

**IMPLEMENTASI ACTION SCRIPT PADA MACROMEDIA FLASH
SEBAGAI BAHASA PEMROGRAMAN BERORIENTASI OBJEK
DALAM RANCANG BANGUN GAME MINIGOLF**

TUGAS AKHIR

**Diajukan sebagai Salah Satu Syarat
Untuk Memperoleh Gelar Sarjana
Jurusan Teknik Informatika**



Oleh:

Nama : Niko Wahyu Nurcahyo
No. Mahasiswa : 01 523 151

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA
YOGYAKARTA
2007**

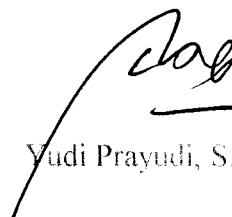
LEMBAR PENGESAHAN PEMBIMBING

IMPLEMENTASI ACTION SCRIPT PADA MACROMEDIA FLASH
SEBAGAI BAHASA PEMROGRAMAN BERORIENTASI OBJEK
DALAM RANCANG BANGUN GAME MINIGOLF



Yogyakarta,

Pembimbing,



Yudi Prayudi, S.Si, M.Kom.

LEMBAR PENGESAHAN PENGUJI

**IMPLEMENTASI ACTION SCRIPT PADA MACROMEDIA FLASH
SEBAGAI BAHASA PEMROGRAMAN BERORIENTASI OBJEK
DALAM RANCANG BANGUN GAME MINI GOLF**

TUGAS AKHIR

Oleh:

Nama : Niko Wahyu Nurcahyo

No. Mahasiswa : 01523151

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

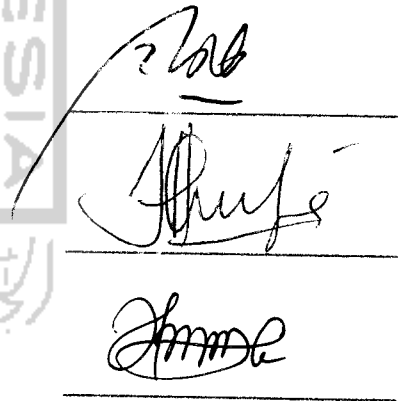
Yogyakarta, 2 Maret 2007

Tim Penguji,

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

Taufiq Hidayat, ST., MCS.
Anggota I

Hendrik, ST.
Anggota II



Mengetahui,
Ketua Jurusan Teknik Informatika
Universitas Islam Indonesia

Yudi Prayudi, S.Si, M.Kom.

Halaman Motto

“ Bacalah, dan Tuhanmulah Yang Maha Pemurah. Yang mengajar (manusia) dengan perantaraan kalam. Dia mengajarkan kepada manusia apa yang tidak diketahuinya. “

(Q.S. Al-‘Alaq: 3-5)

“ ...sesungguhnya sesudah kesulitan itu ada kemudahan. Maka apabila kamu telah selesai (dari suatu urusan), kerjakanlah dengan sungguh-sungguh (urusan) yang lain. “

(Q.S. Alam Nasyrah: 6-7)

“Allah selalu membantu hamba-Nya, selama hamba itu membantu saudaranya.”

(HR. Bukhari dan Muslim)

“ Dalam meraih sukses, yang penting goalnya jelas dan fokus! “

(Tung Desem Waringin)

KATA PENGANTAR

Assalamu'alaikum Wr. Wb.

Segala puji bagi Allah swt. Tuhan seru sekalian alam. Berkat hidayah dan izin-Nya penyusun dapat menyelesaikan laporan penelitian yang berjudul “ IMPLEMENTASI ACTION SCRIPT PADA MACROMEDIA FLASH SEBAGAI BAHASA PEMROGRAMAN BERORIENTASI OBJEK DALAM RANCANG BANGUN GAME MINIGOLF “ sebagai salah satu syarat untuk memperoleh gelar sarjana.

Penulisan laporan ini bertujuan untuk mendokumentasikan beberapa hal yang berkaitan dengan penelitian yang dilakukan oleh penyusun, sehingga memungkinkan untuk dikembangkan lebih lanjut dikemudian hari.

Selama proses penyusunan berlangsung, penulis mendapat banyak dukungan dan bantuan dari berbagai pihak, sehingga penelitian ini dapat terlaksana sesuai dengan yang diharapkan. Oleh karena itu penulis ingin mengucapkan terimakasih yang sebesar-besarnya kepada:

1. Bapak Yudi Prayudi, S.Si, M.Kom., yang telah membimbing penulis dari awal sampai akhir pelaksanaan tugas akhir dan juga selaku Ketua Jurusan Teknik Informatika.
2. Bapak Fathul Wahid, ST., MSC., selaku dekan Fakultas Teknologi Industri.
3. Bapak Prof. Dr. Edy Suandi Hamid, M.Ec., selaku Rektor Universitas Islam Indonesia.
4. Mas David atas kesediannya berbagi pengalaman hidup.

5. Adikku Syarifa, yang tak kenal lelah memberi semangat, kasih sayang dan nasihat-nasihat berguna sehingga saya tetap istiqomah dalam mensikapi kewajiban ini dan bisa segera menyelesaikannya.
6. Ustadz Hendri, atas diskusi yang menyenangkan dan menenangkan.
7. Syaikh Ahmad, terima kasih atas tausiyah-tausiyahnya.
8. Cak Agung, Oom Toshi dan Pakde Sigit, atas segala saran, dukungan dan berbagai pengalaman berharga yang syarat hikmah.
9. Teman-teman seperjuangan yang tak bisa saya sebutkan seluruhnya satu persatu, terima kasih atas segala keindahan, cinta dan persahabatan yang dahsyat!

Penulis menyadari bahwa masih banyak kekurangan dalam penelitian maupun penyusunan laporan ini, untuk itu penulis mengharapkan kritik dan saran yang membangun agar penelitian ini menjadi lebih berguna di masa yang akan datang.

Akhirnya, penulis berharap semoga penelitian ini dapat bermanfaat serta memperkaya khazanah ilmu pengetahuan di Indonesia.

Wassalam

Yogyakarta, Februari 2007

Penyusun

SARI

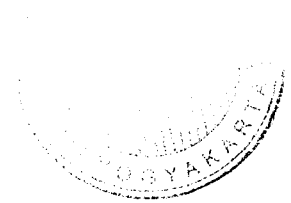
Game atau permainan adalah suatu hal yang selalu diminati oleh berbagai kalangan. Dalam dunia pemrograman *game*, para pengembang aplikasi *game* tentu saja diharapkan mampu menciptakan inovasi-inovasi baru dalam menggunakan teknik pemrograman, sehingga mampu menghasilkan *game-game* canggih dengan *game playing* yang mudah dan *familiar* bagi pengguna *game*. Proses pembuatan *game*, terutama yang menggunakan berbagai macam objek animasi, memerlukan kreatifitas tinggi, dan tentu saja *syntax* program yang efisien dalam penggunaannya, sehingga para pengembang *game* lebih mudah jika dalam pembuatan *game* mereka tidak perlu mengulang-ulang untuk membuat suatu fungsi yang sama. Para pengembang *game* di Indonesia khususnya yang semakin banyak memasukkan unsur-unsur animasi menarik dalam membuat *game*, menjadikan *software macromedia flash* semakin banyak diminati. Melalui bahasa *action script* yang terdapat dalam segi pemrogramannya, *macromedia flash* mampu untuk digunakan sebagai *software* yang tidak hanya mudah digunakan atau dikembngkan, tapi juga mampu untuk dijadikan sebagai sarana menuangkan ide-ide kreatif para pengembang *game* yang menginginkan hasil *game* yang ringan, menarik, mudah dimainkan oleh semua kalangan, dan tidak memerlukan *hardware* komputer dengan kemampuan yang tinggi untuk menjalankannya.

Kata kunci : *game, action script, macromedia flash*

TAKARIR

behavior	perilaku
controller	alat input pengontrol
developer	pengembang
engineer	perancang bangun
entity	entitas, sesuatu yang sungguh-sungguh ada
inheritance	pewarisan
parent	orang tua, induk
path	jalan
reference	referensi, rekomendasi
reusability	dapat digunakan kembali
root	akar
state	keadaan
valid	sah, absah, benar

DAFTAR ISI



JUDUL.....	i
LEMBAR PERNYATAAN KEASLIAN.....	ii
LEMBAR PENGESAHAN PEMBIMBING.....	iii
LEMBAR PENGESAHAN PENGUJI.....	iv
HALAMAN PERSEMBAHAN.....	v
HALAMAN MOTTO.....	vi
KATA PENGANTAR.....	vii
SARI.....	ix
TAKARIR.....	x
DAFTAR ISI.....	xi
DAFTAR GAMBAR.....	xiv
DAFTAR TABEL.....	xvi
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Batasan Masalah.....	2
1.4 Tujuan Penelitian.....	2
1.5 Manfaat Penelitian.....	3
1.6 Metodologi Penelitian.....	3
1.7 Sistematika Penulisan.....	4
BAB II LANDASAN TEORI.....	6
2.1 Definisi Game.....	6
2.1.1 Perancangan.....	7
2.1.2 Implementasi.....	8
2.1.3 Kerangka.....	9
2.1.4 Level.....	9
2.1.5 Kontrol kualitas.....	10
2.2 Langkah-langkah mempersiapkan program game oleh komputer.....	11
2.2.1 Game loop.....	11
2.2.2 Game shutdown.....	12
2.2.3 Hal-hal yang diperlukan pada saat memulai pemrograman game.....	12
2.3 Macromedia Flash MX.....	14
2.4 Pemrograman Action Script pada Flash.....	15
BAB III ANALISIS KEBUTUHAN PERANGKAT LUNAK.....	16
3.1 Metode Analisis.....	16
3.2 Metode Pengumpulan Data.....	17
3.3 Hasil Analisis.....	18
3.4 Kebutuhan Masukan.....	18
3.5 Kebutuhan Fungsi.....	18
3.6 Kebutuhan Keluaran.....	19
3.7 Kebutuhan Antarmuka.....	19
3.8 Aturan Permainan.....	19

BAB IV PERANCANGAN PERANGKAT LUNAK	21
4.1 Metode Perancangan	21
4.2 Prosedur Sistem	21
4.3 Hasil Perancangan	23
4.3.1 Desain arsitektur perangkat lunak	23
4.3.1.1 DFD level konteks (0)	23
4.3.1.2 DFD level 1 proses permainan dalam game golf	24
4.4 Desain Antarmuka	26
BAB V IMPLEMENTASI	34
5.1 Batasan Implementasi	34
5.1.1 Kebutuhan perangkat lunak	34
5.1.2 Kebutuhan perangkat keras	34
5.2. Implementasi Perangkat Lunak	35
5.2.1 Implementasi antarmuka	35
5.2.1.1 Tampilan utama	35
5.2.1.2 Tampilan instruksi	36
5.2.1.3 Tampilan awal menu hole 1 – ruang depan	37
5.2.1.4 Tampilan awal menu hole 2 – ruang santai	38
5.2.1.5 Tampilan awal menu hole 3 – ruang makan	39
5.2.1.6 Tampilan awal menu hole 4 – dapur	39
5.2.1.7 Tampilan awal menu hole 5 – garasi	40
5.2.1.8 Tampilan awal menu hole 6 – ruang kerja	41
5.2.1.9 Tampilan awal menu hole 7 – kamar tidur	42
5.2.1.10 Tampilan awal menu hole 8 – kamar mandi	43
5.2.1.11 Tampilan awal menu hole 9 – gudang	44
5.2.1.12 Tampilan hole 1 – ruang depan	45
5.2.1.13 Tampilan hole 2 – ruang santai	46
5.2.1.14 Tampilan hole 3 – ruang makan	47
5.2.1.15 Tampilan hole 4 – dapur	48
5.2.1.16 Tampilan hole 5 – garasi	49
5.2.1.17 Tampilan hole 6 – ruang kerja	50
5.2.1.18 Tampilan hole 7 – kamar tidur	51
5.2.1.19 Tampilan hole 8 – kamar mandi	52
5.2.1.20 Tampilan hole 9 – gudang	53
5.2.1.21 Tampilan skor akhir	54
5.2.2 Implementasi fungsi-fungsi dalam action script	56
5.2.2.1 Pengelompokan fungsi pada movie clip joueur(pemain)	56
5.2.2.2 Pengelompokan fungsi pada movie clip balle(bola)	57
5.2.2.3 Pengelompokan fungsi pada movie clip cercle(lingkar putih)	57

BAB VI ANALISIS KINERJA	59
6.1 Pengujian Menu Game.....	59
6.1.1 Pemanggilan pada tiap movieclip pada posisi x dan y dengan menyesuaikan jarak movieclip.....	59
6.1.2 Pemanggilan pada tiap movieclip dari radius yg telah ditentukan.....	59
6.1.3 Penilaian pada par.....	60
6.1.4 Mulai penentuan untuk posisi pemukulan yg tepat	64
6.1.5 Perlambatan pada gerakan bola.....	66
BAB VII SIMPULAN DAN SARAN.....	68
7.1 Simpulan	68
7.2 Saran.....	69
DAFTAR PUSTAKA.....	70



DAFTAR GAMBAR

Gambar 4.2 DFD level 1	26
Gambar 4.3 Perancangan antarmuka menu utama	27
Gambar 4.4 Perancangan antarmuka instruksi game.....	27
Gambar 4.5 Perancangan antarmuka hole 1	28
Gambar 4.6 Perancangan antarmuka hole 2.....	28
Gambar 4.7 Perancangan antarmuka hole 3.....	29
Gambar 4.8 Perancangan antarmuka hole 4.....	29
Gambar 4.9 Perancangan antarmuka hole 5.....	30
Gambar 4.10 Perancangan antarmuka hole 6.....	30
Gambar 4.11 Perancangan antarmuka hole 7.....	31
Gambar 4.12 Perancangan antarmuka hole 8.....	31
Gambar 4.13 Perancangan antarmuka hole 9.....	32
Gambar 4.14 Perancangan antarmuka tabel skor.....	32
Gambar 4.15 Perancangan antarmuka hasil skor akhir.....	33
Gambar 5.1 Tampilan menu utama	36
Gambar 5.2 Tampilan petunjuk permainan.....	36
Gambar 5.3 Tampilan awal menu hole 1 – ruang depan.....	37
Gambar 5.4 Tampilan awal menu hole 2 – ruang santai.....	37
Gambar 5.5 Tampilan awal menu hole 3 – ruang makan.....	38
Gambar 5.6 Tampilan awal menu hole 4 – dapur.....	39
Gambar 5.7 Tampilan awal menu hole 5 – garasi	39
Gambar 5.8 Tampilan awal menu hole 6 – ruang kerja	40
Gambar 5.9 Tampilan awal menu hole 7 – kamar tidur.....	41
Gambar 5.10 Tampilan awal menu hole 8 – kamar mandi	42
Gambar 5.11 Tampilan awal menu hole 9 – gudang.....	42
Gambar 5.12 Tampilan hole 1 – ruang depan.....	43
Gambar 5.13 Tampilan hole 2 – ruang santai.....	44
Gambar 5.14 Tampilan hole 3 – ruang makan.....	45
Gambar 5.15 Tampilan hole 4 – dapur.....	46
Gambar 5.16 Tampilan hole 5 – garasi	47
Gambar 5.17 Tampilan hole 6 – ruang kerja.....	47
Gambar 5.18 Tampilan hole 7 – kamar tidur	48
Gambar 5.19 Tampilan hole 8 – kamar mandi.....	49
Gambar 5.20 Tampilan hole 9 – gudang	49
Gambar 5.21 Tampilan skor akhir	50
Gambar 6.1 Tombol mulai	54
Gambar 6.2 Tampilan tabel pertama kali	55
Gambar 6.3 Tombol go.....	55
Gambar 6.4 Tampilan hole.....	56
Gambar 6.5 Tombol pukul bola.....	56
Gambar 6.6 Keadaan saat bola terjebak di sudut.....	57
Gambar 6.7 Saat bola masuk.....	57

Gambar 6.8 Tabel skor	58
Gambar 6.9 Pengaturan power penuh.....	58
Gambar 6.10 Pengaturan power tidak penuh	59
Gambar 6.11 Mouse digerakkan di luar batas area.....	59
Gambar 6.12 Perlambatan pada karpet.....	58
Gambar 6.13 Perlambatan pada oli.....	58
Gambar 6.13 Hasil game.....	59



DAFTAR TABEL

Tabel 4.1 Daftar class actionscript.....	22
--	----



LEMBAR PERNYATAAN KEASLIAN
HASIL TUGAS AKHIR

Saya yang bertandatangan di bawah ini,

Nama : Niko Wahyu Nurcahyo

No. Mahasiswa : 01 523 151

Menyatakan bahwa seluruh komponen dan isi dalam Laporan Tugas Akhir ini adalah hasil karya saya sendiri. Apabila dikemudian hari terbukti 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, Februari 2007



(Niko wahyu Nurcahyo)

Halaman Persembahan

Tugas Akhir ini saya persembahkan kepada :

- *Ibu dan Bapak, kedua orang tua yang selama ini telah banyak memberikan dukungan baik moril maupun materiil, sehingga saya mampu untuk terus berjuang di jalan yang benar.*
- *Seluruh praktisi dalam dunia pemrograman di Indonesia.*



BAB I

PENDAHULUAN

1.1 Latar Belakang

Teknologi *object-oriented programming (OOP)* merupakan paradigma baru dalam rekayasa perangkat lunak yang didasarkan pada objek dan kelas. Diakui para ahli bahwa *object-oriented* merupakan metodologi terbaik yang ada saat ini dalam rekayasa perangkat lunak. *Object-oriented* memandang *software* bagian per bagian, dan menggambarkan satu bagian tersebut dalam satu objek. Satu objek dalam sebuah model merupakan suatu fokus selama dalam proses analisis, desain, dan implementasi dengan menekankan pada *state*, perilaku (*behavior*), dan interaksi objek-objek dalam model tersebut. Keuntungan yang diperoleh dari pendekatan *object-oriented* ini adalah kemudahan untuk mengelola kompleksitas alokasi yang dibuat dan kemudahan untuk melakukan perubahan dan pengembangan dari aplikasi tersebut.

Action Script merupakan salah satu bahasa pemrograman yang menggunakan konsep *object-oriented*. Dalam segi pemrograman, *Action Script* menggunakan objek sebagai acuan untuk membuat suatu *project* animasi atau *game*. Objek-objek yang telah dibuat, kemudian disimpan dalam suatu pustaka (*library*) yang dapat digunakan kembali atau diubah sesuai keinginan *programmer*. Kemampuan *Action Script* yang demikian, memudahkan para pengembang animasi dan *game* dan lebih efisien dalam hal pengerjaan *project*

karena semua objek yang telah dibuat dan tersimpan dalam *library* baik itu gambar, suara, dan animasi *movie* bisa digunakan kembali sesuai kebutuhan.

Pengembangan aplikasi *game minigolf* didasarkan pada banyaknya minat para *gamer* terhadap permainan tersebut. Sebagai alternatif *game* yang cukup menarik, *minigolf* telah banyak diminati oleh para *gamer* tidak hanya para pecinta *golf* saja. Bahkan di *internet*, *minigolf* dapat dimainkan secara *on-line*. Maka dari itu, perlu dikembangkan lebih lanjut alternatif-alternatif *game* semacam itu agar para pencintanya mendapatkan referensi *game* yang lebih banyak lagi dengan *game playing* dan *setting* yang berbeda-beda.

1.2 Rumusan Masalah

Bagaimana membangun sebuah *game minigolf* dengan menerapkan teknik-teknik pemrograman menggunakan bahasa *Action Script*.

1.3 Batasan Masalah

1. Analisis aplikasi berdasarkan fungsi-fungsi game secara *stand-alone* dan satu pemain;
2. Aplikasi *game* yang dibangun adalah bentuk permainan *minigolf* dalam ruangan (*in-door*);
3. Skor akhir permainan bersifat *temporary*, ditunjukkan selama *game* dijalankan.

1.4 Tujuan Penelitian

Tujuan dari pengembangan *game* ini adalah menerapkan dan memaksimalkan teknik-teknik pemrograman pada *Flash* dengan *Action Script* sebagai bahasa pemrograman yang berorientasi pada objek agar mampu menghasilkan *game-game* animasi yang canggih, mudah dimainkan dan tentunya dalam segi tampilan dapat lebih menarik lagi.

1.5 Manfaat Penelitian

Adapun manfaat penggunaan *Action Script* dalam rancang bangun *game minigolf* adalah sebagai berikut :

1. Sebagai bahasa yang berorientasi pada objek, *Action Script* mampu memberikan kemudahan dan efisiensi penulisan *syntax script* untuk penggunaan aturan-aturan dan fungsi-fungsi program dalam membangun *game*;
2. Sebagai suatu referensi pemrograman *game* bagi para pembuat *game*.

1.6 Metodologi Penelitian

1.6.1 Metode Pengumpulan Data

Metode pengumpulan data meliputi kepustakaan dan referensi. Pada penelitian ini dilakukan melalui buku-buku teks, makalah-makalah internet, *ebook*, dan artikel-artikel yang berhubungan dengan *object-oriented programming*, *game programming* dan animasi yang keseluruhannya dalam lingkup kerja *Macromedia Flash MX*.

1.6.2 Metode Pengembangan Perangkat Lunak

Pengembangan aplikasi game dalam penelitian dibagi dalam empat tahap mengacu pada standar pengembangan perangkat lunak.

1. Analisis kebutuhan perangkat lunak, untuk pendefinisian ruang lingkup, kebutuhan sistem beserta fungsinya, unjuk kerja, dan antarmuka yang dibutuhkan.
2. Desain/perancangan aplikasi *game*, untuk penentuan arsitektur dan *game playing* untuk *game minigolf* beserta antarmuka dan detail fungsionalnya.
3. Implementasi/pembuatan aplikasi, merupakan tahap penerapan dari hasil analisis dan desain menjadi sebuah aplikasi *game golf* yang dapat difungsikan sesuai dengan tujuan pengembangannya.
4. Pengujian, untuk melakukan verifikasi dan validasi terhadap aplikasi *game* yang dihasilkan.

1.7 Sistematika Penulisan

Bab 1 pendahuluan, memberikan penjelasan tentang latar belakang penelitian, rumusan masalah yang dihadapi, batasan serta asumsi dari penyelesaian masalah yang telah dirumuskan sebelumnya, tujuan dan manfaat dari penelitian, metodologi penelitian yang digunakan serta sistematika penulisan laporan penelitian.

Bab 2 landasan teori, memuat uraian tentang teori-teori yang berhubungan dengan serta dijadikan bahan acuan untuk penelitian. Teori-teori yang

BAB 11

LANDASAN TEORI

2.1 Definisi *Game*

Game adalah program yang memungkinkan pemain / *gamers* untuk berinteraksi dengan simulasi lingkungan virtual dalam komputer. *Game* memiliki elemen-elemen strategi, aksi dan fantasi. Elemen-elemen inilah yang membuat *gamers* seakan berada di suatu dunia lain dengan waktu dan realita yang berbeda dengan kenyataan. Dan alasan kenapa kita memainkan *game* adalah untuk *fun*. Pemrograman *game* sebenarnya hanya bagian dari sebuah rangkaian proses pembangunan *game*. Pekerjaannya dimulai dari membuat desain meliputi awal sampai akhir permainan. Pengembangan *game* yang profesional memang dimulai dari desain *game* dengan tema hasil imajinasi atau bisa jadi adopsi dari sebuah novel.

Game designer dewasa ini berupaya membuat *game* yang dapat mencapai kemampuan maksimum sebuah komputer dalam hal grafik, *sound*, dan tenaga komputer itu sendiri. Tentunya banyak keterbatasan dan kendala yang akan dialami, tetapi seorang *game programmer* yang cerdas dapat mengupayakan apapun menggunakan *PC* sebagai *tools* untuk membawa *gamers* masuk kedalam dunia lain yang diinginkannya.

Kita telah melihat bayangan mengenai *game* itu sendiri, sekarang kita lihat bagaimana pemrograman di dalamnya. Kita coba melihat dengan pendekatan arsitektur dari sebuah *game*. Arsitektur sebuah *game* sangatlah kompleks. Tetapi

sama seperti aplikasi *real-time* lainnya, *game* memiliki interaksi timbal-balik utama (*main event loop*) yang diproses berulang kali. Interaksi inilah inti dari sebuah *game*. *Loop* dari *game* diperoleh dari referensi internal pemrograman. Referensi ini bisa berupa waktu, *vertical blank* atau lainnya. Interaksi ini mengakomodasi tampilan grafik pada *monitor* dan suara pada *speaker*. Melalui bagan arsitektur tersebut, berikut adalah uraian proses yang terbagi menjadi beberapa bagian.

2.1.1 Perancangan

Ide dasar dan keseluruhan fitur dituliskan dalam sebuah *prototype* sehingga antara pemrogram dan perancangannya dapat bereksperimen dan bereksplorasi dengan berbagai *algoritma* dan skenario. *Prototype* ini sangat membantu dalam penyempurnaan desain utama untuk menspesifikasikan lebih jauh fitur-fitur seperti apa yang akan dibuat, dan bisa memberikan gambaran awal seperti apa kira-kira hasil akhir *game* ini.

Bisa juga *prototype* ini dibuat di tengah-tengah proses implementasi untuk tes sebuah ide baru yang mungkin bisa diterapkan pada *game* tersebut. Yang pasti, *prototype* ini dibuat secara sederhana dalam waktu singkat dan tidak terikat pada satu lingkungan pemrograman karena bisa jadi akan diimplementasikan dalam lingkungan yang berbeda berdasarkan pertimbangan teknis yang menyertainya.

Seorang seniman *game* (*game artist*) bertugas memberikan sentuhan nilai estetika yang tinggi pada *game* dan bertanggung jawab terhadap seluruh aspek pada pengembangan *game* yaitu *visual art*. Si pemrogram yang pekerjaannya utamanya tidak membuat desain, biasanya juga turut memberikan saran pada

desain. Desain yang dibuat bukanlah harga mati tapi terus mengalami revisi. Apalagi dalam implementasinya nanti biasanya ditemui beberapa hambatan atau bisa juga ide baru sehingga desain juga mengalami perubahan.

2.1.2 Implementasi

Setelah desain disepakati bersama, proses implementasinya atau pemrogramannya mulai didiskusikan seperti menentukan bahasa apa yang akan dipakai. Beberapa pertimbangannya antara lain bahasa apa yang dikuasai dan sudah *familiar* bagi para *programmer*, kemudian di *platform* apa *game* ini akan berjalan misalnya di *Play Station* atau *Windows*, kecepatan eksekusi, dan kesesuaian dengan *API* (*application programming interface*) yang digunakan.

Tentu sebuah *game* di mana setiap objek berinteraksi satu sama lain, paradigma yang digunakan adalah *object oriented*. Bahasa yang paling populer digunakan untuk implementasi *game* adalah *C++* dan diikuti oleh *Java* dan *C#*. Sedangkan untuk memprogram *game* yang dijalankan dengan perangkat *video game*, biasanya digunakan bahasa *assembly* yang menawarkan kecepatan eksekusi dan ruang memori relatif kecil, tetapi mensyaratkan pemrogramnya mengerti arsitektur mesin. Namun, beberapa *vendor game* juga menciptakan bahasa sendiri untuk *game* produksinya. Seperti pada masa dewasa ini, para pemrogram *game* telah banyak menggunakan *Flash* dengan *Action Script*. Jika biasanya *Flash* hanya digunakan untuk membuat situs beranimasi, sebenarnya *Flash* memiliki potensi lebih dari itu. *Game-game* animasi ringan berbasis *Flash* telah banyak digunakan sebagai daya tarik pengunjung *web*. *Game* menjadi tidak terpisahkan dari *web*. Setelah sebelumnya seorang *desainer web* dapat membuat betah orang

yang mengunjungi situsnya dengan animasi, musik, dan gambar-gambar, akhirnya sebuah pemikiran muncul bahwa tentu dengan menambahkan *games*, pengunjung akan lebih betah menonton situsya.

2.1.3 Kerangka

Satu yang menarik, secara konseptual kerangka dasar *game* merupakan sebuah *loop* (putaran). Tanpa aksi yang dilakukan pemain, program *game* harus terus berjalan dimulai dari mengecek apakah ada aksi dari pemain, menjalankan program inteligensia buatan, musuh terus bergerak, menggambar grafik, memutar suara, dan lain-lain. Demikian seterusnya kembali lagi pada pengecekan aksi dari pemain dan perputaran ini akan berhenti jika pemain keluar dari *game*. Oleh sebab itu, jika pemain meninggalkan permainan beberapa saat, karakter yang berada di luar kendali pemain seperti musuh dan latar belakang akan terus bergerak diiringi ilustrasi suara. Tanpa aksi dari pemain, kehidupan dalam *game* tersebut masih terus berjalan.

2.1.4 Level

Permainan sebuah *game* biasanya terbagi dalam beberapa *level*. Satu *level* merepresentasikan satu area atau episode dalam dunia *game* yang ditandai dengan perbedaan tata lokasi, musuh, dan makin tinggi *level*, maka tingkat kesulitan semakin tinggi. Dengan sistem *level* ini, pemain yang menang bisa bermain lagi dalam model dan aturan main yang sama tapi dengan tantangan yang berbeda. Kehidupan *game* pun terasa lebih panjang dan menarik.

Di samping itu, *level* bisa menjadi parameter prestasi permainan, yaitu sampai *level* berapa ia sanggup menyelesaikan. Belum lagi dalam sebuah *game*, si

pemain bisa merasakan kepuasan berkali-kali setiap berhasil melampaui satu *level*. Jika sebuah *game* tak ada sistem *level* dan alur terlalu panjang, tentu si pemain tak tahu kapan ia bisa berhenti.

2.1.5 Kontrol kualitas

Meski belum selesai penuh, biasanya *vendor* mulai membuat versi demo *game* yang akan diluncurkannya. Cukup dengan menampilkan fitur-fitur yang dimiliki *game* tersebut sebagai daya tarik untuk penjualan *full version* nanti. Sebagai bagian dari *marketing*, tentu versi demonya diberikan secara cuma-cuma dan sebisa mungkin dicoba banyak orang.

Pada proses finalisasi perangkat lunak, tahap evaluasi atau biasa disebut *testing* memegang peranan lebih besar dibanding implementasi. Bagaimana sedapat mungkin perangkat lunak tersebut tidak banyak *bug* (kutu) atau kesalahan. Untuk itu, menjelang akhir proses pengembangannya pemrogram harus mau dan siap menjalankan tugasnya setiap waktu untuk mengatasi *bug* yang masih ada mulai dari yang ukurannya besar hingga yang mikroskopis.

Kalau pernah menjumpai versi *beta* sebuah *game*, artinya *game* tersebut sudah disertai fitur lengkap tetapi tidak terjamin bebas *bug* karena untuk mencapai bebas *bug* tentu membutuhkan proses lebih lanjut sampai mendapatkan versi stabil. Pada tahap inilah tim *QA* (*quality assurance*) bertanggung jawab atas kualitas *game* yang akan diluncurkan.

2.2 Langkah-langkah mempersiapkan program *game* oleh komputer

Komputer akan melakukan beberapa langkah berikut untuk memulai sebuah *game*:

- a. Alokasi memori. Di sini program melakukan alokasikan *video memory*, inisialisasi *sound card*, inialisasi port, objek dalam *game*, dan lain sebagainya.
- b. Membuka file. Membaca atribut *setting option*, *level data*, *script* dan lainnya.
- c. Men-set variabel awal. Menentukan posisi awal objek, menentukan *level* dan sebagainya.
- d. Membaca gambar. Membuka file gambar yang akan digunakan dalam *game*.
- e. Membaca suara. Membuka file suara yang akan digunakan dalam *game*.
- f. Menentukan *score* awal.

2.2.1 *Game loop*

- a. *User input*. Mengambil data dari user melalui input devices.
- b. *Game logic*. *Artificial Intelligence (AI)*, *algoritma*.
- c. *Object Transform*. Translasi, rotasi, perbesar (*scaling*), dan sebagainya terhadap objek.
- d. *Collision detection*. Melakukan pendeteksian benturan antara dua atau lebih objek - objek.
- e. *Transform player*. Meletakkan posisi objek sebagaimana mestinya setelah ditransform dan *collision detection* (batasan). Contohnya; mencegah posisi pemain keluar dari lingkungan *virtual*.

- f. *Render background, picture and sound*. Menampilkan gambar objek beserta kelengkapannya ke layar dan memainkan suara.

2.2.2 *Game shutdown*

Membersihkan memori yang telah dialokasi. Contohnya pada saat bagian inisialisasi melakukan alokasi gambar, suara, level dan objek lainnya. Maka pada tahap ini alokasi alokasi yang telah dilakukan tersebut akan dibersihkan.

2.2.3 Hal-hal yang diperlukan pada saat memulai pemrograman *game*

Ruang lingkup *game* dan kompleksitasnya akan timbul pada saat memulai membuat sebuah *game*. *Game programmer* menjadi penentu hal ini. Suatu *game* biasanya harus dibuat sesuai dengan kemampuan perangkat keras dan perangkat lunak yang ada sehingga memungkinkan suatu *game* dapat berjalan sebagaimana mestinya. Sebuah *game* akan menjadi semakin kompleks apabila perangkat keras dan perangkat lunak yang diimplementasikan juga kompleks. Untuk itu sebelum memprogram sebuah *game*, seorang *game programmer* dan *game designer* menentukan terlebih dahulu batasan-batasan *game* tersebut dari segi perangkat keras dan perangkat lunak.

Keterbatasan perangkat keras :

1. Memori. Ukuran memori yang didukung.
2. Ukuran display dan resolusi warna. Ukuran tampilan dan jumlah warna biasanya berhubungan dengan ukuran memori.
3. Suara. Kualitas suara, mono atau stereo.

4. *Input device*. 2 tombol, 3 tombol atau lainnya. Tombol yang banyak lebih mudah untuk berinteraksi dengan lingkungan virtual. Misal; game *online* lebih unggul dimainkan pada PC karena untuk melakukan *chatting* lebih mudah menggunakan *keyboard* daripada *joystick*.
5. *CPU power*. *Game* yang kompleks membutuhkan *CPU* yang cepat untuk menghitung dan memroses game.
6. *3D accelerator*. Difungsikan untuk membantu mempercepat proses transformasi dan operasi geometri 3D.
7. *Bandwith*. Kecepatan transfer antar komputer dan antar *device*.

Keterbatasan perangkat lunak :

1. (*Application Programming Interface - Software Development Kit*) API/SDK yang dimiliki. Hardware yang canggih tanpa didukung oleh API yang mendukung hanya akan mempersulit programmer, karena diperlukan waktu lebih untuk membuat *device driver* yang menggerakkan *hardware* tersebut.
2. *Algoritma* yang diterapkan. Algoritma akan membantu apabila yang diterapkan sesuai dengan ruang lingkup permasalahan dan juga biasanya dikaitkan dengan kemampuan *hardware*. Contoh contoh *algoritma*; algoritma untuk AI, *sorting*, *searching* dan sebagainya.
3. *Error rate*. Suatu *game* bisa disebut layak apabila *game* tersebut mempunyai tingkat *error* yang masih dapat ditoleransi.

2.3 Macromedia Flash MX

Macromedia Flash MX adalah program canggih untuk membuat aplikasi dan animasi *movie*, *game* dan *web* profesional. Untuk industri animasi dan *web*, *Flash* telah mejadi salah satu standar. Sedangkan dalam *game*, *Flash* telah banyak dikembangkan menjadi bentuk aplikasi-aplikasi *game* yang umumnya dijadikan sebagai pelengkap untuk menambah fitur dalam *web* profesional. Tidak menutup kemungkinan, teknologi *Flash*, khususnya *game*, bisa juga berdiri sendiri sebagai suatu bentuk aplikasi *game* yang tidak kalah dengan *game-game* yang dibuat menggunakan bahasa pemrograman lainnya.

Area kerja dalam *Macromedia Flash MX* dibagi menjadi beberapa bagian yang dirancang agar mudah diakses untuk segala keperluan dalam membuat *movie*. Bagian-bagian area kerja pada *Macromedia Flash MX* adalah sebagai berikut :

1. *Stage* : adalah tempat untuk menggambar dan mengimport artwork, membuat teks dan suara, serta menambahkan fitur seperti tombol navigasi atau komponen-komponen interface.
2. *Toolbox* : berisi tool yang akan digunakan untuk membuat, menempatkan dan memodifikasi teks dan gambar.
3. *Panels* : berfungsi membantu dalam mengerjakan objek di dalam *Stage* Dokumen, *Timeline*, dan *Action*.
4. *Timeline* : digunakan untuk mengorganisasikan dan mengontrol isi *movie*. Sama seperti film, *Flash* membagi panjang waktu ke dalam frame.

5. *Layer : Timeline* dibagi menjadi beberapa *layer*. Setiap objek dapat berada pada *layer* yang berbeda.
6. *Library* : tempat menyimpan elemen-elemen yang akan digunakan kembali yang disebut *symbol*.
7. *Property Inspector* : digunakan untuk menampilkan semua *setting* yang telah digunakan pada *tool* atau objek
8. *ActionScript* : merupakan bahasa pemrograman yang telah disiapkan untuk membuat aplikasi.

2.4 Pemrograman *Action Script* pada *Flash*

Untuk membuat aplikasi *game* yang tentu saja dapat berinteraksi dengan *user / pemain*, *movie Flash* memerlukan sejumlah kode perintah yang disebut *ActionScript*. *ActionScript* adalah bahasa pemrograman *Flash* yang digunakan untuk mengontrol objek, membuat navigasi dan elemen interaktif lainnya. *ActionScript* terbagi menjadi dua, yaitu :

1. *Action Frame* : adalah action yang diberikan pada *frame*.
2. *Action Object* : adalah action yang diberikan pada objek, baik berupa tombol maupun *movie clip*.

BAB III

ANALISIS KEBUTUHAN PERANGKAT LUNAK

Game atau permainan adalah suatu hal yang selalu diminati oleh berbagai kalangan. Dalam dunia pemrograman *game*, para pengembang aplikasi *game* tentu saja diharapkan mampu menciptakan inovasi-inovasi baru dalam menggunakan teknik pemrograman, sehingga mampu menghasilkan *game-game* canggih dengan *game playing* yang mudah dan *familiar* bagi pengguna *game*. Proses pembuatan *game*, terutama yang menggunakan berbagai macam objek animasi, memerlukan kreatifitas tinggi, dan tentu saja *syntax program* yang efisien dalam penggunaannya, sehingga para pengembang *game* lebih mudah jika dalam pembuatan *game* mereka tidak perlu mengulang-ulang untuk membuat suatu fungsi yang sama.

3.1 Metode Analisis

Metode analisis terstruktur, pengembangan sistem dalam bentuk modul-modul, mudah dites secara terpisah. Metode ini berorientasi kepada aliran data yang masuk dan keluar dari sistem. Metode ini menganut sistem dekomposisi, yaitu menggambarkan terlebih dahulu sistem secara keseluruhan sebagai tingkat tertinggi kemudian memecahnya menjadi bagian-bagian yang lebih terinci. Metodologi pengembangan sistem merupakan kesatuan metode-metode, konsep-konsep pekerjaan dan aturan-aturan ataupun prosedur-prosedur yang akan digunakan untuk mengembangkan sistem, karenanya metode ini adalah metode

yang dibutuhkan untuk dapat melakukan suatu pengembangan sistem perangkat lunak (*software development*).

Tahap analisis dilakukan untuk mengetahui dan menterjemahkan permasalahan dan kebutuhan perangkat lunak serta kebutuhan sistem yang dirancang dengan mendapatkan data-data yang diperlukan untuk membangun sistem.

Metode yang digunakan adalah metode analisis berarah data, yaitu metode analisis secara terstruktur dengan membuat diagram arus data yang akan mendeskripsikan input, proses dan output berdasarkan aliran informasi yang telah ada. Dengan metode ini akan dihasilkan sistem yang lebih terstruktur karena metode ini memberikan gambaran terperinci mengenai komponen-komponen yang digunakan.

3.2 Metode Pengumpulan Data

Dalam mengumpulkan data yang diperlukan untuk melakukan penelitian ini, digunakan metode observasi, yaitu suatu metode pengumpulan data dengan cara mengamati secara langsung terhadap objek yang ditinjau untuk mendapatkan data-data yang berhubungan dengan penelitian yang dilakukan.

3.3 Hasil Analisis

Hasil dari analisis kebutuhan adalah kebutuhan perangkat lunak yang hendak dibangun. Kebutuhan yang dimaksud adalah kebutuhan masukan, keluaran, kebutuhan antarmuka, dan kebutuhan proses atau fungsi. Perangkat lunak yang dihasilkan dalam penelitian ini berupa suatu game bertema *minigolf*

sederhana yang pemrogramannya dibuat menggunakan bahasa *ActionScript* yang terdapat pada *Macromedia Flash*.

3.4 Kebutuhan Masukan

Masukan yang dibutuhkan oleh perangkat lunak yang dibangun dalam penelitian ini adalah :

- a. *Input* dari tombol-tombol pada keyboard dan mouse dalam menjalankan game, yaitu pada saat memukul bola golf dari mulai permainan hingga akhir.
- b. *File-file object* yang dibuat dengan *ActionScript* secara terpisah, berada di luar *file* utamanya (*file-file* dengan ekstensi **.as*).

3.5 Kebutuhan Fungsi

- a. Proses yang terjadi digambarkan menggunakan *DFD* yang menjelaskan alur *algoritma* dari rancang bangun aplikasi *game minigolf* yang dibuat.
- b. Tingkat kesulitan yang semakin tinggi dengan bertambahnya level game.
- c. Adanya suara yang dapat didengar selama game yaitu suara efek-efek dari hasil kejadian yang terjadi selama game berlangsung
- d. Adanya animasi-animasi yang menambah menarik game ini.

3.6 Kebutuhan Keluaran

Output yang terdapat dalam *game* yaitu berupa gambar-gambar objek, animasi kejadian-kejadian, suara efek, dan papan skor yang menunjukkan hasil akhir skor yang dicapai oleh pemain.

3.7 Kebutuhan Antarmuka

Antarmuka yang ditampilkan dalam *game* yaitu berupa gambar-gambar dan animasi yang menarik. Agar pengguna dapat mengeksplorasi sistem dengan mudah maka diperlukan sebuah antarmuka yang tidak membingungkan (*user friendly*). Pembuatan gambar, animasi dan penggunaan suara, seluruhnya diatur dan diolah menggunakan fasilitas-fasilitas yang terdapat dalam *Macromedia Flash*.

3.8 Aturan Permainan

Aturan permainan yang dibuat sedikit menyesuaikan pada permainan golf yang sebenarnya. Namun, tidak semua digunakan oleh karena melihat tujuan awal dalam rancang bangun *game* ini, yang utama adalah menunjukkan implementasi object oriented programming dalam bahasa ActionScript. Beberapa aturan permainan golf yang dipakai adalah sebagai berikut :

- a. *Par* : adalah batas pukulan yang ditentukan pada suatu arena permainan (hole) untuk memasukkan bola.
- b. *Birdie* : adalah keadaan di mana bola dapat dimasukkan oleh pemain dalam pukulan yang jumlahnya kurang satu dari par yang ditentukan.
- c. *Bogey* : adalah keadaan di mana bola baru dimasukkan oleh pemain dalam pukulan yang jumlahnya lebih satu dari par yang ditentukan.
- d. *Hole in one* : adalah keadaan di mana bola langsung masuk saat pemain mulai memukul bola untuk pertama kali.

BAB IV

PERANCANGAN PERANGKAT LUNAK

4.1 Metode Perancangan

Setelah diperoleh spesifikasi kebutuhan aplikasi perangkat lunak pada tahap analisis kebutuhan, langkah selanjutnya adalah perancangan perangkat lunak game. Tahap ini merupakan langkah awal untuk implementasi dan pengembangan perangkat lunak. Dalam rekayasa perangkat lunak, tahap perancangan merupakan tahap yang sangat menentukan kualitas perangkat lunak yang dihasilkan. Oleh karena itu diusahakan tahap perancangan ini dibuat dan didokumentasikan sebaik mungkin agar perangkat lunak yang dihasilkan sesuai dengan yang diharapkan.

Aplikasi perangkat lunak ini didesain berdasarkan metode desain berorientasi pada aliran data. Game dideskripsikan sebagai proses atau kumpulan proses transformasi yang mengubah data masukan menjadi data keluaran. Diagram aliran data digunakan untuk menggambarkan aliran informasi dan transformasi data yang terjadi di dalam game.

Dari bagan alir ini kita bisa mengamati dan menentukan aliran kendali program sehingga dengan berdasar pada bagan alir tersebut kita bisa menyusun baris-baris programnya satu demi satu.

4.2 Prosedur Sistem

Program game yang dibuat diharapkan bisa menjadikan sebagai suatu contoh aplikasi yang didalamnya terdapat unsur-unsur pemrograman berarah

dimaksudkan mencakup baik teori *game programming* maupun pustaka-pustaka ActionScript dan Macromedia Flash MX untuk aplikasi *game*.

Bab 3 analisis kebutuhan, memberikan penjelasan tentang kebutuhan perangkat lunak yang akan dikembangkan selama penelitian, meliputi kebutuhan masukan, fungsi, keluaran, dan antarmuka yang dibutuhkan.

Bab 4 perancangan perangkat lunak, berisi uraian tentang metode perancangan perangkat lunak yang dipakai, hasil perancangan perangkat lunak, yang merupakan terjemahan kebutuhan perangkat lunak, yang meliputi struktur data yang digunakan, arsitektur perangkat lunak, rincian prosedur-prosedur atau metode-metode, dan antarmuka yang akan dikembangkan.

Bab 5 implementasi, memuat dokumentasi dari implementasi perangkat lunak yang dikembangkan, meliputi batasan pengembangan, implementasi struktur data dan prosedur dalam ActionScript, serta antarmuka yang dihasilkan.

Bab 6 analisis kinerja, berisi dokumentasi selama pengujian perangkat lunak yang dikembangkan selama penelitian, meliputi pengujian data masukan dan penanganan kesalahan yang terjadi dalam pengoperasian aplikasi *game*.

Bab 7 simpulan dan saran, berisi simpulan laporan penelitian tugas akhir dan saran-saran untuk pengembangan aplikasi sejenis yang lebih baik.

objek, di mana terdapat prosedur-prosedur yang mempunyai kemampuan untuk digunakan kembali berkali-kali, dengan cara memanggil parent (nama kelas) dari prosedur yang telah dibuat. Kelas-kelas objek tersebut tersimpan secara terpisah dalam suatu file dengan ekstensi *.as. Dalam proses pemrogramannya, syntax yang digunakan bertindak memanggil kelas-kelas yang telah dibuat tersebut untuk menjalankan fungsi baru yang merupakan turunannya. Dalam pemrograman ActionScript, ada beberapa class penting yang digunakan dalam menggunakan teknik pemrograman berorientasi objek. Berikut adalah tabel yang berisi nama-nama class dan kegunaannya :

Tabel 4.1 Daftar class actionscript

Nama Class	Deskripsi
Arguments	Sebuah array yang berisi nilai-nilai yang dikirim sebagai parameter ke berbagai fungsi.
Array	Class array yang merepresentasikan array di ActionScript dan semua objek array adalah instance class ini.
Boolean	Class boolean adalah sebuah wrapper untuk nilai boolean (true atau false).
Date	Class date menampilkan cara tanggal dan waktu direpresentasikan dalam ActionScript.
Error	Class error berisi informasi tentang error pada script yang terjadi saat runtime.
Function	Class fungsi adalah class yang merepresentasikan semua fungsi ActionScript dan yang didefinisikan.
Math	Class math adalah class yang menyediakan cara mudah mengakses konstanta matematika.
Number	Class number adalah wrapper untuk tipe data number yang primitive.
Object	Class object adalah root hirarki class pada ActionScript. Semua class adalah turunan metode dan properti class ini.
String	Class string adalah wrapper untuk tipe data string primitif.
System	Class system menyediakan informasi tentang flash player dan sistem tempat flash player sedang berjalan.

4.3 Hasil Perancangan

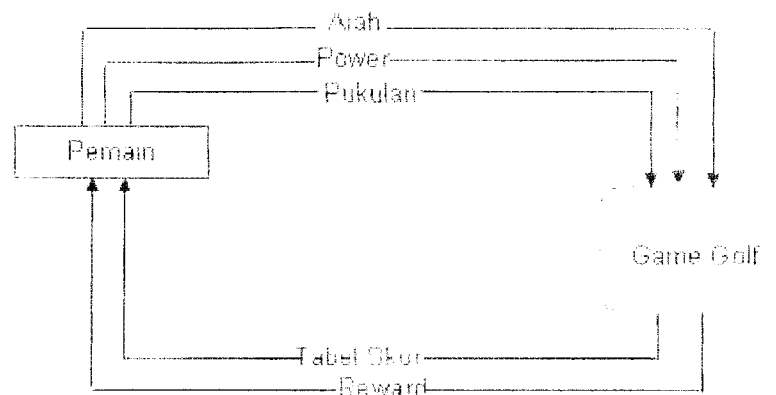
Bab ini berisi dokumentasi hasil perancangan perangkat lunak yang dibangun dalam penelitian. Dokumentasi terdiri dari model data yang hendak digunakan, desain arsitektur perangkat lunak, dan desain antarmuka aplikasi yang hendak dibangun. Hasil perancangan merupakan terjemahan kebutuhan perangkat lunak yang dibangun dalam penelitian.

4.3.1 Desain arsitektur perangkat lunak

Diagram aliran data / *data flow diagram* (DFD) adalah sebuah teknik grafis yang menggambarkan aliran informasi dan transformasi yang diaplikasikan pada saat data bergerak dari input menjadi output [PRE02a]. Agar transformasi data dapat dilakukan maka dibutuhkan beberapa proses transformasi. Berikut diagram alir data yang menjelaskan proses transformasi yang terjadi di dalam aplikasi game golf.

4.3.1.1 DFD level konteks (0)

Dalam gambar 4.1 diilustrasikan abstraksi dari sistem secara keseluruhan. Entiti eksternal yang berhubungan dengan game adalah pemain. Masukan yang diterima berupa penentuan arah, pengaturan power, dan eksekusi pukulan terhadap bola dalam permainan. Keluaran yang diinformasikan kepada pemain yaitu berupa tampilan table skor dan hasil reward (penghargaan) terhadap prestasi.



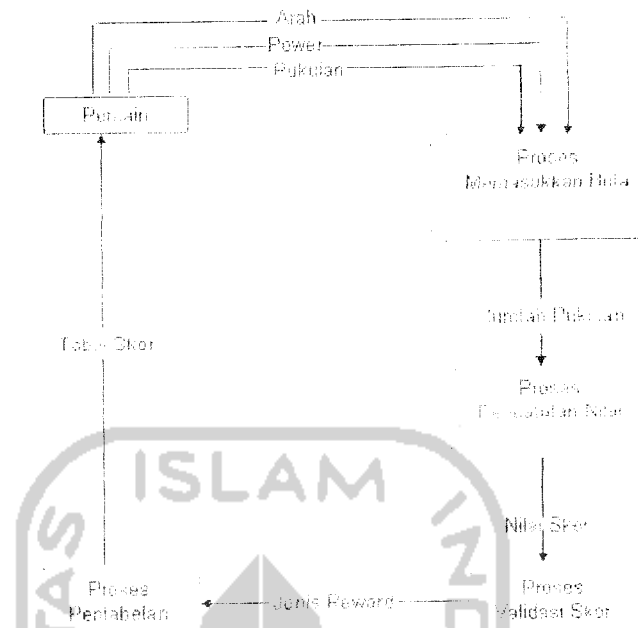
Gambar 4.1 DFD level konteks (0)

4.3.1.2 DFD level 1 proses permainan dalam game golf

Dalam gambar 4.2 mengilustrasikan beberapa proses. berikut uraian dari proses-proses tersebut:

1. Proses memasukkan bola, yang merupakan inti dari permainan, bertujuan menampung proses masukan berupa pukulan dan efek-efek yang terjadi selama permainan berjalan. Efek-efek tersebut adalah: bola bergerak, bola berhenti, bola memantul, bola bergerak cepat, bola bergerak lambat, gerak bola mendapat percepatan, gerak bola mendapat perlambatan, bola masuk ke dalam lubang.
2. Proses pencatatan nilai, bertujuan menampung jumlah pukulan yang masukannya berasal dari banyaknya pukulan yang dilakukan pemain untuk memasukan bola dalam suatu hole.
3. Proses validasi skor, bertujuan menghitung hasil skor yang diperoleh berdasarkan aturan yang telah ditetapkan dalam game. Aturan-aturan tersebut adalah :

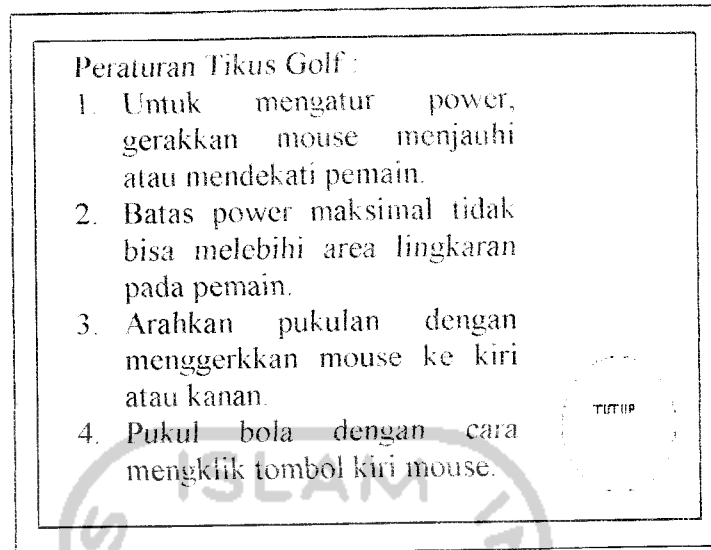
- a. *Par* : keadaan di mana bola berhasil dimasukkan pemain ke dalam lubang dengan melalui pukulan yang jumlahnya sama dengan *par* pada hole yang dimainkan.
 - b. *Hole in one* : keadaan di mana bola berhasil dimasukkan pemain ke dalam lubang dengan hanya melalui satu pukulan.
 - c. *Birdie* : keadaan di mana bola berhasil dimasukkan pemain ke dalam lubang dengan melalui pukulan yang jumlahnya sama dengan *par*-1. Contohnya, jika pada suatu hole ditentukan *par* 4, lalu pemain mampu memasukkan bola pada pukulan ke-3, maka prestasi ini disebut *birdie*.
 - d. *Bogey* : keadaan di mana bola berhasil dimasukkan pemain ke dalam lubang dengan melalui pukulan yang jumlahnya sama dengan *par*+1. Contohnya, jika pada suatu hole ditentukan *par*=4, lalu pemain baru bisa memasukkan bola pada pukulan ke-5, maka prestasi ini disebut *bogey*.
 - e. *Game Over* : keadaan di mana pemain terlalu banyak melakukan pukulan dalam suatu hole, dengan selisih 5 pukulan. Jika hal ini terjadi maka pemain harus mengulang permainan dari awal.
4. Proses pentabelan, bertujuan sebagai informasi kepada pemain yang menyelesaikan permainan. Hasil yang diperoleh yaitu berupa tabel skor yang memuat selisih antara jumlah pukulan yang dilakukan dengan *par* yang ditentukan. Hasilnya bisa positif (+) atau negatif (-). Pada tabel skor juga ditampilkan prestasi yang dicapai oleh pemain.



Gambar 4.2 DFD level 1

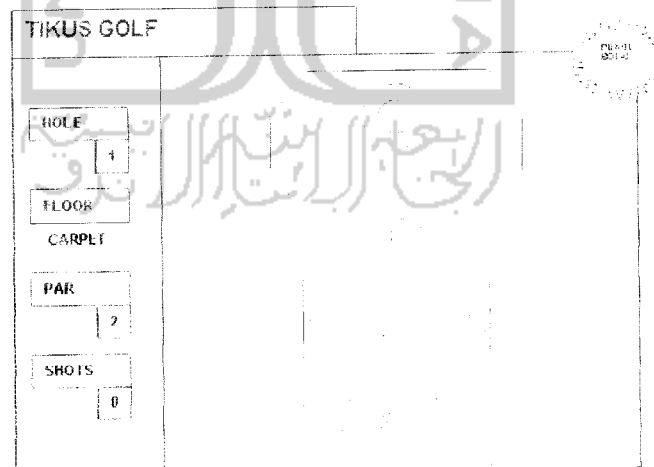
4.4 Desain Antarmuka

Antarmuka pemakai dengan sistem merupakan bagian yang tidak terpisahkan dari sebuah aplikasi perangkat lunak. Antarmuka digunakan sebagai media interaksi antara pemakai dengan sistem. Dengan adanya desain antarmuka diharapkan antarmuka yang dihasilkan bersifat ramah, efektif, dan efisien. Desain antarmuka yang baik memungkinkan pemakai untuk memanfaatkan fungsi-fungsi yang disediakan oleh perangkat lunak dengan mudah. Sehingga pemakai dapat mengoptimalkan fungsi-fungsi tersebut tanpa merasa cepat lelah dan bosan. Berikut ini adalah perancangan tampilan untuk fungsi-fungsi dasar yang tersedia di dalam perangkat lunak yang dihasilkan.



Gambar 4.4 Perancangan antarmuka instruksi game

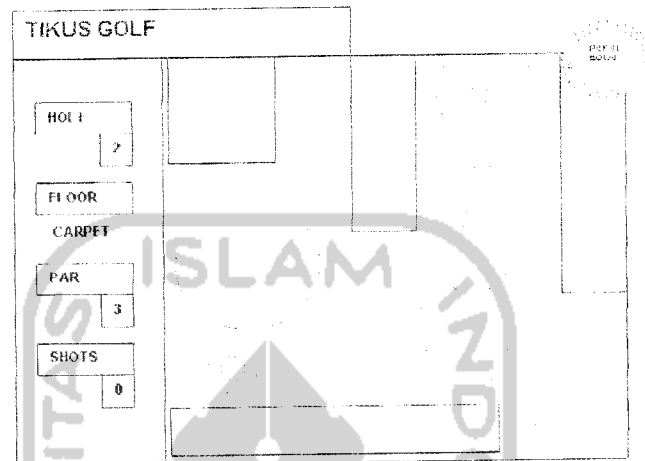
Pada sisi ini game menampilkan instruksi/petunjuk tentang bagaimana cara memainkan game golf.



Gambar 4.5 Perancangan antarmuka hole 1

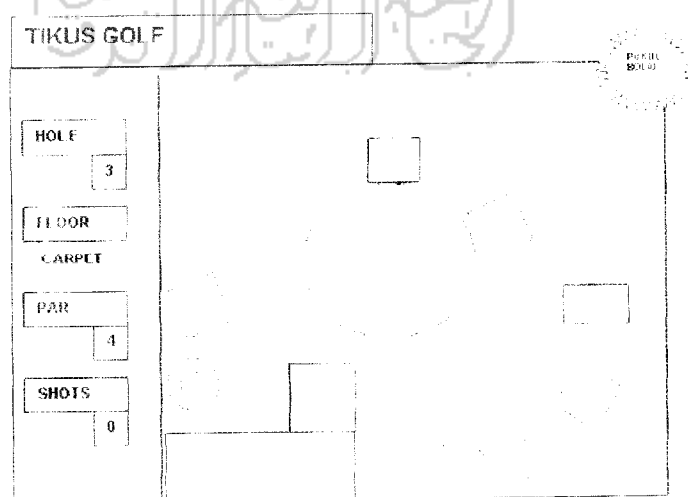
Perancangan *hole* ini berupa ruang depan sebuah rumah yang di dalamnya terletak beberapa barang yang masing-masing memiliki karakteristik bentuk dan

sudut yang berbeda. Benda-benda itu antara lain: tempat payung, sepatu, sandal, pintu dan rak.



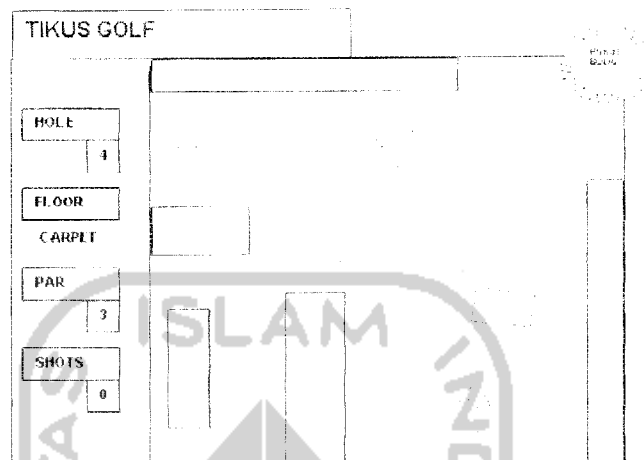
Gambar 4.6 Perancangan antarmuka hole 2

Dalam ruangan ini terdapat seekor kucing yang sedang duduk di atas akarpet berbentuk lingkaran. Efek yang terjadi saat bola melewati karpet tersebut adalah gerakan bola mengalami perlambatan atau bahkan berhenti.



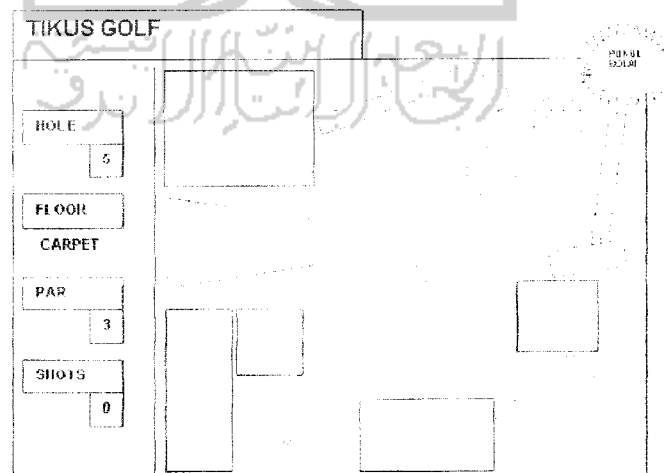
Gambar 4.7 Perancangan antarmuka hole 3

Pada *hole* ini pemain juga terdapat sebuah ban berjalan yang mana jika terlewati oleh bola, maka bola akan mengalami percepatan, seperti meluncur.



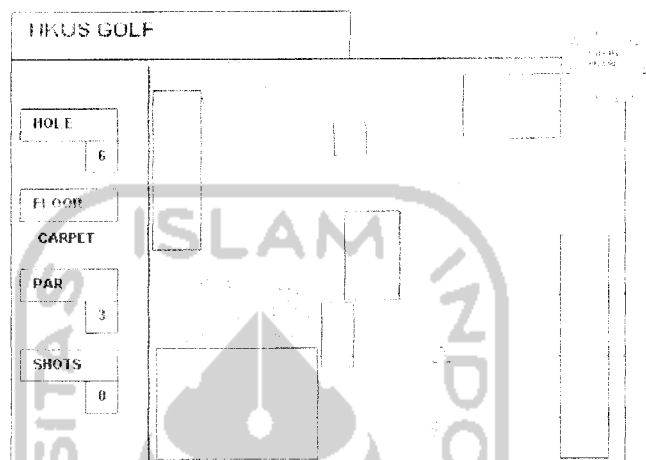
Gambar 4.8 Perancangan antarmuka hole 4

Pada hole yang keempat di dalamnya terdapat beberapa rintangan seperti: meja, kursi, dan meja untuk memasak yang memiliki sudut yang lebih sulit dari arena-arena sebelumnya.



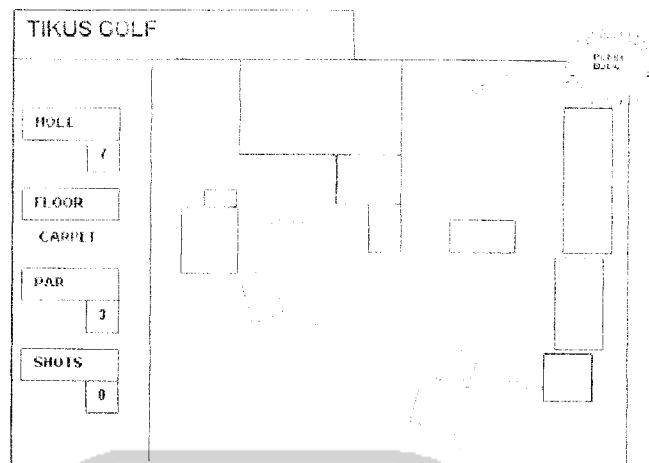
Gambar 4.9 Perancangan antarmuka hole 5

Pada hole yang kelima di dalamnya terdapat benda-benda agak besar, seperti mobil, sepeda dan alat-alat perbengkelan. Terdapat pula tetesan oli yang mana jika terlewati oleh bola, maka bola akan bergerak melambat.



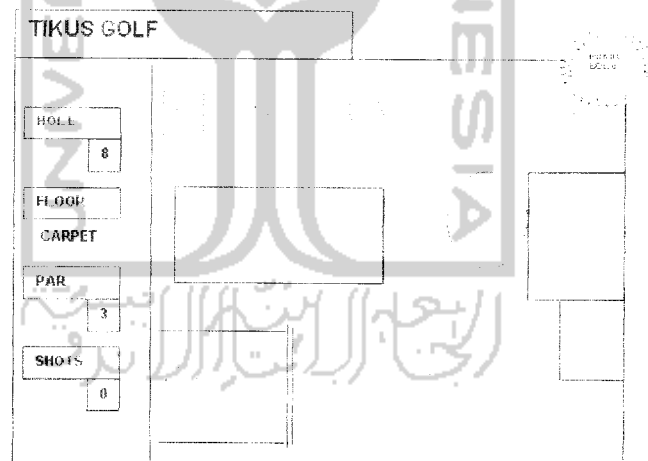
Gambar 4.10 Perancangan antarmuka hole 6

Pada ruangan ini terdapat benda-benda perkantoran dan seekor kucing yang sedang duduk di atas bantalan kursinya. Dalam arena ini pemain hanya cukup mengarahkan pukulan-pukulan dengan tepat sehingga bisa memasukkan bola sesuai par atau bahkan bisa menghemat jumlah pukulan.



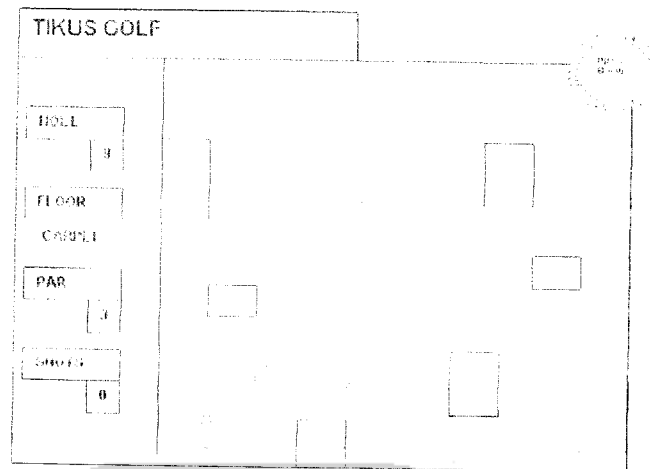
Gambar 4.11 Perancangan antarmuka hole 7

Di dalam ruangan ini, terdapat lebih banyak lagi benda-benda yang dijadikan sebagai rintangan.



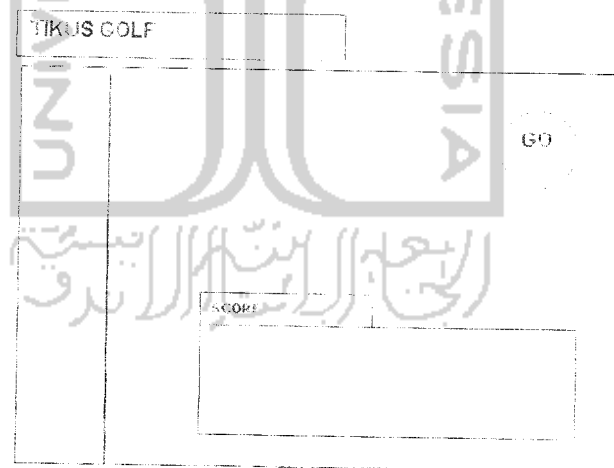
Gambar 4.12 Perancangan antarmuka hole 8

Selain terdapat ban berjalan yang bisa mempercepat gerakan bola, dalam ruangan ini juga terdapat air yang menggenang di lantainya. Efek yang terjadi saat bola melewati air tersebut adalah gerakan bola mengalami perlambatan atau bahkan berhenti.



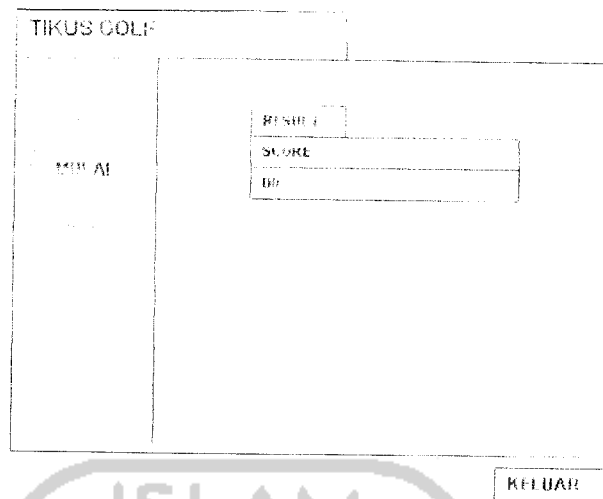
Gambar 4.13 Perancangan antarmuka hole 9

Pada arena terakhir ini termasuk pada tingkat yang tidak terlalu sulit, atau bisa juga dikatakan hanya sebagai pelengkap saja.



Gambar 4.14 Perancangan antarmuka tabel skor

Seluruh jumlah skor yang dihasilkan oleh pemain akan terantum dalam sebuah tabel yang di dalamnya terdapat keterangan-keterangan tentang jumlah pukulan dan jumlah skor.



Gambar 4.15 Perancangan antarmuka hasil skor akhir

Setelah pemain melewati 9 buah hole (ruangan), maka skor akhir langsung ditampilkan sebagai informasi kepada pemain. Setelah tampilan tabel skor ini, pemain bisa mengulang lagi permainan dari awal atau keluar dari permainan dengan menekan tombol “keluar”.

BAB V

IMPLEMENTASI

5.1 Batasan Implementasi

Batasan-batasan implementasi *game* dari sisi perangkat keras (*hardware*) dan perangkat lunak (*software*) yang digunakan dalam pengembangan perangkat lunak serta penjabaran tentang spesifikasi hardware maupun software yang diperlukan untuk menjalankan *game* yang dihasilkan akan dijelaskan dalam pembahasan di bawah ini.

5.1.1 Kebutuhan perangkat lunak

Perangkat lunak yang digunakan untuk rancang bangun *game minigolf* menggunakan bahasa ActionScript adalah sebagai berikut:

- a. Microsoft Windows XP Professional 2002
- b. Macromedia Flash 8 Professional
- c. Microsoft Word 2003
- d. ACDSee Foto Canvas versi 3.0.2.0007
- e. Microsoft Paint

5.1.2 Kebutuhan perangkat keras

Kebutuhan perangkat keras untuk pengembangan aplikasi dalam penelitian ini adalah sebagai berikut:

- a. *Processor* Intel Pentium M 1500 MHz
- b. Memori DDRAM 256 MB
- c. Intel 82852 Internal Graphic Controller

d. *Monitor, Keyboard, dan mouse*

Spesifikasi minimal perangkat keras yang diperlukan untuk menjalankan aplikasi tersebut adalah sebagai berikut:

- a. Processor pentium III 800 MHz atau yang setara
- b. Memori minimal DDRAM 128 MB atau yang setara
- c. Ruang Hardisk 40 MB
- d. *Monitor, Keyboard, dan mouse*

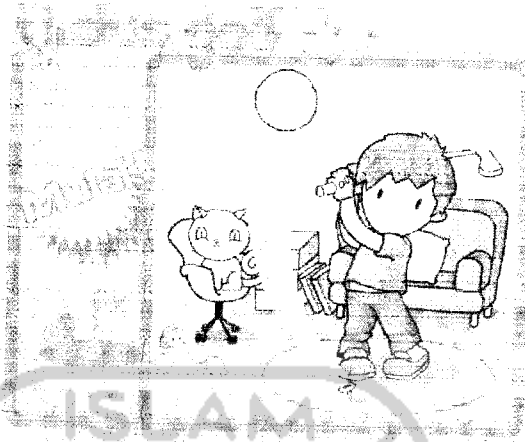
5.2. Implementasi Perangkat Lunak

Bab ke lima ini akan menjelaskan mengenai implementasi perangkat lunak berdasarkan pada analisis kebutuhan dan perancangan yang sudah diterangkan dalam dua bab terdahulu. Implementasi game akan menggunakan perangkat lunak Macromedia Flash 8 Professional dengan bahasa pemrograman ActionScript.

5.2.1 Implementasi antarmuka

5.2.1.1 Tampilan utama

Tampilan utama Game Golf menggunakan Macromedia Flash 8 Professional diilustrasikan pada gambar 5.1 berikut ini:

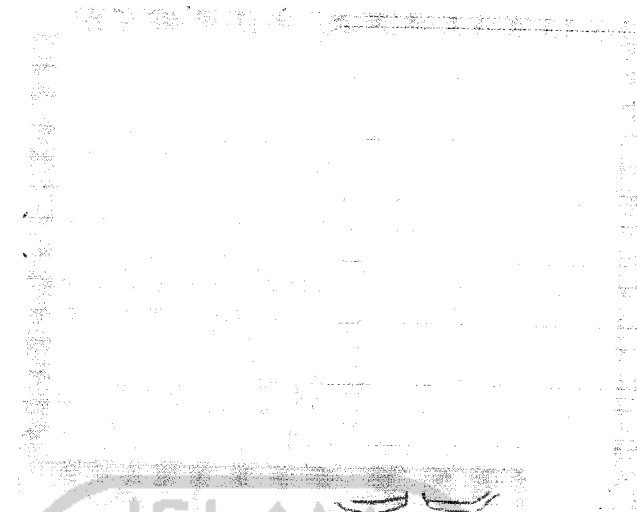


Gambar 5.1 Tampilan menu utama

Menu menjadi alat navigasi utama dalam menjalankan aplikasi ini. Pada tampilan ini terdapat tombol menu, tombol instruksi dan tombol keluar. Tombol menu berisi menu utama untuk menjalankan game. Tombol instruksi berisi petunjuk cara memainkan game. Tombol keluar digunakan untuk menutup aplikasi game.

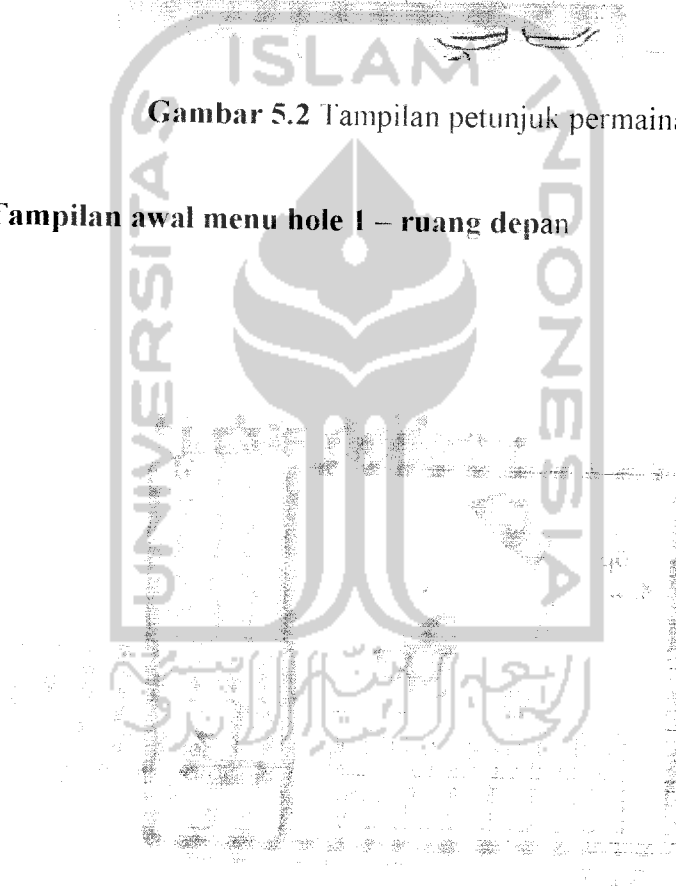
5.2.1.2 Tampilan instruksi

Pada sisi ini game menampilkan instruksi/petunjuk tentang bagaimana cara memainkan game golf.



Gambar 5.2 Tampilan petunjuk permainan

5.2.1.3 Tampilan awal menu hole 1 – ruang depan



Gambar 5.3 Tampilan awal menu hole 1 – ruang depan

Saat pemain memilih hole pertama, terdapat anak panah yang menunjuk ke sebuah ruangan, dengan keterangan yang berisikan tulisan “ruang depan”. Setelah

memilih, nantinya pemain akan menjumpai sebuah ruangan yang terlihat seperti ruang depan.

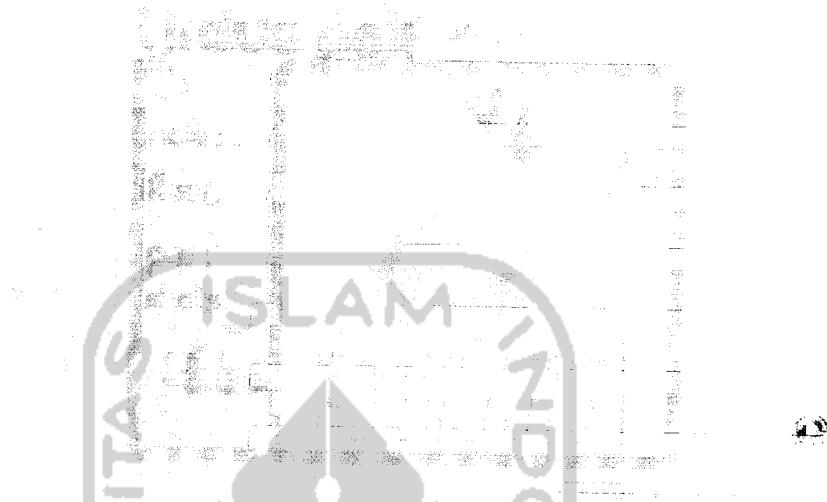
5.2.1.4 Tampilan awal menu hole 2 – ruang santai



Gambar 5.4 Tampilan awal menu hole 2 – ruang santai

Saat pemain memilih hole kedua, terdapat anak panah yang menunjuk ke sebuah ruangan, dengan keterangan yang berisikan tulisan “ruang santai”. Setelah memilih, nantinya pemain akan menjumpai sebuah ruangan yang terlihat seperti ruang santai.

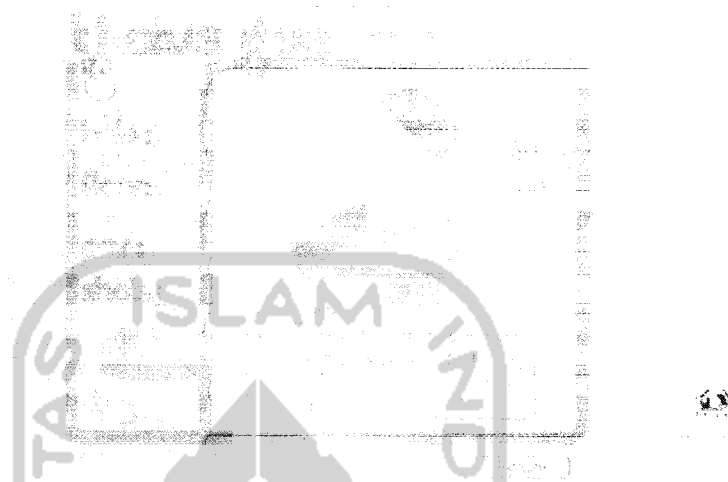
5.2.1.5 Tampilan awal menu hole 3 – ruang makan



Gambar 5.5 Tampilan awal menu hole 3 – ruang makan

Saat pemain memilih hole kedua, terdapat anak panah yang menunjuk ke sebuah ruangan. dengan keterangan yang berisikan tulisan "ruang makan". Setelah memilih, nantinya pemain akan menjumpai sebuah ruangan yang terlihat seperti ruang makan.

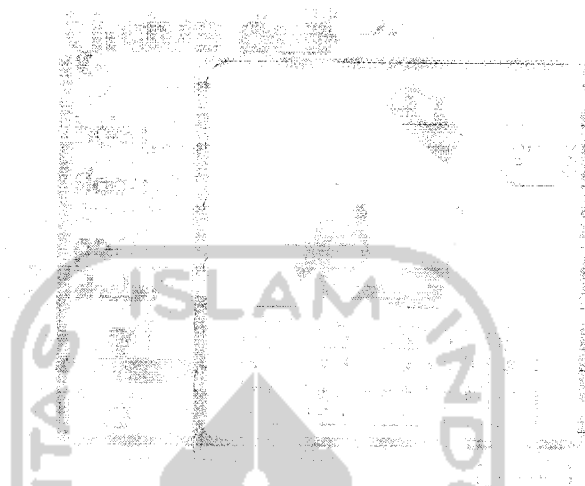
5.2.1.7 Tampilan awal menu hole 5 – garasi



Gambar 5.7 Tampilan awal menu hole 5 – garasi

Saat pemain memilih hole kedua, terdapat anak panah yang menunjuk ke sebuah ruangan, dengan keterangan yang berisikan tulisan “ruang garasi”. Setelah memilih, nantinya pemain akan menjumpai sebuah ruangan yang terlihat seperti garasi.

5.2.1.8 Tampilan awal menu hole 6 – ruang kerja

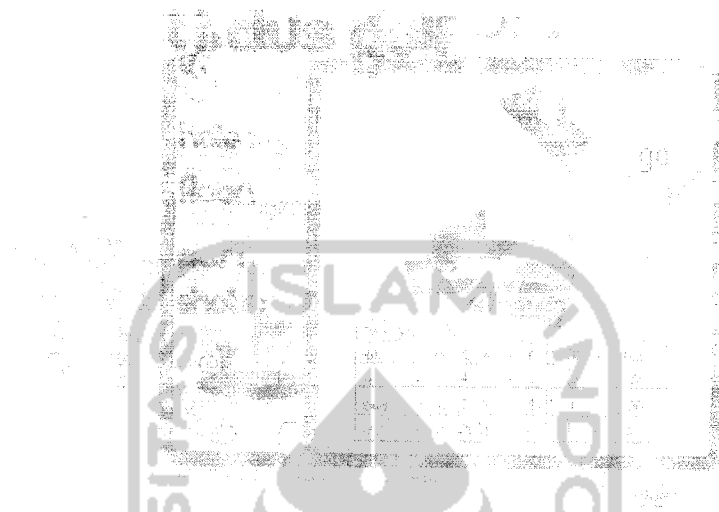


Gambar 5.8 Tampilan awal menu hole 6 – ruang kerja

Saat pemain memilih hole kedua, terdapat anak panah yang menunjuk ke sebuah ruangan, dengan keterangan yang berisikan tulisan “ruang kerja”. Setelah memilih, nantinya pemain akan menjumpai sebuah ruangan yang terlihat seperti ruang kerja.

الرَّبِّهِمْ اَللّٰهُمَّ صَلِّ وَسَلِّمْ عَلٰى رَسُوْلِكَ الْكَرِيْمِ

5.2.1.9 Tampilan awal menu hole 7 – kamar tidur



Gambar 5.9 Tampilan awal menu hole 7 – kamar tidur

Saat pemain memilih hole kedua, terdapat anak panah yang menunjuk ke sebuah ruangan, dengan keterangan yang berisikan tulisan “ruang tidur”. Setelah memilih, nantinya pemain akan menjumpai sebuah ruangan yang terlihat seperti kamar tidur.

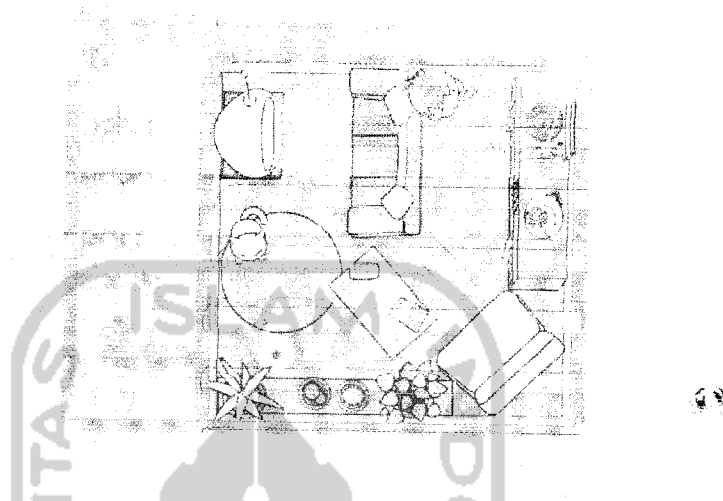
5.2.1.12 Tampilan hole 1 – ruang depan



Gambar 5.12 Tampilan hole 1 – ruang depan

Pertama kali, pemain akan menjumpai suatu hole (arena bermain golf) yang sederhana, yaitu suatu ruangan dengan rintangan yang sangat sedikit. Arena ini berupa ruang depan sebuah rumah yang di dalamnya terletak beberapa barang yang masing-masing memiliki karakteristik bentuk dan sudut yang berbeda. Benda-benda itu antara lain: tempat payung, sepatu, sandal, pintu dan rak. Terdapat sebuah lubang untuk memasukkan bola golf yang dipukul dan diarahkan oleh pemain. Dengan mengayunkan tongkat (stik) golf, efek yang terjadi adalah bola bergerak sesuai kekuatan (power) yang diatur menggunakan gerakan mouse.

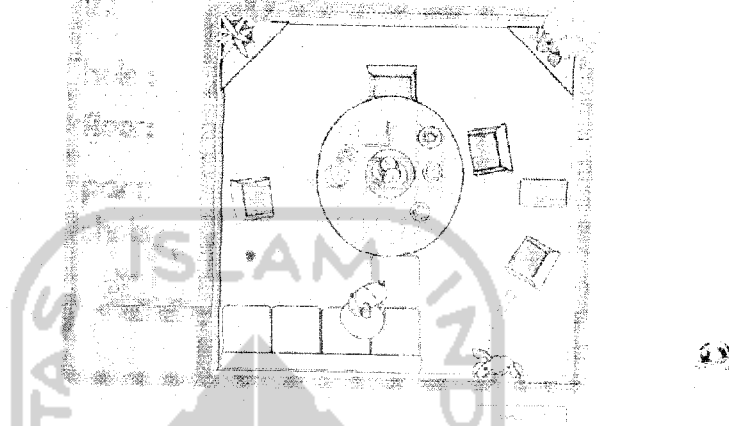
5.2.1.13 Tampilan hole 2 – ruang santai



Gambar 5.13 Tampilan hole 2 – ruang santai

Pada hole yang kedua, pemain akan menjalani permainan di dalam suatu ruangan yang tampak seperti ruang santai, yang tentu saja dengan rintangan yang lebih banyak dari pada hole yang pertama. Dalam ruangan ini terdapat seekor kucing yang sedang duduk di atas akarpet berbentuk lingkaran. Efek yang terjadi saat bola melewati karpet tersebut adalah gerakan bola mengalami perlambatan atau bahkan berhenti.

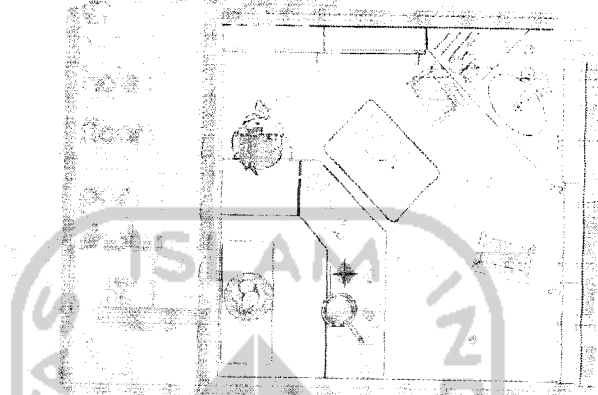
5.2.1.14 Tampilan hole 3 – ruang makan



Gambar 5.14 Tampilan hole 3 – ruang makan

Pada hole yang ketiga, pemain akan menjalani permainan di dalam suatu ruangan yang tampak seperti ruang makan, yang tentu saja dengan rintangan yang lebih banyak dari pada hole yang pertama dan kedua. Di sini juga terdapat sebuah ban berjalan yang mana jika terlewati oleh bola, maka bola akan mengalami percepatan, seperti meluncur.

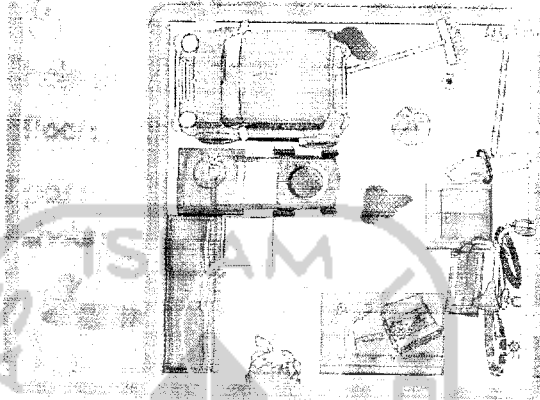
5.2.1.15 Tampilan hole 4 -- dapur



Gambar 5.15 Tampilan hole 4 -- dapur

Pada hole yang keempat, pemain akan menjalani permainan di dalam suatu ruangan yang tampak seperti dapur. Di dalamnya terdapat beberapa rintangan seperti: meja, kursi, dan meja untuk memasak yang memiliki sudut yang lebih sulit dari arena-arena sebelumnya.

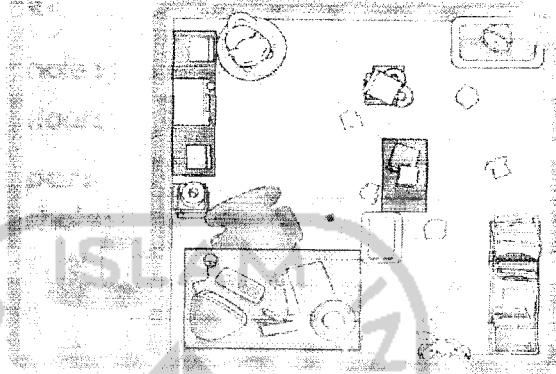
5.2.1.16 Tampilan hole 5 – garasi



Gambar 5.16 Tampilan hole 5 – garasi

Pada hole yang kelima, pemain akan menjalani permainan di dalam suatu ruangan yang tampak seperti garasi. Di dalamnya terdapat benda-benda agak besar, seperti: mobil, sepeda dan alat-alat perbengkelan. Terdapat pula tetesan oli yang mana jika terlewati oleh bola, maka bola akan bergerak melambat.

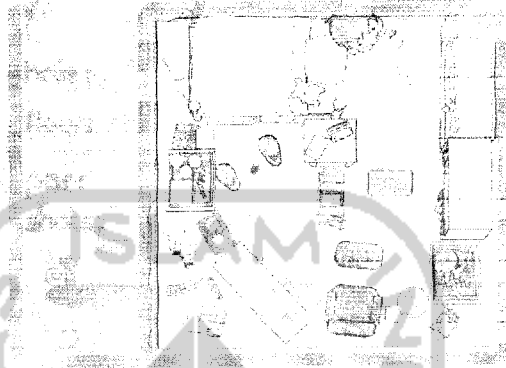
5.2.1.17 Tampilan hole 6 – ruang kerja



Gambar 5.17 Tampilan hole 6 – ruang kerja

Pada hole yang keenam, pemain akan menjalani permainan di dalam suatu ruangan yang tampak seperti ruang kerja. Di dalam ruangan ini terdapat benda-benda perkantoran dan seekor kucing yang sedang duduk di atas bantalan kursinya. Dalam arena ini pemain hanya cukup mengarahkan pukulan-pukulan dengan tepat sehingga bisa memasukkan bola sesuai par atau bahkan bisa menghemat jumlah pukulan.

5.2.1.18 Tampilan hole 7 – kamar tidur

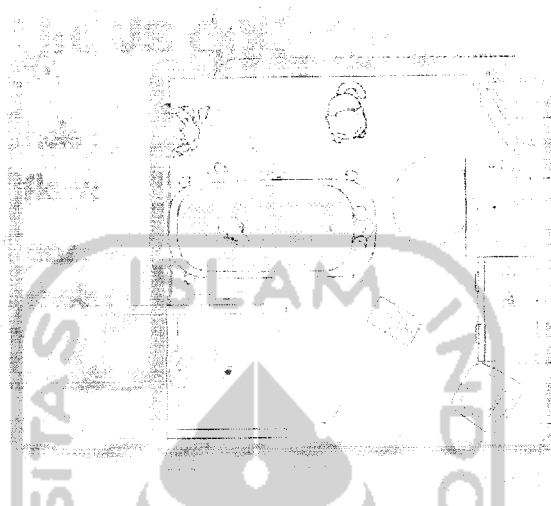


Gambar 5.18 Tampilan hole 7 – kamar tidur

Pada hole yang ketujuh, pemain akan menjalani permainan di dalam suatu ruangan yang tampak seperti kamar tidur. Di dalam ruangan ini, terdapat lebih banyak lagi benda-benda yang dijadikan sebagai rintangan.

الجامعة الإسلامية
البراقية

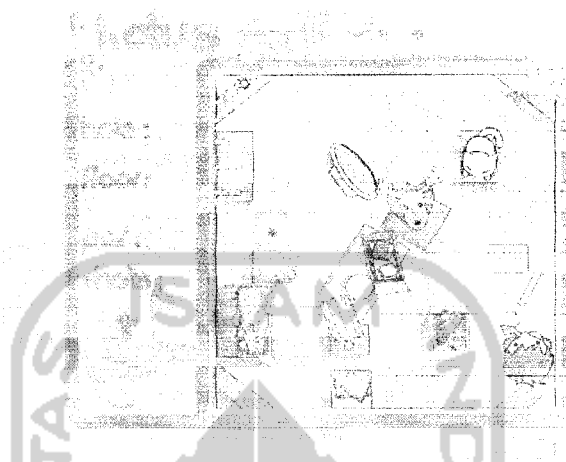
5.2.1.19 Tampilan hole 8 – kamar mandi



Gambar 5.19 Tampilan hole 8 – kamar mandi

Pada hole yang kedelapan, pemain akan menjalani permainan di dalam suatu ruangan yang tampak seperti kamar mandi. Selain terdapat ban berjalan yang bisa mempercepat gerakan bola, dalam ruangan ini juga terdapat air yang menggenang di lantainya. Efek yang terjadi saat bola melewati air tersebut adalah gerakan bola mengalami perlambatan atau bahkan berhenti.

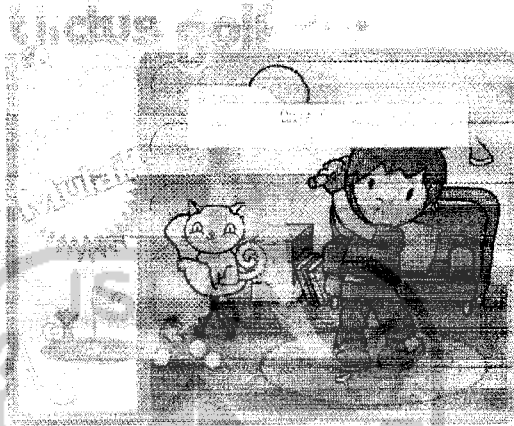
5.2.1.20 Tampilan hole 9 – gudang



Gambar 5.20 Tampilan hole 9 – gudang

Pada hole yang kesembilan yang merupakan hole terakhir, pemain akan menjalani permainan di dalam suatu ruangan yang tampak seperti gudang yang di dalamnya terdapat barang-barang bekas seperti: peti kayu, cermin, kepala hewan buruan, dan juga terdapat ban berjalan. Pada arena terakhir ini termasuk pada tingkat yang tidak terlalu sulit, atau bisa juga dikatakan hanya sebagai pelengkap saja.

5.2.1.21 Tampilan skor akhir



Gambar 5.21 Tampilan skor akhir

Setelah pemain melewati 9 buah hole (ruangan), maka skor akhir langsung ditampilkan sebagai informasi kepada pemain. Setelah tampilan tabel skor ini, pemain bisa mengulang lagi permainan dari awal atau keluar dari permainan dengan menekan tombol “keluar”.

5.2.2 Implementasi fungsi-fungsi dalam action script

Bentuk implementasi pemrograman bahasa *Action Script* akan dijelaskan pada pembahasan bab ini. *Action Script* memandang *software* bagian per bagian, dan menggambarkan satu bagian tersebut dalam satu objek. Dalam rancang bangun game golf, objek-objek baru bisa dibuat dengan menggunakan salah satu class pada syntax *Action Script* yaitu di dalam file dengan ekstensi *.as. Letak file tersebut tidak berada di dalam file utama *project* game Macromedia Flash, namun terletak di luar file. File *.as yang berisi *class-class* kemudian bisa dipanggil untuk digunakan oleh *project* utamanya. Berikut adalah contoh-contoh pembuatan *class* dan implentasi pemanggilannya dalam game golf.

5.2.2.1 Pengelompokan fungsi pada *movie clip joueur* (pemain)

```

vitesse = Math.sqrt(vx * vx + vy * vy);
if (_root.ballIn)
{
    _root.joueur._alpha = 0;
    _root.ombre._alpha = 0;
    _root.cercle._alpha = 0;
}
else if (vitesse > 0.200000)
{
    _root.joueur._alpha = 100 - 90 * vitesse;
    _root.ombre._alpha = 95 - 90 * vitesse;
    _root.cercle._alpha = 100 - 90 * vitesse;
}
else
{
    vx = 0;
    vy = 0;
    _root.joueur._alpha = 100;
    _root.ombre._alpha = 95;
    _root.cercle._alpha = 100;
} // end if
};

```

5.2.2.2 Pengelompokan fungsi pada *movie clip balle* (bola)

```

{
    xinit = x;
    yinit = y;
    balle._x = yinit;
    balle._y = yinit;
    joueur._x = yinit;
    joueur._y = yinit;
    balle.alpha = 100;
    joueur.alpha = 100;
    cercle._x = 1000;
    cercle._y = 1000;
    cercle.alpha = 100;
    ballin = false;
    vx = 0;
    vy = 0;
}
root.onEnterFrame = function ()
{
    if (_root.cercle.hitTest(_root.cible._x, _root.cible._y, true))
    {
        root.delta = Math.sqrt((_root.cible._x - _root.cercle._x) *
        (_root.cible._x - _root.cercle._x) + (_root.cible._y - _root.cercle._y) *
        (_root.cible._y - _root.cercle._y));
        _root.joueur.pawa._yscale = _root.delta;
    }
    else
    {
        _root.delta = 0;
        _root.joueur.pawa._yscale = _root.delta;
    } // end if
    if (_root.ballin.hitTest(_root.trou) && vitesse < vitesse_in)
    {
        _root.joueur.alpha = 0;
        _root.ombre.alpha = 0;
        _root.cercle.alpha = 0;
        _root.balle._x = _root.trou._x;
        _root.balle._y = _root.trou._y;
        _root.balle.alpha = 0;
        if (_root.ballin != true)
        {
            _root.inside.gotoAndPlay(2);
        } // end if
    } // end if
}

```

5.2.2.3 Pengelompokan fungsi pada *movie clip cercle* (lingkaran putih)

```

anz = this.balle._x;
any = this.balle._y;
long = Math.sqrt(vx * vx + vy * vy);
nn = Math.floor(long / r2);
px = vx * r2 / long;
py = vy * r2 / long;
for (i = 1; i <= nn; i++)
{
    if (i <= nn)
    {
        x = anz + i * px;

```



```

        y = any + vy;
    }
    else
    {
        x = any + vx;
        y = any + vy;
    } // end if
    if (!this.terrain.hitTest(x, y, true))
    {
        ax = ax + (i - 1) * px;
        ay = any + (i - 1) * py;
        vx = vx * perchehit;
        vy = vy * petlehit;
        _recb.cureband.play();
        rebendir();
        break;
    }
    continue;
} // end if
if (i == un + 1)
{
    atfdir();
} // end if
} // end of for

```



BAB VI

ANALISIS KINERJA

Analisis kinerja perangkat lunak digunakan untuk menguji validitas perangkat lunak yang dihasilkan, terutama fungsi-fungsi yang ada didalamnya. Setelah melewati proses analisis ini, diharapkan perangkat lunak yang dihasilkan dapat memenuhi kebutuhan yang diharapkan.

Pengujian GUI atau *Graphical User Interface* bertujuan memeriksa kebenaran dari antarmuka yang dihasilkan. Kebenaran yang dimaksud adalah kesesuaian antara antarmuka terhadap fungsi-fungsi yang seharusnya ada dan diakomodasi oleh antarmuka tersebut.

6.1 Pengujian Menu Game

Pengujian dilakukan dengan cara pemanggilan fungsi-fungsi pada file Script1.as yang memuat class-class utama. Berikut adalah fungsi-fungsi tersebut :

6.1.1 Pemanggilan pada tiap movieclip pada posisi x dan y dengan menyesuaikan jarak movieclip

```
Movieclip.prototype.calcdistance = function (mvc)
{
    var dx = this._x - mvc._x;
    var dy = this._y - mvc._y;
    return (Math.sqrt(dx * dx + dy * dy));
};
```

6.1.2 Pemanggilan pada tiap movieclip dari radius yg telah ditentukan

```

Movieclip.prototype.calcradian = function (mvc)
{
    var dx = this._x - mvc._x;
    var dy = this._y - mvc._y;
    return (Math.atan2(dy, -dx));
};
pertehit = 0.800000;
pertesol = 0.960000;
vit = 0.330000;
vitesse_in = 7;
coup = 0;
//nm hole
niveau = 1;
for (i = 1; i <= 9; i++)
{
    _root["score" + i] = 0;
} // end of for

```

Fungsi-fungsi di atas dipanggil oleh game menggunakan perintah:

```
include Script1.as
```

Ketika game dijalankan, maka tampilan utama akan muncul. Pada tampilan utama game, terdapat tombol mulai, digunakan untuk memulai permainan.

6.1.3 Penilaian pada *par*

```

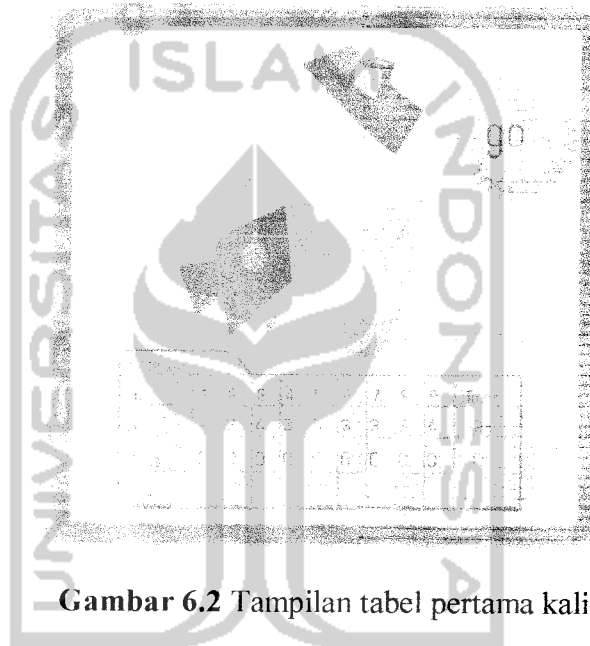
scoretotal = 0;
_root.par1 = 2;
_root.par2 = 3;
_root.par3 = 4;
_root.par4 = 3;
_root.par5 = 3;
_root.par6 = 3;
_root.par7 = 3;
_root.par8 = 3;
_root.par9 = 4;
k = -1;
r = 10;
r2 = r - 2;
inca = 0.080000;
repositionner = function (x, y)

```



Gambar 6.1 Tombol mulai

Saat tombol mulai diklik, maka akan muncul tampilan tabel skor (Score Card) yang masih kosong. Tab *Hole* menunjukkan arena (ruangan) yang akan dijalani pemain untuk menyelesaikan permainan. Tab *Par* menunjukkan jumlah maksimal batas pukulan yang seharusnya dilakukan. Tab *Shots* menunjukkan jumlah pukulan yang dilakukan oleh pemain dalam suatu hole.



Gambar 6.2 Tampilan tabel pertama kali

Pada tampilan awal tabel skor, pemain memulai untuk membuka Hole pertama dengan mengklik tombol “go”. Setelah tombol “go” diklik, maka tampilan Hole pertama akan muncul dengan beberapa keterangan.



Gambar 6.3 Tombol go

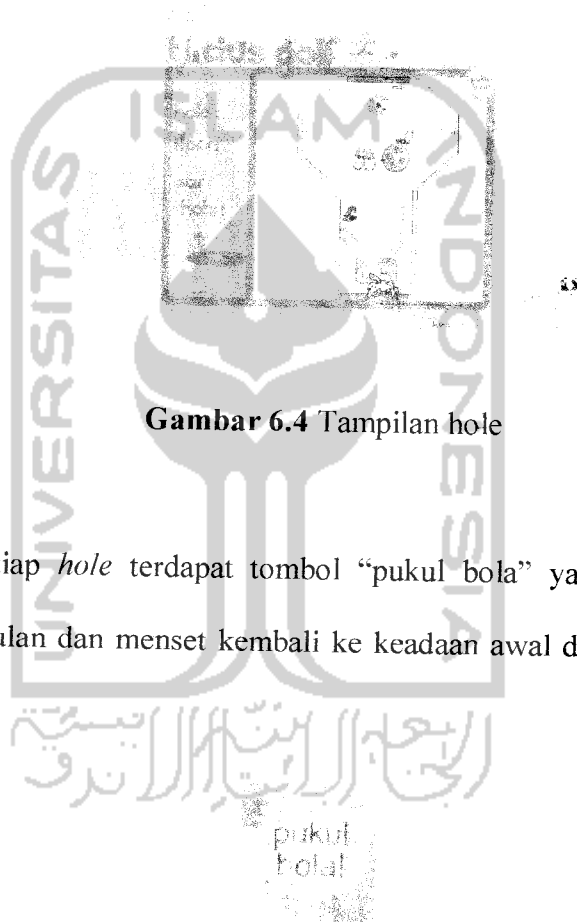
Keterangan yang terdapat dalam *hole* :

Hole : menunjukkan hole ke-berapa yang sedang dijalankan oleh pemain

Floor : menunjukkan jenis lantai pada hole yang dijalankan

Par : menunjukkan batas pukulan yang telah ditentukan

Shots : menunjukkan jumlah pukulan yang dilakukan oleh pemain

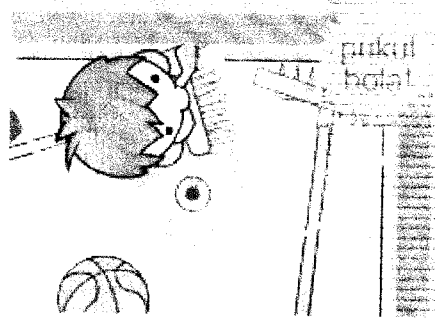


Gambar 6.4 Tampilan hole

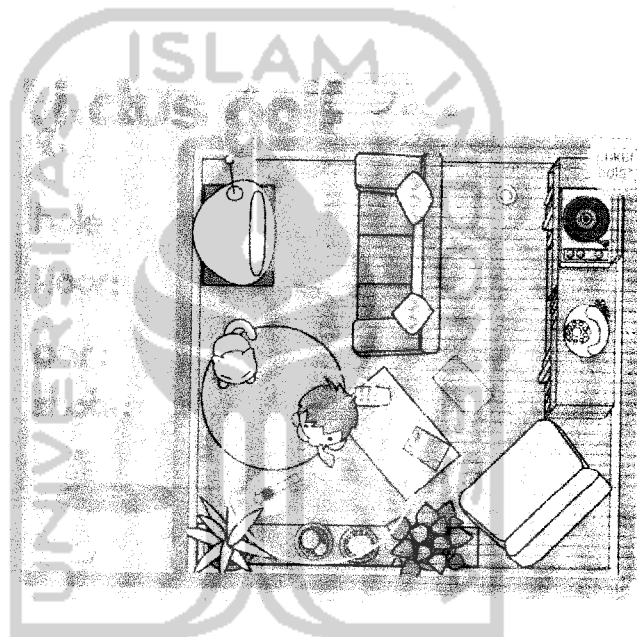
Pada setiap *hole* terdapat tombol “pukul bola” yang digunakan untuk mengulang pukulan dan menset kembali ke keadaan awal di mana bola pertama kali dipukul.

Gambar 6.5 Tombol pukul bola

Saat tombol “pukul bola” diklik, yang terjadi adalah bola kembali ke posisi awal permainan. Tombol ini digunakan saat bola tersangkut di suatu sudut ruangan atau terjebak pada suatu benda dan tidak bisa bergerak lagi.



Gambar 6.6 Keadaan saat bola terjebak di sudut



Gambar 6.7 Saat bola masuk

Saat bola masuk ke lubang, yang terjadi adalah perubahan pada tabel skor. Semula, jumlah pukulan dan selisih pada tabel skor adalah 0. Setelah bola masuk, otomatis hole pertama juga selesai dan banyaknya pukulan yang telah dilakukan selama hole pertama akan tercatat pada tabel skor.

Hit	1	2	3	4	5	6	7	8	9	Total
P1	2	3	4	3	3	3	3	4		29
P2	3	0	0	0	0	0	0	0		3

Gambar 6.8 Tabel skor

6.1.4 Mulai penentuan untuk posisi pemukulan yg tepat

Sebelum pemain melakukan pukulan, ada beberapa poin penting yang harus dilakukan, yaitu menentukan arah pukulan, mengatur power. Berikut fungsi class untuk mengatur arah yang tepat sesuai keinginan pemain.

```

angle = function (aa, inc)
{
  do
  {
    t1 = this.terrain.hitTest(anx + Math.cos(aa) * r, any +
Math.sin(aa) * r, true);
    t2 = this.terrain.hitTest(anx + Math.cos(aa + inc) * r, any +
Math.sin(aa + inc) * r, true);
    aa = aa + inc;
  } while (aa < 6.280000 && (!t1 && !t2 || t1 && t2))
  return (t1 ? (aa - inc) : (aa));
};

//
rebondir = function ()
{
  for (i = 1; i < 3; i++)
  {
    x0 = (x + anx) / 2;
    y0 = (y + any) / 2;
    if (!this.terrain.hitTest(x0, y0, true))
    {
      x = x0;
      y = y0;
      continue;
    } // end if
    anx = x0;
    any = y0;
  } // end of for
  a = angle(0, inca);
  if (a >= 6.280000 || !t1 && !t2 || t1 && t2)
  {
    vx = vx * -1;
    vy = vy * -1;
    x = anx;
    y = any;
  }
  else
  {
    ca = Math.cos(a);
    sa = Math.sin(a);
  }
}

```

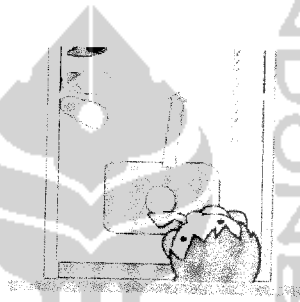
```

    tpx = -sa * k * (-vx * sa + vy * ca) + ca * (vx * ca + vy * sa);
    vy = ca * k * (-vx * sa + vy * ca) + sa * (vx * ca + vy * sa);
    vx = tpx;
    long = Math.sqrt(vx * vx + vy * vy);
} // end if
afficher();
};

```

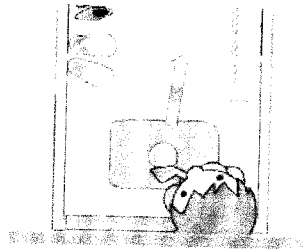
Untuk *script* di atas, cara pemanggilannya sama yaitu dengan memasukkan perintah `include` pada program utama game.

Pengaturan power untuk kecepatan bola diatur menggunakan gerakan mouse. Dalam hal ini juga diberi batasan yaitu pemain hanya bisa memukul jika mouse digerakkan selama mouse berada dalam daerah batasan yang berupa lingkaran.



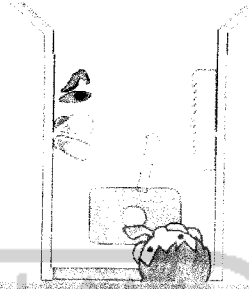
Gambar 6.9 Pengaturan power penuh

Jika mouse digerakkan hingga batas lingkaran, yang arahnya yaitu dari dalam ke luar, maka power akan maksimal. Artinya, stik golf akan memiliki tenaga penuh dan bola akan bergerak dengan cepat.



Gambar 6.10 Pengaturan power tidak penuh

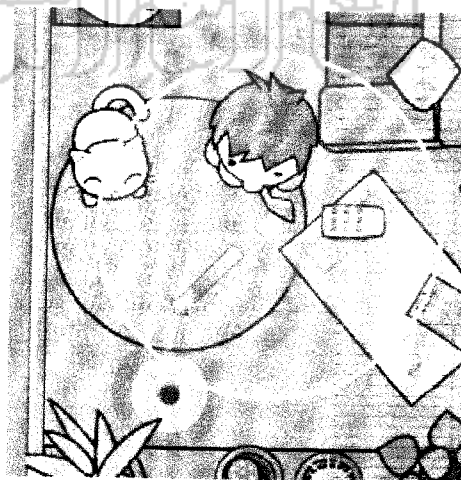
Jika mouse digerakkan tidak sampai pada batas lingkaran dan arahnya yaitu dari dalam ke luar, maka power tidak maksimal. Besar kecilnya power bisa diatur menurut kemauan pemain yaitu dengan cara menjauhkan atau mendekatkan arah mouse dari posisi awal bola.



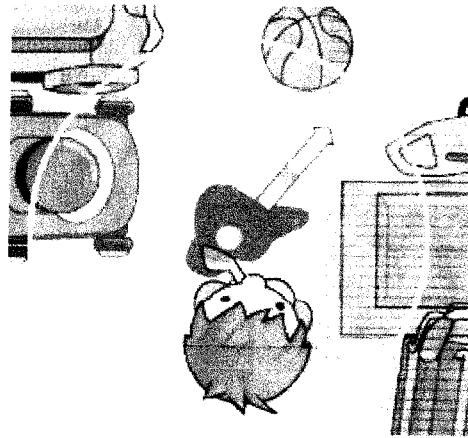
Gambar 6.11 Mouse digerakkan di luar batas area

6.1.5 Perlambatan pada gerakan bola

Perlambatan akan terjadi pada saat bola dipukul dan bergerak melewati karpet. Efek tersebut dibuat karena menyesuaikan dengan keadaan nyata, yaitu bahan karpet secara otomatis akan memperlambat gerakan bola yang melewati. Begitu juga terjadi jika bola melewati benda-benda tertentu, contohnya seperti oli kotor. Benda tersebut juga akan memperlambat gerakan bola, bahkan bisa membuat bola berhenti.

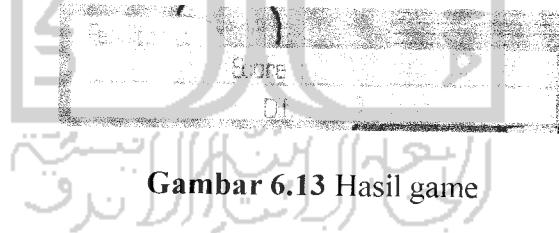


Gambar 6.12 Perlambatan pada karpet



Gambar 6.13 Perlambatan pada oli

Suatu game akan lebih menarik jika pada saat akhir permainan, nilai ahsil yang diperoleh pemain mampu untuk ditampilkan sebagai penghargaan. Hal ini akan membuat pemain ingin mengulang lagi permainan karena terdorong untuk lebih meningkatkan kemampuanny dalam game tersebut.



Gambar 6.13 Hasil game

BAB VII

SIMPULAN DAN SARAN

7.1 Simpulan

Hasil dari rancang bangun game ini adalah sebuah aplikasi *game minigolf* yang dalam rancang bangunnya menggunakan bahasa *Action Script* pada *Macromedia Flash*. Tentunya penelitian ini diharapkan mampu menjadi referensi bagi para pengembang *game* (*game developer*) dalam mendalami *Action Script* sebagai bahasa pemrograman *game* yang berorientasi pada objek. Keunggulan dan kemudahan *Action Script* bermanfaat untuk membantu para pengembang *game* agar lebih efisien dalam merancang bangun *game-game* mereka, baik itu digunakan sebagai fitur tambahan pada *website* atau dalam kegunaan lainnya selain itu diharapkan pula para pengembang *game* dapat menghasilkan berbagai macam *game* yang canggih berbasis pada teknologi *object-oriented*. Pengujian telah dilakukan untuk mengetahui kinerja aplikasi *game*, baik itu dalam segi tampilan ataupun *game playing*. Dari hasil pengujian didapat kesimpulan sebagai berikut:

- a. Proses *game playing* yang diimplementasikan dalam tampilan yang *user friendly*, memudahkan pemain untuk menjalankan/memainkan *game* hingga akhir permainan.
- b. *Reward* (penghargaan) yang diterima pemain apabila telah menyelesaikan *game minigolf* ini, berupa hasil skor akhir dan penobatan juara.

Secara keseluruhan fitur yang tersedia dapat berjalan dengan benar, hanya saja mungkin aplikasi *game* tidak mempunyai kemampuan untuk menampilkan dan menunjukkan seluruh fungsi dalam pemrograman dengan metode pengembangan berorientasi objek.

7.2 Saran

Beberapa saran untuk para pengembang aplikasi *game* terutama bagi mereka yang berbasis pada multimedia dan animasi melalui penelitian ini adalah sebagai berikut:

- a. Penerapan teknologi *game programming* menggunakan metode-metode yang baru sehingga fitur yang dibangun dapat lebih bervariasi dan lebih canggih.
- b. *Game playing* diharapkan bisa dibuat lebih menarik dengan tampilan-tampilan yang mudah dimengerti (*user friendly*) bagi para *user / gamer*.



DAFTAR PUSTAKA

[HAR04]	Hariyanto, Bambang, <i>Rekayasa Sistem Berorientasi Objek, Informatika, Bandung, 2004</i>
[JEP04]	Jeprie, Mohammad, <i>Membuat Games dengan Flash MX, PT Elex Media Komputindo, Jakarta, 2004</i>
[SYA05]	Syarif, Ary Maulana, <i>Mastering Action Script Macromedia Flash MX 2004, PT Elex Media Komputindo, Jakarta, 2005</i>
[ZEE05]	Zeembry, <i>123 Tip & Trik Action Script Flash MX 2004, PT Elex Media Komputindo, Jakarta, 2005</i>
[GIT05]	Gittings, Gabriel, <i>OOP and Inheritance in Flash,</i> http://www.actionscript.org , Diakses tanggal 23 Desember 2006
[CUR05]	Currie, Christoper, <i>Advanced Actionscript 2.0,</i> http://www.actionscript.org , Diakses tanggal 23 Desember 2006
[XAV05]	Xavier, <i>Class and Object in Games,</i> http://www.actionscript.org , Diakses tanggal 23 Desember 2006