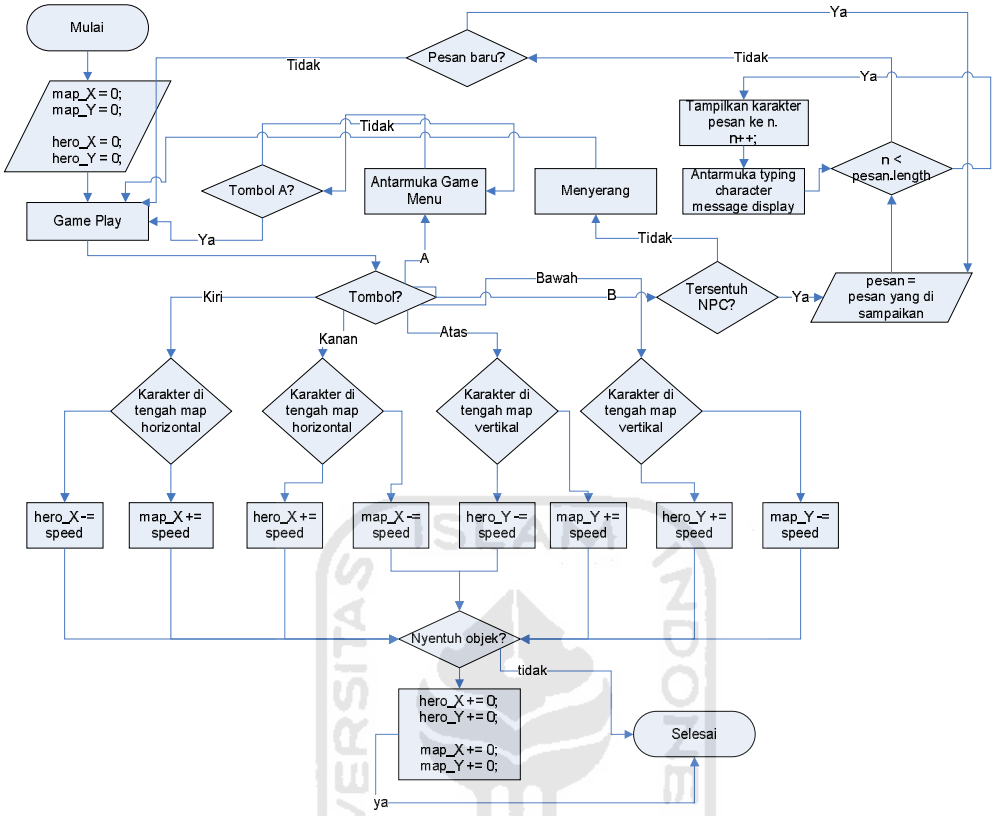


Gambar 3.2 Proses pemilihan menu pada *Game Menu*

PosisiXY_1 hingga PosisiXY_4 merupakan sebuah posisi dari menu-menu yang terdapat dalam antarmuka yang ditampilkan. Kemudian di*inisialisasi* dengan titik koordinat x dan y dari menunya. Sehingga saat *user* menekan tombol kebawah maupun ke atas, perubahan *pointer* akan sesuai dengan koordinat menu-menu yang sudah di kelompokkan menjadi satu tipe data variabel yang memiliki anggota tipe data yang sama yaitu *Array*.

3.4.2 Flow Chart Pergerakan Karakter

Flowchart ini menjabarkan tentang alur proses pergerakkan karakter, karena dalam pembangunan *Game RPG* ini pengkodeannya adalah manual, tidak menggunakan bantuan *Game Engine* yang pada umumnya sudah terdapat fungsi pergerakan. Proses yang terdapat pada *flowchart* ini adalah proses pergerakan karakter itu sendiri, *camera view* yang merupakan sebuah sistem yang dibuat agar karakter tetap berada dalam layar meskipun *map* yang digunakan sangat besar, karena pergerakan karakter memungkinkan karakter berjalan dan menelusuri keseluruhan map. Kemudian *Collision Detection* atau sistem yang mendeteksi adanya tubrukan antara karakter dengan objek lainnya, entah itu NPC, maupun komponen-komponen *map*nya seperti pohon, pagar, kayu bakar, boneka salju, dan sebagainya. Jika karakter menyentuh objek tersebut maka pergerakan karakter dihentikan, seolah-olah objek tersebut bukanlah gambar namun dapat di kenali sebagai objek yang tidak bisa di tembus dan nyata. Kemudian *Character Message Display* yang merupakan sistem pembicaraan secara textual oleh karakter dengan *NPC,* jika karakter menyentuh *NPC* dan melakukan sebuah aksi dengan *NPC* yang di sentuhnya. Berikut adalah gambar dari *flowchart*nya :



Gambar 3.3 Proses pergerakkan karakter pada *Game Play*

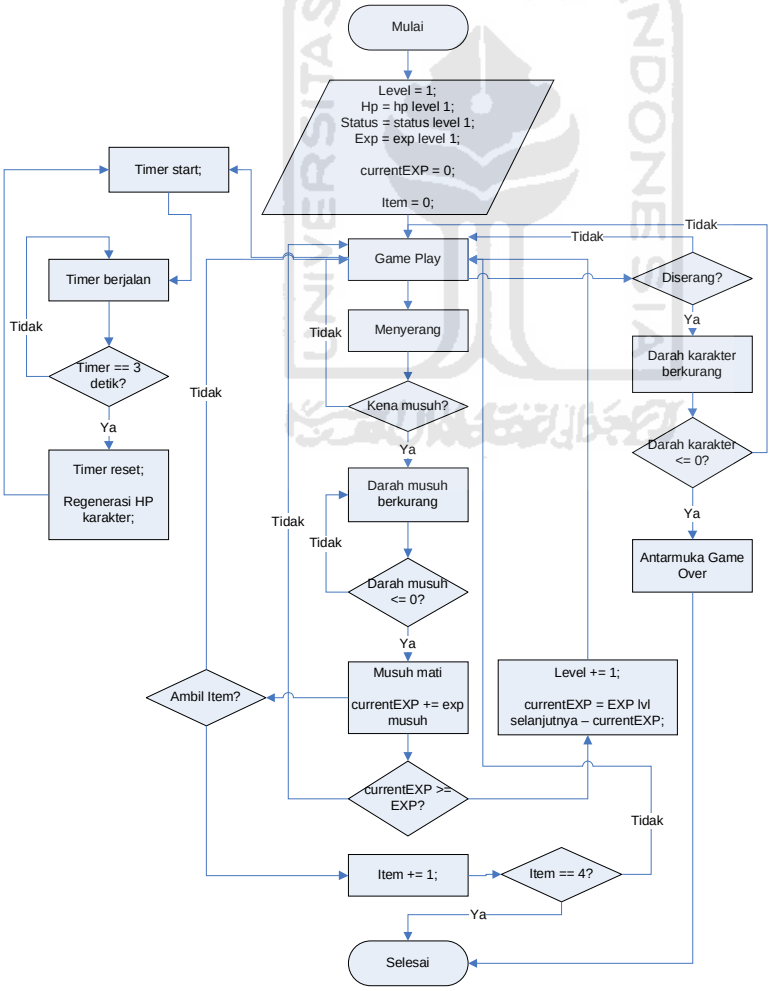
Pada alur proses di atas, variabel koordinat karakter dan *map*nya di inisialisasi dengan 0, kemudian perubahannya sesuai dengan inputan yang akan di terima yaitu berupa tombol. Disaat tombol-tombol itu di tekan, maka posisi koordinat juga akan berubah, perubahan koordinat juga tergantung oleh posisi karakternya, jika karakter berada pada tengah layar secara horisontal maupun secara vertikal maka koordinat yang di gerakkan adalah koordinat dari mapnya, sehingga seolah-olah karakter tetap berjalan. Teknik ini disebut dengan teknik *Scrolling Background*.

Saat tombol B di tekan, maka karakter akan melakukan sebuah aksi tergantung apakah karakter menyentuh *NPC (Non-Player-Character)* atau tidak. Jika karakter menyentuh *NPC* maka akan ditampilkan pembicaraan antara karakter dan *NPC* secara textual, dan akan berakhir hingga tidak terdapat pesan lagi, pesan yang di sampaikan akan ditampilkan per huruf, sehingga seolah-olah

pesan tersebut tampil seperti mesin ketik (Typing Message), dan akan berhenti hingga huruf terakhir. Kemudian jika karakter tidak menyentuh *NPC* maka karakter akan melakukan serangan.

3.4.3 Flow Chart Battle System

Flowchart ini terdiri dari beberapa proses, yang pertama adalah proses regenerasi darah karakter menggunakan *timer*, kemudian perang antara musuh dengan karakter, yang menentukan perubahan darah, level, dan status dari karakter, kemudian terselesaikannya permainan jika *user* berhasil mengumpulkan *item* atau barang yang dibutuhkan. Berikut adalah *flowchart*nya :



Gambar 3.4 Proses *Battle System* dan manajemen *item*

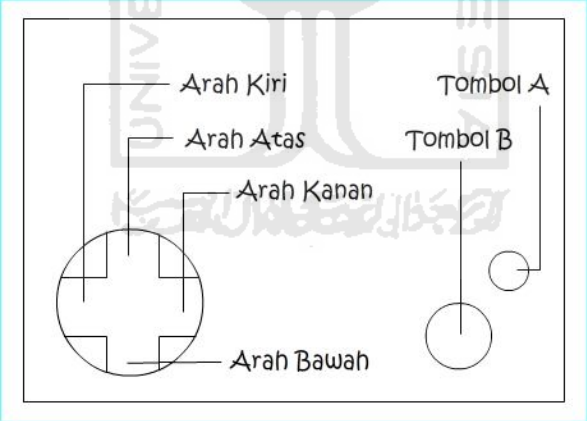
3.5 Perancangan Antarmuka

Antarmuka (*interface*) merupakan bagian dimana terjadi komunikasi antara pengguna dengan sistem. Antarmuka dirancang semudah dan semenarik mungkin agar pengguna dapat dengan mudah menggunakan sistem.

Antarmuka yang sulit dipahami akan membingungkan penggunanya dan menyebabkan sistem tidak dapat digunakan dengan sempurna. Kemudahan pengguna dalam menggunakan sistem dapat dikatakan sebagai keberhasilan antarmuka dalam melakukan komunikasi dengan pengguna.

3.5.1 Antarmuka *Controller*

Antarmuka *Controller* merupakan antarmuka utama untuk mengkontrol segala jenis aksi yang terdapat dalam *Game RPG* yang akan dibangun ini. Karena berbasis *Controller PSP* maka implementasi *Controller*nya di letakkan langsung pada *stage* atau layar, sehingga *user* bisa menggunakannya dengan interaksi *Touchscreen,* yang terlihat seperti sedang bermain pada *PSP*.

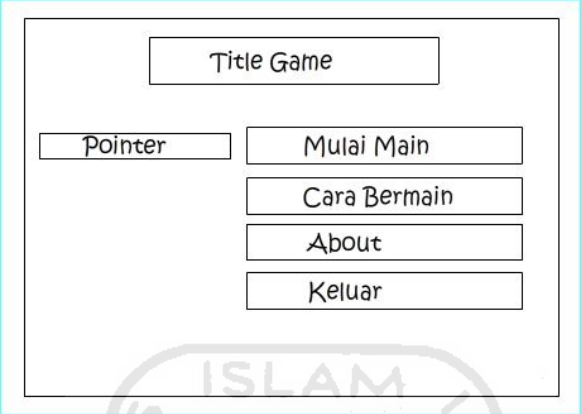


Gambar 3.5 Rancangan Antarmuka *Controller*

3.5.2 Antarmuka Menu Utama (*Main Menu*)

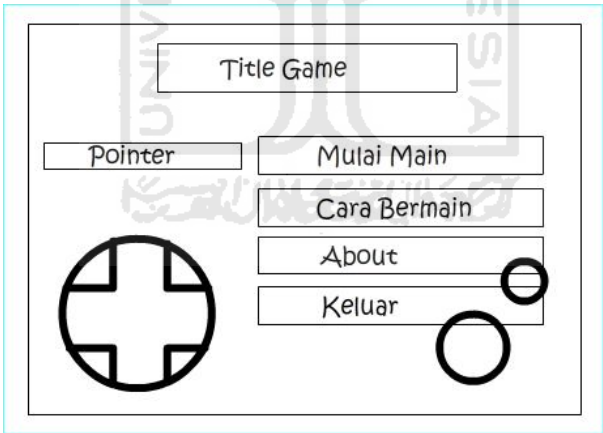
Tampilan halaman menu merupakan tampilan yang pertama kali muncul ketika aplikasi ini dijalankan. Melalui halaman menu ini, pengguna dapat memilih apa yang akan dia lakukan dalam aplikasi ini. Terdapat 4 pilihan yang dapat

dipilih oleh user, yaitu : mulai main, cara bermain, about dan keluar. Tampilan rancangan antarmuka halaman menu dapat dilihat pada gambar 3.6.



Gambar 3.6 Rancangan Antarmuka Menu Utama

Kemudian jika di satukan dengan Controller yang telah dibuat sebelumnya, maka akan tampil seperti pada gambar 3.7.

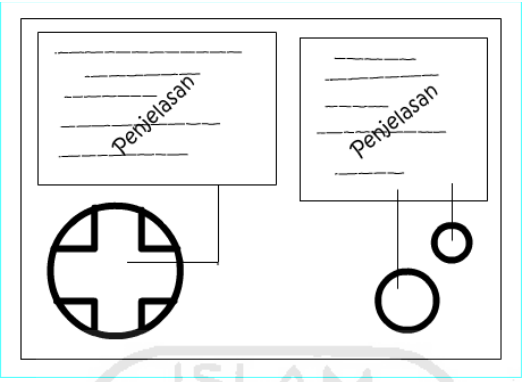


Gambar 3.7 Rancangan Antarmuka Menu Utama dengan Controller

Pada gambar di atas, *Controller* dengan Antarmuka Halaman Utama terlihat seperti menumpuk, Oleh karena itu *Controller* akan dibuat semi transparansi agar tidak menutupi tampilan yang ada di belakangnya.

3.5.3 Antarmuka Menu Cara Bermain

Jika pada Halaman Utama *user* memilih menu "Cara Bermain", maka akan tampil halaman seperti pada gambar 3.8.



Gambar 3.8 Rancangan Antarmuka Cara Bermain

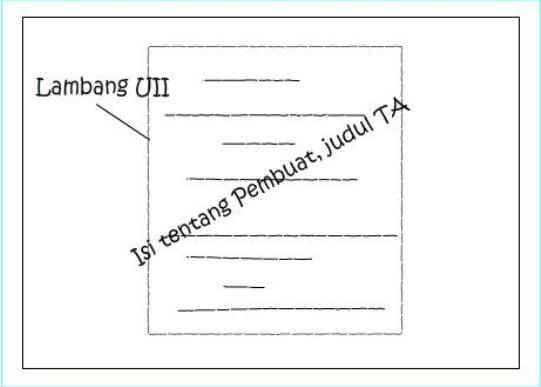
Rancangan Antarmuka Cara Bermain di atas akan dibuat semi transparansi, dan tampil langsung di depan Menu Utama, sehingga tampilan Menu Utama masih terlihat dengan di tutupi oleh Antarmuka Cara Bermain yang semi transparansi, sehingga akan terlihat lebih menarik.

Controller pada gambar di atas berfungsi sebagai berikut :

a. Tombol A : Untuk kembali ke antarmuka *Main Menu*.

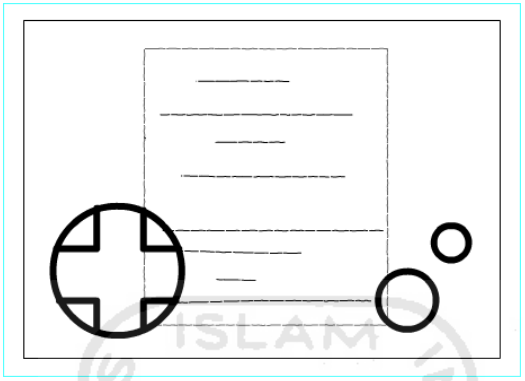
3.5.4 Antarmuka Menu *About*

Berikut ini adalah tampilan rancangan Menu About yang akan tampil jika *user* memilih menu About pada Halaman Utama :



Gambar 3.9 Rancangan Antarmuka *About*

Pada tampilan Antarmuka *About,* akan ditampilkan judul Tugas Akhir, nama dari pemilik Software dan Nomor Induk Mahasiswa. Kemudian jika di kombinasikan dengan *Controller*nya maka tampilannya menjadi seperti pada gambar 3.10.



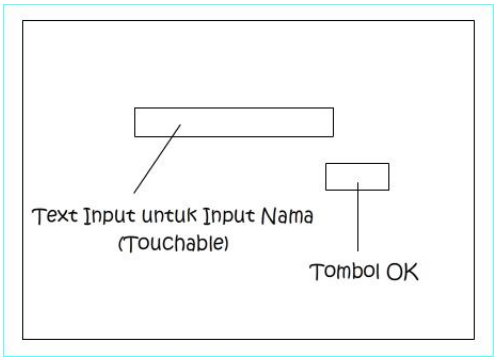
Gambar 3.10 Rancangan Antarmuka *About* dengan *Controller*

Controller pada gambar di atas berfungsi sebagai berikut :

a. Tombol A : Untuk kembali ke Antarmuka *Main Menu*.

3.5.5 Antarmuka Menu *Name Input*

Saat *user* memilih opsi “Mulai Main” pada Halaman Menu Utama, maka *user* akan langsung menuju pada tampilan input nama, dimana pada tampilan ini *user* memberi nama pada karakter yang akan dimainkan. Rancangan tampilan antarmuka *Name Input* dapat dilihat pada gambar 3.11.

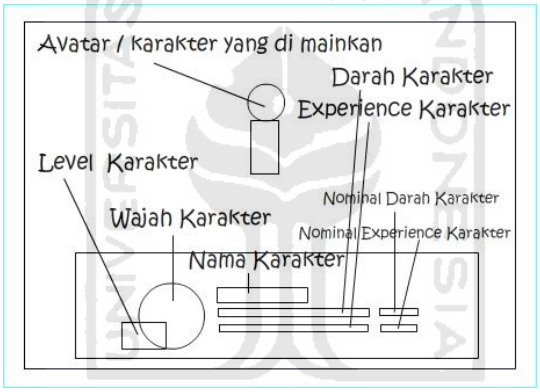


Gambar 3.11 Rancangan Antarmuka *Name Input*

Pada menu *Name Input, user* menyentuh text area yang ada pada tampilan tersebut (*Touchable*), kemudian akan muncul *keyboard* dari *Mobile Android*nya dan *user* mengisikan nama karakter yang diinginkan dan akan di berikan sebuah *filter* untuk mendeteksi apakah nama yang di inputkan masih kosong atau tidak, jika nama yang di inputkan masih kosong maka "tombol OK" tidak akan berfungsi sampai nama yang di inputkan terisi.

3.5.6 Antarmuka *Game Play*

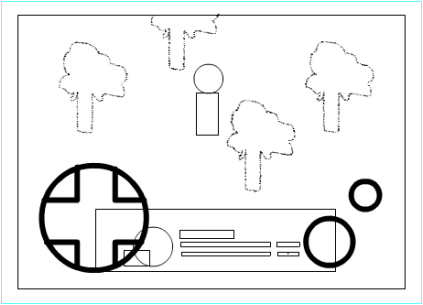
Setelah *user* telah mengisi nama karakter yang akan dimainkan dengan benar, maka *user* langsung masuk dalam permainan yang sesungguhnya, rancangan tampilan *Game Play*nya dapat dilihat pada gambar 3.12.



Gambar 3.12 Rancangan Antarmuka *Game Play*

Pada gambar rancangan Antarmuka *Game Play* di atas terdapat banyak sekali menu-menu dan tampilan-tampilan yang akan ditemui. Status dan parameter karakter akan ditampilkan pada bawah layar yang berisikan nama karakter, wajah karakter, level karakter, darah karakter, *experience* karakter, nominal darah karakter yang merupakan tampilan yang menunjukkan berapakah jumlah maksimal darah karakter saat ini, dan berapakah jumlah darah karakter yang tersisa, dan nominal *experience* karakter dimana berfungsi sebagai informasi kepada *user* untuk mengetahui berapakah jumlah *experience* yang telah didapatkan dan berapakah jumlah *experience* yang dibutuhkan untuk naik ke level berikutnya.

Kemudian berikut ini adalah tampilan *Game Play* jika di kombinasikan dengan *Controller*nya :



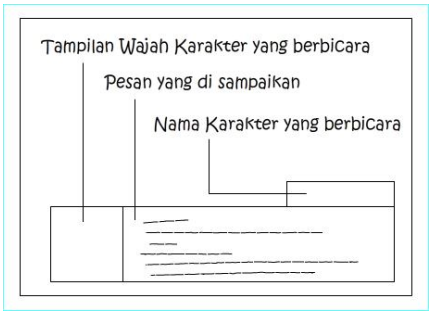
Gambar 3.13 Rancangan Antarmuka *Game Play* dengan *Controller*

Controller pada gambar di atas berfungsi sebagai berikut :

a. Tombol Kanan : Menggerakkan karakter ke kanan.
b. Tombol Kiri : Menggerakkan karakter ke kiri.
c. Tombol Atas : Menggerakkan karakter ke atas.
d. Tombol Bawah : Menggerakkan karakter ke bawah.
e. Tombol A : Masuk ke tampilan Game Menu.
f. Tombol B : Untuk melakukan serangan.

3.5.7 Antarmuka *Character Message Display*

Character Message Display merupakan tampilan yang akan muncul jika karakter melakukan pembicaraan pada karakter yang lain, dalam hal ini adalah *NPC(Non-Player-Character).* Di saat karakter mendekati *NPC* yang terdapat pada *Game Play* dan melakukan aksi terhadap *NPC* tersebut maka akan tampil sebuah antarmuka seperti pada gambar 3.14.

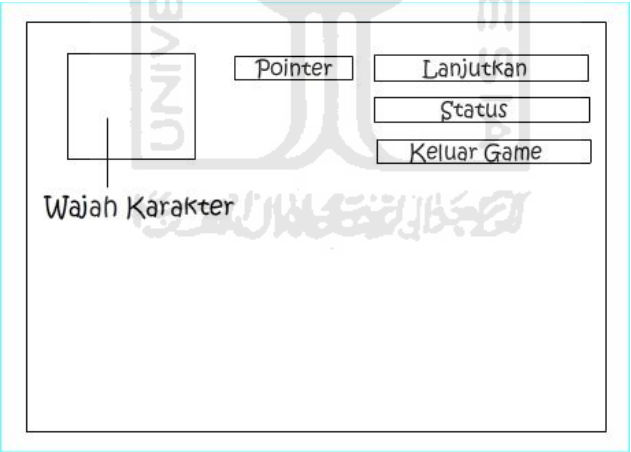


Gambar 3.14 Rancangan Antarmuka *Character Message Display*

Pada tampilan *Character Message Display, Controller* akan *invisible* atau tidak terlihat, agar mempermudah *user* dalam membaca pesan yang di sampaikan. Informasi yang akan ditampilkan adalah tampilan wajah karakter yang berbicara entah itu karakter utama atau *avatar* maupun karakter *NPC,* tergantung siapakah yang memberikan pesan, kemudian nama karakter yang berbicara, dan tampilan pesan yang di sampaikan. Pesan yang ditampilkan akan tampil secara *continue* mulai dari huruf pertama hingga huruf terakhir, sehingga pesan akan terlihat seperti pesan yang di ketikkan (*Typing Text*). Ketika satu pesan habis maka akan dilanjutkan dengan pesan selanjutnya, hingga tidak terdapat pesan lagi. Setelah tidak terdapat pesan lagi, maka tampilan ini akan hilang.

3.5.8 Antarmuka *Game Menu*

Antarmuka *Game Menu* merupakan sebuah tampilan menu yang akan keluar jika *user* menekan tombol A pada saat *Game Play* berlangsung. Rancangan tampilan *Game Menu* dapat dilihat pada gambar 3.15.

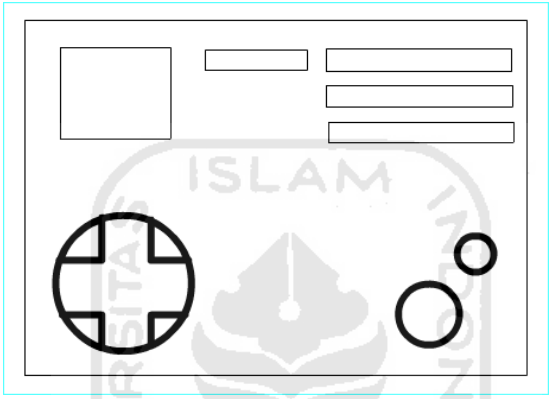


Gambar 3.15 Rancangan Antarmuka *Game Menu*

Pada rancangan antarmuka *Game Menu* terdapat beberapa menu, diantaranya adalah "Lanjutkan", merupakan sebuah menu untuk kembali ke kondisi *Game Play,* karena di saat tampil antarmuka *Game Menu*, *Game Play* akan berada dalam kondisi berhenti sementara hingga *user* keluar dari antarmuka *Game Menu.* Jadi segala jenis pergerakan karakter, musuh, dan segala objek di

sekitarnya akan berhenti (*freeze* atau *pause)*. *Status* merupakan sebuah menu untuk menampilkan beberapa paramater karakter yang akan lebih dijelaskan pada Antarmuka *Game Menu Status*. *Exit Game* merupakan sebuah menu untuk keluar dari permainan, untuk lebih detailnya akan dijelaskan pada antarmuka *Game Menu Exit.*

Kemudian berikut adalah tampilan kombinasi antara Antarmuka *Game Menu* dengan *Controller*nya.



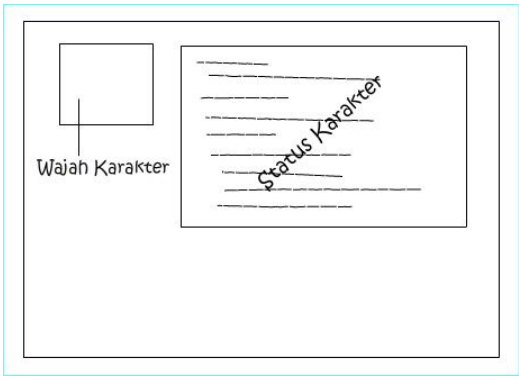
Gambar 3.16 Rancangan Antarmuka *Game Menu* dengan *Controller*

Controller pada gambar di atas berfungsi sebagai berikut :

a. Tombol Atas : Menggerakkan *pointer* ke atas.
b. Tombol Bawah : Menggerakkan *pointer* ke bawah.
c. Tombol A : Untuk kembali ke kondisi *Game Play*.
d. Tombol B : Untuk memilih menu yang diinginkan.

3.5.9 Antarmuka *Game Menu Status*

Saat *user* memilih menu *Status* pada *Game Menu*, maka akan tampil antarmuka *Game Menu Status*, dimana pada halaman ini terdapat beberapa informasi dan parameter-parameter yang di miliki oleh karakter yang kita mainkan. Berikut adalah rancangan tampilan antarmuka *Game Menu Status* :

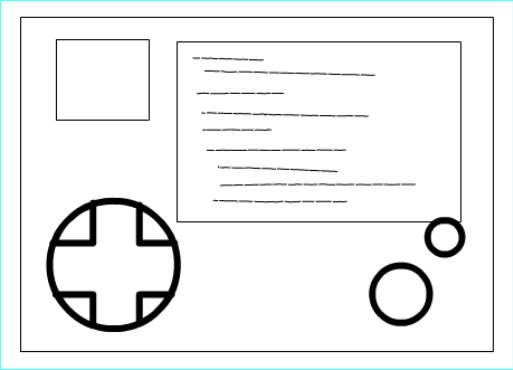


Gambar 3.17 Rancangan Antarmuka *Game Menu Status*

Parameter-parameter dan informasi yang ditampilkan adalah :

a. Wajah Karakter : Menampilkan wajah dari karakter yang kita mainkan.
b. Nama Karakter : Nama dari karakter yang kita mainkan.
c. *Level* Karakter : Merupakan *level* dari karakter yang kita mainkan, semakin level kita tinggi, maka semakin kuat karakter yang kita mainkan. *Level* ini bisa di naikkan dengan mengumpulkan *point Experience.*
d. *HP (Hit Point)* : Darah dari karakter yang kita mainkan.
e. *EXP (Experience)* : point atau jumlah yang harus di kumpulkan untuk naik ke level selanjutnya, didapatkan dengan cara membunuh musuh yang ada. Jumlah *Experience* akan meningkat tergantung dengan *level* karakter.
f. Kekuatan : Kekuatan pukulan dari karakter dengan batas minimum dan batas maksimumnya. *Attack* akan meningkat jika *level* dari karakter juga meningkat.
g. Pertahanan : Pertahanan dari karakter yang kita mainkan, semakin *level* kita tinggi maka pertahanan yang di miliki juga semakin tinggi.
h. Barang terkumpul :Merupakan jumlah dari item-item yang telah terkumpul. Jika item yang telah terkumpul sudah memenuhi syarat, maka permainanpun berakhir.

Berikut adalah rancangan tampilan antarmuka Game Menu Status, yang di kombinasikan dengan Controller :



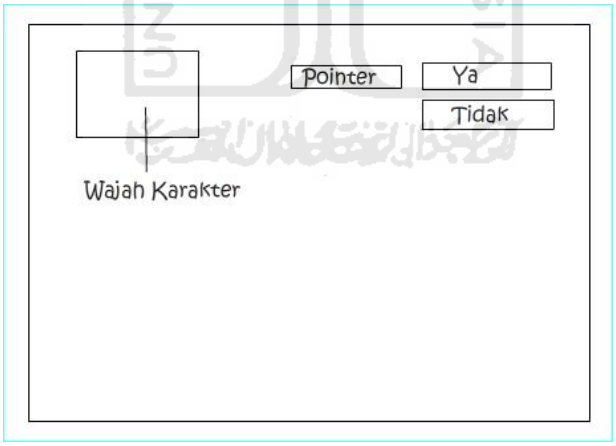
Gambar 3.18 Rancangan Antarmuka *Game Menu Status* dengan *Controller*

Controller pada gambar di atas berfungsi sebagai berikut :

a. Tombol A : Untuk kembali ke antarmuka *Game Menu*.

3.5.10 Antarmuka *Game Menu Exit*

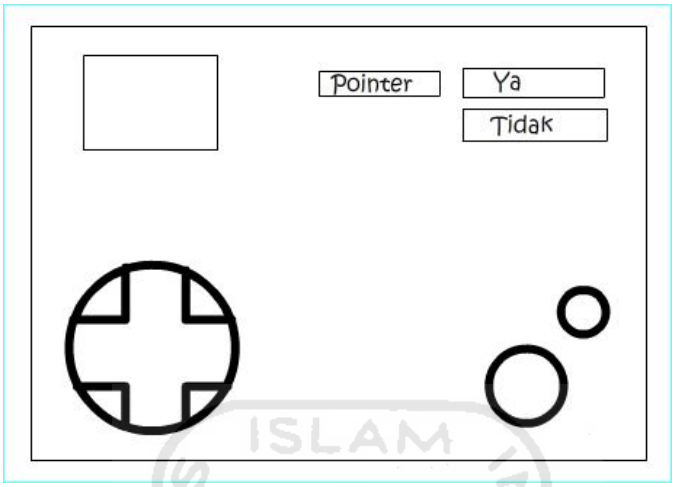
Ketika *user* memilih menu "*Keluar Game*" pada *Game Menu*, maka akan tampil antarmuka *Game Menu Exit*. Rancangan dari tampilan *Game Menu Exit* dapat dilihat pada gambar 3.19.



Gambar 3.19 Rancangan Antarmuka *Game Menu Exit*

Pada antarmuka *Game Menu Exit* terdapat 2 pilihan yaitu "Ya" dan "Tidak". Jika *user* memilih menu "Ya", maka akan langsung keluar dari permainan dan secara bersamaan menutup aplikasi yang dimainkan. Jika *user* memilih menu "Tidak", maka akan kembali ke antarmuka *Game Menu*.

Berikut adalah rancangan tampilan *Game Menu Exit* yang di kombinasikan dengan *Controller* :



Gambar 3.20 Rancangan Antarmuka *Game Menu Exit* dengan *Controller*

Controller pada gambar di atas berfungsi sebagai berikut :

a. Tombol Atas : Menggerakkan *pointer* ke atas.
b. Tombol Bawah : Menggerakkan *pointer* ke bawah.
c. Tombol A : Untuk kembali ke antarmuka *Game Menu*.
d. Tombol B : Untuk memilih menu yang diinginkan.

3.5.11 Antarmuka *Game Over*

Game Over merupakan sebuah kondisi dimana *user* gagal dalam permainan, dalam *Game RPG* ini *user* akan gagal jika terbunuh oleh *monster-monster* yang ada. Jika darah karakter yang kita mainkan kurang dari atau sama dengan 0, maka antarmuka *Game Over* akan muncul. Rancangan tampilan *Game Over* dapat dilihat pada gambar 3.21.



Gambar 3.21 Rancangan Antarmuka *Game Over*

Pada tampilan *Game Over*, semua objek termasuk *Controller*nya akan tertutupi dengan antarmuka *Game Over* ini secara perlahan dengan menggunakan perubahan nilai transparansi, mulai dari transparansi yang sedikit hingga tidak transparansi. Dengan menyentuh sembarang layar pada antarmuka *Game Over* ini, maka tampilan antarmuka *Game Over* ini akan hilang secara perlahan dan akan kembali ke antarmuka *Main Menu* (Menu Utama).

BAB IV
HASIL DAN PEMBAHASAN

4.1 Batasan Implementasi

Hasil dan pembahasan mengutarakan tentang implementasi perangkat lunak yang meliputi batasan implementasi, *story board*, implementasi komponen-komponen *Game RPG* yang dibangun, implementasi antarmuka, pengujian program dan pengujian menu, dan analisis kinerja program.

4.2 *Game RPG* Berbasis *Controller PSP* pada *Sistem Operasi Android*

Game RPG ini merupakan sebuah game yang berjalan pada Sistem Operasi Mobile Android. Nama Game ini adalah "Yotuhir" yang merupakan singkatan dari "Yogie Tugas Akhir", *Game RPG* yang dibangun merupakan sebuah demo dan memiliki cerita yang sangat singkat, karena dalam pembangunan dan dilihat dari batasan masalahnya penulis tidak membangun sebuah aplikasi *Game RPG* dari segi skenario atau jalan ceritanya, namun dari segi bagaimanakah pembuatan *Game RPG* beserta komponen-komponennya yang sama sekali tidak menggunakan bantuan library ataupun *Game Engine*. Karena *Game* yang dibuat menggunakan *Game Engine* sangatlah mudah, pengguna sudah di suguhi berbagai macam fitur yang tinggal digunakan saja.

Game RPG ini memiliki alur cerita yang bersifat dongeng atau legenda, hal yang bukan merupakan kejadian yang benar-benar terjadi. Skenario seperti ini sering ditemukan di berbagai *Game RPG* agar pengguna memiliki imajinasi yang bervariasi, memiliki wawasan yang luas dalam dunia *Game*, dan kreatif dalam menyelesaikan permasalahan yang ada. Dalam *Game RPG* ini tidak terdapat *Load/Save System* yang berfungsi untuk penyimpanan maupun pengambilan data.

4.2.1 Alur Cerita

Dalam *Game RPG* ini pengguna akan di berikan satu alur cerita, yaitu mengumpulkan 4 batu elemen api, air, angin dan tanah. Keempat batu elemen ini akan digunakan untuk menetralkan sebuah baru Kristal Hitam yang jatuh dari

langit, batu kristal hitam itu merupakan sebuah penghubung antara dunia monster dengan dunia bumi, selama kristal itu masih berwarna hitam maka monster-monster akan bisa terus masuk ke Bumi. Sehingga pengguna di tugaskan untuk mendapatkan keempat batu elemen itu untuk membersihkan kembali batu kristal hitam itu.

Penggunaan skenario atau alur cerita yang bersifat dongeng atau legenda ini ditujukan untuk implementasi komponen-komponen yang dibangun dalam *Game RPG* ini, sehingga pengguna juga akan mengeksplor keseluruhan komponen-komponen yang dibangun dalam *Game RPG* ini.

4.3 Implementasi Komponen-komponen *Game RPG*

Komponen-komponen yang dibangun merupakan implementasi dari rancangan fitur yang akan dibangun seperti pergerakan karakter dan aksi, *NPC, Map,* sistem *inventory,* sistem pertarungan, dan kecerdasan dasar musuh dalam bertarung.

Dalam pembuatan komponen-komponennya sprite atau gambar dari *Game RPG* yang dibangun mayoritas menggunakan Sprite dari RPG Maker XP 1.02 dan RMVX, yang merupakan aplikasi *Game Engine*. Kemudian suara yang digunakan mulai dari "*Sound Effect*" dan "*Background Music*" menggunakan suara yang terdapat pada beberapa Game seperti Seal Online, Ragnarok Online, Xian Online, ElSword Online, Xenosaga, dan dari beberapa *Game Engine* seperti RPG Maker XP 1.02 dan RMVX.

4.3.1 Pergerakan Karakter dan Aksi

Membangun Pergerakan dan aksi karakter menggunakan teknik *Image Blitting* dan teknik animasi *Pixel Movement. Image Blitting* merupakan sebuah teknik pengambilan perbagian dari sebuah gambar sesuai dengan ukuran lebar dan tinggi yang diinginkan. Gambar karakter utama dalam kondisi menyerang dapat dilihat pada gambar 4.1.



Gambar 4.1 Sprite karakter utama kondisi menyerang

Secara keseluruhan gambar di atas merupakan sebuah file gambar yang berekstensi (.png) dan gambar ini didapatkan dari RPG Maker XP 1.02. Penggunaan file gambar berekstensi (.png) adalah untuk gambar-gambar yang mendukung sifat transparansi. Dalam satu gambar terdapat 16 jenis gambar yang berbeda-beda, dengan menggunakan teknik *Image Blitting* kita bisa hanya mengambil 1 dari 16 posisi gambar yang terdapat dalam 1 file gambar itu. Penggunaan teknik animasi *Pixel Movement*, merupakan sebuah teknik dimana peletakan *Image Blitting* dibuat dinamis dan bergerak sesuai ukuran pixel yang diinginkan, sehingga yang di hasilkan adalah gambar karakter utama yang bergerak. Berikut adalah penggalan *ActionScript* pada *Image Blitting* dan *Pixel Movement :*

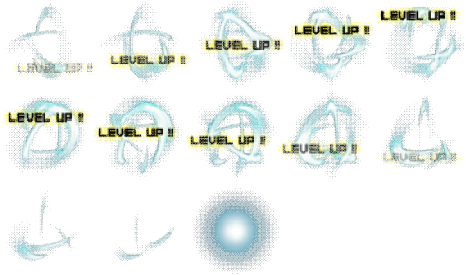
```
if (hinoAttackAnimationCounter == hinoAttackAnimation) {
                    hinoAttackAnimationCounter = 0;
            }

hinoHeroRect.x = hinoHeroWidth*hinoAttackAnimationCounter;
hinoHeroRect.y = hinoHeroHeight * hinoHeroDirectionData;

hinoAttackAnimationCounter++;

hinoHeroPoint = new Point(hinoHeroX, hinoHeroY);
```

Pergeseran pixel sesuai dengan lebar dan panjang dari gambar karakter yang diinginkan berdasarkan perkalian antara lebar karakter dengan `hinoAttackAnimationCounter`. Kemudian Sprite atau gambar yang digunakan untuk animasi naik level karakter dapat dilihat pada gambar 4.2.



Gambar 4.2 Sprite animasi naik level

Kemudian berikut adalah potongan *ActionScript* untuk pergerakan Karakternya :

```
            if (direction == "UP") {
      HinoMain.hinoHeroStatus = "MOVE";
      HinoMain.hinoHeroDirectionData = 3;

            } else if (direction == "DOWN") {
      HinoMain.hinoHeroStatus = "MOVE";
      HinoMain.hinoHeroDirectionData = 0;

            } else if (direction == "LEFT") {
      HinoMain.hinoHeroStatus = "MOVE";
      HinoMain.hinoHeroDirectionData = 1;

            } else if (direction == "RIGHT") {
      HinoMain.hinoHeroStatus = "MOVE";
      HinoMain.hinoHeroDirectionData = 2;

            } else if (direction == "ACCEPT") {
      if (HinoMain.hinoNpcChat == false) {
      HinoMain.hinoHeroStatus = "ATTACK";
                                    }
```

Pada *ActionScript* di atas terdapat `HinoMain.hinoHeroDirectionData` yang berfungsi untuk menentukan kemanakah arah wajah karakter sesuai dengan `direction,` yang dimana berfungsi sebagai input arah terhadap karakternya.

Kemudian berikut adalah *ActionScript* pergerakan karakter sesuai dengan kecepatan yang sudah ditentukan :

```
if (direction == "LEFT") {
                  //hinoCollisionDetector = false;
                  hinoNpcChat = false;
                  hinoHeroMovementCounter++;
                  hinoXVel = hinoSpeedVelocity;
                  };
}
```

Kita ambil contoh pergerakan ke arah kiri, `hinoHeroMovementCounter` digunakan untuk pergerakan pixel karakter ke arah kanan, merupakan sebuah animasi karakternya. Kemudian `hinoSpeedVelocity` merupakan kecepatan karakter yang sudah di definisikan.

4.3.2 *NPC (Non-Player Character)*

Membangun sebuah karakter yang tidak dikendalikan oleh player dan bisa berinteraksi dengan player. Sprite gambar *NPC* juga diambil dari *Game Engine* RPG Maker XP 1.02. Sprite gambar *NPC* dapat dilihat pada gambar 4.3.



Gambar 4.3 Sprite *Non-Player-Character*

Berikut adalah *ActionScript* pergerakan *NPC*nya :

```
if(HinoNPC.hinoNPC_AnimationCounter==HinoNPC.hinoNPC_MaxAnim
ation) {
      HinoNPC.hinoNPC_AnimationCounter = 0;
      }

HinoNPC.cuteMage_NPC_Rect = new
Rectangle(HinoNPC.hinoNPC_AnimationCounter*HinoNPC.hinoNPC_W
idth,HinoNPC.hinoNPC_Dir*HinoNPC.hinoNPC_Height,HinoNPC.hino
NPC_Width,HinoNPC.hinoNPC_Height);

HinoNPC.hinoNPC_AnimationCounter++;
```

Kemudian untuk interaksinya terhadap pengguna adalah saat pengguna menyentuh *NPC*, dan menekan tombol B :

```
if(hinoHeroSpriteSensorUp.hitTestObject(hinoNPCSpriteSensor)
&& hinoHeroDirectionData == 3) { //hero hadap atas

      HinoNPC.hinoNPC_Dir = 0;
            }

chatStartFunction(null);
```

`chatStartFunction(null)` merupakan sebuah fungsi untuk menampilkan pembicaraan secara tekstual antara karakter dengan *NPC*nya yang akan membangkitkan fitur *Typing Character Message Display.*

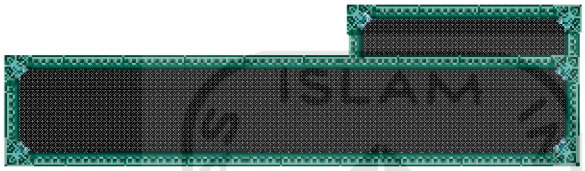
4.3.3 *Typing Character Message Display*

Membangun salah satu komponen utama dalam game RPG, yaitu pembicaraan pada setiap karakter, entah itu katakter utama maupun tidak (NPC). Gambar sprite wajah karakter yang digunakan, dapat dilihat pada gambar 4.4.



Gambar 4.4 Sprite wajah karakter

Sprite gambar di atas diambil dari gambar yang terdapat pada RMVX. Gambar tampilan belakang Typing Character Message Display dapat diihat pada gambar 4.5.



Gambar 4.5 Sprite *Background Typing Character Message Display.*

Kemudian berikut adalah potongan *ActionScript* untuk *Typing Character Message Display* :

```
hinoWindowSkin.hinoWinName.text = charName.toString();
hinoTypeSpeedTimer.start();

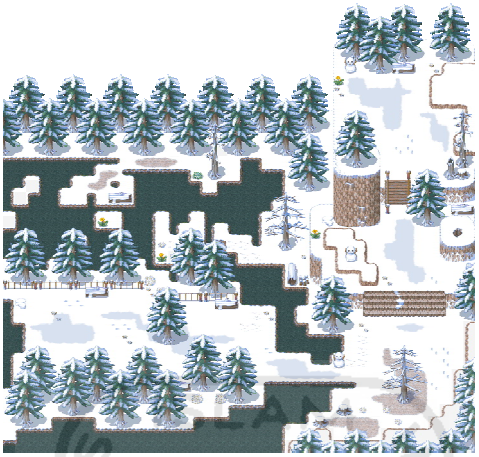
//fungsi yang digunakan untuk menampilkan pesan perhuruf
function typeText(event:TimerEvent):void {

hinoWindowSkin.hinoWinText.appendText(hinoTextTemp.charAt(hinoText
Counter));
      hinoTextCounter++;
}
```

4.3.4 *Map*

Membangun sebuah peta untuk karakter dan penempatan keseluruhan objek berdasarkan dengan file gambar yang digunakan, ukuran dari map ini adalah sebesar 1000x1000 pixel, peletakan dan pergerakannya juga menggunakan

teknik *Image Blitting* dan *Pixel Movement*. Pembuatan *Map* ini menggunakan gambar map dari RPG Maker XP 1.02.



Gambar 4.6 Sprite Snow Map

Dalam *Game RPG* ini terdapat 3 layer, gambar di atas merupakan layer paling bawah, yang dimana penempatan karakter dan objek lainnya terdapat pada layer kedua. Gambar *Map* untuk layer teratas yaitu layer ketiga, dapat dilihat pada gambar 4.7:



Gambar 4.7 Sprite Snow Map layer atas

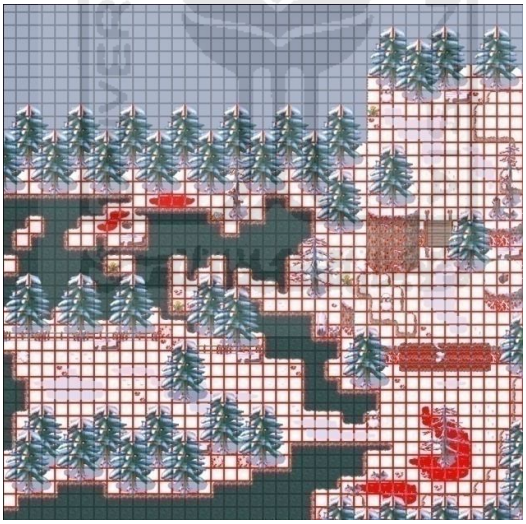
Gambar *Map* diatas diambil dari *Game Engine* RPG Maker XP 1.02. Gambar di atas digunakan untuk beberapa bagian map seperti pohon yang

seharusnya terletak menutupi karakter. 2 gambar di atas yang dijadikan satu rangkap akan membuat 2 gambar tersebut seperti satu dan terlihat hidup. Karena objek yang berada dalam *Map* tersebut berfungsi sebagaimana mestinya.

Untuk interaksi karakter dengan semua objek yang ada dalam *Map* itu digunakan teknik *Collision Detection*, yang dimana karakter akan mengenali manakah objek-objek yang tidak bisa dilewati seperti kayu, pohon, sungai, batu, maupun *NPC*nya.

4.3.5 *Collision Detection*

Membangun sebuah kolisi atau tubrukan antara karakter utama dengan objek-objek yang ada di sekitarnya, seperti yang sudah dijelaskan pada sub bab Map di atas. Ilustrasi gambar Map untuk peletakan objek-objek dan kolisinya, dapat dilihat pada gambar 4.8.



Gambar 4.8 Sprite Snow Map Block

Pada gambar di atas, terdapat garis-garis merah yang merupakan pembagian block pixel. Tiap block pixel berukuran 25 pixel. Alasan mengapa *Map* yang digunakan berukuran 1000x1000 pixel agar memudahkan dalam meletakkan kolisi dan beberapa objek dalam *Game RPG* ini. Kolisi disini

merupakan sebuah objek yang tidak terlihat, dimana blok yang terdapat kolisi akan di kenali sebagai lokasi yang tidak bisa dilewati oleh karakter utama.

Block pixel direpresentasikan sebagai Array 2 dimensi yang ordo arraynya adalah 40x40, dan setiap index menyimpan satu objek yang memiliki ukuran 25x25 pixel. Sehingga jika dikombinasikan lebar dan tinggi pixel setiap blok dengan panjang dari array tersebut maka lebar keseluruhan objek dari array adalah 40x25 yaitu 1000 pixel, demikian juga dengan tingginya. Perhitungan tersebut sama dengan panjang dan lebar dari *Map* yang dibuat. Kemudian untuk perulangan pada baris terakhir digunakan untuk peletakan objeknya berdasarkan index berapakah terdapat objek yang akan diletakkan, dikalikan dengan lebar bloknya. Misalkan dalam array itu kita menginisialisasi satu index yang bernilai "X", yang dimana nilai tersebut merupakan objek-objek yang memiliki kolisi atau objek yang akan di kenali sebagai objek yang tidak dapat dilewati oleh Karakter Utama. Misalkan saja nilai "X" terdapat pada index 20,33, berarti peletakan objek kolisi itu berada pada koordinat 500,825, dengan index pertama sebagai koordinat x, dan index kedua sebagai koordinat y.

4.3.6 *Camera View*

Membagun sebuah kamera sisi untuk menghindari adanya map yang lebih besar dari ukuran layarnya, maksudnya adalah agar karakter yang dimainkan tetap berada dalam satu layar ketika pengguna ingin menjelajahi keseluruhan *Map*nya. Misalkan saja ukuran layarnya adalah 240x320 dan ukuran *Map*nya adalah 1000x1000. Berikut adalah potongan *ActionScript* untuk arah ke kanan dengan menggunakan teknik *Camera View* :

```
if (direction == "RIGHT") {
            hinoHeroMovementCounter++;
if ((hinoHeroX >= hinoHalfStageWidth)&&(HinoMapSprite.hinoCameraX
>= HinoMap_Snow.hinoMapSnow_MinCamera_X)) {
                  hinoXVel = 0;
                  hinoXCamVel = -HinoMapSprite.hinoCameraVelocity;
                  } else {
```

```
                hinoXCamVel = 0;
                hinoXVel = hinoSpeedVelocity;
                }
        }
        }
```

Camera View disini menggunakan 2 macam pergerakkan, yang pertama adalah pergerakkan karakter utama, pada potongan *ActionScript* di atas kita definisikan `hinoXVel = hinoSpeedVelocity` dengan `hinoXVel` sebagai koordinat x dari karakter, dan `hinoSpeedVelocity` sebagai kecepatan karakter dengan kondisi posisi karakter tidak pada tengah layar dan posisi kamera masih lebih kecil atau sama dengan posisi *Map*nya, sebaliknya jika posisi karakter berada di tengah layar dan posisi kamera lebih besar dari `HinoMap_Snow.hinoMapSnow_MinCamera_X` yang merupakan batas kamera mulai bergerak, maka pergerakkan karakter dihentikan dan pergerakan *Map* yang dijalankan dengan kecepatan yang sama dengan karakter utama dan arah yang berlawanan, sehingga seakan karakter masih berjalan seperti biasanya.

4.3.7 Sistem *Inventory*

Membangun sistem *inventory* untuk menampung barang-barang yang ditemukan. Dalam *Game RPG* ini, *inventory* akan digunakan untuk menampung 4 batu elemental yaitu batu api, air, angin, dan tanah. Keempat batu elemen ini akan digunakan untuk menetralkan atau membersihkan batu kristal hitam agar permainan selesai.



Gambar 4.9 Tampilan Sistem *Inventory* pada *Game Play*

Pada sistem *inventory* menggunakan sistem yang dinamis, barang yang didapatkan terlebih dahulu akan terletak pada *block inventory* lebih awal. *Block*

Inventory merupakan sebuah *block* untuk menampung barang-barang yang didapatkan, dalam *Game RPG* ini terdapat 4 *block inventory*.

4.3.8 Sistem Pertarungan

Membangun Sistem pertarungan antar karakter dengan musuh. Dalam Game RPG ini menggunakan teknik pertarungan *Action Battle System*, yang merupakan sebuah sistem pertarungan dimana karakter akan langsung bertemu dengan musuhnya dan pertarungan terjadi pada *Map* yang sama dan akan selesai jika salah satu dari musuh maupun karakter utama kehabisan darah.

Berikut adalah potongan *ActionScript Battle System* untuk karakter :

```
if (direction == "ACCEPT") {
                  if (hinoNpcChat == false) {
                  if (slashEnabled) {
                  slashEnabled = false;
                  hinoSlashDelay.start();
                  HinoSceneEvent.hinoSlash.play();

                  HinoMain.hinoHeroStatus = "ATTACK";
                  }
                  }
}
```

Pada potongan *ActionScript* di atas terjadi sebuah kondisi dimana ketika pengguna menekan tombol B / *Accept Button,* dan karakter yang kita mainkan tidak menyentuh *NPC (Non-Player Character)* maka karakter akan melakukan serangan. Serangan itu ditandai dengan perubahan variabel `HinoMain.hinoHeroStatus = "ATTACK";`, kemudian akan di cek ketika karakter statusnya adalah “ATTACK”, maka sprite karakter diubah menjadi sprite karakter yang melakukan serangan. Saat karakter melakukan serangan juga terdapat suara tebasan pedang agar terlihat lebih menarik.

4.3.9 Kecerdasan Dasar untuk Musuh dalam Bertarung

Membangun musuh yang bisa berjalan, bergerak dan melakukan aksi sendiri. Salah satu sprite dari musuhnya, dapat dilihat pada gambar 4.10.



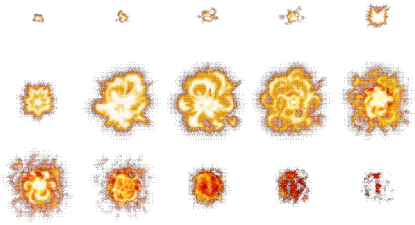
Gambar 4.10 Sprite musuh (Devil Terbang) kondisi diam

Gambat Sprite musuh yang digunakan berasal dari Game XAS Hero. Gambar sprite tebasan ketika serangan musuh atau karakter mengenai lawannya, dapat dilihat pada gambar 4.11.



Gambar 4.11 Sprite animasi serangan

Kemudian dibawah ini merupakan gambar sprite ledakan musuh ketika kehabisan darah dan hancur :



Gambar 4.12 Sprite animasi ledakan

Dalam *Game RPG* ini musuh memiliki beberapa kecerdasan sendiri mulai dari pergerakkan musuh mendekati karakter, musuh akan bergerak mendekati karakter jika jarak antara musuh tersebut dengan karakter lebih kecil dari jarak yang sudah ditentukan. Kemudian arah dari musuhnya, musuh akan terus menghadap ke arah karakter utama dengan parameter perbedaan jarak antara koordinat x dan koordinat y. Jika selisih jarak koordinat x lebih kecil dari pada koordinat y maka musuh akan menghadap ke arah y entah menghadap ke atas ataupun kebawah, begitu juga sebaliknya. Berikut adalah potongan *ActionScript*nya :

```
//DIRECTION Y
if    ((Math.abs(hinoSpriteSensor_Core.x-enemyGroup[d].id.x))    <
(Math.abs(hinoSpriteSensor_Core.y-enemyGroup[d].id.y))) {
if (HinoEnemy.hinoEnemy_Devil_1_Y > hinoHeroY) {
            HinoEnemy.hinoDevil_1_Dir = 3;
} else if (HinoEnemy.hinoEnemy_Devil_1_Y < hinoHeroY) {
            HinoEnemy.hinoDevil_1_Dir = 0;
                  }
                  } else {
                  //DIRECTION X
if (HinoEnemy.hinoEnemy_Devil_1_X > hinoHeroX) {
HinoEnemy.hinoDevil_1_Dir = 1;
} else if (HinoEnemy.hinoEnemy_Devil_1_X < hinoHeroX) {
                        HinoEnemy.hinoDevil_1_Dir = 2;
                  }
                  }
```

Musuh akan mulai mengejar karakter dengan jarak yang telah ditentukan, perhitungan jaraknya menggunakan metode deteksi jarak menggunakan *Euclidean Distance*. Musuh juga dapat mengenali adanya kolisi, dan selama karakter masih dalam jangkauan musuh maka musuh akan mencari jalan untuk mencapai karakter utamanya. Berikut adalah potongan *ActionScript*nya :

```
if((Math.sqrt((hinoSpriteSensor_Core.x-
enemyGroup[d].id.x)*(hinoSpriteSensor_Core.x-
enemyGroup[d].id.x)+(hinoSpriteSensor_Core.y-
```

```
enemyGroup[d].id.y)*(hinoSpriteSensor_Core.y-enemyGroup[d].id.y))
< charDistance) || gameEndEnemyDestroy) {
            enemyGroup[d].ngejar = true;
      } else {
            enemyGroup[d].ngejar = false;
      }
```

Kemudian musuh akan mulai melakukan serangan terhadap karakter, jika musuh mulai menyentuh sensor dari karakter utamanya dengan formula :

```
hinoHero_CurrentHP -= HinoEnemy.enemy_Devil_1_MinAttack +
(Math.random() * ((HinoEnemy.enemy_Devil_1_MaxAttack  -
HinoEnemy.enemy_Devil_1_MinAttack ) - hinoHero_Defend / 2));
```

Darah karakter akan berkurang sesuai dengan jumlah kekuatan serangan yang dimiliki oleh musuh tersebut di kurangi dengan *defend* (pertahanan) dari karakter utama. Pertahanan dari karakter utama adalah dinamis, akan berubah jika level dari karakter meningkat. Kemudian untuk baris *ActionScript* tampilan pengurangan darah dari musuhnya :

```
HinoEnemy.enemy_Devil_Bar.enemyHPBar.scaleX =
HinoEnemy.enemy_Devil_1_CurrentHP/HinoEnemy.enemy_Devil_1_HP;
```

Berikut adalah baris dimana letak pengurangan darah karakter utama ketika menerima serangan dari musuh tersebut :

```
HinoEnemy.hinoEnemy_Attacked = true;
HinoEnemy.enemy_Devil_1_CurrentHP -= hinoHero_MinAttack +
(Math.random() * ((hinoHero_MaxAttack - hinoHero_MinAttack) -
HinoEnemy.enemy_Devil_1_Defend / 2));
```

Saat musuh mati, akan terpicu *Timer* untuk musuh tersebut selama 8 detik, ketika *Timer* itu selesai maka akan memanggil sebuah fungsi sebagai berikut :

```
public static function enemyRevive(event:TimerEvent):void {
if (!HinoMain.gameEndEnemyDestroy) {
if (hinoDevil_Die_Status) {
            enemy_Devil_1_CurrentHP = enemy_Devil_1_HP;
```

```
            hinoDevil_Die_Status = false;

            hinoDevil_1 = enemy_Devil_IdleMove;

            enemyReviveTimer_Devil.reset();

            HinoEnemy.enemy_Devil_Bar.visible = true;
            }
      }
      }
```

Fungsi di atas digunakan untuk membangkitkan lagi musuh yang sudah mati di posisi yang sama saat musuh mati. Musuh akan terus bangkit selama batu kristal dalam permainan masih berwarna hitam.

`HinoMain.gameEndEnemyDestroy` merupakan sebuah kondisi yang menandakan game sudah berakhir atau belum tergantung pada kondisi batu kristal hitam, jika bernilai ”true”, maka musuh tidak akan pernah bangkit lagi untuk selama-lamanya.

4.4 Implementasi Antarmuka

Implementasi Antarmuka adalah hasil tampilan pembuatan game sesuai dengan rancangan antarmuka sebelumnya. Setelah pembuatan maka dapat dipresentasikan hasil dari perancangan yang telah dibuat.

4.4.1 Menu Utama

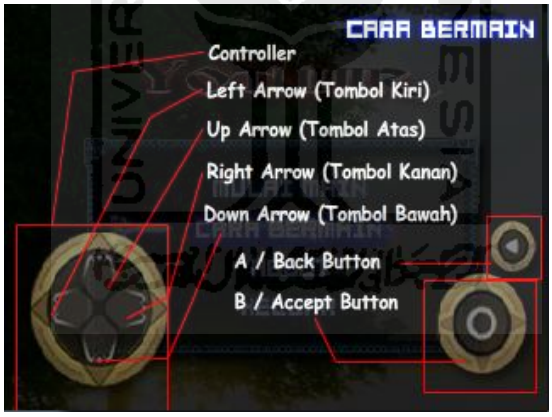
Menu utama merupakan menu yang pertama kali, tampilan ini juga bisa di sebut dengan *title menu.* Dalam menu utama ini terdapat 4 sub menu, yaitu menu mulai main, cara bermain, about, dan keluar. Tampilan Menu Utama dapat dilihat pada gambar 4.13.



Gambar 4.13 Tampilan Halaman Menu Utama

4.4.2 Halaman Cara Bermain

Halaman ini merupakan halaman yang muncul setelah menu ”Cara Bermain” terseleksi pada halaman menu utama. Pada halaman ini akan ditampilkan fungsi-fungsi *Controller* yang akan digunakan. Tampilan dari halaman Cara Bermain dapat dilihat pada gambar 4.14.



Gambar 4.14 Tampilan Halaman Cara Bermain

Fungsi Controller pada halaman ini hanya tombol A yang aktif. Tombol A disini berguna untuk kembali ke halaman menu utama. Berikut adalah potongan *ActionScript* untuk *Controller* pada halaman Cara Bermain :

```
case "Show Stuff": {
if (direction == "BACK"){
hinoBack.play();
HinoMain.hinoActiveScene = "Menu";
```

```
MainMenu.hinoSelectedMenu = "None";
                                }
} break;
```

4.4.3 Halaman About

Halaman ini merupakan halaman yang muncul setelah menu ”About” terseleksi pada halaman menu utama. Pada halaman ini akan ditampilkan fungsi-fungsi *Controller* yang akan digunakan. Tampilan dari halaman About dapat dilihat pada gambar 4.15.



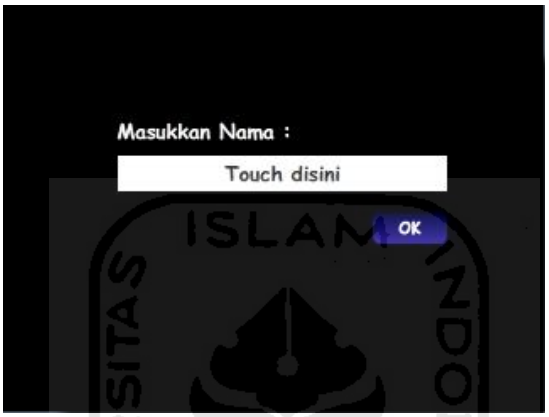
Gambar 4.15 Tampilan Halaman About

Fungsi Controller pada halaman ini hanya tombol A yang aktif. Tombol A disini berguna untuk kembali ke halaman menu utama. Berikut adalah potongan *ActionScript* untuk *Controller* pada halaman About :

```
case "Show Stuff": {
if (direction == "BACK"){
hinoBack.play();
HinoMain.hinoActiveScene = "Menu";
MainMenu.hinoSelectedMenu = "None";
                                }
} break;
```

4.4.4 Halaman Input Nama Karakter

Halaman Input nama karakter merupakan halaman yang muncul setelah pengguna memilih menu ”Mulai Main” pada menu utama. Pada halaman ini terdapat filter untuk pengisian nama. Jika nama yang diisikan masih kosong maka pengguna tidak dapat lanjut ke halaman yang selanjutnya. Tampilan dari halaman Input nama karakter dapat dilihat pada gambar 4.16.



Gambar 4.16 Tampilan Halaman Input Nama Karakter

Pada tampilan halaman menu, tidak terdapat *Controller*, namun pengguna menyentuh textbox input nama yang bertuliskan ”Touch disini”. Maka akan muncul *Keyboard* dari mobile *Android* yang digunakan yang berguna untuk pengisian nama karakter yang akan dimainkan.

4.4.5 *Game Play*

Game Play merupakan tampilan utama dalam *Game RPG* ini, disinilah tempat pengguna manjalankan karakter, bertarung, dan melakukan aksi-aksi tertentu. Tampilan *Game Play*nya dapat dilihat pada gambar 4.17.



Gambar 4.17 Tampilan *Game Play*

Pada tampilan *Game Play*, terdapat tampilan baru yaitu indikasi dari parameter karakter berupa nama, jumlah darah, *experience*, wajah, dan level karakter. Berikut adalah potongan *ActionScript* untuk tampilan parameter karakter pada *Game Play* :

```
hinoController.hinoGamePlayMenu.hinoLVLBar.text = "LVL " +
hinoHero_Level;

hinoController.hinoGamePlayMenu.hinoHPBar_txt.text =
hinoHero_CurrentHP+"/"+hinoHero_HP;
hinoController.hinoGamePlayMenu.hinoEXPBar_txt.text =
hinoHero_CurrentEXP + "/" + hinoHero_EXP;

hinoController.hinoGamePlayMenu.hinoHPBar.scaleX =
hinoHero_CurrentHP / hinoHero_HP;
hinoController.hinoGamePlayMenu.hinoEXPBar.scaleX =
hinoHero_CurrentEXP / hinoHero_EXP;
```

Pada potongan *ActionScript* di atas, akan ditampilkan secara terus menerus parameter karakter pada *Game Play*, berfungsi untuk menampilkan parameter karakter yang berubah-ubah. Seperti jumlah darah, *experience* dan *level* karakter yang berubah-ubah selama proses pertarungan antara karakter dan musuh berlangsung. Fungsi *Controller* pada *Game Play* adalah untuk tombol arah

berfungsi sebagai penggerak karakter utama yang kita mainkan, tombol B berguna untuk melakukan serangan, jika saat menyentuh tombol B dan karakter menyentuh *NPC*, maka akan terjadi pembicaraan secara tekstual antara karakter dengan *NPC*. Kemudian jika pengguna menekan tombol A, akan muncul halaman *Game Menu*.

4.4.6 Halaman *Game Menu*

Halaman *Game Menu* merupakan halaman yang akan tampil setelah pengguna menekan tombol A pada saat *Game Play*. Dalam *Game Menu* terdapat 3 menu yaitu ”Lanjutkan”, ”Status”, dan ”Keluar Game”. Tampilan halaman *Game Menu* dapat dilihat pada gambar 4.18.



Gambar 4.18 Tampilan Halaman *Game Menu*

Saat pengguna masuk pada halaman *Game Menu*, mode *Game Play* akan masuk dalam mode *pause* atau berhenti. Pada halaman *Game Menu*, menu ”Lanjutkan” berfungsi untuk kembali ke mode *Game Play*, menu ”*Status*” berfungsi untuk menampilkan halaman status dari karakter berupa parameter-parameter yang dimiliki oleh karakter. Kemudian untuk menu ”Keluar *Game*”, berfungsi untuk keluar dari permainan sekaligus menutup aplikasinya.

4.4.7 Halaman *Game Menu Status*

Halaman *Game Menu Status* merupakan halaman yang muncul setelah pengguna memilih menu ”*Status*” pada halaman *Game Menu,* dalam halaman *Game Menu Status* akan ditampilkan parameter-parameter dari karakter utama yang kita mainkan. Tampilan halaman *Game Menu Status* dapat dilihat pada gambar 4.19.



Gambar 4.19 Tampilan Halaman *Game Menu Status*

ActionScript Controller pada *Game Menu Status,* sama seperti fungsi *Controller* pada halaman *Game Menu,* tetapi hanya tombol A yang berfungsi yaitu untuk kembali ke halaman *Game Menu*.

4.4.8 Halaman *Game Menu Exit*

Halaman *Game Menu Exit* merupakan halaman yang akan muncul jika pengguna memilih menu ”Keluar Game” pada halaman *Game Menu,* dalam halaman ini hanya terdapat 2 menu, yaitu ”Ya” dan ”Tidak”. Tampilan halaman *Game Menu Exit* dapat dilihat pada gambar 4.20.



Gambar 4.20 Tampilan Halaman *Game Menu Exit*

Saat pengguna memilih menu ”Ya”, maka pengguna akan langsung keluar dari aplikasi. Jika menekan tombol A dan memilih menu ”Tidak”, maka pengguna akan kembali ke halaman *Game Menu*.

4.4.9 Halaman *Game Over*

Halaman *Game Over* adalah halaman yang akan muncul jika karakter kalah dalam pertarungan. Tampilan halaman *Game Over* dapat dilihat pada gambar 4.21.



Gambar 4.21 Tampilan Halaman *Game Over*

Saat pengguna masuk ke halaman *Game Over,* berarti pengguna gagal menyelesaikan misi yang sudah di berikan. Kemudian pengguna akan langsung dibawa ke halaman utama.

4.5 Analisis Kinerja *Game RPG* Berbasis *Controller PSP* pada *Android*

4.5.1 Pengujian *Game* pada *Mobile Android*

Setelah terselesaikannya Game RPG berbasis *Controller PSP* pada *Sistem Operasi Android*, maka *game* akan dicoba untuk diujikan secara langsung pada beberapa mobile yang bersistem operasi *Android.* Berikut adalah beberapa foto mobile Android yang sedang menjalankan aplikasi Game RPG berbasis *Controller PSP* pada *Sistem Operasi Android* ini:



Gambar 4.22 Pengujian *Game* pada Samsung Galaxy GT-I5500

Pada Mobile Samsung Galaxy GT-I5500 yang memiliki *Sistem Operasi Android Froyo 2.2, prosessor* sebesar 600 Mhz, *game* dapat berjalan dengan baik. Dengan didukung dengan *VGA* di dalamnya dan resolusi layar sebesar 240x320 maka kualitas gambar juga sangat baik.



Gambar 4.23 Pengujian *Game* pada AHA Touch

Pengujian kedua dilakukan pada Mobile AHA Touch, yang merupakan sebuah mobile untuk modem namun memiliki *Sistem Operasi Android Froyo 2.2*. Dengan jumlah *prosessor* yang kurang dari 600Mhz, *game* masih berjalan dengan baik, namun pergerakan animasinya sedikit lambat. Mobile ini juga tidak mendukung *VGA* sehingga kualitas grafiknya sedikit pecah tetapi karena mobile ini memiliki resolusi layar sebesar 240x320, maka kualitas gambar juga tidak terlalu buruk.



Gambar 4.24 Pengujian *Game* pada Samsung Galaxy Mini

Pengujian ketiga dilakukan pada Mobile Samsung Galaxy Mini, Mobile ini juga memiliki *Sistem Operasi Android Froyo 2.2*, jumlah *prosessor* pada mobile ini adalah 600Mhz, sehingga *game* berjalan dengan baik. Dengan didukung dengan *VGA* dan resolusi layar sebesar 240x320, maka kualitas grafik dari *game*nya juga sangat baik.



Gambar 4.25 Pengujian *Game* pada Samsung Galaxy GT-I5501

Pengujian keempat dilakukan Mobile Samsung Galaxy GT-I5501 yang memiliki *Sistem Operasi Android Froyo 2.2*, *prosessor* sebesar 600 Mhz, *game* dapat berjalan dengan baik. Dengan didukung dengan *VGA* di dalamnya dan resolusi layar sebesar 240x320 maka kualitas gambar juga sangat baik.



Gambar 4.26 Pengujian *Game* pada Samsung Galaxy GIO

Pengujian terakhir dilakukan pada *Samsung Galaxy GIO*. Mobile ini ber*sistem operasi Android Froyo 2.2*, dan jumlah prosessornya sangat baik yaitu 800Mhz. Dengan kemampuan *prosessor* yang baik maka *game* berjalan dengan sangat baik, ditambah lagi dengan dukungan VGA sehingga kualitas gambar juga baik, namun karena ukuran resolusinya layarnya sebesar 320x480 maka gambar menjadi sedikit pecah.

Pada saat proses pengujian pada beberapa mobile yang bersistem operasi *Android* di atas, tahap instalasi *game*nya sama yaitu dengan penginstalan aplikasi adobe AIR 2.5, dan dilanjutkan dengan penginstalan aplikasi gamenya. Dapat disimpulkan bahwa game yang berjalan pada mobile tidak akan secepat berjalan pada komputer, karena perbedaan jumlah prosesor yang lumayan jauh. Maka dari itu digunakan perbedaan penggunaan *FPS* (*Frame Rate Per Second*) untuk *game* yang dijalankan pada komputer dan pada mobile, sehingga dibutuhkan jumlah *FPS* yang sedikit besar. *FPS* yang digunakan pada *game* ini adalah sebesar 60 *FPS*. Dengan digunakannya *FPS* yang besar, *game* bisa berjalan lebih halus dibandingkan dengan *FPS* yang sedikit jumlahnya.

4.5.2 Kelebihan dan Kekurangan Sistem

Pada bagian ini akan dijelaskan tentang kelebihan dan kekurangan sistem yang dibangun, dilihat dari segi performa dan pengguna. Berikut adalah ulasannya:

A. Kelebihan Sistem

Kelebihan dari Aplikasi *Game RPG* berbasis *Controller PSP* pada sistem operasi *Android* adalah :

- Pergerakan animasi pada *Game Play* dibangun menggunakan teknik pergerakan pixel dari gambar, sehingga aplikasi berjalan lebih ringan dibandingkan dengan pergerakan animasi yang menggunakan objek-objek seperti *MovieClip*.
- Sistem pertarungan dalam *Game RPG* yang dibangun merupakan pertarungan secara langsung, sehingga akan mempermudah pengguna dalam bertarung karena pengguna akan lebih mudah menganalisa keberadaan musuh dan bergerak sesuai dengan yang diinginkan. Sistem pertarungan secara langsung ini juga tidak membutuhkan perpindahan lokasi, sehingga pertarungan berlangsung di lokasi yang sama.
- Animasi dan karakter yang bervariasi akan membuat pengguna tidak cepat merasa bosan dalam memainkan *game* ini, mulai dari animasi

naik level, animasi terkena serangan, ledakan musuh, juga animasi aura saat karakter yang kita mainkan mencapai level maksimalnya.

- *Game RPG* yang dibangun menggunakan beberapa efek suara yang bervariasi seperti suara monster saat melakukan serangan, suara karakter saat melakukan serangan, suara ledakan, teriakan musuh saat kalah dalam pertarungan membuat *game* terkesan lebih hidup.

B. Kekurangan Sistem

Kekurangan dari Aplikasi *Game RPG* berbasis *Controller PSP* pada sistem operasi *Android* adalah :

- Dibutuhkan pengalaman dalam memainkan *Game* pada perangkat layar sentuh, karena *Controller* yang dibangun merupakan *Controller* yang langsung bersatu dengan layar permainannya dan penggunaannya juga dibutuhkan ketelitian dalam menekan tombol *Controller*nya.
- Game yang dibangun tidak dirancang khusus alur ceritanya namun pada pembangunan sistemnya, sehingga dengan alur cerita yang sangat singkat, *game* juga akan berakhir dengan cepat.
- Musuh yang kalah dalam pertarungan, akan dibangkitkan tidak secara acak, namun dibangkitkan dari posisi musuh sebelum kalah dalam pertarungan. Sehingga pengguna akan sangat mudah menganalisa keberadaan musuh dengan cepat.
- Ketika *game* ini dijalankan pada resolusi layar yang lebih besar daripada 240x320, maka kualitas gambar akan sedikit pecah.

BAB V
KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil penelitian, analisis, perancangan sistem dan pembuatan program sampai dengan tahap penyelesaian, maka dapat ditarik beberapa kesimpulan antara lain sebagai berikut:

1. Aplikasi Game RPG berbasis *Controller PSP* pada *Sistem Operasi Android* telah berhasil dibangun dan diterapkan pada Mobile yang ber*sistem operasi Android 2.2 (Froyo).*
2. Setelah diujikan dari beberapa *mobile* yang ber*sistem operasi Android*, proses installasinya sama, dan *game* juga dapat berjalan dengan baik pada minimal prosessor sebesar 600Mhz.
3. Dengan dukungan *VGA* pada mobile yang ber*sistem operasi Android*, kualitas gambarnya sangat baik, namun jika resolusinya lebih besar daripada 240x320, kualitas gambar menjadi sedikit pecah.

5.2 Saran

Berdasarkan pada pengujian yang telah dilakukan terhadap *game* yang dibuat, perlu pengembangan lagi agar menjadi lebih baik, yaitu agar fitur-fitur pendukung dalam *Game RPG* yang dibangun terutama pada Sprite-spritenya tidak menggunakan sprite dari luar tetapi membuat sendiri. Kemudian untuk manajemen *script* masih tidak bersifat kurang dinamis, dalam artian jika ingin di kembangkan secara luas maka akan banyak perubahan dalam penulisan *Script*nya. Untuk perkembangannya *Game RPG* yang dibangun jika *Script* di olah secara umum, sehingga jika terdapat penambahan fitur maupun lama permainannya tidak perlu menduplikasi fungsi yang dibuat sebelumnya, namun cukup menggunakan 1 fungsi saja.

Sistem pertarungannya masih tergolong *Action Battle System* yang sangat sederhana, hanya terbatas pada serangan biasa dan kurang bervariasi, sehingga akan lebih baik jika terdapat pengembangan *Battle System* dan juga penambahan

parameter-parameter karakter dan statusnya bisa dinaikan secara manual oleh pengguna sehingga pengguna akan lebih tertarik dan lebih kreatif dalam membuat karakter utamanya sesuai dengan yang diinginkan.

DAFTAR PUSTAKA

Fazri, A. 2009. *Penggunaan Ruby Game Sscripting System Untuk Pengembangan Menu Game System Pada RPG Maker XP*. Skripsi, tidak diterbitkan. Yogyakarta: Jurusan Teknik Informatika, Fakultas Teknologi Industri, Universitas Islam Indonesia.

Sungwon, P. C., Hyukjae, J., Junehwa, S. 2011. *Roleplaying gamification to encourage social interactions at parties.* Diakses pada 18 Februari 2011 dari http://gamification-research.org/wp-content/uploads/2011/04/07-Choe.pdf.

Creative Commons Attribution-Share Alike 3.0 Unported. 2010. *Essentials of Adobe Flash.* Diakses pada 18 Februari 2011 dari http://www.culturalview.com/books/flash.pdf.

Ed, B. 2008. *Hello Android.* Diakses pada 19 Februari 2011 dari http://kronox.org/documentacion/Hello.Android.new.pdf.

Damianos, G., Daphne, E. 2011. *Development Platforms for Mobile Applications.* Diakses pada 19 Februari 2011 dari http://www2.aegean.gr/dgavalas/en/iframe_files/papers/2010/IEEE_Soft.pdf.

SEGA. 2000. *A Tabletop RPG Based on Phantasy Star Online.* Diakses pada 19 Februari 2011 dari http://www.pso-world.com/downloads/fantheater/jeffwmoore-Phantasy_Star_Ragols_Curse.pdf.

Dull, B. 2010. *Pie Role Playing Game.* Diakses pada 19 Februari 2011 dari http://www.pierpg.com/pierpg.pdf.

Simon, K. 2005. *An introduction to SDL.* Diakses pada 20 Februari 2011dari http://www.ipd.bth.se/ska/sdl_intro.slides.pdf.

Fabian, F. 2006. *Simple Direct Media Layer*. Diakses pada 21 Februari 2011 dari http://www.cs.helsinki.fi/u/ffagerho/coursenotes/seminars/ohjelmistotuotanto-pelit/sdl-talk.pdf.

UNIVERSITAS ISLAM INDONESIA