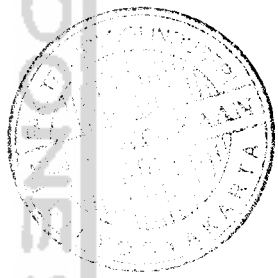
RANCANG BANGUN APLIKASI GAME MOBIL BALAP DENGAN BERBASIS AGENT MENGGUNAKAN MACROMEDIA FLASH MX 2004 DAN BAHASA ACTION SCRIPT

TUGAS AKHIR

Diajukan sebagai Salah Satu Syarat
untuk Memperoleh Gelar Sarjana Bidang Studi Teknik Informatika





Oleh:

Nama : **Yulin Arryany**
No. Mahasiswa : **03523233**

JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA
YOGYAKARTA
2007

LEMBAR PERNYATAAN KEASLIAN HASIL TUGAS AKHIR

Yang bertanda tangan di bawah ini
Nama : Yulin Arryany
NIM : 03 523 233
Tugas Akhir dengan judul :

RANCANG BANGUN APLIKASI GAME MOBIL BALAP DENGAN BERBASIS AGENT MENGGUNAKAN MACROMEDIA FLASH MX 2004 DAN BAHASA ACTION SCRIPT

Dengan ini saya menyatakan bahwa seluruh komponen dan isi dalam Tugas Akhir ini adalah hasil karya saya sendiri.
Apabila di kemudian hari 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, 21 Agustus 2007
Yang Membuat Pernyataan

Yulin Arryany

Saksi-saksi

Taufiq Hidayat ST, MCS.
Saksi I

Sri Kusumadewi,SSi., MT.
Saksi II

Nur Wijayaning R. S.Kom
Saksi III

LEMBAR PENGESAHAN PENGUJI

RANCANG BANGUN APLIKASI GAME MOBIL BALAP DENGAN BERBASIS AGENT MENGGUNAKAN MACROMEDIA FLASH MX 2004 DAN BAHASA ACTION SCRIPT

TUGAS AKHIR

Oleh :

Nama : Yulin Arryany

NIM : 03 523 233

Telah Dipertahankan di Depan Sidang Penguji Sebagai Salah Satu Syarat
untuk Memperoleh Gelar Sarjana Teknik Informatika
Fakultas Teknologi Industri Universitas Islam Indonesia

Yogyakarta, 21 Agustus 2007

Tim Penguji

Taufiq Hidayat ST, MCS.
Ketua

Sri Kusumadewi,SSi., MT.
Anggota I

Nur Wijayaning R. S.Kom
Anggota II

Mengetahui,
Ketua Jurusan Fakultas Teknologi Industri
Universitas Islam Indonesia

Yudi Prayudi, SSi.,M.Kom

PERSEMBAHAN

Rasa syukur Kehadirat Allah SWT atas karunianya dan seluruh keridhoannya

Bapak Djuari Bunasir dan Ibu Erwina yang selalu setia mendukung tiap waktu terima kasih atas segalanya, doa ,didikan serta kasih sayang yang tak terhingga. semoga pahala yang berlipat yang mereka dapatkan,karena ketidakmungkinnanku untuk membalasnya untuk selamanya.....

Untuk kedua Kakak-qu yang kusayang terima kasih atas dorongannya,

Dan sahabat sahabat dekat yang selalu mendukungqu semoga

Apa yang kalian harapkan dapat terwujud.....

Amin.....

MOTTO

""Hai orang-orang beriman, mintalah pertolongan kepada Allah dengan sabar dan sholat, sesungguhnya Allah beserta orang-orang yang sabar."

(QS Al-Baqarah[2]:153)

"...Sesungguhnya sesudah kesulitan itu ada kemudahan. sesungguhnya sesudah kesulitan itu ada kemudahan."

(QS Al-Insyirah[94]:5-6)

Kemenangan dan keberhasilan hanya dapat dicapai dengan kesabaran.

(HR At Tirmidzi)

KATA PENGANTAR

بسم الله الرحمن الرحيم

Assalamu'alaikum Wr. Wb

Alhamdulillah, segala puji syukur hanyalah kepada Allah SWT dan semoga sholawat serta salam dilimpahkan kepada junjungan kita Nabi Muhammad SAW, keluarga, sahabat, dan pengikut-pengikut beliau (amin). Sehingga penulisan laporan tugas akhir yang berjudul : *Rancang Bangun Aplikasi Game Mobil Balap dengan Berbasis Agent menggunakan Macromedia Flash MX 2004 dan bahasa Action* Script dapat penulis selesaikan dengan baik.

Laporan tugas akhir ini disusun untuk melengkapi salah satu syarat guna memperoleh gelar Sarjana Teknik Informatika pada Universitas Islam Indonesia dan atas apa yang telah diajarkan selama perkuliahan baik teori maupun praktek, disamping laporan itu sendiri yang merupakan rangkaian kegiatan yang harus dilakukan setelah tugas akhir ini selesai.

Dalam pembuatan tugas akhir ini, penulis mendapat banyak bimbingan dan pengarahan serta bantuan yang sangat bermanfaat dari berbagai pihak. Oleh karena itu penulis ingin menyampaikan ucapan terima kasih yang sebesar-besarnya kepada :

1. Keluarga saya, bapak ibu dan kedua kakak saya yang selalu mendukung dengan semangat dan doa.
2. Bapak Fathul Wahid, ST. MSc., selaku Dekan Fakultas Teknologi Industri UII dan seluruh jajaran dekanat Fakultas Teknologi Industri.

3. Bapak Yudi Prayudi, S.Si.,M.Kom., selaku Ketua Jurusan Teknik Informatika, Universitas Islam Indonesia.
4. Taufiq Hidayat, ST. MCS., selaku dosen pembimbing yang telah memberikan banyak petunjuk dan semangat.
5. Teman-teman kontrakan 69 B Rizka, Farah, Heny, Eka Domesti dan si Irma anak baru di kontrakan kami. Terima kasih telah membantu memberikan semangat'45, ide, saran, dan kritik dalam pembuatan tugas akhir ini.
6. Genk Derezan : kak chun, kak ani, kak aghie, mas anton, ncut, etc yang dah memberikan support dari sana.
7. Teman-teman UII Informatika 2003.
8. Sahabat-sahabat dekat walaupun lain kampus : Ajai, Henk, Anda, Yudi, Attienk, Nita, Nani, Sinta, Lina,Yuni, dan semuanya.
9. Teman-teman Al-Manar terutama anak-anak Insight Photography, Jurnal, etc.

Penulis menyadari bahwa tugas akhir ini masih jauh dari sempurna. Untuk itu segala saran dan kritik yang sifatnya membangun sangat penulis harapkan guna penyempurnaan dimasa mendatang. Akhir kata, semoga laporan tugas akhir ini dapat bermanfaat bagi kita semua. Amiin.

Yogyakarta, 1 Agustus 2007

Penulis

ABSTRAKSI

Teknologi dalam bidang komputer yang berkembang saat ini semakin pesat ditunjang dengan media *hardware* yang memungkinkan dibuatnya beragam jenis aplikasi *game*. Macromedia Flash MX 2004 yang didukung dengan bahasa *Action*Script merupakan program grafis multimedia dan animasi yang dapat dipergunakan sebagai program pembuat *game*. *Game* juga merupakan salah satu sarana hiburan. Kini banyak dikembangkan *game-game* baru yang bisa dipakai untuk mengetes kecerdasan seseorang. Salah satu permainan yang dapat dikembangkan adalah *Game Mobil Balap Racing Cars* yang lebih menarik dikarenakan dilengkapi dengan basis *Agent*.

Game Mobil Balap Racing Cars merupakan suatu permainan yang terdiri dari beberapa obyek yang mempunyai karakter yang berbeda - beda dan digunakan menjadi satu dalam suatu alur yang di implementasikan di tiap - tiap *level*. Musuh akan dilengkapi dengan skrip *berbasis agent*, yaitu suatu skrip pintar atau perintah untuk melakukan perlawanan terhadap pemain.

Hasil rancang bangun *Game Mobil Balap Racing Cars* telah menjadi salah satu alternatif pilihan lain dalam memilih jenis permainan dan telah bersifat *user friendly*.

Kata Kunci : *game*, *script berbasis agent*, *obyek* dan *user*

2.6 Macromedia Flash MX 2004 13
2.6.1 Definisi Flash MX 13
2.6.2 Teknologi *ActionScript* 14
2.6.3 Pemrograman *ActionScript* 15
2.6.4 Sintaks *ActionScript* 16
2.6.4.1 Dot Syntax 16
2.6.4.2 Kurung Kurawal 18
2.6.5 Tipe Data 18
2.7 *Script* Basis *Agent* 19

BAB III METODOLOGI 20
3.1 Metode Analisis 20
3.2 Hasil Analisis 20
3.2.1 Analisis Kebutuhan Proses 20
3.2.2 Analisis Kebutuhan Input 22
3.2.3 Analisis Kebutuhan Output 23
3.2.4 Analisis *User Interface* 24
3.2.5 Analisis Kebutuhan Perangkat Lunak 24
3.2.6 Analisis Kebutuhan Perangkat Keras 25
3.3 Perancangan Perangkat Lunak 25
3.3.1 Motode Perancangan 25
3.3.2 Hasil Perancangan 26
3.3.2.1 *Use Case* Diagram 26
3.3.2.2 Diagram Kelas 27
3.3.2.3 Diagram Sekuen 29
3.3.2.4 Diagram Aktivitas 38
3.4 Perancangan Antarmuka (*interface*) 40
3.4.1 Perancangan Tampilan Menu Utama 40
3.4.2 Perancangan Tampilan *Single Game* 41
3.4.3 Rancangan Tampilan *Setting* 43
3.4.4 Rancangan Tampilan *Help* 43

3.4.5 Rancangan Tampilan *High Score* 44
3.4.6 Rancangan Tampilan *About* 44

BAB IV IMPLEMENTASI DAN PEMBAHASAN 45
4.1 Implementasi 45
4.1.1 Batasan Implementasi 45
4.1.1.1 Perangkat Keras 46
4.1.1.2 Perangkat Lunak 46
4.1.2 Implementasi Rancangan Antarmuka 46
4.1.2.1 Antarmuka Menu Utama 47
4.1.2.2 Antarmuka Permainan 47
4.1.2.3 Antarmuka *Setting* 52
4.1.2.4 Antarmuka *Help* 52
4.1.2.5 Antarmuka *High Score* 53
4.1.2.6 Antarmuka *About* 53
4.1.3 Implementasi Prosedural 54
4.1.3.1 Script Main 54
4.1.3.2 Script Pilihan Modifikasi 54
4.1.3.3 Script Tombol RACE 61
4.1.3.4 Script pergerakan mobil pemain 62
4.1.3.5 Script pergerakan mobil musuh 65
4.2 Pembahasan 66
4.2.1 Analisis Kinerja Perangkat Lunak 66
4.2.1.1 Pengujian Perangkat Lunak 66
4.2.1.2 Hasil Pengujian Perangkat Lunak 67
4.2.1.3 Hasil Pengujian terhadap Responden 69
4.2.2 Analisis Kesesuaian 69
4.2.3 Perbandingan dengan *game* sejenis 71

BAB V KESIMPULAN DAN SARAN 74
5.1 Kesimpulan 74
5.2 Saran 75

DAFTAR PUSTAKA 76
LAMPIRAN 78

DAFTAR GAMBAR

Gambar 3.1 Diagram *Use Case* 27
Gambar 3.2 Diagram *Kelas* 28
Gambar 3.3 Diagram sekuen *pemanggilan game* 30
Gambar 3.4 Diagram sekuen pilih *single game* 31
Gambar 3.5 Diagram sekuen *melakukan setting* 32
Gambar 3.6 Diagram sekuen *melihat bantuan* 33
Gambar 3.7 Diagram sekuen melihat *about* 34
Gambar 3.8 Diagram sekuen *melihat High Score* 35
Gambar 3.9 Diagram sekuen *memulai game* 36
Gambar 3.10 Diagram sekuen *menjalankan mobil* 37
Gambar 3.11 Diagram sekuen *input score* 38
Gambar 3.12 Diagram aktivitas 39
Gambar 3.13 Rancangan Tampilan Menu Utama 41
Gambar 3.14 Rancangan Tampilan Single *Game level* biasa 42
Gambar 3.15 Rancangan Tampilan Single *Game level* bonus 42
Gambar 3.16 Rancangan Tampilan *Setting* 43
Gambar 3.17 Rancangan Tampilan *help* 43
Gambar 3.18 Rancangan Tampilan *High Sroce* 44
Gambar 3.19 Rancangan Tampilan *about* 44
Gambar 4.1 Antarmuka *Menu Utama* 47
Gambar 4.2 Antarmuka modifikasi mobil 48
Gambar 4.3 Antarmuka indikasi 48
Gambar 4.4 Antarmuka permainan *level* 1 49
Gambar 4.5 Antarmuka bonus *level* 1 49
Gambar 4.6 Antarmuka permainan *level* 2 50
Gambar 4.7 Antarmuka bonus *level* 2 50
Gambar 4.8 Antarmuka permainan *level* 3 51
Gambar 4.9 Antarmuka bonus *level* 3 51

Gambar 4.10 Antarmuka *Setting* 52
Gambar 4.11 Antarmuka *Help* 52
Gambar 4.12 Antarmuka *High Score* 53
Gambar 4.13 Antarmuka *About* 53
Gambar 4.14 Tampilan Menu utama *game* "*Racing Cars*" 73
Gambar 4.15 Tampilan Menu utama *game* "*Mode7Racing*" 73

LEMBAR PENGESAHAN PEMBIMBING

RANCANG BANGUN APLIKASI GAME MOBIL BALAP DENGAN BERBASIS AGENT MENGGUNAKAN MACROMEDIA FLASH MX 2004 DAN BAHASA ACTION SCRIPT

TUGAS AKHIR

Oleh:

Nama : Yulin Arryany

No. Mahasiswa : 03523233

Yogyakarta, 1 Agustus 2007

Pembimbing,



Taufiq Hidayat, ST, MCS.

DAFTAR TABEL

Tabel 4.1 Tabel hasil analisis 68
Tabel 4.2 Tabel perbandingan *game* 72

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Semakin majunya teknologi informasi dan komputer saat ini mendorong setiap pengguna komputer untuk dapat memanfaatkan dan terus mengikuti perkembangan teknologi tersebut. Salah satu yang paling berkembang dalam dunia teknologi adalah bidang hiburan, karena bidang ini akan terus dibutuhkan oleh setiap orang. Salah satu hiburan yang mempunyai media teknologi kompuer yang maju adalah *game*. *Game* merupakan media hiburan yang sangat digemari dari anak-anak sampai dengan orang dewasa, karena *game* juga dapat dijadikan sebagai sarana hiburan itu sendiri. *Game* bisa dimainkan kapan saja selain waktu senggang, bahkan ada pula yang sampai kecanduan untuk terus bermain *game*. Hal ini disebabkan *game* mengalami perkembangan yang menarik sehingga membuat orang penasaran untuk dapat menyelesaikan *game* tersebut.

Semakin pesatnya perkembangan teknologi dalam bidang komputer yang ditunjang dengan media *hardware* yang memadai, diharapkan dapat tercipta beragam jenis aplikasi *game*. Tetapi yang lebih menyenangkan apabila dapat membuat aplikasi *game* sendiri. Untuk dapat membuat *game* tidaklah mudah, beberapa *game* komersil yang sudah ada saat ini dibuat dan dikembangkan oleh banyak orang dengan keahlian pemrograman yang tinggi dan dibuat dengan biaya yang relatif besar, misalnya *game* yang bernuansa 3D yang biasanya

membutuhkan spesifikasi komputer yang tinggi untuk dapat menjalankan *game* tersebut.

Salah satu alternatif jenis *game* yang banyak diminati adalah jenis *game* 2D. *Game* jenis ini selain tidak membutuhkan spesifikasi komputer yang tinggi untuk menjalankannya, *game* ini juga cukup menarik untuk dimainkan. Dalam perkembangan aplikasi *game* 2D, Macromedia Flash MX merupakan salah satu program grafis multimedia dan animasi yang ditunjang dengan bahasa pemrograman Action Script yang dapat juga dimanfaatkan sebagai software pembuat *game*, khususnya aplikasi *game* 2D.

Game juga merupakan salah satu sarana untuk mengembangkan sumber daya manusia yaitu psikologi. Psikologi merupakan cabang ilmu pengetahuan yang abstrak. Berbagai hal mengenai perilaku dan lingkungannya dapat dipelajari dan diuji. Banyak yang dianggap membosankan sehingga banyak dikembangkan cara melakukan tes kepandaian seseorang dengan menciptakan *game-game* yang menarik. Salah satu permainan yang dapat dikembangkan adalah *game* Mobil Balap yang lebih menarik dikarenakan dilengkapi dengan basis Agent.

Menggunakan sebuah mobil sebagai pembalap yang akan dimainkan untuk mengumpulkan Bonus sebanyak-banyaknya. Pembalap akan ditantang oleh dua musuh lainnya yang saling bersaing mengumpulkan bonus. Musuh inilah yang akan dilengkapi dengan 'basis agent'. Musuh akan diberikan perintah (script) untuk bermain dan dapat bersaing dengan pembalap. Kepintaran tiap musuh akan semakin meningkat dalam tiap level yang berbeda. Tingkat kepandaian dan reflek pemain akan terlihat berdasarkan jumlah bonus yang diperoleh.

1.2 Rumusan Masalah

Berdasarkan uraian dari latar belakang masalah dapat dirumuskan sebuah permasalahan yang dapat dijadikan acuan dalam pembuatan program, yaitu bagaimana membangun aplikasi *game* Mobil Balap dengan Berbasis Agent Menggunakan Macromedia Flash MX 2004.

1.3 Batasan Masalah

Dalam membangun suatu aplikasi diperlukan adanya batasan-batasan agar aplikasi yang dibangun tidak menyimpang dari yang telah direncanakan sehingga tujuan yang sesungguhnya dapat tercapai. Adapun batasan-batasannya adalah sebagai berikut :

1. Pembuatan aplikasi *game* ini menggunakan Macromedia Flash MX dan menggunakan bahasa pemrograman Action Script.
2. Permainan dibatasi dengan proses pengumpulan bonus sebanyak banyaknya pada level biasa maupun level bonus. Permainan tiap level akan berakhir ketika waktu habis.
3. Tiap level *game* balap ini akan menampilkan dua jenis sudut pandang, yaitu level biasa dan level bonus. Level biasa akan menampilkan pemandangan depan yang pada dasarnya adalah menggerakkan background menuju pemain. Dengan gerakan tersebut seolah-olah mobil tersebut bergerak maju. Level bonus akan menampilkan pandangan dari atas.

4. Pada level biasa pembalap dan musuh hanya dapat menjalankan mobil ke arah atas (maju), kanan dan kiri, sedangkan pada level bonus mobil dapat bergerak mundur.
5. Jumlah pemain adalah 1 (satu) orang pemain.
6. Jumlah musuh adalah 2 (dua) orang pemain.
7. Jumlah level yang dimainkan adalah 3 (tiga) level.

1.4 Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah :

1. Mempelajari dan memanfaatkan teknologi yang berbasis komputer dalam pembuatan aplikasi *game* dengan menggunakan program Macromedia Flash MX 2004.
2. Membangun sebuah aplikasi *game* yang menarik dan menghibur.

1.5 Manfaat Penelitian

Adapun manfaat dari penelitian ini adalah :

1. Diharapkan dengan dibuatnya aplikasi *game* ini, maka peneliti tidak hanya menjadi pemakai (*user*) dari *game* yang sudah ada melainkan juga dapat membuat aplikasi *game* sendiri.
2. Sebagai salah satu alternatif pilihan lain dalam memilih jenis permainan yang semakin banyak macamnya.

1.6 Metodologi Penelitian

Metodologi yang digunakan dalam penelitian ini meliputi metode pengumpulan data dan pengembangan perangkat lunak.

1.6.1 Metode Pengumpulan Data

Pengumpulan data pada penelitian ini menggunakan metode studi literatur dengan mengumpulkan literatur yang layak dijadikan landasan, melalui buku-buku referensi, internet, dan forum diskusi, terutama yang banyak diterapkan oleh para praktisi pengembang *game*.

1.6.2 Metode Pengembangan Perangkat Lunak

Metode pengembangan perangkat lunak yang digunakan meliputi analisis kebutuhan perangkat lunak, perancangan perangkat lunak, implementasi perangkat lunak dan analisis kinerja perangkat lunak.

1.7 Sistematika Penulisan

Dalam penyusunan tugas akhir ini, sistematika penulisan dibagi menjadi beberapa bab sebagai berikut:

BAB I PENDAHULUAN

Bab ini berisi pembahasan masalah umum yang meliputi latar belakang masalah, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metodologi penelitian dan sistematika penulisan.

BAB II LANDASAN TEORI

Bagian ini memuat tinjauan pustaka, definisi *game*, jenis, elemen, dasar teori pembuatan *game*, *Action Script* dan definisi *Script Agent* dalam Macromedia Flash MX.

BAB III METODOLOGI

Bab ini terdiri dari analisis kebutuhan perangkat lunak, perancangan perangkat lunak dan implementasi perangkat lunak untuk membangun *game* Mobil Balap dengan basis agent menggunakan Macromedia Flash MX.

BAB IV IMPLEMENTASI DAN PEMBAHASAN

Bab ini membahas tentang analisis kerja dari perangkat lunak yang mengulas analisis hasil pengujian *game* yang dibangun dibandingkan dengan kebenaran dan kesesuaiannya dengan kebutuhan perangkat lunak yang telah dituliskan dalam bagian sebelumnya.

BAB V KESIMPULAN DAN SARAN

Merupakan simpulan dan saran yang merupakan rangkuman dari hasil analisis kinerja yang perlu diperhatikan berdasar keterbatasan- keterbatasan yang ditemukan dan asumsi-asumsi yang dibuat selama pengembangan perangkat lunak.

BAB II

LANDASAN TEORI

2.1 Tinjauan Pustaka

Beberapa *game* dapat dijadikan contoh dalam rancang bangun *game* Mobil Balap berbasis agent ini, diantaranya adalah *game* Perang Robot dan *game* Skateboard Jet.

1. *Game* Perang Robot

   Game ini ditulis oleh Deydri Adha, dengan judul *Rancang Bangun Aplikasi Action Game Perang Robot menggunakan bahasa Action Script pada Macromedia Flash MX.*

   Game Perang Robot ini berisi tentang karakter robot yang bertarung dengan karakter robot lain hingga salah satu karakter kalah. Sebelum pertarungan dimulai, terdapat *option* untuk memilih robot yang akan digunakan untuk bertarung. Gerakan yang dapat dilakukan robot adalah gerakan menyerang, bertahan, gerakan ketika terkena senjata, gerakan ketika menang dan kalah. Permainan akan berakhir apabila salah satu karakter dinyatakan kalah. Musuh digerakkan dengan script berbasis agent. Musuh akan diberikan script untuk melakukan perlawanan terhadap serangan yang digerakkan oleh *user*. [DEY05]

dengan bentuk yang sesuai dengan si pembuatnya, yang jelas untuk ukuran *game* bervariasi dan dapat dioperasikan di PC yang sederhana.

Filosofi *game* sendiri dibagi menjadi dua yaitu stand – alone pada *game* yang bersifat single player dan network application yang bersifat multiplayer. Salah satnya dalam jenis *game* single adalah *game* Mobil Balap dengan berbasis agent.

Game Mobil Balap memiliki *game*play yang sangat sederhana dengan alur penyelesaian melewati finish pada level terakhir dan memenangkan pertarungan.

2.3 Jenis *Game*

Sebuah PC tidak akan sempurna bila tanpa *game*. Ungkapan tersebut menggambarkan keadaan saat ini di mana hampir setiap PC sudah memiliki *game*, baik *game* berukuran besar, ataupun kecil.

Game yang dapat dipasang di PC dapat diklasifikasikan menjadi beberapa tipe, yaitu [CHA05] :

1. **Side Scrolling *Game***. Jenis *game* Side Scrolling, karakter dapat bergerak ke samping diikuti dengan gerakan background. Cotoh *game* dari tipe ini adalah Super Mario, Metal Slug dan sebagainya.
2. **Shooting *Game***. Ada beberapa macam *game* shooting, di antaranya adalah First Person Shooting danThird Person Shooting. Pada dasarnya kedua jenis *game* tersebut sama, yaitu menembak musuh. Contoh *game* dari tipe ini adalah Counter Strike, Doom, Chicken Shoot, Virtual Cop dan sebagainya.

3. **RPG (RolePlay *Game*)**, adalah *game* memainkan peran suatu karakter dalam menjalankan misi tertentu. Contoh dari *game* ini antara lain Final Fantasy, Lords of The Ring dan sebagainya.
4. **Simulation**. *Game* ini merupakan simulasi dari keadaan riil. Ada beberapa karakter pembagian *game* simulasi antara lain adalah *game* manusia seperti The SIMS, Ace Combat, dan *game* simulasi lainnya seperti sim City, Champhionship Manager dan *game* sim lainnya.
5. **Racing**. Contoh dari *game* racing antara lain Need for Speed, F1, Grandprix 500 cc dan sebagainya.
6. **Fighting**. Dalam *game* fighting pemain memainkan satu atau lebih karakter unuk bertarung dengan karakter lain. Contoh dari *game* ini antara lain : Tekken, Street Fighter dan sebagainya.

2.4 Dasar dan Elemen penyusun *Game*

Memainkan *game* di komputer sangatlah menyenangkan. Namun demikian, akan lebih menyenangkan lagi bila *game* dapat dibuat sendiri dan dapat dimainkan oleh orang lain. Untuk membuat *game* memang tidak mudah. Beberapa *game* komersil yang pernah yang ada saat ini dibuat dan dikembangkan oleh banyak orang dengan keahlian pemrograman yang tinggi dan dibuat dengan biaya yang relatif besar.

Dalam suatu *game* terdapat beberapa elemen yang menyusun *game* tersebut, seperti : jenis *game*, karakter dalam *game*, background, elemen sound/suara, dan gerakan-gerakan. Suatu *game* akan membosankan dan kurang

menarik apabila elemen-elemen yang menyusun *game* tersebut kurang dikelola dengan baik.

2.5 Teori Dalam Pembuatan Program Permainan

Urutan pembuatan *game* dan pengembangan program *game* adalah sebagai berikut [HEN89] :

1. Tentukan tipe permainan yang ingin dibuat

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

2. 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.

3. Definiskan secara jelas *Game Worlds*-nya

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

   a. *Game board*

      Game board merupakan bentuk tampilan, latar belakang, dan lainnya.

   b. Instruksi untuk permainan

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

c. Informasi untuk pemain

Informasi ini penting ditampilkan dalam program permainan ketika sedang berjalan.

d. Penghargaan

Penghargaan digunakan untuk pemain ketika menyelesaikan *level* tertentu dan mendorong untuk melanjutkan ke *level* berikutnya.

e. Variasi

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

f. Tingkat Kesulitan

Tingkat kesulitan dari program permainan akan membuat pemain manjadi bergairah jika melewati tingkat kesulitan yang diberikan.

4. Pastikan permainan bisa dimainkan

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

5. Rancang program sebaik mungkin

Gunakan teknik pemrograman yang sesuai dalam membuatnya dan buat program yang mudah dimodifikasi dan dikembangkan.

6. Pengujian program

Program yang selesai harus diuji. Pertama, program diuji oleh perancangnya sendiri (*alpha test*) untuk menentukan kesalahan logika yang mungkin terjadi. Kedua, program diuji oleh orang lain (*beta test*)

untuk mengetahui kesalahan logika yang tidak terlihat oleh perancangnya sehingga memberi masukan, saran atau ide dalam pengembangan program permainan.

2.6 Macromedia Flash MX 2004

2.6.1 Definisi Flash MX

Macromedia Flash merupakan program grafis multimedia dan animasi yang dapat dipergunakan untuk membuat aplikasi web interaktif yang menarik. Selain digunakan sebagai media pembuat web interaktif, Flash juga dapat dimanfaatkan sebagai program pembuat *game*. Macromedia Flash MX memiliki beberapa kemampuan, antara lain :

1. Animasi dan gambar yang dibuat dengan Flash terlihat bagus pada ukuran window dan resolusi layar berapapun. Hal ini karena Flash merupakan suatu program grafis dengan sistem vektor.
2. Waktu loading untuk animasi ataupun *game* sangat cepat, lebih cepat dibandingkan program sejenis lainnya.
3. Kemampuannya sebagai program pembuat web interaktif, karena ditunjang beberapa action script penting sehingga dapat dimanfaatkan untuk pembuatan *game*.
4. Mampu menganimasikan grafis, sekalipun dalam ukuran besar, dengan cepat dan mampu mengerjakan sejumlah frame dengan urutan.

2.6.2 Teknologi *Action Script*

Meskipun untuk membuat *game* bisa dikatakan sangat rumit, namun juga tidak boleh berkecil hati. *Game* sederhana dapat dibuat dengan cara yang mudah. Macromedia Flash ditunjang oleh Script-nya sehingga dapat dimanfaatkan untuk membuat suatu *game* yang diinginkan. Untuk membuat suatu *game*, hanya beberapa script saja yang akan digunakan sehingga tidak mengharuskan pembuat *game* untuk mengetahui seluruh action script yang ada.

Teknologi *Action Script* yang diharapkan sebagai teknologi masa depan merupakan salah satu teknologi yang saat ini banyak digunakan dan dikembangkan karena memiliki beberapa kelebihan, beberapa faktor yang mendukung teknologi dalam Flash antara lain [NAV02] :

1. Format Grafis

   Secara grafis, grafik berbasis *vektor* dimana obyek didefinisikan dengan persamaan matematis atau vektor yang didalamnya mengandung ukuran objek, bentuk, warna, garis tepi (*outline*) dan posisi.

2. Kapasitas File

   Pada Flash, obyek atau elemen yang berbentuk *vektor*, *bitmap* maupun suara hanya akan didefinisikan sekali. Jika membutuhkan objek lain yang bentuk asalnya sama dengan obyek yang pernah dibuat sebelumnya maka obyek asal dapat dimodifikasi tanpa menambah ukuran file.

3. Interaktif

   Dukungan multimedia dan animasi berbasis pemrograman membuat Flash dapat berinteraksi secara langsung dengan *user*. Hal ini sangat penting

ketika membangun sebuah web yang interaktif. Bahkan Flash juga mendukung *HTML (HyperText Markup Language)* yang merupakan bahasa pembuat situs web.

4. Kemampuan Design

   Objek – objek dalam halaman Flash dapat ditempatkan secara bebas sesuai dengan keinginan pembuat. Halaman tersebut dapat terdiri dari beberapa layer termasuk obyek yang dijadikan background.

5. Keanekaragaman

   Flash dapat menangani *project* atau aplikasi obyek dalam skala besar maupun kecil, tingkat kesukaran tinggi atau sederhana. Adapun bentuk aplikasi yang dapat dibuat dengan Flash antara lain :

   a. *Games*
   b. Kartun
   c. Web Site dan Web Database
   d. Aplikasi Multimedia
   e. Media Promosi
   f. Video

2.6.3 Pemrograman *ActionScript*

Bahasa pemrograman yang digunakan dalam membangun aplikasi ini menggunakan bahasa pemrograman *ActionScript* yang terdapat pada Macromedia Flash. Pemrograman pada Macromedia Flash merupakan pemrograman yang berbasis visual, sehingga dengan fasilitas visual tersebut akan mempermudah

dalam membangun aplikasi *game* yang diinginkan. Pemrograman dengan *ActionScript* dapat digunakan untuk pembuatan visualisasi di segala bidang terutama aplikasi interaktif berbasis web. *AclionScript* adalah pemrograman visual berorientasi object. Seperti *JavaScript*, *ActionScript* mempunyai sintaks, tata bahasa, dan struktur yang mirip dengan bahasa pemrograman C++.

ActionScipt merupakan bahasa pemrograman (*scripting*) yang digunakan dalam Macromedia Flash. Penggunaan *ActionScript* mempunyai tujuan [SUT02]:

1. memberikan kebebasan berkreasi bagi desainer
2. membuat animasi yang interaktif
3. dapat menampilkan animasi tertentu (non-linier)

2.6.4 Sintaks *ActionScript*

ActionScript mempunyai aturan tata bahasa dan tanda baca tersendiri sehingga dapat mengenali awal dan akhir sebuah pernyataan.

2.6.4.1 Dot Syntax

Dalam *ActionScript*, suatu *dot* (.) digunakan untuk [SUT02]:

1. menunjukkan *property method* dari *object* atau *movie clip*
2. mengenali *target path* ke suatu *movie clip* atau variabel

Dot syntax dimulai dengan nama dari *object* atau *movie clip*, diikuti dengan *dot*, dan diakhiri dengan *property*, *method*, atau variabel.

Contohnya, *movie clip* mempunyai *property* :

1. _x menunjukkan posisi *ordinat* x pada *stage*
2. *setproperty*(_root.D4,_x,465), berarti menempatkan *movie clip* D4 dengan *ordinat* x bernilai 465

Selain *property*, *movie clip* juga mempunyai *method play*{). Ekspresi suatu *method* dari *object* atau *movie clip* mengikuti pola yang sama. *Dot syntax* juga digunakan untuk dua alias khusus yaitu *_parent* dan *_root*. Contohnya, *method play* dari *instance* tokoh1 menggerakkan *playhead* pada *Timeline* dari tokoh1, seperti pernyataan berikut :

```
tokoh1.play();
```

1. _root

a. *Alias _root* berhubungan dengan *Timeline* utama.

b. *Alias root* dapat digunakan untuk membuat *target path* yang *absolut.*

Contohnya, pernyataan berikut memanggil fungsi unpause pada *movie clip* choose.

```
_root.choose.unpause();
```

2. _parent

a. *Alias parent* mengacu pada *movie clip* bila mempunyai hirarki.

b. *Alias _parent* dapat digunakan untuk membuat *target path* yang relatif.

Contohnya, bila *movie clip* burung terdapat dalam *movie clip* binatang, pernyataan *instance movie clip* burung memerintahkan binatang untuk berhenti :

```
_parent.stop ( );
```

2.6.4.2 Kurung Kurawal

Pernyataan pada *ActionScript* dikelompokkan bersama ke dalam satu blok dengan kurung kurawal ({ }), seperti *script* berikut :

```
on (press) {
    startDrag(_root.A3);
    }
```

2.6.5 Tipe Data

Tipe data menggambarkan macam elemen informasi yang dapat ditangani oleh variabel dalam *ActionScript*. Terdapat dua macam tipe data yaitu [SUT02] :

1. Tipe data primitif (string, number, dan Boolean)

   Tipe data yang mempunyai nilai konstanta, sehingga dapat menangani nilai aktual dan elemen yang ada.

2. Tipe data referensi (*movie clip* dan *object*)

   Tipe data yang mempunyai nilai yang dapat berubah, sehingga berisi referensi dan nilai aktual yang dimiliki oleh elemen.

Parameter Event dapat berisi :

1. *Load* dan *Unload* perintah akan dijalankan ketika pertama kali movie clip di-load dan di-remove oleh time line.
2. *enterFrame*, perintah akan dijalankan secara terus-menerus sepanjang frame yang aktif.
3. *mouseMove*, *mouseDown*, *mouseUp* perintah akan dijalankan ketika mouse digerakkan, ditekan dan dilepaskan.

4. *keyDown*, perintah akan dijalankan ketika mendeteksi adaya tombol yang ditekan dengan menggunakan action Key.getCode.
5. *keyUp*, perintah akan dijalankan ketika mendeteksi adanya tombol yang dilepaskan dengan menggunakan action Key.getCode.

5.7 *Script* Basis *Agent*

Dalam suatu perancangan aplikasi pembuatan *game*, agent juga memegang peranan yang sangat penting. Dalam rancang bangun *game* mobil balap ini, musuh akan digerakkan dan diperintah oleh *script agent*. *Script agent* diberikan untuk mengatur seluruh kepintaran dan kemampuan musuh selama *game* dijalankan. *Script agent* game ditempatkan di awal permainan dan pada *movie clip* mobil musuh yang akan ditampilkan pada area *race*. *Script agent* pada musuh akan diberikan berbeda-beda di tiap level berdasarkan tingkat kepandaian dan kesulitan area *race*. Perintah *agent* yang diberikan yaitu untuk menghalangi dan mengikuti pergerakan mobil pemain. Bila pemain bergerak ke kanan, musuh akan ikut ke kanan dan sebaliknya. Musuh juga dapat dengan sengaja melakukan benturan atau tabrakan dengan mobil pemain yang dapat menyebabkan pemain keluar dar area *rece*. Musuh tiap level akan ditampilkan secara *random*.

BAB III

METODOLOGI

3.1 Metode Analisis

Metode analisis yang digunakan adalah metode analisis dengan pendekatan terstruktural (*structural approach*) yang lengkap dengan alat dan teknik yang disesuaikan dengan kebutuhan. Arsitektur berusaha didefinisikan dengan sebaik mungkin agar pengembangan perangkat lunak dapat berjalan dengan terstruktur dan jelas meskipun mengalami peningkatan kompleksitas yang signifikan.

Pada tahap analisis ini, digunakan suatu alat untuk melakukan pemodelan agar pengembangan perangkat lunak dapat memenuhi semua kebutuhan pengguna dengan lengkap dan tepat. Pemodelan yang dilakukan menggunakan notasi-notasi UML (*Unified Modelling Languange*) yang merupakan standar dalam dunia industri perangkat lunak untuk melakukan visualisasi, perancangan dan pendokumentasian.

3.2 Hasil Analisis

3.2.1 Analisis Kebutuhan Proses

Aplikasi yang akan dibangun nantinnya diusahakan agar dapat menangani hal-hal yang sudah lazim terdapat pada sebuah *game* mobil balap. Beberapa proses yang dapat dilakukan adalah sebagai berikut :

1. Proses menampilkan *level*

   Pada proses ini, *game* akan menampilkan *level* tertentu dengan elemen-elemen pembangunnya, seperti gambar *background*, pergerakan jalan, objek meliputi pohon dan rumah, spidometer, bonus dan tampilan mobil musuh.

2. Proses menampilkan mobil

   Pada proses ini, *game* akan menampilkan mobil pemain yang akan ditampilkan pada layar.

3. Proses pergerakan mobil

   Proses ini dilakukan untuk menggerakan mobil pemain, sesuai dengan script yang diinputkan oleh pemain.

4. Proses menggerakan jalan

   Proses ini dilakukan untuk menggerakan jalan lurus dan berbelok untuk menimbulkan kesan mobil berjalan yang pada dasarnya adalah menggerakkan background menuju pemain.

5. Proses pergerakan objek

   Proses ini dilakukan untuk menggerakan pohon, orang dan objek lainnya untuk menimbulkan kesan mengikuti gerakan mobil yang berjalan.

6. Proses pergerakan musuh

   Proses ini dilakukan untuk menggerakkan karakter musuh. Musuh diberikan perintah (script) untuk dapat bersaing dengan pembalap, menentukan aksinya dengan karakter pemain.

7. Proses pengumpulan *score*

   Proses ini dilakukan untuk menampilkan bonus yang akan dikumpulkan oleh pemain dan musuh pada tiap level.

8. Proses penghitungan dan penyimpanan nilai

   Proses ini dilakukan untuk menyimpan nama pemain yang masuk daftar high *score*.

3.2.2 Analisis Kebutuhan Input

Beberapa komponen masukan utama yang diperlukan dalam membangun aplikasi yaitu :

1. Menu

   Beberapa masukan menu yang disediakan, antara lain : menu *Single Player*, menu *Setting*, menu *Help*, menu *High Score* dan menu *About*.

2. *Single Player*

   Masukan permainan, meliputi : jenis, warna dan pilihan modifikasi mobil .

3. *Setting*

   Masukan pengaturan terhadap *sound*, dimana pemain dapat mematikan atau menghidupkan suara selama permainan berlangsung.

4. Kendali mobil

   Masukan untuk mengahkan mobil sesuai dengan arah yang telah diinputkan (sesuai script).

5. Jumlah pembalap

   Masukan pembalap sebanyak tiga buah mobil. Pemain dan dua musuh.

3.2.3 Analisis Kebutuhan Output

Keluaran pada sistem ini adalah status permainan pada suatu kondisi tertentu ketika permainan sedang berlangsung.

1. *High Score*

   Menampilkan keluaran daftar score tertinggi yang pernah dicapai.

2. Status posisi

   Menampilkan keluaran status mobil pemain, ditandai dengan urutan angka di samping spidometer.

3. *Help*

   Menampilkan keluaran halaman bantuan.

4. Pergerakan jalan

   Menampilkan keluaran pergerakan jalan antara lain : lurus, tikungan kanan dan kiri.

5. Pergerakan objek

   Menampilkan keluaran pergerakan pohon, orang dan objek lainnya yang mengikuti pergerakan mobil yang melaju.

6. Background

   Menampilkan keluaran background yang berbeda dalam tiap level.

7. Bonus score

   Menampilkan keluaran bonus yang akan muncul secara random.

3.2.4 Analisis *User Interface*

Suatu sistem yang memiliki *interface* menarik dan informatif pada tampilan saat bermain, seperti :

1. Adanya indikasi untuk informasi jumlah score yang telah dikumpulkan.
2. Adanya indikasi untuk informasi geer melalui tampilan spedometer.

3.2.5 Analisis Kebutuhan Perangkat Lunak

Untuk pengembangan aplikasi *game* Mobil Balap dibutuhkan perangkat lunak lainnya, antara lain :

1. Sistem Operasi Windows

   Sistem operasi Windows merupakan sistem operasi yang *user friendly*, serta mendukung bahasa pemrograman yang akan digunakan dalam membangun sistem.

2. Macromedia Flash MX 2004

   Program ini adalah salah satu program terbaik dan terpopuler saat ini dalam pembuatan animasi. Serta mendukung bahasa pemrograman *action script*, sehingga program ini sangat cocok digunakan untuk membuat aplikasi *game* sederhana.

3. Adobe Photoshop CS

   Adobe Photoshop CS adalah program pengolah grafis dan editor gambar yang sangat baik dan mendukung hampir semua format gambar.

3.2.6 Analisis Kebutuhan Perangkat Keras

Untuk dapat melakukan pengembangan perangkat lunak dengan nyaman dan memadai, spesifikasi perangkat keras komputer juga harus diperhatikan. Berikut spesifikasi yang direkomendasikan:

1. Processor Pentium 4 1.4 GHz
2. RAM 512 MB
3. Free Hard Disk memory 300 MB
4. CD-ROM drive
5. VGA 128 MB
6. Monitor VGA dengan resolusi 1024 x 768
7. Keyboard dan mouse
8. Speaker

3.3 Perancangan Perangkat Lunak

3.3.1 Motode Perancangan

Metode perancangan perangkat lunak yang digunakan dalam pembuatan aplikasi *game* perang mobil ini menggunakan *Unified Modeling Language* (UML). Karena penggunaan UML (*Unified Modelling Language*) untuk melakukan analisis berorientasi obyek lebih sesuai dan lebih tepat daripada penggunaan model perancangan lainnya seperti DFD (*Data Flow Diagram*). Ketika membangun model dengan menggunakan UML, tidak hanya diciptakan melalui satu model tunggal. Perlu didekati melalui sekumpulan pandangan model

yang hampir independent, karena tidak ada satu pandangan tunggal yang memadai untuk sistem yang kompleks.

3.3.2 Hasil Perancangan

UML memiliki beberapa konsep dasar yang diabstraksikan dalam bentuk *structural classification*, *dynamic behavior*, dan *model management.* Hal terpenting dalam penggunaan UML adalah pembuatan diagram yang sesuai dengan analisis dan pengembangan sistem.

Rancang bangun aplikasi *game* Mobil Balap akan dibagi menjadi beberapa diagram, yaitu :

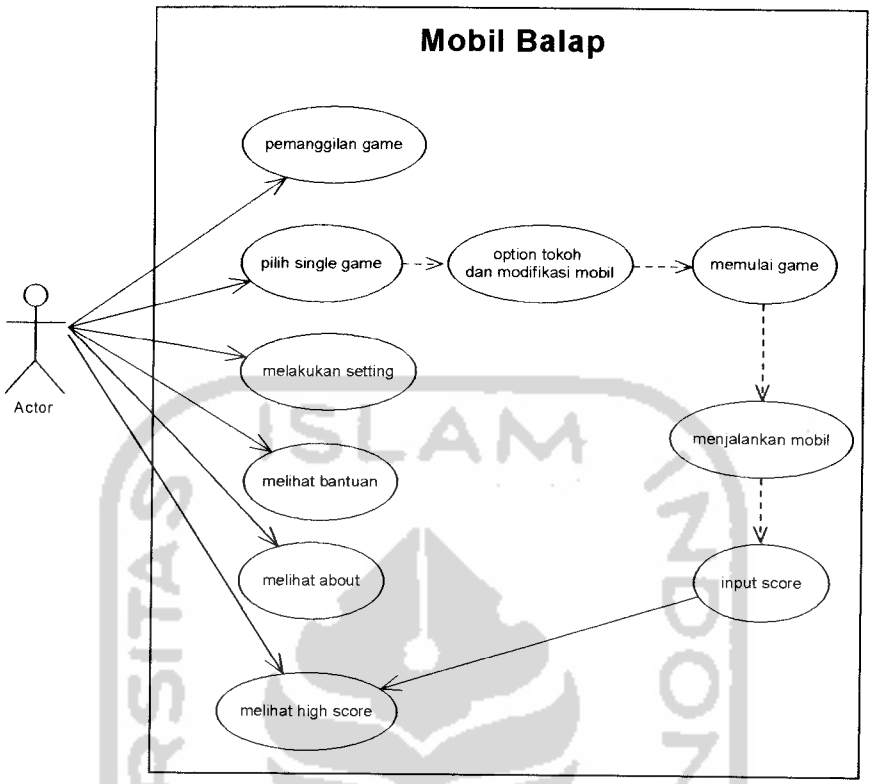
1. Diagram Use case
2. Diagram Kelas
3. Diagram Sekuen
4. Diagram Aktifitas

3.3.2.1 Use Case Diagram

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah "apa" yang diperbuat sistem, dan bukan "bagaimana". Sebuah *use case* merepresentasikan sebuah interaksi atau hubungan antara *aktor* dengan sistem.

Dalam perancangan sistem dengan diagram Use case ini, hanya terdapat satu *actor* saja, yaitu pemain (*player*). Didalam diagram Use case, digambarkan kebutuhan sistem dan interaksi antara *actor* dengan sistem.

Hubungan antara sistem dan pemain ditunjukkan oleh gambar 3.1.

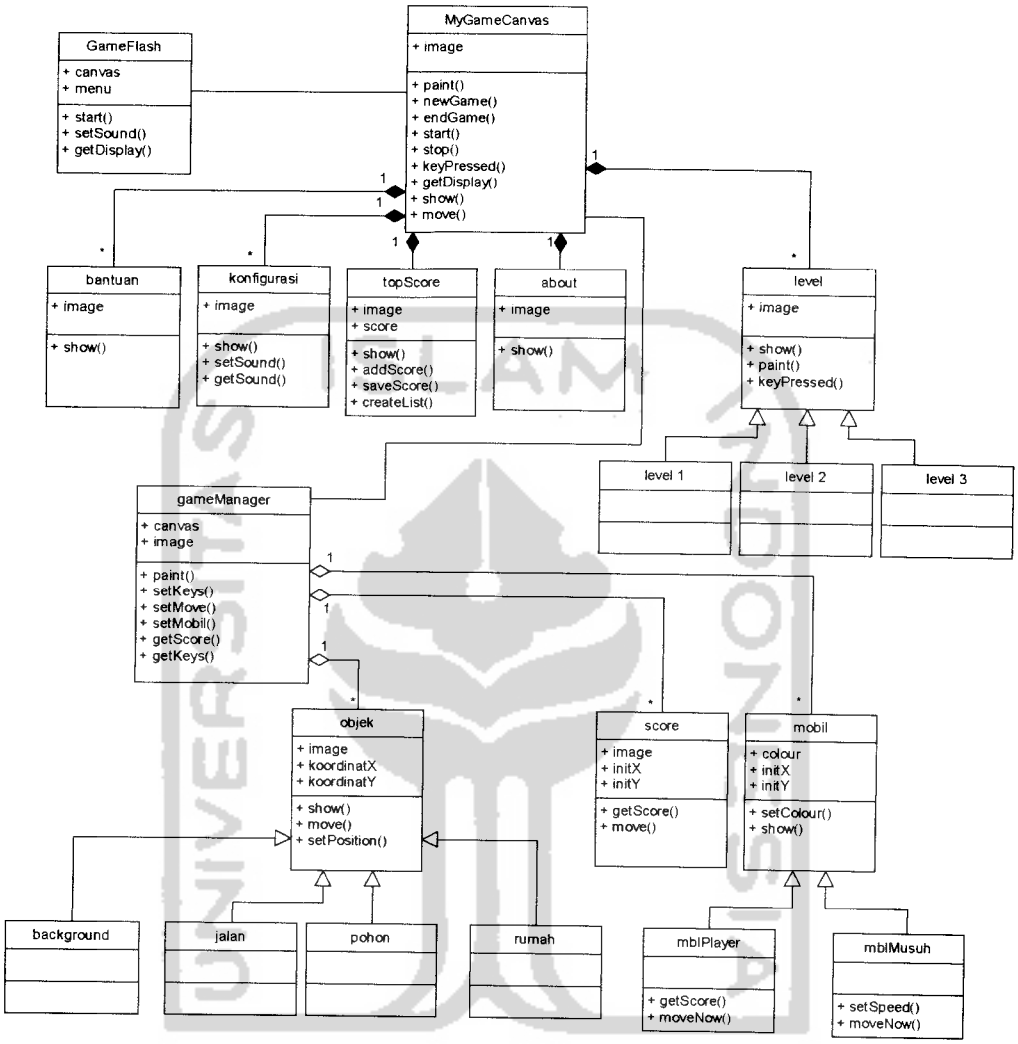


Gambar 3.1 Diagram *Use Case*

3.3.2.2 Diagram Kelas

Dalam UML digunakan notasi diagram Kelas untuk mengilustrasikan suatu aplikasi yang akan dibangun. Penggambaran dilakukan dengan visualisasi pada komponen aplikasi yang akan dibangun kedalam kelas-kelas yang memiliki nama kelas, atribut, *methode* dan tipe. Komponen tesebut saling terkoneksi sesuai dengan aturan dari aplikasi yang akan dibangun. Pada aplikasi *game* mobil balap ini terdapat beberapa kelas yang saling terkolaborasi baik dalam bentuk *Composition* maupun *Generalization* yang masing-masingnya digambarkan oleh

tanda garis penghubung yang berbeda. Lebih jelasnya dapat dilihat pada gambar 3.2.



Gambar 3.2 Diagram *Kelas*

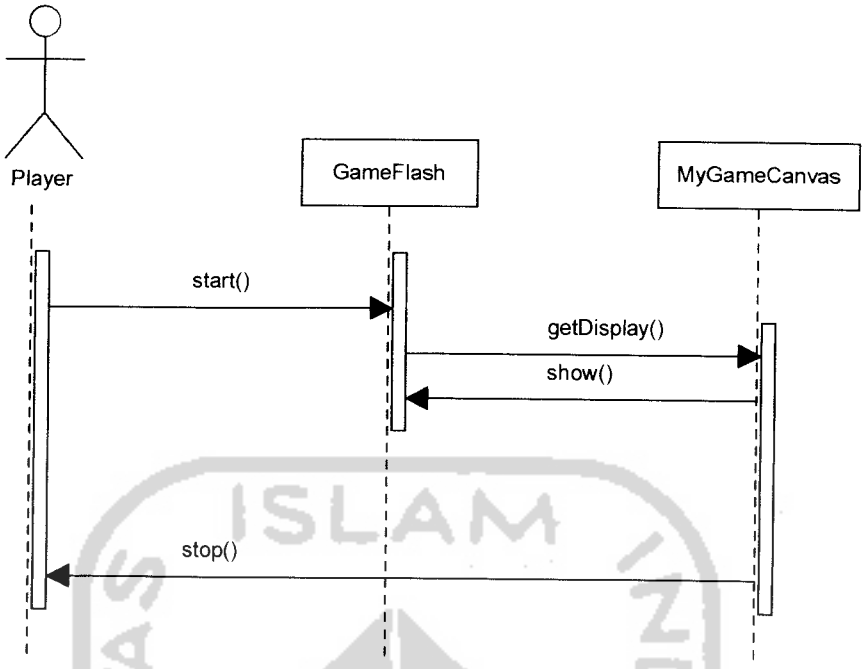
3.3.2.3 Diagram Sekuen

Diagram Sekuen menggambarkan interaksi antar objek di dalam dan di sekitar sistem berupa message yang digambarkan terhadap waktu. Diagram sekuen terdiri atas dimensi vertical (waktu) dan dimensi horizontal (objek-objek yang terkait).

Diagram sekuen digunakan untuk memodelkan skenario penggunaan. Skenario penggunaan adalah barisan kejadian yang terjadi selama satu eksekusi sistem. Dalam perancangan aplikasi mobil balap ini terdapat sembilan diagram sekuen yang merupakaran representasi dari sejumlah Use case yang telah dirancang pada diagram Use case sebelumnya, yaitu panggil *game*, single *game*-memilih tokoh (pemain) dan modifikasi, setting suara, help, about, memulai *game*, menjalankan mobil, *input score*, dan *high score*. Berikut ini adalah penjabaran diagram sekuen berdasarkan Use case nya masing-masing.

a. Pemanggilan *Game*

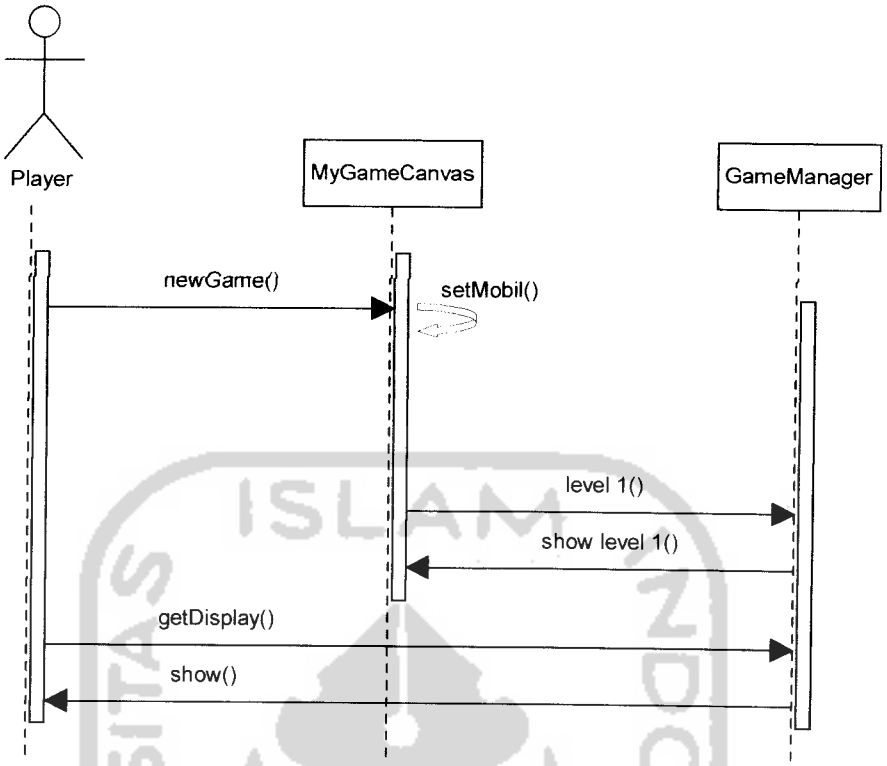
   Ketika pemain menjalankan *game* pertama kali, akan dipanggil kelas MainMenu untuk memunculkan menu-menu permainan yang terdiri dari *Single Player*, *Setting*, *Help*, *High Score* dan *About*.



Gambar 3.3 Diagram sekuen *pemanggilan game*

b. Pilih Single *Game*

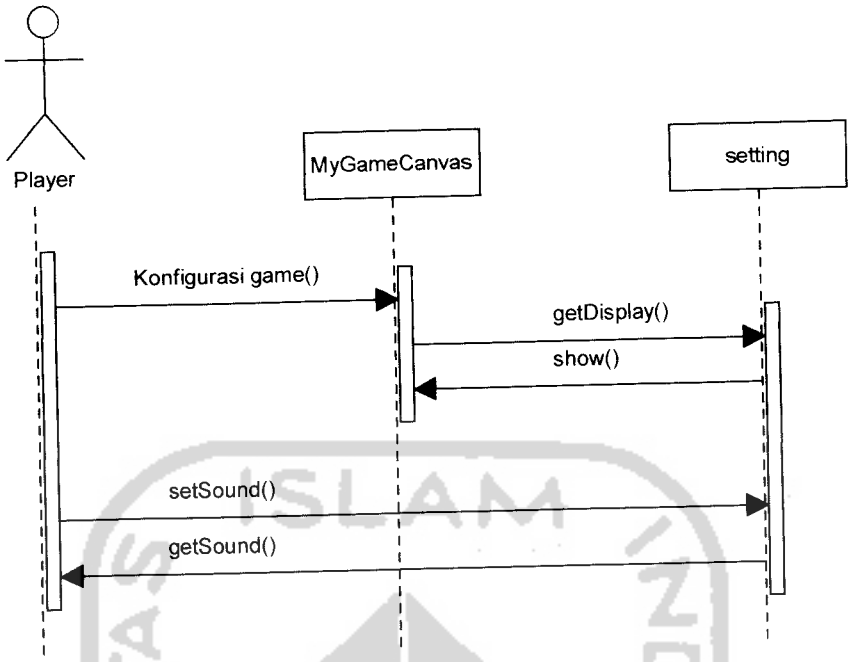
Apabila pemain memilih Single *Game*, maka akan tampil *kelas* halaman option tokoh dan modifikasi mobil. Setelah memilih tokoh dan mobil maka permainan akan dimulai dari *level* 1. Kelas menu akan memanggil kelas yang mengatur pemanggilan *level* 1, yaitu kelas My*Game*Canvas. Di dalam kelas My*Game*Canvas berisi aturan pada *level* 1 yang kemudian akan diterapkan pada kelas *Game*Manager, seperti jumlah musuh, jumlah score, pergerakan jalan, maupun kemampuan musuh berdasarkan basis agent yang diberikan (di input). Diagram sekuen *game* baru ditunjukkan oleh gambar 3.4.



Gambar 3.4 Diagram sekuen pilih *single game*

c. Melakukan Setting

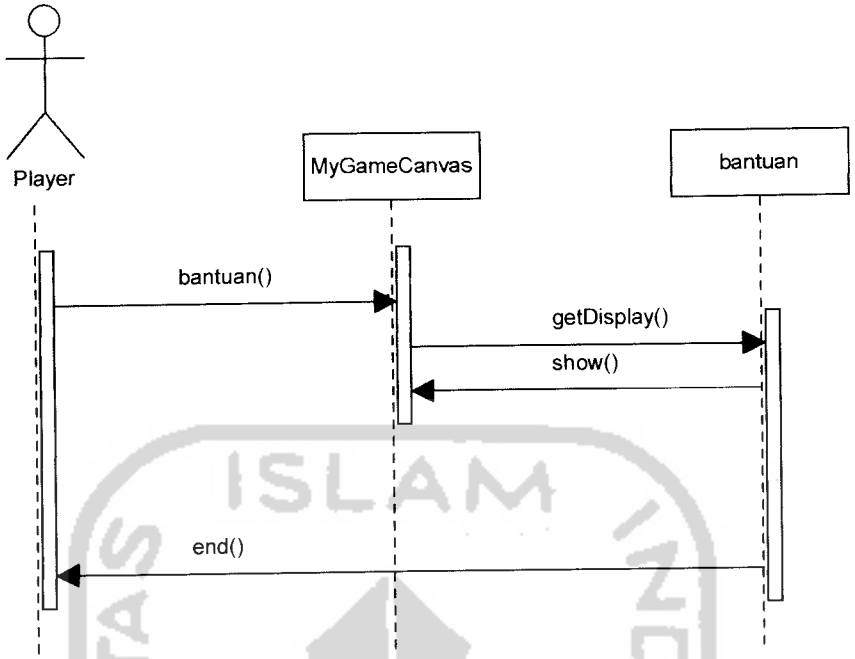
Diagram sekuen *setting* suara, yang berfungsi untuk mengaktifkan atau tidak mengaktifkan suara. Masukan dari pemain akan diatur oleh *method* setSound(). Diagram sekuen yang mendeskripsikan *option game* ditunjukkan oleh gambar 3.5.



Gambar 3.5 Diagram sekuen *melakukan setting*

d. Melihat Bantuan

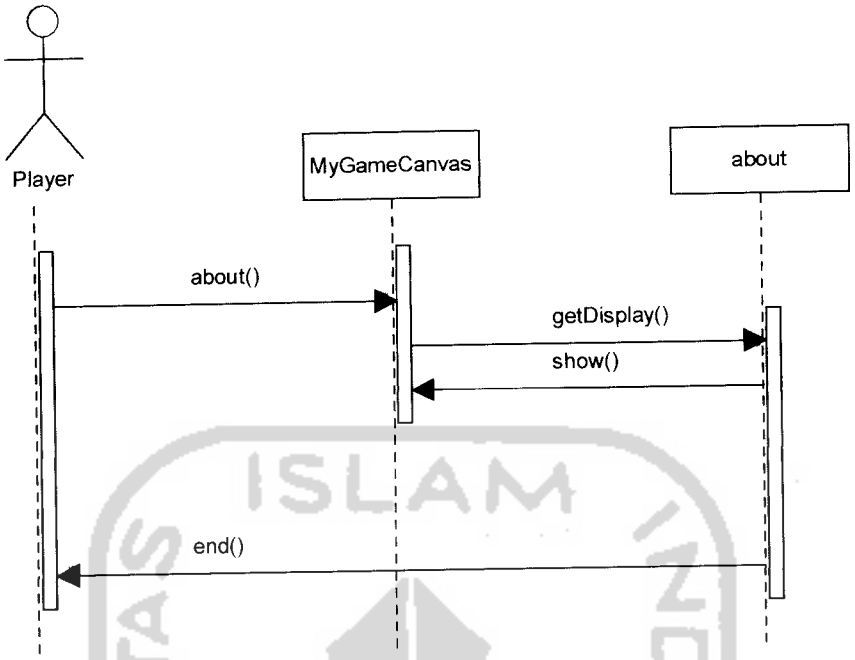
Diagram sekuen Bantuan, yang berfungsi untuk menampilkan informasi cara bermain. Dengan ini, pemain akan mengetahui cara bermain. Diagram sekuen yang mendeskripsikan bantuan ditunjukkan oleh gambar 3.6.



Gambar 3.6 Diagram sekuen *melihat bantuan*

e. Melihat *About*

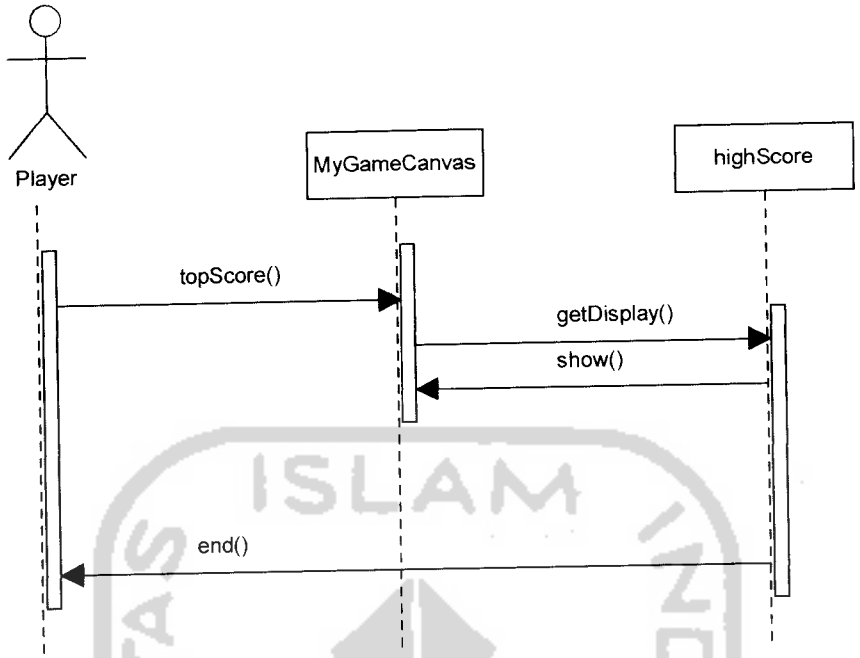
Diagram sekuen *about,* yang berfungsi untuk melihat informasi tentang pembuat *game* ini. Diagram sekuen yang mendeskripsikan about *game* ditunjukkan oleh gambar 3.7.



Gambar 3.7 Diagram sekuen melihat *about*

f. Melihat *High Score*

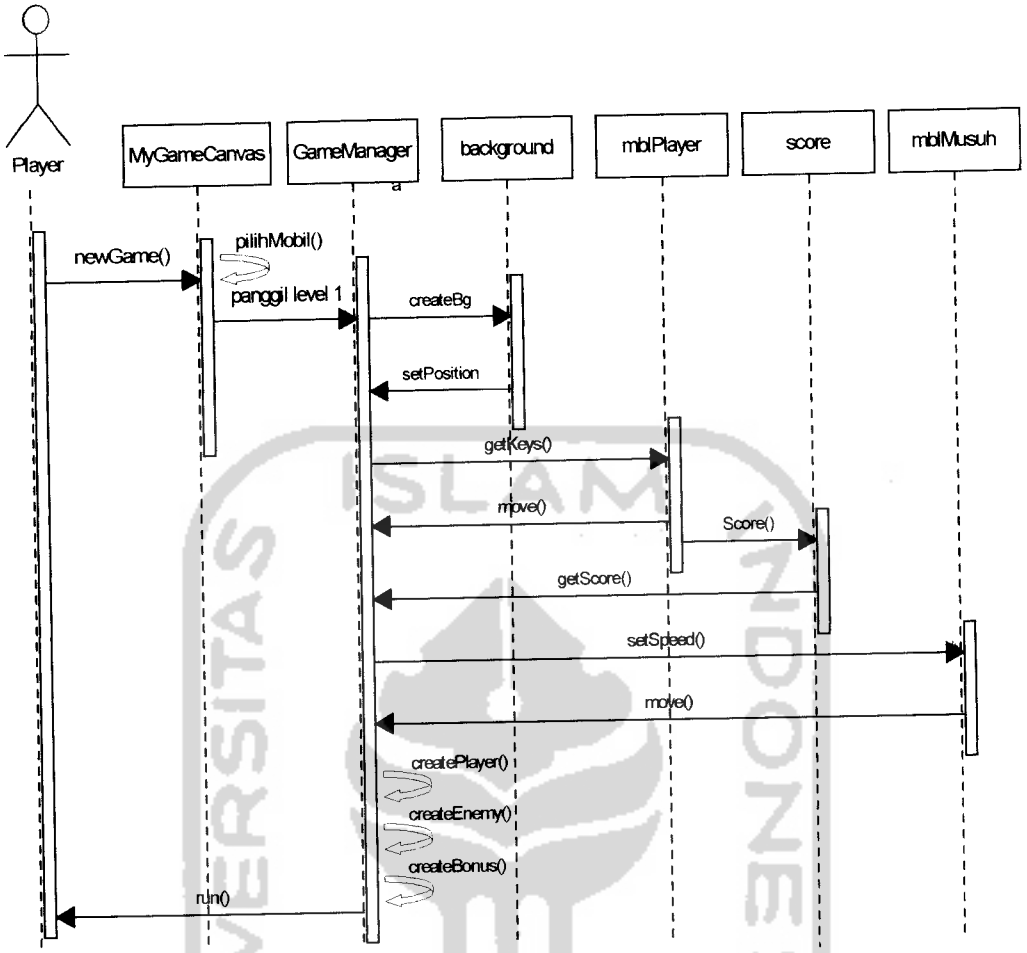
Diagram sekuen *high score*, yang berfungsi untuk menampilkan isi daftar pemain yang pernah bermain *game* ini. Dengan ini, pemain akan mengetahui berapa catatan *score* tertinggi yang pernah dicatatkan pemain. Diagram sekuen yang mendeskripsikan tampilkan *high score* ditunjukkan oleh gambar 3.8.



Gambar 3.8 Diagram sekuen *melihat High Score*

g. Memulai *Game*

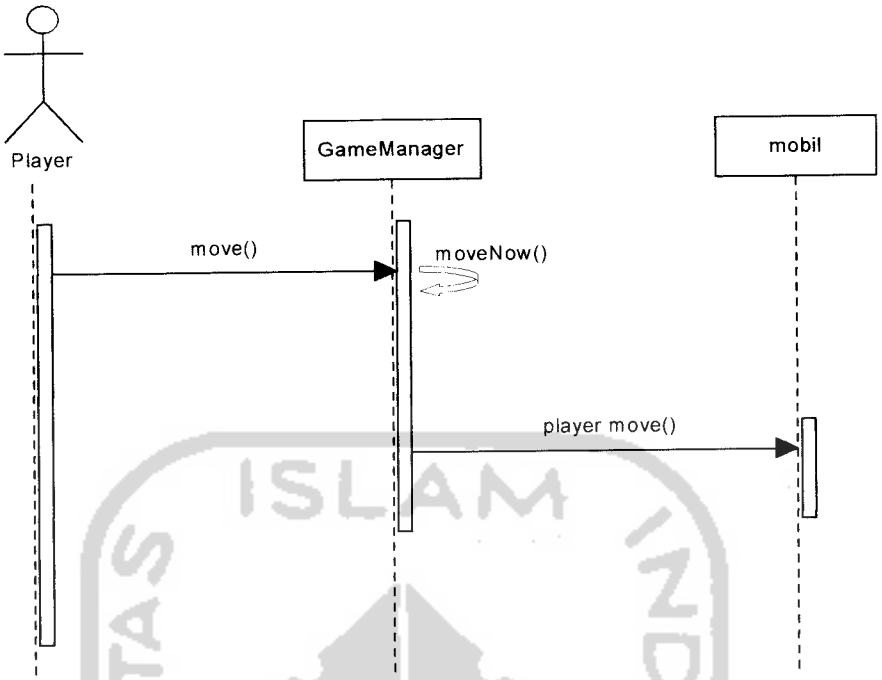
Diagram sekuen memulai *game*, yang berfungsi untuk menampilkan semua karakter yang dipanggil dalam *game* ini khususnya *level* 1. Dengan ini, pemain akan mengetahui tampilan awal saat *game* akan dimainkan. Diagram sekuen yang mendeskripsikan memulai *game* ditunjukkan oleh gambar 3.9.



Gambar 3.9 Diagram sekuen *memulai game*

h. Menjalankan mobil

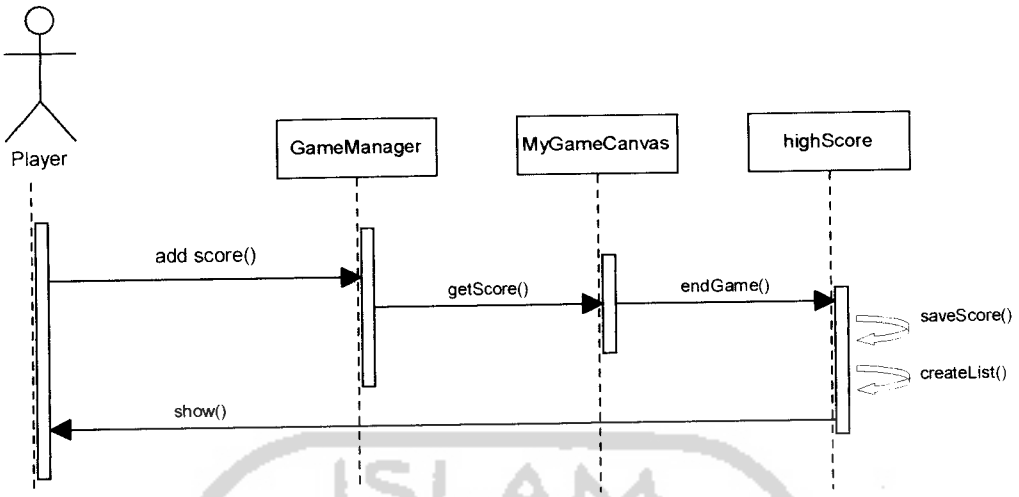
Diagram sekuen menjalankan mobil pemain, yang berfungsi untuk menggerakkan mobil. Dengan ini, pemain dapat menjalankan mobil maju, mundur (rem), kekanan ataupun kekiri sesuai keinginan. Diagram sekuen yang mendeskripsikan menjalankan mobil ditunjukkan oleh gambar 3.10.



Gambar 3.10 Diagram sekuen *menjalankan mobil*

i. Input Score

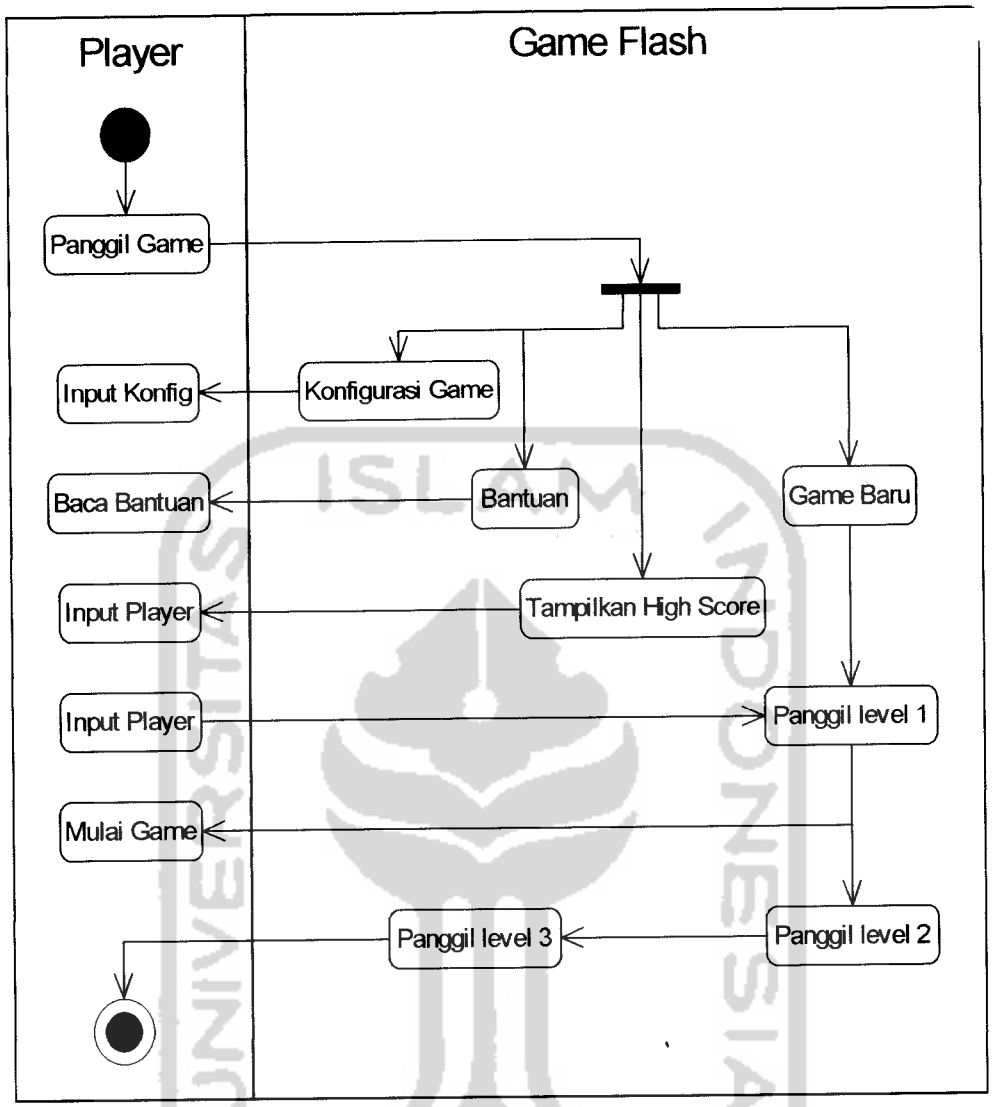
Diagram sekuen *input score*, yang berfungsi untuk menginputkan *score* pada *form high score* setelah pemain menyelesaikan *game* atau pemain kalah. Pada proses ini pemain dapat menginputkan nama, dan nama tersebut akan terpampang pada form high *score*. Diagram sekuen yang mendeskripsikan *input score* ditunjukkan oleh gambar 3.11.



Gambar 3.11 Diagram sekuen *input score*

3.3.2.4 Diagram Aktivitas

Diagram aktivitas mendekripsikan aksi-aksi dan hasilnya. Diagram aktivitas memiliki tiga elemen, yaitu *activity state*, transisi, objek. Dalam perancangan *game* ini, diagram aktivitas digunakan untuk menggambarkan bagaimana objek-objek di dalam *game* ini bekerja. Aktivitas-aktivitas dalam *game* ini ditunjukkan oleh gambar 3.12.



Gambar 3.12 Diagram aktivitas

3.4 Perancangan Antarmuka (*interface*)

Antarmuka (*interface*) merupakan bagian dimana terjadi komunikasi antara pengguna dengan sistem. Kemudahan pengguna dalam menggunakan sistem dapat dikatakan sebagai keberhasilan antarmuka dalam melakukan komunikasi dengan pengguna.

Pembentukan Agen dilakukan oleh perancang sistem dengan menempatkan *script agent* pada *button* perintah yang akan diberikan oleh pengguna. *Script agent* diberikan pada awal permainan terutama pada proses pemanggilan musuh secara *random* dan pengaturan tingkat kepandaian musuh tiap level yang berbeda dan saling terkait.

3.4.1 Perancangan Tampilan Menu Utama

Interface menu utama mempunyai beberapa menu yang dapat dapat dipilih salah satunya oleh player, yaitu:

a. Single *Game*

   Untuk memulai *game* baru.

b. Setting

   Untuk melakukan setting terhadap sound maupun music *game*.

c. Help

   Untuk melihat petunjuk permainan, baik itu tombol-tombol pengendali maupun misi dari permainan

d. High Score

   Untuk melihat nilai tertinggi dari permainan.

e. About

Untuk melihat nama dan versi *game*, tahun pembuatan *game* hingga nama pembuatnya.

f. Exit

Untuk keluar dari aplikasi *game*.

Rancangan interface menu utamanya seperti yang ditunjukkan pada gambar 3.13.



Gambar 3.13 Rancangan Tampilan Menu Utama

3.4.2 Perancangan Tampilan Single *Game*

Antarmuka inilah yang menjadi inti dari program permainan ini. Disinilah terjadi pertandingan mobil balap. Permainan pada single *game* terbagi dua bagian meliputi level biasa dan level bonus. Pada antarmuka ini terdapat beberapa item, yaitu:

a. Spidometer

Ketika mobil berjalan, jarum spidometer akan bergerak dan geer juga akan berpindah antara kecepatan 1 – 5. Spidometer hanya akan tampil pada level biasa.

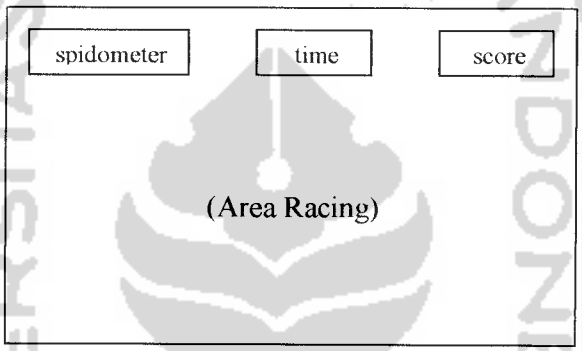
b. Score

Ketika permainan berlangsung, nilai score akan terus bertambah sesuai dengan banyaknya bonus yang berhasil dikumpulkan.

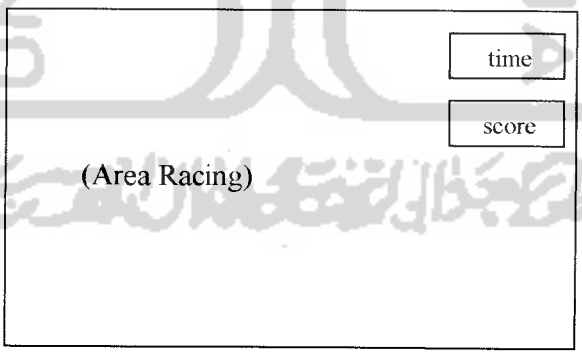
c. Time

Time akan berjalan mundur sesuai input waktu yang telah dimasukkan. Permainan akan berakhir ketika waktu yang ditentukan telah habis.

Tampilan ini dapat dilihat seperti pada gambar 3.14.



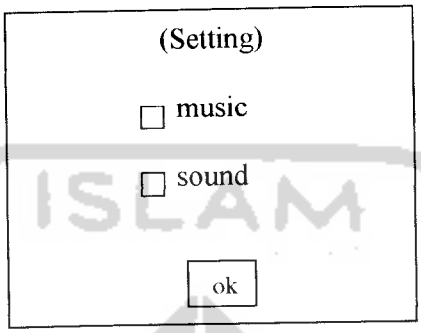
Gambar 3.14 Rancangan Tampilan Single *Game* Level Biasa



Gambar 3.15 Rancangan Tampilan Single *Game* Level Bonus

3.4.3 Rancangan Tampilan Setting

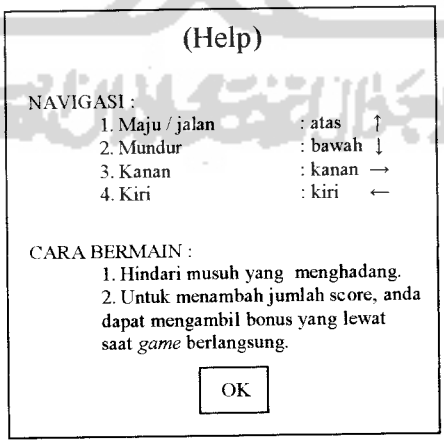
Interface setting digunakan untuk menyalakan atau mematikan suara dan music *game* selagi permainan berlangsung. Rancangan interface *setting game* ditunjukkan oleh gambar 3.15.



Gambar 3.16 Rancangan Tampilan *Setting*

3.4.4 Rancangan Tampilan Help

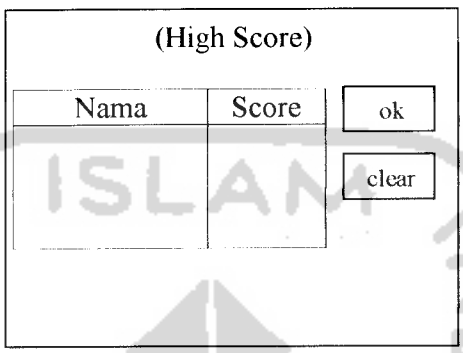
Interface *help* digunakan untuk menampilkan berbagai informasi tentang *game*, diantaranya petunjuk tentang apa yang seharusnya dilakukan oleh player disaat *game* berlangsung, hingga tombol-tombol yang harus dikuasai oleh player untuk mengendalikan mobil selagi *game* berlangsung. Rancangan interface *help* ditunjukkan oleh gambar 3.16.



Gambar 3.17 Rancangan Tampilan *help*

3.4.5 Rancangan Tampilan High Score

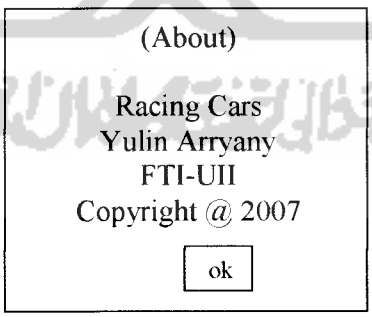
Interface *high sroce* digunakan *player* untuk melihat catatan score tertinggi yang pernah dibuat oleh *player*. Rancangan interface *high Score* ditunjukkan oleh gambar 3.17.



Gambar 3.18 Rancangan Tampilan *High Sroce*

3.4.6 Rancangan Tampilan About

Interface *about* digunakan untuk menampilkan informasi tentang nama dan versi *game*, tahun pembuatan serta nama pembuat *game*. Rancangan interface *about* ditunjukkan oleh gambar 3.18.



Gambar 3.19 Rancangan Tampilan *about*

BAB IV

IMPLEMENTASI DAN PEMBAHASAN

4.1 Implementasi

Tahap implementasi merupakan proses dimana rancangan bangun suatu perangkat lunak telah dibuat dan diterjemahkan ke dalam bahasa pemrograman tertentu sehingga menjadi suatu aplikasi yang siap dioperasikan, serta sesuai dengan fungsi dan tujuan dibangunnya perangkat lunak tersebut.

4.1.1 Batasan Implementasi

Untuk dapat mengimplementasikan perancangan sistem yang telah dilakukan diperlukan beberapa hal yaitu perangkat keras, perangkat lunak, dan antar muka.

Dalam proses pembuatan dan pengembangan sistem ini, batasan yang telah direncanakan antara lain :

1. Tingkat kesulitan (*difficulty*) *game* ini ditentukan dengan kepintaran musuh dalam tiap *level*, pergerakan musuh yaitu kemampuan musuh untuk dapat saling menghalangi dan memperebutkan bonus.
2. Permainan pada tiap *level* akan berhenti jika waktu telah habis. Tujuan atau *goal* tiap *level* adalah sama, yaitu mengumpulkan dan memperebukan bonus sebanyak banyaknya.

4.1.2.1 Antarmuka Menu Utama

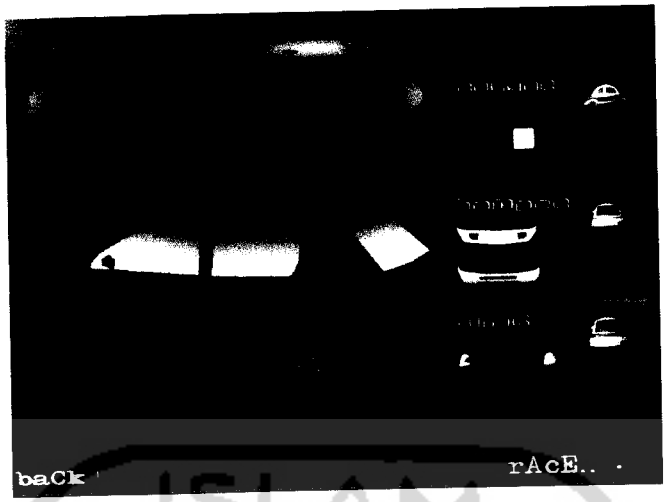
Pada tampilan menu terdapat tujuh menu utama, yaitu *Single Player*, *Setting*, *Help*, *High Score*, *About* dan *Exit*. Gambar 4.1 adalah gambar antarmuka menu utama.



Gambar 4.1 Antarmuka *Menu Utama*

4.1.2.2 Antarmuka Permainan

Antarmuka permainan ini akan muncul jika menu *single game* dipilih. Pada antarmuka ini akan menampilkan modifikasi mobil meliputi pemilihan jenis mobil, warna, bumber dan sayap. Pada tahap modifikasi level 1, pemain dibatasi dengan pilihan jenis dan warna mobil saja. Untuk modifikasi bumper, pemain harus menyelesaikan level 2. Begitu pula dengan modifikasi sayap, pemain harus melalui level 2 dan 3 terlebih dahulu. Gambar 4.2 adalah gambar antarmuka modifikasi mobil.



Gambar 4.2 Antarmuka modifikasi mobil

Indikasi jumlah waktu dan score terletak di bagian atas dari area permainan. Spidometer menujukkan kecepatan laju mobil di lintasan. Tombol *Quit* yang berfungsi untuk membatalkan permainan dan langsung kembali ke menu utama. Gambar 4.3 adalah gambar antarmuka indikasi waktu, score, spidometer dan tombol *Quit*.



Gambar 4.3 Antarmuka indikasi

Antarmuka permainan setelah pemilihan modifikasi yaitu tampilan-tampilan pada tiap *level*, yaitu *level* 1, *level* 2, dan *level* 3. Jika pemain dapat melewati *level* 1 dan bonus *level* 1, maka akan masuk ke *level* 2, dan seterusnya. Berikut ini merupakan tampilan *game* pada *level* 1, *level* 2 dan *level* 3. Gambar 4.4 dan 4.5 adalah gambar antarmuka permainan *level* 1.

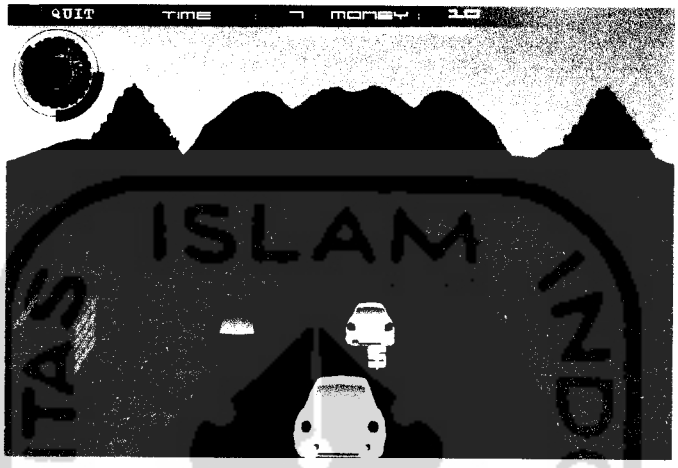


Gambar 4.4 Antarmuka permainan *level* 1

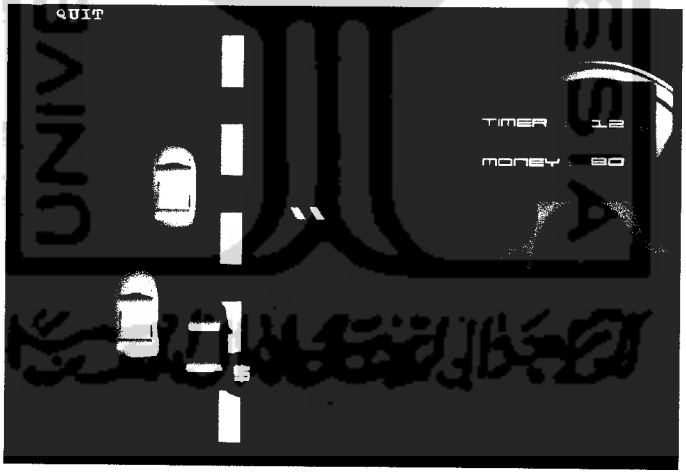


Gambar 4.5 Antarmuka bonus *level* 1

Pada *level* 2 hampir sama dengan tahap level 1, pada level 2 mobil dapat di modifikasi ulang dengan tambahan pilihan *bumper* pada modifikasi mobil. Musuh yang dihadapi bertambah pintar begitu juga dengan pola pergerakannya. Gambar 4.6 dan 4.7 adalah gambar antarmuka permainan *level* 2.

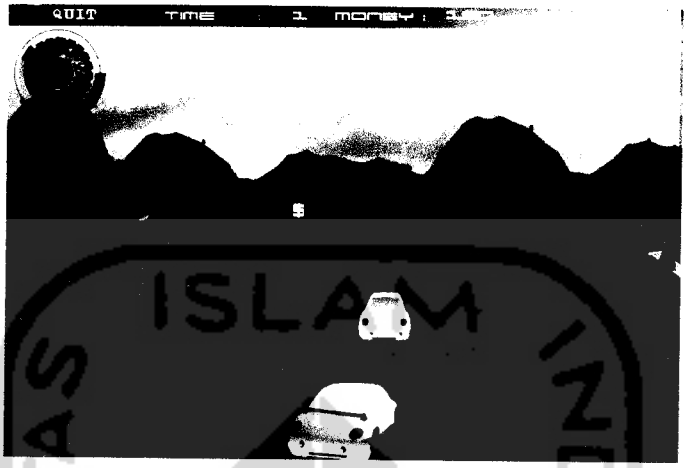


Gambar 4.6 Antarmuka permainan *level* 2

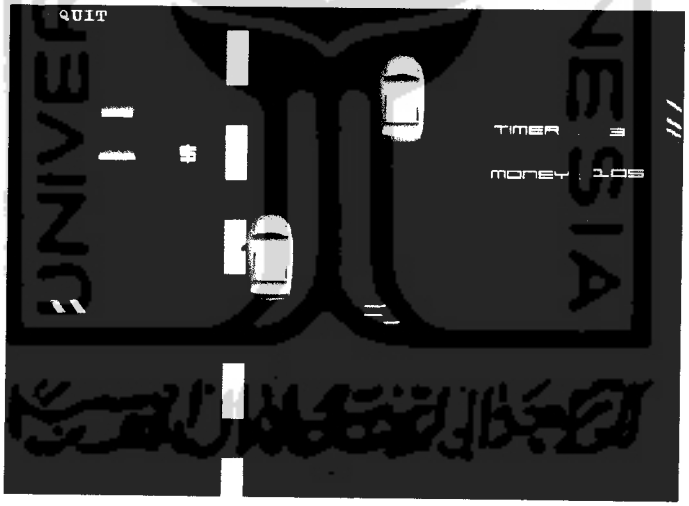


Gambar 4.7 Antarmuka bonus *level* 2

Pada *level* 3 hampir serupa pada *level 1 dan level 2*. Kepintaran musuh akan semakin bertambah. Tambahan pada modifikasi mobil yaitu pada pilihan sayap. Gambar 4.8 dan 4.9 adalah gambar antarmuka permainan *level* 3.



Gambar 4.8 Antarmuka permainan *level* 3



Gambar 4.9 Antarmuka bonus *level* 3

4.1.2.3 Antarmuka Setting

Antarmuka *Setting* digunakan untuk menyalakan atau mematikan musik dan suara permainan. Gambar 4.10 adalah gambar antarmuka *Setting*.



Gambar 4.10 Antarmuka *Setting*

4.1.2.4 Antarmuka Help

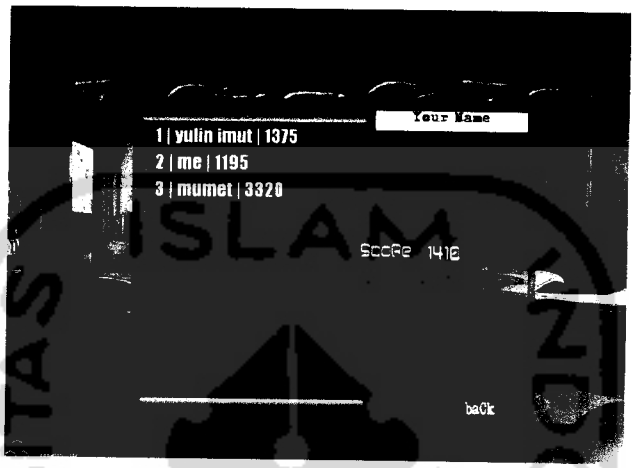
Antarmuka *Help* digunakan untuk menampilkan petunjuk permainan mulai dari tombol-tombol pengendali permainan. Gambar 4.11 adalah gambar antarmuka *Help*.



Gambar 4.11 Antarmuka *Help*

4.1.2.5 Antarmuka High Score

Antar muka *High Score* digunakan untuk mengisi, menampilkan dan menghapus nama dengan nilai tertinggi. Terdapat juga tombol untuk kembali ke menu utama. Gambar 4.12 adalah gambar antarmuka *High Score*.



Gambar 4.12 Antarmuka *High Score*

4.1.2.6 Antarmuka About

Antar muka *About* digunakan untuk menampilkan nama *game*, perancang *game* serta tahun pembuatannya. Gambar 4.13 adalah gambar antarmuka *About*.



Gambar 4.13 Antarmuka *About*

4.1.3 Implementasi Prosedural

4.1.3.1 Script Main

Script Main memberikan perintah awal untuk jalannya permainan. Meliputi kecepatan awal dan maksimum, jarak, time, score dan hasil akhir permainan.

```
kecepatan = 0;
rusak = 100;
jarak = 100;

money = 0;
scorelev1 = 0;
scorelev2 = 0;
scorelev3 = 0;
score2 = 0;
time = 10;
time2 = 10;
time3 = 10;
hasil = "";
stop();
```

4.1.3.2 Script Pilihan Modifikasi

Script Modifikasi mobil terdiri dari 3 pilihan yaitu pilihan warna, bumper dan sayap. Script untuk pilihan warna terbagi menjadi 3, meliputi warna hitam, putih dan biru.

Berikut script untuk pilihan warna hitam :

```
on (release) {
      if (mobil == 2)gotoAndPlay("satu");
      if (mobil == 3)gotoAndPlay("satu");
      if (mobil == 5)gotoAndPlay("empat");
      if (mobil == 6)gotoAndPlay("empat");
      if (mobil == 8)gotoAndPlay("tujuh");
      if (mobil == 9)gotoAndPlay("tujuh");
}
on (rollOver) {

}
on (rollOut) {
      hint = "";
}
```

Berikut script untuk pilihan warna putih :

```
on (release) {
      if (mobil == 1)gotoAndPlay("dua");
      if (mobil == 3)gotoAndPlay("dua");
      if (mobil == 4)gotoAndPlay("lima");
      if (mobil == 6)gotoAndPlay("lima");
      if (mobil == 7)gotoAndPlay("dlpn");
      if (mobil == 9 )gotoAndPlay("dlpn");
}
on (rollOver) {

}
on (rollOut) {
      hint = "";
}
```

Berikut script untuk pilihan warna biru :

```
on (release) {
      if (mobil == 1)gotoAndPlay("tiga");
      if (mobil == 2)gotoAndPlay("tiga");
      if (mobil == 4)gotoAndPlay("enam");
      if (mobil == 5)gotoAndPlay("enam");
      if (mobil == 7)gotoAndPlay("smbl");
      if (mobil == 8 )gotoAndPlay("smbl");
}
on (rollOver) {

}
on (rollOut) {
      hint = "";
}
```

Script untuk pilihan bumper terbagi menjadi 2 pilihan meliputi bumper1 dan bumper 2.

Berikut script untuk pilihan modifikasi bumper 1 :

```
on (release) {
      if (mobil == 1) {
            gotoAndPlay("htm1BA");
      }
      if (mobil == 13) {
            gotoAndPlay("htm1BA");
      }
      if (mobil == 2) {
            gotoAndPlay("pth1BA");
      }
      if (mobil == 14) {
            gotoAndPlay("pth1BA");
      }
      if (mobil == 3) {
            gotoAndPlay("br1BA");
      }
      if (mobil == 15) {
            gotoAndPlay("br1BA");
      }
      if (mobil == 4) {
            gotoAndPlay("htm2BA");
      }
      if (mobil == 19) {
            gotoAndPlay("htm2BA");
      }
      if (mobil == 5) {
            gotoAndPlay("pth2BA");
      }
      if (mobil == 20) {
            gotoAndPlay("pth2BA");
      }
      if (mobil == 6) {
            gotoAndPlay("br2BA");
      }
      if (mobil == 21) {
            gotoAndPlay("br2BA");
      }
      if (mobil == 7) {
            gotoAndPlay("htm3BA");
      }
      if (mobil == 25) {
            gotoAndPlay("htm3BA");
      }
      if (mobil == 8) {
            gotoAndPlay("pth3BA");
      }
```

```
        if (mobil == 26) {
              gotoAndPlay("pth3BA");
        }

        if (mobil == 9) {
              gotoAndPlay("br3BA");
        }
        if (mobil == 27) {
              gotoAndPlay("br3BA");
        }
        if (mobil == 31) {
              gotoAndPlay("htm1BSA");
        }
        if (mobil == 32) {
              gotoAndPlay("pth1BSA");
        }
        if (mobil == 33) {
              gotoAndPlay("br1BSA");
        }
        if (mobil == 37) {
              gotoAndPlay("htm2BSA");
        }
        if (mobil == 38) {
              gotoAndPlay("pth2BSA");
        }
        if (mobil == 39) {
              gotoAndPlay("br2BSA");
        }
        if (mobil == 43) {
              gotoAndPlay("htm3BSA");
        }
        if (mobil == 44) {
              gotoAndPlay("pth3BSA");
        }
        if (mobil == 45) {
              gotoAndPlay("br3BSA");
        }
}
```

Berikut script untuk pilihan modifikasi bumper 2 :

```
on (release) {
      if (mobil == 1) {
            gotoAndPlay("htm1BBA");
      }
      if (mobil == 10) {
            gotoAndPlay("htm1BBA");
      }
      if (mobil == 2) {
            gotoAndPlay("pth1BBA");
      }
      if (mobil == 11) {
            gotoAndPlay("pth1BBA");
      }
      if (mobil == 3) {
            gotoAndPlay("br1BBA");
      }
      if (mobil == 12) {
            gotoAndPlay("br1BBA");
      }
      if (mobil == 4) {
            gotoAndPlay("htm2BBA");
      }
      if (mobil == 16) {
            gotoAndPlay("htm2BBA");
      }
      if (mobil == 5) {
            gotoAndPlay("pth2BBA");
      }
      if (mobil == 17) {
            gotoAndPlay("pth2BBA");
      }
      if (mobil == 6) {
            gotoAndPlay("br2BBA");
      }
      if (mobil == 18) {
            gotoAndPlay("br2BBA");
      }
      if (mobil == 7) {
            gotoAndPlay("htm3BBA");
      }
      if (mobil == 22) {
            gotoAndPlay("htm3BBA");
      }
      if (mobil == 8) {
            gotoAndPlay("pth3BBA");
      }
      if (mobil == 23) {
            gotoAndPlay("pth3BBA");
      }
      if (mobil == 9) {
            gotoAndPlay("br3BBA");
      }
```

```
        if (mobil == 24) {
            gotoAndPlay("br3BBA");
        }
        if (mobil == 28) {
            gotoAndPlay("htm1BBSA");
        }
        if (mobil == 29) {
            gotoAndPlay("pth1BBSA");
        }
        if (mobil == 30) {
            gotoAndPlay("br1BBSA");
        }
        if (mobil == 34) {
            gotoAndPlay("htm2BBSA");
        }
        if (mobil == 35) {
            gotoAndPlay("pth2BBSA");
        }
        if (mobil == 36) {
            gotoAndPlay("br2BBSA");
        }
        if (mobil == 40) {
            gotoAndPlay("htm3BBSA");
        }
        if (mobil == 41) {
            gotoAndPlay("pth3BBSA");
        }
        if (mobil == 42) {
            gotoAndPlay("br3BBSA");
        }
}
```

Script untuk pilihan sayap hanya terdiri 1 pilihan. Berikut script untuk pilihan modifikasi sayap:

```
on (release) {
      if (mobil == 10) {
            gotoAndPlay("htm1BSA");
      }
      if (mobil == 11) {
            gotoAndPlay("pth1BSA");
      }
      if (mobil == 12) {
            gotoAndPlay("br1BSA");
      }
      if (mobil == 13) {
            gotoAndPlay("htm1BBSA");
      }
      if (mobil == 14) {
            gotoAndPlay("pth1BBSA"); }
```

```
        if (mobil == 15) {
            gotoAndPlay("br1BBSA");
        }
        if (mobil == 16) {
            gotoAndPlay("htm2BSA");
        }
        if (mobil == 17) {
            gotoAndPlay("pth2BSA");
        }
        if (mobil == 18) {
            gotoAndPlay("br2BSA");
        }
        if (mobil == 19) {
            gotoAndPlay("htm2BBSA");
        }
        if (mobil == 20) {
            gotoAndPlay("pth2BBSA");
        }
        if (mobil == 21) {
            gotoAndPlay("br2BBSA");
        }
        if (mobil == 22) {
            gotoAndPlay("htm3BSA");
        }
        if (mobil == 23) {
            gotoAndPlay("pth3BSA");
        }
        if (mobil == 24) {
            gotoAndPlay("br3BSA");
        }
        if (mobil == 25) {
            gotoAndPlay("htm3BBSA");
        }
        if (mobil == 26) {
            gotoAndPlay("pth3BBSA");
        }
        if (mobil == 27) {
            gotoAndPlay("br3BBSA");
        }
}
```

4.1.3.3 Script Tombol RACE

Script RACE digunakan untuk memulai permainan suatu level. Perintah panggilan *game* tergantung level yang akan dimainkan. Mobil yang telah di modifikasi akan dipanggil ke area permainan berdasarkan nama variable mobil.

Berikut script untuk tombol race :

```
on (release) {
      //nama mobil
      if (mobil == 1 ) {
            gotoAndStop("readylev1");
            this.mobilnya.attachMovie("mobil1", "mobila", 1);
            this.musuhx.attachMovie("mobil" + _root.randomEnemy, "mobilb", 2);
            this.musuhx2.attachMovie("mobil" + _root.randomEnemy2, "mobilc", 3);
      } else if (mobil == 2) {
            gotoAndStop("readylev1");
            this.mobilnya.attachMovie("mobil2", "mobila", 1);
            this.musuhx.attachMovie("mobil" + _root.randomEnemy, "mobilb", 2);
            this.musuhx2.attachMovie("mobil" + _root.randomEnemy2, "mobilc", 3);
      } else if (mobil == 3) {
            gotoAndStop("readylev1");
            this.mobilnya.attachMovie("mobil3", "mobila", 1);
            this.musuhx.attachMovie("mobil" + _root.randomEnemy, "mobilb", 2);
            this.musuhx2.attachMovie("mobil" + _root.randomEnemy2, "mobilc", 3);
      } else if (mobil == 4) {
            gotoAndStop("readylev1");
            this.mobilnya.attachMovie("mobil4", "mobila", 1);
            this.musuhx.attachMovie("mobil" + _root.randomEnemy, "mobilb", 2);
            this.musuhx2.attachMovie("mobil" + _root.randomEnemy2, "mobilc", 3);
      } else if (mobil == 5) {
            gotoAndStop("readylev1");
            this.mobilnya.attachMovie("mobil5", "mobila", 1);
            this.musuhx.attachMovie("mobil" + _root.randomEnemy, "mobilb", 2);
            this.musuhx2.attachMovie("mobil" + _root.randomEnemy2, "mobilc", 3);
      }
      .
      .
      .
      .
      else if (mobil == 45) {
            gotoAndStop("readylev1");
            this.mobilnya.attachMovie("mobil45", "mobila", 1);
            this.musuhx.attachMovie("mobil" + _root.randomEnemy, "mobilb", 2);
            this.musuhx2.attachMovie("mobil" + _root.randomEnemy2, "mobilc", 3);
      }
}
```

4.1.3.4 Script pergerakan mobil pemain

Script pergerakan mobil terbagi menjadi 2 bagian, script pada level biasa dan level bonus. Script pada level biasa menggambarkan pergerakan mobil saat bergerak ka kanan dan ke kiri. Script pada level bonus hanya menggambarkan pergerakan mobil dari sudut pandang atas.

Berikut script mobil pemain pada level biasa :

```
onClipEvent (load) {
	kecepatanmax = 100;
	kec = 0;
	kekanan = 0;
	kekiri = 0;
	geser = 10;
	gas = 0;
	jarak = 0;
	this.gotoAndStop(1);
}

// control mobil
onClipEvent (enterFrame) {
	if (Key.isDown(Key.UP)) {
		if (_root.kecepatan<kecepatanmax) {
			_root.kecepatan += 1;
			gas = 1;
		}
	}
	if (Key.isDown(Key.DOWN)) {
		if (_root.kecepatan>=5) {
			_root.kecepatan -= 6;
		}
	}
	if (Key.isDown(Key.LEFT)) {
		kekanan = 0;
		kekiri += 1;
		if (_x>75) {
			_x -= geser;
		}
		if (_root.jalan.geserkiri=1) {
			_root.jalan._x += 3;
		}
	}
	if (Key.isDown(Key.RIGHT)) {
		kekiri = 0;
		kekanan += 1;
		if (_x<350) {
```

```
                _x += geser;
        }
if (_root.jalan.geserkanan=1) {
                _root.jalan._x -= 3;
        }
}

//normal
if (kekanan == 0 or kekiri == 0) {
        this.mobila.gotoAndStop(1);
}

// miring kanan
if (kekanan>1) {
        this.mobila.gotoAndStop(1);
}
if (kekanan>3) {
        this.mobila.gotoAndStop(2);
        geser = 8;
}
if (kekanan>6) {
        this.mobila.gotoAndStop(3);
        geser = 5;
}
if (kekanan>10) {
        this.mobila.gotoAndStop(4);
        geser = 2;
}

// miring kiri
if (kekiri>1) {
        this.mobila.gotoAndStop(5);
}
if (kekiri>3) {
        this.mobila.gotoAndStop(6);
        geser = 8;
}
if (kekiri>6) {
        this.mobila.gotoAndStop(7);
        geser = 5;
}
if (kekiri>10) {
        this.mobila.gotoAndStop(8);
        geser = 2;
}
if (_root.kecepatan>1 and gas == 0) {
        kec += 1;
        if (kec>1) {
                _root.kecepatan -= 3;
                kec = 0;
        }
}
```

```
//status posisi
onClipEvent (enterFrame) {
      if ((this._y<_root.musuhx._y) and
      (this._y<_root.musuhx2._y)) {
            _root.raceposition = "1/3";
      } else if ((this._y<_root.musuhx._y) and
      (this._y>=_root.musuhx2._y)) {
            _root.raceposition = "2/3";
      } else if ((this._y<_root.musuhx2._y) and
      (this._y>=_root.musuhx._y)) {
            _root.raceposition = "2/3";
      } else {
            _root.raceposition = "3/3";
      }

//tabrakn dgn musuh
            if (this.hitTest(_root.musuhx)) {
                  _root.musuhx._x += 4;
                  this._x += 4;
            } else if (this.hitTest(_root.musuhx2)) {
                  _root.musuhx2._x -= 4;
                  this._x -= 4;
            }
            if
(this.hitTest(_root.jalan.road1.grs_kiri)) {

//jika menabrak sisi jalan sebelah kiri
                  this._x += 10;
            }
            if
(this.hitTest(_root.jalan.road1.grs_knn)) {

//jika menabrak sisi jalan sebelah knan
                  this._x -= 10;
            }
      }
}
```

Berikut script mobil pemain pada level bonus :

```
onClipEvent (enterFrame) {
      if(Key.isDown(Key.LEFT)&& _root.jalann.speed &&
      this._x > -6.65)
         this._x -= 10
      if(Key.isDown(Key.RIGHT)&& _root.jalann.speed &&
      this._x < 515)
         this._x += 10
      if(Key.isDown(Key.UP))
         this._y -= 5
       if(Key.isDown(Key.DOWN))
          this._y += 5
      if(Key.isDown(Key.SPACE))
         this._y -= 10 }
```

4.1.3.5 Script pergerakan mobil musuh

Script pergerakan musuh terbagi 2 berdasarkan jenis levelnya. Script musuh pada level biasa dan pada level bonus.

Berikut script area lintasan pada level biasa :

```
onClipEvent (enterFrame) {
      if (_root.enStart<=80) {
            _root.enStart++;
      } else {
            this._visible = true;
            if (this._x<(_root.mobilnya._x-1)) {
                  this._x++;
            } else if (this._x>(_root.mobilnya._x+1)) {
            this._x--;
            }
            if (this._xscale<=0) {
                  this._visible = false;
            }
            if (_root.keyup == false) {
                  this._y--;
                  this._xscale -= 0.2;
                  this._yscale -= 0.2;
                  } else {
                  this._y += 2;
                  this._xscale += 0.4;
                  this._yscale += 0.4; }
            if (this.hitTest(_root.jalan.road1.grs_kiri)) {
                  this._x += 10;
            }
            if (this.hitTest(_root.jalan.road1.grs_knn)) {
                  this._x -= 10;
            }
            if (this.hitTest(_root.mobilnya)) {
                  if (this._x>=_root.mobilnya._x) {
                        _root.mobilnya._x -= 10;
                        this._x += 10;
                  } else {
                        _root.mobilnya._x += 10;
                        this._x -= 10; }
            }
            if (this.hitTest(_root.musuhx2)) {
                  if (this._x<_root.musuhx2._x) {
                        _root.musuhx2._x += 8;
                        this._x -= 8;
                  } else {
                        _root.musuhx2._x -= 8;
                        this._x += 8; }
            }
      }
}
```

Berikut script area lintasan pada level bonus :

```
onClipEvent (enterFrame) {
      if (this.hitTest(_root.atasnya)) {
            _root.atasnya._x += 7;
            this._x -= 4;
      }
      if (this.hitTest(_root.carx)) {
            _root.carx._x += 7;
            this._x -= 4;
      }
      if (this._y<=_root.atasnya._y) {
            if (this._x<(_root.atasnya._x-3)) {
                  this._x = thisx+3;
            } else if (this._x>(_root.atasnya._x+3)) {
                  this._x = thisx-3;
            }
      }
}
```

4.2 Pembahasan

Pada tahap ini, akan dibahas tentang analisis kinerja perangkat lunak, meliputi tahap pengujian, hasil yang diperoleh, analisis kesesuaian dengan teori sebelumnya dan perbandingan dengan *game* sejenis.

4.2.1 Analisis Kinerja Perangkat Lunak

4.2.1.1 Pengujian Perangkat Lunak

Game yang telah dibangun akan melalui tahap pengujian. Pengujian ini digunakan untuk mengetahui kekurangan pada *game* yang telah dibuat, dan untuk dapat mengetahui secara langsung apakah *game* telah bersifat *user friendly* atau tidak.

Proses pengujian dilakukan dengan mengundang maupun mengunjungi responden. Responden yang diundang dengan usia 15 sampai 25 tahun. Kuisioner yang telah diisi oleh sepuluh orang responden tersebut dapat segera

Tabel 4.1 Tabel hasil analisis

NO	Jawaban responden / Soal	1	2	3	4	5	6	7	8	9	10	Rata2
1	Bagaimana susunan menu utama *game* ini ?	4	4	4	4	4	4	5	3	4	4	4,0
2	Bagaimana kemudahan mengakses menu pada tampilan utama *game* ?	4	4	4	3	4	3	4	4	4	4	3,8
3	Bagaimana informasi pada menu Help ?	4	4	3	4	3	3	4	3	3	4	3,5
4	Apakah tingkat kepintaran musuh sudah cukup ?	3	4	2	3	4	3	3	3	4	4	3,3
5	Apakah kesukaran melawan musuh untuk tiap level-nya telah sesuai ?	3	3	3	4	3	3	4	2	3	4	3,2
6	Apakah ketersediaan bonus pada saat *game* berlangsung sudah cukup ?	3	4	3	3	4	3	3	4	3	3	3,3
7	Apakah informasi status *game* yang muncul sudah jelas (spidometer, time, bonus) ?	4	4	3	4	3	4	4	5	4	4	3,9
8	Bagaimanakan keragaman mobil musuh dalam *game* ini ?	4	4	4	3	4	4	4	4	4	3	3,8
9	Bagaimana pola pergerakan musuh ?	2	4	3	4	4	3	3	3	3	4	3,3
10	Bagaimana keragaman backsound *game* ini?	5	4	4	3	4	4	4	4	4	4	4,0
11	Bagaimana tampilan background / canvas pada menu utama dan tiap levelnya ?	4	4	3	3	3	4	4	3	4	4	3,5
12	Apakah model-model karakter mobil dalam *game* ini sudah bagus?	4	3	4	3	4	4	4	3	4	4	3,7
13	Bagaimana kemudahan mengendalikan mobil pemain?	4	4	3	4	3	3	4	3	3	4	3,5

4.2.1.3 Hasil pengujian terhadap responden

Proses pengujian yang telah dilakukan pada kesepuluh responden dengan usia 15 sampai 25 tahun. Pemilihan usia responden dimaksudkan sebagai proses perbandingan perbedaan pendapat-pendapat yang diutarakan responden. Terdapat tiga responden yang kurang merespon terhadap rancang bangun *game Racing Cars* ini. Setelah di adakan beberapa pertanyaan secara langsung, perancang dapat mengetahui kekurangan-kekurangan yang ada pada *game* ini dan dapat menyimpulkan bahwa kurangnya ketertarikan responden terhadap *game Racing Cars* dikarenakan tidak adanya fasilitas komputer dan minimnya pengetahuan tentang kemajuan teknologi saat ini, yang menyebabkan responden dinyatakan kurang tertarik dalam memainkan *game* tersebut.

4.2.2 Analisis Kesesuaian

Sesuai dengan teori dalam pembuatan *game* pada bab II, analisis yang dilakukan meliputi beberapa komponen berikut :

1. *Game board*

   Game board merupakan bentuk tampilan, dan untuk *game* ini menggunakan tampilan 2D yang *full colour*. Menurut responden tampilan antarmuka *game* ini sudah baik, hal itu dapat dilihat dari nilai rata-rata yang diberikan oleh responden pada pertanyaan nomor 1, 8, 11 dan 12 yaitu dengan nilai rata-rata 3,8.

2. Instruksi untuk pemain

   Game telah dilengkapi dengan instruksi cara bermain yang dapat diakses pemain sebelum mulai bermain pada menu *Help*. Menurut responden instruksi yang ada sudah baik, hal itu dapat dilihat dari nilai rata-rata yang diberikan oleh responden pada pertanyaan nomor 2 dan 3 yaitu dengan nilai rata-rata 3,7.

3. Informasi untuk pemain

   Informasi pada *game* ini adalah informasi mengenai tombol yang akan digunakan dalam permainan yang dapat dilihat pada menu *Help*. Menurut responden informasi yang ada sudah baik, hal itu dapat dilihat dari nilai rata-rata yang diberikan oleh responden pada pertanyaan nomor 1 dan 3 yaitu dengan nilai rata-rata 3,8.

4. Penghargaan

   Ketika pemain berhasil memenangkan *game* ini maka pemain dapat mencatatkan namanya pada daftar high score, sedangkan untuk yang belum berhasil memenangkan *game* ini, maka akan kembali ke halaman menu.

5. Variasi

   Disediakannya variasi sound berfungsi agar permainan tidak berjalan monoton. Menurut responden variasi sound yang ada dinilai cukup, hal itu dapat dilihat dari nilai rata-rata yang diberikan oleh responden pada pertanyaan nomor 10, yaitu dengan nilai 4,0.

6. Tingkat kesulitan

   Tingkat kesulitan dalam menyelesaikan *game* ini terletak pada jumlah bonus maupun musuh, pola pergerakan serta kemampuan musuh untuk menghalangi pergerakan. Menurut responden tingkat kesulitan yang ada sudah baik, hal itu dapat dilihat dari nilai rata-rata yang diberikan oleh responden pada pertanyaan nomor 4, 5, 6, 8, 9 dan 13 yaitu dengan nilai rata-rata 3,1.

4.2.3 Perbandingan dengan *game* sejenis

Game "Racing Cars" merupakan *game* mobil balap yang dapat dijalankan melaui PC. Pemain hanya akan memperebutkan bonus yang ada dengan kedua mobil musuh dengan waktu yang telah ditentukan. *Game* ini memiliki 3 level yang terdiri dari level biasa dan bonus level, dimana perbedaan setiap level ditentukan oleh kepandaian musuh, skill musuh (pola pergerakan) dan jenis *background* serta objek-objek yang lain. Setiap bonus yang didapatkan akan menghasilkan *score* untuk pemain. Pemain yang berhasil menyelesaikan permainan, akan menginputkan nama yang kemudian akan ditampilkan pada *form high score*.

Pada antarmuka saat *game* sedang berjalan disediakan tampilan yang informatif, seperti ditampilkannya indikasi spidometer, time (waktu) permainan, jumlah bonus, maupun indikasi status posisi pemain. Selain itu disediakan juga efek sound pada antarmuka menu utama maupun antarmuka *game* pada saat sedang berjalan.

Perancang memilih *game Mode7Racing* untuk melakukan perbandingan dikarenakan adanya beberapa persamaan tampilan, musuh, pola pergerakan dan beberapa tampilan antarmuka antara *game* mobil balap "*Racing Cars*" dengan *game* " *Mode7Racing*" yang merupakan *game* sejenis, dapat dilihat pada tabel 4.2.

Tabel 4.2 Tabel perbandingan *game*

NO	Pembanding	*Game* "Racing Cars"	*Game* "Mode7Racing"
1	Variasi background	Ada	Tidak
2	Indikasi waktu / time	Ada	Ada
3	Indikasi score	Ada	Tidak
4	Konfigurasi Setting	Ada	Tidak
5	Tampilan high score	Ada	Ada
6	Tampilan help / control	Ada	Tidak
7	Variasi level	Ada (3 level)	Ada (1 level)
8	Random musuh	Ada	Ada
9	Tampilan menu utama	Ada	Tidak
10	Variasi mobil	Ada	Tidak

Untuk lebih jelasnya akan ditampilkan antarmuka tampilan area permainan pada *game* "*Racing Cars*" dan " *Mode7Racing*" pada gambar 4.14 dan 4.15.

Berikut perbedaan tampilan antarmuka permainan :

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Pada bab ini dijelaskan beberapa kesimpulan sesuai dengan uraian yang telah dipaparkan pada bab-bab sebelumnya. Dari hasil perancangan sistem hingga tahap implementasi sistem dan dengan pengujian pada perangkat lunak yang telah dibuat, maka diperoleh beberapa kesimpulan, antara lain :

1. Aplikasi *game* mobil balap "*Racing Cars*" dibuat dengan tujuan dapat menjadi sarana hiburan dan digemari dalam beberapa kalangan. Tampilan *game* "*Racing Cars*" dapat dikatakan masih sangat sederhana, namun susunan kombinasi warna dibuat semenarik mungkin sehingga menarik untuk dimainkan.
2. Hasil rancang bangun *Game Mobil Balap Racing Cars* telah menjadi salah satu alternatif dalam memilih jenis permainan dan dari hasil uji responden *game* dapat dikatakan *user friendly.*
3. Hasil uji dengan *game* sejenis, *game Racing Cars* memiliki analisis kesesuaian yang lebih menarik dan sesuai dengan dasar dan elemen penyusun *game* dibandingkan dengan *game Mode7Racing* yang tidak memiliki kelengkapan dari segi *interface* maupun antarmuka pada tampilannya.
4. Aplikasi *game* mobil balap "*Racing Cars*" memiliki tingkat kesulitan yang berbeda-beda dalam tiap levelnya.

5. Gerakan pemain maupun musuh dalam tiap levelnya adalah statis, dimana *background* seolah olah bergerak mendekati pemain yaitu sudut pandang dari belakang, sedangkan pada bonus level akan tampak tampilan dari sudut pandang atas.

5.2 Saran

Berdasarkan pada pengujian yang telah dilakukan pada perangkat lunak yang dibuat, masih terdapat banyak kekurangan dan kelemahan sehingga perlu dikembangkan lagi agar menjadi lebih baik dan menarik, adapun hal tersebut adalah antara lain :

1. Memberikan efek suara yang lebih beragam, pergerakan objek serta karakter pemain dan musuh yang lebih bervariasi atau random.
2. Membuat *game* mobil balap "*Racing Cars*" dengan tampilan yang lebih menarik dan ditambahkan animasi flash *movie*.
3. Dalam pembuatan model-model mobil serta objek lainnya agar lebih dikembangkan lagi tenik-teknik perancangannya sehingga *game* lebih menarik untuk dimainkan.
4. Memberikn tampilan 3 D agar tampilan permainan terasa lebih nyata.

DAFTAR PUSTAKA

[CHA05] Chandra. Membuat Sendiri Game Flash MX 2004. Palembang : Mazikom, 2005.

[CRA97] Crawford, Cris. *Teori Game, The Art of Game Design*. Vancouver : 1997.

[DEY05] Deydri Adha S.T. 2005. *Rancang Bangun Aplikasi Action Games Perang Robot menggunakan bahasa Action Script pada Macromedia Flash MX. Skripsi*, tidak diterbitkan. Yogyakarta : Fakultas Teknologi Industri Universitas Islam Indonesia. (01523284)

[DIK03] Didik, Wijaya. *Macromedia Flash MX dengan Action Script*. Jakarta : PT Elex Media Komputindo, 2003

[HAR04] Hartono, B. dan Wijaya, F. *The Magic of Flash MX 2004*. Jakarta : Elex Media Komputindo, 2004.

[HEN89] Hendra. *Membuat Program Permainan*. Jakarta : Elex Media Komputindo, 1989.

[HUT03] Hutasoit, Andar Parulian, dan Wijaya, Didik. *Tip dan Trik Macromedia Flash MX dengan Action Script*. Jakarta : Elex Media Komputindo, 2003.

[LET01] Lethbridge, T. C., Laganiere, R. *Object-Oriented Softwere Engineering*. England: McGraw-Hill, 2001.

[NAV02] Noviyanto Fiftin. *Pemrograman Action Script dengan Flash 5 dan aplikasinya*. Yogyakarta : Penerbit ANDI Yogyakarta, 2002.

[ROL03] Rollings, Andrew, dan Adams, Ernest. *Andrew Rollings and Ernest Adams on Game Design*. USA : New Riders Publishing, 2003.

[SUT02] Sutepo, Ariesta Hadi. *Animasi dengan Macromedia Flash Berikut Action Script*. Jakarta : Salemba Infotek, 2002.

[SUT03] Suciadi, Andreas Andi. *Menguasai Pembuatan Animasi dengan Macromedia Flash Berikut Action Script*. Jakarta : Salemba Infotek, 2002.

[TEG06] Teguh Adi Wardoyo, S.T. 2006. *Program Game Skateboard Jet dengan menggunakan Action Script pada Flash MX. Skripsi*, tidak diterbitkan. Yogyakarta : Fakultas Teknologi Industri Universitas Islam Indonesia. (97523089)

[WIJ01] Didik Wijaya. *Macromedia Flash 5.0 dengan Action Script*. Jakarta : Elex Media Komputindo, 2001.

[WIJ03] Wijaya, D. dan Hutasoit, A. P. *Tip dan Trik Macromedia Flash MX dengan ActionScript*. Jakarta : Elex Media Komputindo, 2003.

LAMPIRAN

KUISIONER GAME "RACING CARS"

Game 'Racing Cars" adalah sebuah Tugas Akhir, dan sebagai evaluasi dan uji hasil dibutuhkan pendapat dan penilaian dari beberapa responden tentang game ini mencakup tampilan, menu, karakter, tingkat kesulitan, informasi bantuan, dan sebagainya.

Petunjuk pengisian :

1. Responden diharapkan dapat mengisi kuisioner ini dengan sejujur-jujurnya dengan *range* nilai :
   Nilai 1 untuk jawaban **tidak baik**
   Nilai 2 untuk jawaban **kurang**
   Nilai 3 untuk jawaban **sedang**
   Nilai 4 untuk jawaban **baik**
   Nilai 5 untuk jawaban **sangat baik**
2. Berilah jawaban untuk setiap pertanyaan (jangan dikosongi)
3. Berikan tanda centang (√) dalam menjawab pertanyaan sesuai dengan apa yang anda ketahui

NO	Jawaban responden / Soal	1	2	3	4	5
1	Bagaimana susunan menu utama game ini ?					
2	Bagaimana kemudahan mengakses menu pada tampilan utama game ?					
3	Bagaimana informasi pada menu Help ?					
4	Apakah tingkat kepintaran musuh sudah cukup ?					
5	Apakah kesukaran melawan musuh untuk tiap level-nya telah sesuai ?					
6	Apakah ketersediaan bonus pada saat game berlangsung sudah cukup ?					
7	Apakah informasi status game yang muncul sudah jelas (spidometer, time, bonus) ?					
8	Bagaimanakan keragaman mobil musuh dalam game ini ?					
9	Bagaimana pola pergerakan musuh ?					
10	Bagaimana keragaman backsound game ini?					
11	Bagaimana tampilan background / canvas pada menu utama dan tiap levelnya ?					
12	Apakah model-model karakter mobil dalam game ini sudah bagus?					
13	Bagaimana kemudahan mengendalikan mobil pemain?					

Berapa usia anda? (. . . – tahun)
Apakah anda suka bermain game? (Ya / Tidak) **coret salah satu*

Kritik dan saran :

_____

_____

_____