

**Rancang Bangun Pembuatan Game Latihan Sepak Bola
Menggunakan Metode Finite State Machine (FSM)**

SKRIPSI



**Disusun Oleh :
Alexander Taka Logo
09.18.002**

**JURUSAN TEKNIK INFORMATIKA S-1
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
2013**

LEMBAR PERSETUJUAN

**RANCANG BANGUN PEMBUATAN GAME LATIHAN SEPAK BOLA
MENGUNAKAN METODE FINITE STATE MACHINE (FSM)**

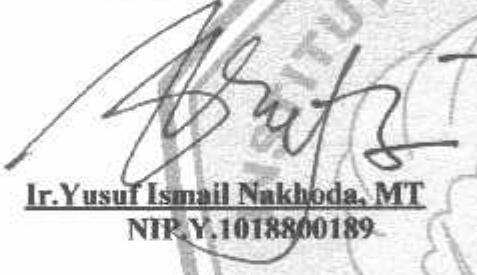
SKRIPSI

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

**Disusun Oleh :
Alexander Taka Logo
09.18.002**

Diperiksa dan disetujui oleh

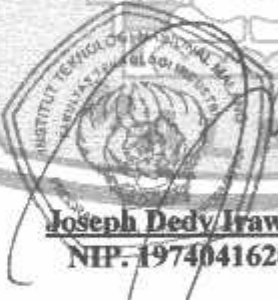
Dosen Pembimbing I


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

Dosen Pembimbing II


Nurlaily Vendyansyah, ST


Mengetahui,
Ketua Jurusan Teknik Informatika S-1


Joseph Dedy Irawan, ST, MT
NIP. 197404162005021002

**JURUSAN TEKNIK INFORMATIKA S-1
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
2013**



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

PERNYATAAN KEASLIAN SKRIPSI

Saya yang bertanda tangan di bawah ini:

Nama : Alexander Taka Logo
Nim : 09.18.002
Program Studi : Teknik Informatika S-1
Fakultas : Teknologi Industri

Menyatakan dengan sesungguhnya bahwa Skripsi saya yang berjudul:

**“RANCANG BANGUN PEMBUATAN GAME LATIHAN SEPAK BOLA
MENGUNAKAN METODE FINITE STATE MACHINE (FSM)”**

Adalah Skripsi saya sendiri bukan duplikat serta mengutip atau menyadur seluruhnya karya orang lain kecuali dari sumber aslinya.

Malang, 1 Agustus 2013

Yang membuat pernyataan

METERAI
TEMPEL
F77CCABF711543796
6000
DJP

Alexander Taka Logo

RANCANG BANGUN PEMBUATAN GAME LATIHAN SEPAK BOLA MENGUNAKAN METODE FINITE STATE MACHINE (FSM)

Alexander Taka Logo
09.18.002

Teknik Informatika S-1, Institut Teknologi Nasional Malang
e-mail : alexander_art05@yahoo.com

Dosen Pembimbing : I. Ir.Yusuf Ismail Nakhoda, MT
II. Nurlaily Vendyansyah,ST

Abstrak

Seiring dengan perkembangan teknologi. *Game* adalah salah satu media hiburan yang telah berkembang pesat di masyarakat. Perkembangan *game* dengan menampilkan grafis, dan sistem yang lebih baik menjadi hal utama untuk menarik minat para pecintanya. Salah satu unsur yang berperan penting dalam merancang sebuah *game* yaitu kecerdasan buatan. *Artificial Intelligence* (kecerdasan buatan) adalah bidang ilmu komputer yang meneliti dan mempelajari bagaimana agar komputer dapat bertindak dan mengerjakan pekerjaan seperti yang dilakukan oleh manusia. Dengan adanya kecerdasan buatan, elemen-elemen dalam *game* dapat berperilaku sealam mungkin layaknya manusia.

Dipilihnya FSM (*Finite State Machine*), sebagai metode perancangan *game* ini karena FSM merupakan sebuah mesin *abstrak* yang berfungsi untuk mendefinisikan sekumpulan kondisi yang menentukan kapan suatu *state* (keadaan) harus berubah. Setiap *state* yang sedang dijalankan tersebut menentukan perilaku yang terjadi pada objek yang bersangkutan.

Suatu objek dapat bergerak menuju posisi tujuan dari area dimana dia berada dengan melakukan improvisasi tertentu, dapat juga disebut dengan istilah *steering behaviors*.

Pengujian metode *finite state machine* dalam *game* ini sesuai, karena pemain kiper dalam *game* ini dapat memberikan respon sesuai dengan gerak bola.

Kata kunci: *Artificial Intelligence, Finite State Machine, Steering Behaviors, Game.*

KATA PENGANTAR

Puji syukur penulis panjatkan ke hadirat Tuhan yang maha esa atas karunia, rahmat dan hidayahNya, sehingga penulis dapat menyelesaikan penelitian yang berjudul

“RANCANG BANGUN PEMBUATAN GAME LATIHAN SEPAK BOLA MENGUNAKAN METODE FINITE STATE MACHINE (FSM)”

Skripsi ini dapat terselesaikan tidak terlepas dari dukungan berbagai pihak. Penulis mengucapkan terima kasih yang sebesar-besarnya kepada :

1. Bapak Ir. Soeparno Djiwo, MT selaku Rektor Institut Teknologi Nasional Malang.
2. Bapak Ir. H. Anang Subardi, MT selaku Dekan Fakultas Teknologi Industri Institut Teknologi Nasional Malang.
3. Bapak Joseph Dedy Irawan, ST, MT selaku Kepala Jurusan Teknik Informatika S-1 Institut Teknologi Nasional Malang.
4. Bapak Ir.Yusuf Ismail Nakhoda,MT selaku Dosen Pembimbing I yang telah memberikan saran dan bimbingannya dalam penyusunan laporan ini.
5. Ibu Nurlaily Vendyansyah,ST selaku Dosen Pembimbing II yang telah memberikan saran dan bimbingannya dalam penyusunan laporan ini.
6. Bapak dan Ibu Dosen yang telah mengajar penulis selama studi di Institut Teknologi Nasional Malang.
7. Rekan-rekan Teknik Informatika angkatan 2009 serta berbagai pihak yang turut membantu dalam penyelesaian laporan ini.
8. Untuk Orang Tua dan Keluarga yang telah mendukung saya sampai sekarang.

Semoga apa yang telah disajikan dapat memberikan manfaat dan pengetahuan bagi para pembaca. Segala kritik dan saran yang bersifat membangun, diterima dengan senang hati sebagai tambahan ilmu pengetahuan.

Malang, Agustus 2013

Penulis

DAFTAR ISI

	Halaman
Halaman Judul	i
Lembar Persetujuan dan Pengesahan	ii
Lembar Keaslian	iii
Abstrak	iv
Kata Pengantar	v
Daftar Isi	vi
Daftar Gambar	x
Daftar Tabel	xiv
Bab I Pendahuluan	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan	2
1.4 Batasan Masalah.....	2
1.5 Metodologi Penelitian.....	2
1.6 Sistematika Penulisan	3
Bab II Landasan Teori	4
2.1 <i>Finite State Machine</i>	4
2.2 <i>State Design Pattern</i>	5
2.3 <i>Steering Behaviors</i>	7
2.4 <i>Artificial Intelligence Agent</i>	8
2.5 <i>Multimedia</i>	9
2.5.1 Jenis Multimedia	10
2.5.2 Elemen Penunjang Multimedia.....	10
2.6 Diagram Alir	15
2.7 Perangkat Lunak Pendukung	17
2.7.1 Adobe Flash Profrrsional CS5	17

Bab III Analisa Dan Perancangan.....	19
3.1 Analisa Perancangan Sistem	19
3.1.1 Kebutuhan Fungsional	19
3.1.2 Kebutuhan Non Fungsional	20
3.2 Perancangan Game	21
3.2.1 Alur Permainan Game (<i>Flowchart</i>).....	22
3.2.2 Alur Proses pada <i>Game Pinalty</i> (<i>Flowchart</i>).....	25
3.2.3 Alur Proses pada <i>Game Shooting</i> (<i>Flowchart</i>).....	26
3.2.4 Alur <i>Game Passing</i> dan <i>Heading</i> (<i>Flowchart</i>).....	27
3.2.5 Perancangan Kecerdasan Buatan	28
3.2.6 Alur <i>Flowchart</i> Metode <i>FSM</i> Pada <i>Game Pinalty</i>	28
3.2.7 Alur <i>Flowchart</i> Metode <i>FSM</i> Pada <i>Game Shooting</i>	30
3.3 Properti Game	31
3.3.1 Perancangan <i>Intro Game</i>	31
3.3.2 Perancangan <i>Menu Game</i>	31
3.3.3 Perancangan <i>Arena Game</i>	32
3.3.4 Perancangan <i>Arena Game Pinalty</i>	33
3.4 Komponen Game	33
3.4.1 Perancangan Karakter Pemain.....	33
3.4.2 Perancangan Posisi Karakter <i>Game Shooting</i>	34
3.4.3 Perancangan Komponen <i>Game</i>	34
3.5 Perancangan Score	35
3.5.1 Perancangan Sistem Nilai	35
3.5.2 Perancangan Poin	36
3.5.3 Perancangan Tampilan Cetak Nilai	36
3.6 Perancangan Game Kuis	37
3.6.1 Perancangan <i>Outro / Exit</i>	38

Bab IV Implementasi Dan Pengujian	39
4.1 Implementasi	39
4.1.1 Implementasi Tampilan Intro.....	39
4.1.2 Implementasi Menu Utama.....	44
4.1.3 Implementasi Tampilan Petunjuk Permainan.....	49
4.1.4 Implementasi Komponen Game	49
4.1.5 Implementasi Tampilan Arena Game	50
4.1.6 Implementasi Game Pinalty	51
4.1.7 Implementasi Tampilan Score	58
4.1.8 Implementasi Game Shooting	58
4.1.9 Implementasi Game Heading	61
4.1.10 Implementasi Game Kuis	64
4.1.11 Implementasi Tampilan Exit Game	65
4.2 Pengujian.....	66
4.2.1 Pengujian Intro Loading	66
4.2.2 Pengujian Menu Game	66
4.2.3 Pengujian Petunjuk Permainan	67
4.2.4 Pengujian <i>Game Pinalty</i>	67
4.2.5 Pengujian <i>Game Shooting</i>	68
4.2.6 Pengujian <i>Game Heading</i>	69
4.2.7 Pengujian <i>Game Passing</i>	69
4.2.8 Pengujian Game Kuis	70
4.2.9 Pengujian Tampilan <i>Exit</i>	71
4.3 Pengujian Fungsional	71
4.3.1 Pengujian Operating Sistem OS	73
4.3.2 Pengujian Sistem Komputer / Spesifikasi Komputer	73
4.3.3 Pengujian User	74

4.3.4 Hasil Pengujian Kuisioner	76
Bab V Penutup	77
5.1 Kesimpulan	77
5.2 Saran.....	78
Daftar Pustaka.....	79
Lampiran	80

DAFTAR GAMBAR

Gambar 2.1	Contoh alur diagram <i>finite state machine</i> sederhana	4
Gambar 2.2	<i>Ilustrasi state</i> pada saat melakukan <i>double passing</i>	5
Gambar 2.3	Struktur <i>design pattern</i>	6
Gambar 2.4	<i>Design artificial intelligence agent</i>	9
Gambar 2.5	Perbedaan gambar vektor dengan <i>bitmap</i>	13
Gambar 2.6	Tampilan <i>Adobe Flash Professional CS.5</i>	17
Gambar 3.1	Bagan alur <i>game</i>	21
Gambar 3.2	<i>Flowchart</i> perancangan alur <i>game</i>	22
Gambar 3.3	<i>Flowchart</i> proses <i>game penalty</i>	25
Gambar 3.4	<i>Flowchart</i> alur <i>game shooting</i>	26
Gambar 3.5	<i>Flowchart</i> alur <i>game passing</i> dan <i>game heading</i>	27
Gambar 3.6	Perancangan gerak kiper	28
Gambar 3.7	<i>Flowchart</i> FSM pada <i>game penalty</i>	29
Gambar 3.8	<i>Flowchart</i> FSM pada <i>game shooting</i>	30
Gambar 3.9	Perancangan tampilan <i>intro game</i>	31
Gambar 3.10	Perancangan tampilan <i>menu game</i>	31
Gambar 3.11	Perancangan <i>button game</i>	32
Gambar 3.12	Perancangan arena <i>game</i>	32
Gambar 3.13	Perancangan arena untuk <i>game pinalty</i>	33
Gambar 3.14	Perancangan karakter <i>game</i>	33
Gambar 3.15	Rancangan posisi karakter <i>game shooting</i>	34
Gambar 3.16	Perancangan komponen <i>game</i>	34
Gambar 3.17	<i>Flowchart</i> alur proses nilai.....	35
Gambar 3.18	Perancangan tampilan hasil poin.....	36
Gambar 3.19	Perancangan tampilan cetak nilai.	36
Gambar 3.20	Perancangan tampilan <i>game kuis</i>	37
Gambar 3.21	Perancangan tampilan total nilai.	37
Gambar 3.22	Perancangan <i>outro exit game</i>	38
Gambar 4.1	Tampilan <i>intro loading game</i>	39
Gambar 4.2	Konfigurasi ukuran lembar kerja.....	40

Gambar 4.3	Tampilan <i>background intro game</i>	40
Gambar 4.4	Step membuat gerak bola	41
Gambar 4.5	Urutan <i>proses</i> perputaran <i>loading</i>	42
Gambar 4.6	<i>Step</i> membuat objek pemain transparan.....	43
Gambar 4.7	Gerak pemain transparan	43
Gambar. 4.8	Tampilan menu utama	44
Gambar 4.9	Layer untuk pembuatan menu utama.....	45
Gambar 4.10	Membuat <i>button menu game</i>	46
Gambar 4.11	Membuat <i>movie clip</i> pada <i>button</i>	46
Gambar 4.12	Membuat efek <i>motion guide</i>	47
Gambar. 4.13	Tampilan keseluruhan <i>button</i> menu.	48
Gambar. 4.14	Petunjuk permainan pada tiap-tiap <i>game</i>	49
Gambar. 4.15	Tampilan komponen <i>game</i>	50
Gambar. 4.16	Tampilan arena 1.	51
Gambar. 4.17	Tampilan arena 2.	51
Gambar 4.18	Membuat <i>movie clip supporter</i> dan animasi.....	52
Gambar. 4.19	Membuat area pilihan pada area gawang.....	53
Gambar 4.20	Objek transparan.....	54
Gambar 4.21	Nama <i>instance name</i> pada setiap objek target.....	54
Gambar 4.22	Area kotak yang sudah diatur posisinya.....	55
Gambar 4.23	Pemain kiper dalam posisi gawang.....	55
Gambar 4.24	Gerak posisi kipper	56
Gambar 4.25	<i>Layer</i> gerak kiper dan <i>label</i> nama <i>layer</i>	56
Gambar 4.26	Kursor target.....	57
Gambar 4.27	<i>Layer sound, layer script</i>	57

Gambar 4.28	<i>Tools</i> untuk memasukkan <i>script</i>	57
Gambar. 4.29	Tampilan nilai <i>score</i>	58
Gambar 4.30	Objek pemain pada <i>game shooting</i>	59
Gambar 4.31	Gerak pemain dalam <i>game shooting</i>	59
Gambar 4.32	Gerak bola dalam <i>game shooting</i>	60
Gambar 4.33	Tiang target dan objek bola.....	60
Gambar 4.34	Pergerakkan target bola	61
Gambar 4.35	Membuat <i>movie clip</i> pemain.	61
Gambar 4.36	Gerak lari pemain.....	62
Gambar 4.37	Gerak pemain pada saat akan <i>heading</i>	62
Gambar 4.38	Membuat <i>movie clip</i> pemain lawan.....	63
Gambar 4.39	Membuat objek area menjadi transparan.....	63
Gambar 4.40	Tampilan <i>game kuis</i>	64
Gambar 4.41	Beberapa <i>layer</i> yang ada pada <i>game kuis</i>	64
Gambar 4.42	Tampilan total nilai.....	65
Gambar 4.43	<i>Script link file</i> soal untuk <i>game kuis</i>	65
Gambar 4.44	Tampilan <i>exit game</i>	65
Gambar. 4.45	Pengujian <i>intro loading game</i>	66
Gambar 4.46	Tampilan menu <i>game</i>	67
Gambar 4.47	Tampilan Petunjuk permainan.....	67
Gambar 4.48	Tampilan pengujian <i>game pinalty</i>	68
Gambar 4.49	Tampilan pengujian <i>game shooting</i>	68
Gambar 4.50	Tampilan pengujian <i>game heading</i>	69
Gambar 4.51	Tampilan Pengujian <i>game passing</i>	69
Gambar 4.52	Tampilan pengujian <i>intro game kuis</i>	70
Gambar 4.53	Tampilan pengujian <i>game kuis</i>	70

Gambar 4.54	Tampilan pengujian total nilai.....	70
Gambar 4.55	Tampilan pengujian <i>exit game</i>	71

DAFTAR TABEL

Tabel 2.1	Diagram Alir	16
Tabel 2.2	Operasi Dasar Adobe Flash Professional Cs.5	18
Tabel 4.1	Action Script Untuk Tombol-tombol Pada Menu	47
Tabel 4.2	Pengujian Fungsional Game	71
Tabel 4.3	Pengujian <i>Sistem OS</i>	73
Tabel 4.4	Pengujian Sistem Komputer / Spesifikasi Komputer	73
Tabel 4.5	Lembar Pengujian Kuisisioner Untuk User	75
Tabel 4.6	Hasil Pengujian Kuisisioner	76
Tabel 4.7	Hasil Kuisisioner Saran / Masukkan	76

BAB I

PENDAHULUAN

1.1 Latar Belakang

Seiring dengan perkembangan teknologi. *Game* adalah salah satu media hiburan yang telah berkembang pesat di masyarakat. Perkembangan *game* dengan menampilkan grafis, dan sistem yang lebih baik menjadi hal utama untuk menarik minat para pecintanya.

Game latihan sepak bola adalah judul yang saya ambil. alasan yang paling utama dalam pengambilan judul ini karena olahraga sepak bola merupakan salah satu olah raga yang paling banyak di minanti oleh kalangan pemuda khususnya anak laki-laki. Oleh karena itu *game* yang saya rancang adalah *game* latihan sepak bola. Salah satu unsur yang berperan penting dalam merancang sebuah *game* yaitu kecerdasan buatan. *Artificial intelligence* (kecerdasan buatan) adalah bidang ilmu komputer yang meneliti dan mempelajari bagaimana agar komputer dapat bertindak dan mengerjakan pekerjaan seperti yang dilakukan oleh manusia. Dengan adanya kecerdasan buatan, elemen-elemen yang ada dalam *game* dapat berperilaku sealami mungkin layaknya manusia. Implementasi kecerdasan buatan dalam *game*, terletak pada salah satu elemennya yaitu NPC (*Non playable character*). Yang mana, NPC mampu mengenali aksi dari pemain sekaligus merespon dengan reaksi sesuai dengan acuannya.

Game ini merupakan *game* yang menggunakan kecerdasan buatan untuk proses permainan yang akan dimainkan, dimana beberapa karakter pemain yang terdapat dalam *game* ini dapat bergerak secara otomatis seperti halnya pemain bola pada dunia nyata, Seperti pada saat penjaga gawang yang bertugas menghalau bola supaya tidak masuk ke gawang. Keseluruhan dari *game* ini di harapkan mampu memberikan suatu permainan yang menyenangkan dan tidak membosankan sehingga pengguna mampu menyelesaikan permainan yang ada.

1.2 Rumusan Masalah

Bagaimana mengimplementasikan *Finite State Machine* (FSM) sebagai kecerdasan buatan untuk menentukan kondisi kapan suatu *state* (keadaan) harus berubah menentukan perilaku yang terjadi pada objek yang ditentukan yang terdapat pada *game* rancang bangun pembuatan *game* latihan sepak bola.

1.3 Tujuan

Adapun tujuan dari penelitian ini adalah mengimplementasikan *Finite State Machine* (FSM) sebagai reaksi terhadap pemain yang terdapat dalam *game* latihan sepak bola.

1.4 Batasan Masalah

Adapun batasan masalah dalam membangun *game* latihan sepak bola ini, antara lain:

1. *Game* yang akan dibangun terdiri dari 5 kategori *game*. Pertama *game pinalty*, kedua *game shooting*, ketiga *game heading*, keempat *game passing* dan kelima *game kuis*.
2. Algoritma yang diterapkan dalam pencarian solusi adalah *Finite State Machine* (FSM) untuk menentukan reaksi pemain.
3. *Artificial intelligence* dalam *game* ini terdapat pada tiga kategori *game* saja yakni *game passing*, *game heading*, dan *game pinalty*.
4. Teknik gerak dalam karakter pemain menggunakan teknik *frame by frame*.

1.5 Metodologi Pemecahan Masalah

Metode yang digunakan dalam pembahasan skripsi ini adalah :

- a) Studi Literatur.

Mencari referensi-referensi yang berhubungan dengan perencanaan dan pembuatan program yang akan dibuat.

b) Perancangan Program.

Sebelum melaksanakan pembuatan program, dilakukan perancangan terhadap program yang meliputi merancang keseluruhan program.

c) Pembuatan Program.

Pada tahap ini realisasi program yang dibuat, dilakukan perakitan sistem terhadap seluruh hasil rancangan yang telah dibuat.

d) Pengujian Program.

Untuk mengetahui cara kerja program, maka dilakukan pengujian secara keseluruhan.

1.6 Sistematika Penulisan

Dalam penyusunan skripsi ini agar lebih mudah dipahami maka dibuatlah suatu sistematika penulisan sebagai berikut:

BAB I : PENDAHULUAN

Berisi latar belakang, perumusan masalah, batasan masalah, tujuan dan metodologi penelitian.

BAB II : LANDASAN TEORI

Berisi teori-teori yang menunjang dalam proses pembuatan Skripsi.

BAB III : PERANCANGAN APLIKASI

Berisi tentang desain dan perancangan sistem aplikasi.

BAB IV : IMPLEMENTASI DAN PENGUJIAN

Berisi tentang implementasi dan uji coba dari aplikasi.

BAB V : PENUTUP

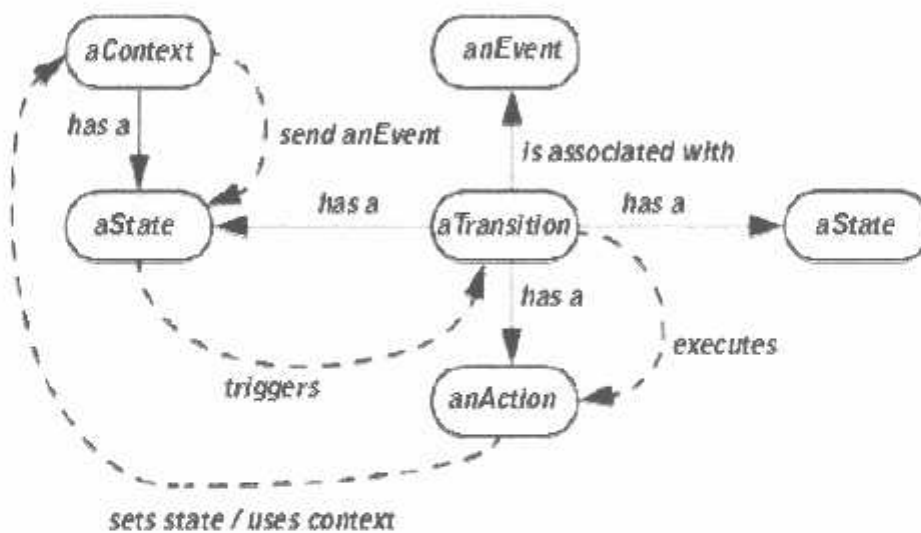
Berisi tentang kesimpulan dan saran-saran yang digunakan untuk pengembangan program selanjutnya.

BAB II

LANDASAN TEORI

2.1 Finite State Machine

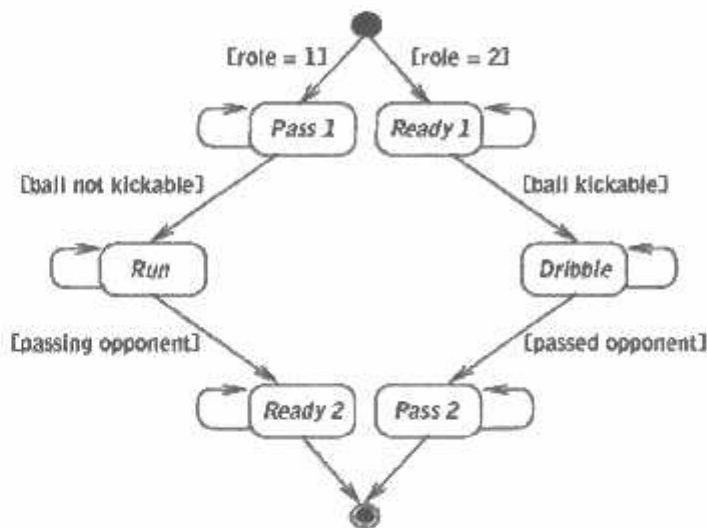
FSM merupakan sebuah mesin abstrak yang berfungsi untuk mendefinisikan sekumpulan kondisi yang menentukan kapan suatu *state* (keadaan) harus berubah. Setiap *state* yang sedang dijalankan tersebut menentukan perilaku yang terjadi pada objek yang bersangkutan [2].



Gambar 2.1 Contoh alur diagram *finite state machine* sederhana.

Seperti yang terlihat pada gambar.2.1. Diagram tersebut memperlihatkan FSM menggunakan *state design pattern*. *State* saat ini merupakan komponen *state* yang memicu terjadinya suatu transisi antar *state*. Transisi yang terjadi sesuai dengan *event* atau peraturan yang berkaitan dengan *state* tersebut sehingga dapat menentukan aksi apa yang akan dieksekusi selanjutnya. Jika aksi tersebut akan merubah *state* saat ini, maka akan digunakan *context* untuk merubah *state* tersebut. FSM ini di implementasikan pada olahraga futsal dengan mendefinisikan pemain-pemain sebagai kecerdasan buatan. Sebuah objek yang perilaku tiap objek

tersebut akan berbeda tergantung kondisi yang mereka alami, kondisi ini yang akan di implementasikan sebagai sebuah *state*. FSM ini berfungsi dalam mengatur perubahan *state* dengan memisahkan perilaku yang akan dilakukan pada setiap *state* objek yang bersangkutan perubahan *state* tersebut akan ditentukan oleh aksi dari objek itu sendiri mengharuskan adanya perubahan ^[2].



Gambar 2.2 Ilustrasi *state* pada saat melakukan *double passing*.

2.2 State Design Pattern

State design pattern adalah *design pattern* yang digunakan jika suatu objek dapat mengubah perilakunya terhadap suatu kondisi pada saat terjadi perubahan *state* di dalamnya. Objek tersebut mengganti kelas yang ada di dalamnya menjadi kelas lain.

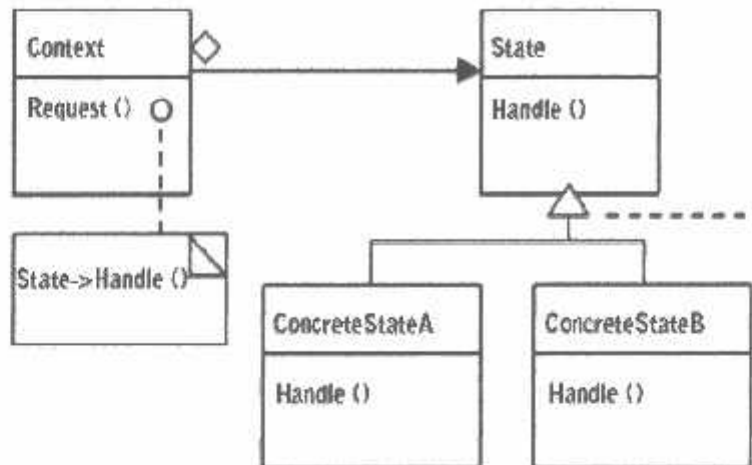
A. Penggunaan.

State design pattern ini digunakan jika menemui kasus berikut:

1. Perilaku dari tiap objek bergantung pada *state* yang dimilikinya dan objek tersebut harus merubah perilakunya pada saat sistem sedang berjalan ^[2].
2. Operasi yang akan dijalankan sangat besar, yang terdiri dari banyak *conditional statement* sesuai dengan *state* yang dimilikinya. *State* ini biasa direpresentasikan sebagai satu atau beberapa *enumerasi* konstan. Dengan *state design pattern* setiap kondisi akan dibedakan menjadi beberapa kelas sehingga memudahkan penggunaan *state* suatu objek sebagai suatu objek tersendiri agar dapat diubah sesuai kebutuhan ^[2].

B. Struktur.

Struktur dari *design pattern*



Gambar 2.3 Struktur *design pattern*.

Pada struktur *state machine* ini terdapat beberapa aspek yang dapat dilihat, yaitu:

1. Context.

Merupakan suatu penghubung yang digunakan untuk interaksi antara *automata* dengan pengguna, aspek ini mendefinisikan suatu *interface* yang berguna sebagai transisi logika dari kelas-kelas yang akan mengaksesnya. Dengan demikian *context* ini dapat mengatur instansiasi dari subkelas *concrete state* yang mendefinisikan *state* yang sedang berjalan saat ini ^[2].

2. State.

Aspek ini mendefinisikan suatu *interface* untuk mengenkapsulasi perilaku dari objek yang berasosiasi dengan *context* saat berada pada suatu *state* tertentu. Aspek ini menyimpan instansiasi dari kelas-kelas yang ada serta kondisi *state* yang sedang berjalan ^[2].

3. Concrete State Subclasses.

Menentukan perilaku dari objek saat berada pada suatu *state* tertentu. Semua subkelas ini akan mengimplementasi perilaku dari *state* yang dimiliki oleh *context*. ^[2]

C. Konsekuensi.

Akibat yang ditimbulkan dari penggunaan *state design pattern* adalah sebagai berikut:

1. Membuat cakupan perilaku pada suatu *state* menjadi lokal dengan mengenkapsulasinya sebagai objek dan membagi perilaku tersebut sesuai dengan *state* tertentu [2].
2. Perubahan *state* menjadi lebih jelas karena setiap *state* direpresentasikan sebagai sebuah objek yang terpisah [2].
3. Objek *state* dapat diakses bersama-sama oleh objek-objek yang membutuhkan [2].

D. Implementasi

Implementasi dilakukan dengan mendefinisikan setiap aspek yang ada pada struktur *state design pattern* agar sesuai dengan masalah yang ingin diselesaikan. Selain itu juga dilakukan perancangan aliran perubahan kondisi yang mungkin terjadi pada sistem agar dapat membagi perilaku sesuai dengan kondisi tersebut. Dalam penelitian ini objek-objek utama disesuaikan dengan objek yang terdapat pada olahraga futsal. *Context* yang berfungsi sebagai transisi logika dapat direpresentasikan sebagai *state machine* dan *state* sebagai *interface* dari setiap kondisi yang ada. Sedangkan untuk *concretestates* ditentukan dari perilaku tiap objek yang dikelompokkan sesuai dengan kondisinya. Setiap subkelas *concretestate* yang ada pada penelitian ini akan mengimplementasi *singleton design pattern*. Hal ini diperlukan agar saat menginstansiasi kelas dari suatu *state* hanya dilakukan satu kali. Sehingga walaupun kelas-kelas *state* yang didefinisikan banyak, tetapi instansiasi dari setiap kelas tidak dilakukan berkali-kali [2].

2.3 Steering Behaviors

Kemampuan suatu agen kecerdasan buatan agar dapat bergerak menuju posisi tujuan dari area dimana dia berada dengan melakukan improvisasi tertentu.

Berikut ini akan dijelaskan beberapa jenis *steering behaviors*:

1. *Seek*.

Steering behavior ini menggerakkan objek langsung menuju posisi target yang ditentukan [2].

2. *Flee*.

Merupakan kebalikan dari *seek*, *steering behavior* ini menggerakkan objek langsung menjauhi posisi target yang ditentukan ^[2].

3. *Arrive*.

Steering behavior ini menggerakkan objek menuju posisi target yang ditentukan sambil sesekali berhenti atau melakukan perlambatan sebelum sampai ke tujuan ^[2].

4. *Pursuit*.

Berguna pada saat objek harus mencegat objek lain yang sedang bergerak. *steering behavior* ini menggerakkan objek menuju target dengan memperhitungkan posisi target selanjutnya ^[2].

5. *Evade*.

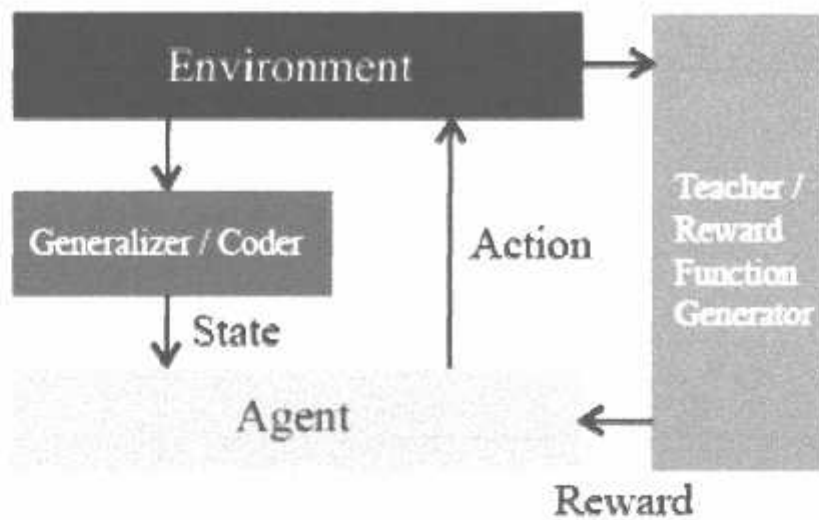
Berprinsip sama dengan *pursuit*, namun *steering behavior* ini menggerakkan objek menjauhi posisi target selanjutnya ^[2].

6. *Wander*.

Steering behavior ini menggerakkan objek secara acak pada lingkungannya sehingga terlihat seperti bergerak tanpa tujuan tertentu. Kombinasi dari *steering behaviors* dapat digunakan untuk mencapai tujuan sesuai dengan yang diharapkan ^[2].

2.4 Artificial Intelligence Agent

Sebuah agen dalam kecerdasan buatan adalah suatu objek yang dapat memahami lingkungannya melalui sensor dan bereaksi dengan menggunakan *effector* yang dimilikinya. Salah satu model pada agen kecerdasan buatan yaitu *reinforcement learning*. Pada model ini setiap agen dapat memutuskan aksi yang akan dilakukan dengan mempelajari aksi-aksi mereka sebelumnya melalui *reward* yang diberikan. Model suatu agen ini dapat dibagi menjadi tiga bagian, yaitu lingkungan, agen itu sendiri, dan *teacher* ^[2].



Gambar 2.4 *Design artificial intelligence agent.*

1. Lingkungan (*environment*)

Merupakan bagian yang tidak diketahui atau tak terdefinisi secara pasti yang berinteraksi dengan agen dimana mempengaruhi reaksi dari suatu agen saat menghadapi permasalahan ^[2].

2. Agen

Objek yang memiliki sensor dalam kaitannya dengan perubahan lingkungan yang terjadi, agar dapat menyelesaikan permasalahan dengan hasil yang optimal ^[2].

3. Teacher

Tugas dari bagian ini adalah memberikan respon terhadap reaksi agen terhadap perubahan lingkungan, agar agen tersebut dapat mengetahui dan mempelajari aksi yang sebelumnya telah dilakukan ^[2].

2.5 Multimedia

Multimedia berasal dari kata multi dan media. Multi dapat berarti banyak, sedangkan media dapat berarti sebagai tempat, sarana atau alat yang digunakan untuk menyimpan suatu informasi. *Multimedia* adalah pemanfaatan perangkat lunak pada komputer untuk membuat dan mengkombinasikan teks, bunyi, *video*, gambar dan animasi dengan menggunakan beberapa *link* (penghubung) dan *tool*

(alat) yang memungkinkan bagi pengguna dapat melakukan navigasi, interaksi, berkreasi dan berkomunikasi ^[1]

2.5.1 Jenis Multimedia

Dalam pengelompokannya, *multimedia* terbagi menjadi beberapa macam jenis. Penggunaan jenis multimedia tersebut, antara lain :

1. *Multimedia Linier.*

Adalah suatu *multimedia* yang tidak dilengkapi oleh pengontrol apapun, *multimedia* ini berjalan secara sekuensial (berurutan), contohnya adalah televisi ^[1].

2. *Multimedia Interaktif.*

Sudah dilengkapi oleh pengontrol yang dapat dioperasikan oleh pengguna, sehingga pengguna dapat memilih apa yang dikehendaki untuk proses interaksi. Contohnya adalah permainan dan pembelajaran interaktif ^[1].

2.5.2 Elemen Penunjang Multimedia

Multimedia adalah suatu istilah generik bagi media yang terdiri dari berbagai macam media, baik untuk tujuan pembelajaran maupun bukan. Elemen penunjang pada multimedia meliputi teks, bunyi, *video*, gambar, animasi dan navigasi ^[1].

1. Teks.

Teks adalah media paling dasar yang bersifat sederhana dan efektif dalam proses penyampaian suatu informasi, media dengan menggunakan teks membutuhkan tempat penyimpanan data yang paling kecil dibandingkan dengan menggunakan media lainnya. Secara umum, teks terbagi menjadi beberapa macam, antara lain :

a. Teks cetak.

Merupakan media yang tercetak di atas kertas.

b. Teks hasil *scan*.

Merupakan media yang tercetak di atas kertas dan dikonversikan dengan menggunakan komputer dalam bentuk gambar *digital* melalui mesin *scanner*.

c. *Teks digital.*

Media yang dibuat dengan menggunakan perangkat lunak pada komputer untuk membuat dan melakukan pengetikan teks tersebut^[1].

d. *Hyper text.*

Merupakan media penghubung pada sebuah teks yang pada umumnya dikombinasikan dengan tombol untuk dapat terhubung dengan informasi lainnya^[1].

2. *Bunyi.*

Bunyi adalah segala sesuatu yang dapat didengar oleh telinga normal pada manusia dalam rentang *frekuensi* antara 20 sampai 20.000 *hertz*. Bunyi dapat berupa kata-kata atau ucapan, efek suara, musik dan lain sebagainya^[1].

3. *Video.*

Bentuk rekaman gambar hidup atau gambar bergerak yang saling berurutan. Dalam fungsinya, terdapat dua macam jenis *video*, yaitu *video analog* dan *video digital*. *Video analog* dibentuk oleh deretan sinyal elektrik (gelombang *analog*) yang direkam dengan menggunakan kamera dan dipancarluaskan melalui gelombang di udara. Sedangkan *video digital* dibentuk oleh sederetan sinyal *digital* yang berbentuk bilangan biner 0 dan 1.

Pada *video digital* terdapat tiga komponen utama yang terbentuk, yaitu *frame rate*, *frame size* dan *datatype*. Pada *frame rate* menggambarkan berapa kali bingkai gambar yang dapat dimunculkan pada setiap detiknya, sementara *frame size* merupakan ukuran fisik yang sebenarnya dari setiap bingkai gambar yang telah disusun dan *datatype* menentukan seberapa banyak perbedaan warna yang dapat dimunculkan pada saat bersamaan^[1].

4. *Gambar.*

Gambar merupakan elemen yang dapat membantu menyampaikan suatu informasi dengan lebih mudah, menarik perhatian dan mengurangi rasa kebosanan. Dalam multimedia, gambar dapat diolah menjadi suatu proses penyampaian informasi dengan cara memahaminya, gambar juga dapat menjadi suatu penghubung kepada informasi yang lainnya. Ketika pengguna melihat sebuah objek gambar, penggunaan mendapatkan informasi dalam jumlah yang lebih besar jika dibandingkan dengan hanya membacanya,

misalnya pada sebuah karya seni lukis, seni grafis dan karya fotografi. Secara umum terdapat beberapa jenis macam gambar, antara lain :

a. Gambar *vektor*.

Merupakan objek gambar yang terbentuk dari sekumpulan kombinasi *node* (titik) dan *line* (garis) yang diperoleh dari suatu rumusan matematika, aljabar linier maupun geometri. *Vektor* menggambarkan sebuah objek gambar yang disesuaikan dengan perhitungan geometri (posisi sumbu x dan y pada koordinat tertentu).

Sebagai contoh, sebuah objek gambar *vektor* yang dibentuk dengan proses perhitungan untuk membentuk suatu kurva, menentukan posisi dengan diisi kombinasi warna tertentu. Gambar tersebut secara *fleksibel* dapat diubah ukurannya, dipindah atau bahkan diubah warnanya tanpa merusak kualitas gambar itu sendiri. Faktor yang menyebabkan kualitas gambar *vektor* tetap terjaga adalah karena gambar *vektor* tidak tergantung pada resolusi.

Gambar *vektor* lebih cocok untuk produksi gambar-gambar yang memiliki tepi tajam. Gambar *vektor* ini dapat dicetak dengan resolusi yang berbeda-beda tanpa kehilangan detail gambar, selain itu gambar *vektor* mempunyai beberapa kelebihan, antara lain :

1. Gambar *vektor* banyak digunakan untuk membuat desain yang membutuhkan soliditas dari bentuk, warna dan fleksibilitas ukuran yang tergantung dari kreatifitas desainer untuk mampu merancang atau membuat suatu objek.
2. Dapat lebih mengeksplorasi kemampuan dalam membuat bentuk, memilih warna, mengatur *layout* (tata letak) dan lain sebagainya.
3. Memungkinkan membuat karakter, jati diri dari desainer yang tidak dapat ditemui pada hasil karya orang lain, dikarenakan sebuah desain *vektor* merupakan sebuah objek ilustrasi, imajinasi dan teknik dari desainer itu sendiri, sehingga kecil kemungkinannya untuk dapat menemukan ilustrasi, imajinasi dan teknik yang benar-benar sama.

b. Gambar *bitmap*

Gambar *bitmap* merupakan rekontruksi dari gambar asli yang tersimpan sebagai serangkaian *pixel* (titik-titik) yang memenuhi bidang *pixel* pada layar komputer. Kelebihannya dalam menggunakan gambar *bitmap* antara lain :

1. Proses pengolahan yang minimal.
2. Lebih cepat untuk ditampilkan, dikarenakan gambar *bitmap* dapat dikirimkan secara langsung dari data ke layar monitor. dikarenakan pada layar monitor menampilkan gambar dengan cara menggunakan jaringan pada *node* (titik), maka gambar *bitmap* dan gambar *vektor* akan diperlihatkan sebagai *pixel*. Begitu juga ketika pengguna hendak menampilkan gambar *vektor*, dimana desainer harus melakukan ekspor dari gambar *vektor* ke dalam *format* data **.jpg*, **.jpeg*, **.gif*, **.png* dan lainnya. Sifat gambar *vektor* yang telah diekspor tersebut, secara otomatis berubah menjadi tipe gambar *bitmap*, meskipun gambar tersebut dibuat dengan menggunakan perangkat lunak gambar *vektor*. Perbedaan gambar *vektor* dan gambar *bitmap* dapat dilihat pada gambar 2.5.

Gambar Vektor



Gambar Bitmap



Gambar 2.5 Perbedaan gambar *vektor* dengan *bitmap*.

5. Animasi

Animasi adalah suatu gerak *visual* tiruan terhadap perubahan bentuk pada suatu objek yang tercipta melalui imajinasi, khayalan, keadaan nyata atau lingkungan. Pada pengelompokannya, animasi terbagi menjadi dua jenis, yaitu animasi dua dimensi dan tiga dimensi, penggunaan jenis pada animasi

tergantung dari kebutuhan. Berdasarkan teknik pembuatan animasi dibedakan menjadi beberapa jenis, meliputi :

a. *Cell Animation.*

Merupakan teknik dasar dalam pembuatan animasi kartun, yang umumnya banyak digunakan pada awal saat pembuatan animasi kartun^[1].

b. *Frame Animation.*

Merupakan teknik animasi yang sederhana. Dimana animasi tersebut dibangun pada setiap *frame* yang saling berkaitan dengan *frame* lainnya^[1].

c. *Sprite Animation.*

Sprite animation merupakan sebuah animasi yang bergerak secara mandiri, sedangkan latar belakang pada animasi tersebut diam^[1].

d. *Path Animation.*

Path animation (animasi lintasan) adalah animasi yang bergerak mengikuti garis pada suatu lintasan yang dibuat.

e. *Spline Animation* ^[1].

Spline animation merupakan sebuah animasi bergerak yang mengikuti garis lintasan dengan membentuk suatu kurva, dimana kurva tersebut didapat dari hasil perhitungan secara sistematis yang dapat menghasilkan gerak animasi lebih halus^[1].

f. *Vector Animation.*

Sebuah animasi yang dikendalikan oleh vektor. Animasi vektor memberikan hasil yang bersih dan halus, dikarenakan gambar yang ditampilkan dan ukuran animasi menggunakan nilai matematika ^[1].

g. *Character Animation.*

Character animation (animasi karakter) umumnya digunakan dalam film kartun berbasis dua dimensi maupun tiga dimensi. Dimana proses pembuatan tiga dimensi dilakukan pada tiga sudut pandang, yaitu pada sudut x, y dan z, sedangkan pada dua dimensi pada sudut x dan y^[1].

h. *Morphing Animation.*

Adalah animasi yang mengubah satu bentuk menjadi bentuk yang lain dan memperlihatkan serangkaian *frame* yang menciptakan suatu gerak yang halus^[1].

i. *Digital Animation.*

Digital animation adalah teknik penggabungan gambar dimana pada setiap gambar terdapat gerakan yang berbeda^[1].

j. *Navigasi.*

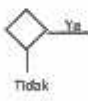
Navigasi merupakan sebuah kontrol untuk mengatur, memfungsikan suatu perintah, maupun informasi yang dapat berupa tombol teks atau simbol^[1].

2.6 Diagram Alir

Diagram alir adalah sekumpulan simbol yang digunakan untuk menggambarkan atau merepresentasikan kegiatan dari suatu sistem, aliran dokumen serta aliran logika yang ditunjukkan dengan arah panah. Penggambaran diagram alir harus menggunakan cara dan ketentuan yang berlaku secara lazim dalam pembuatan aplikasi, sehingga tidak akan menimbulkan kebebasan yang tidak mempunyai standar dalam menggambarkan dan merepresentasikan suatu sistem. Simbol diagram alir dapat dilihat pada tabel 2.2

Tabel 2.1 Diagram Alir

Simbol	Nama	Keterangan
	Dokumen	Digunakan untuk semua jenis dokumen.
	Dokumen rangkap	Menggambarkan dokumen asli beserta tembusannya.
	Berbagai dokumen	Menggambarkan berbagai jenis dokumen yang digabungkan bersama dalam satu paket.
	<i>Input</i> atau <i>output</i> data	Proses <i>input output</i> data, parameter dan informasi.
	Penghubung	Penghubung bagian-bagian diagram alir yang

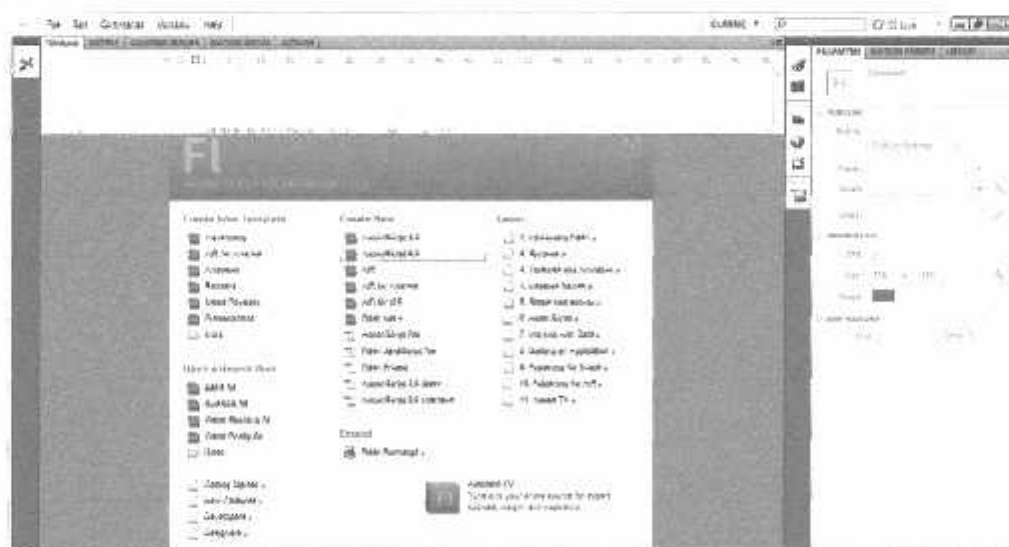
		berada pada satu halaman.
	Kegiatan manual	Untuk menggambarkan suatu kegiatan manual.
	Arsip sementara	Menunjukkan tempat penyimpanan dokumen sementara.
	Arsip permanen	Menunjukkan tempat penyimpanan dokumen secara permanen yang tidak akan diproses lagi.
	Proses	Proses perhitungan dan proses pengolahan data.
	Keying atautyping	Menggambarkan input data ke dalam komputer melalui on-line terminal.
	Pita magnetic	Menggambarkan arsip pada komputer yang berbentuk pita magnetik.
	On-line storage	Menggambarkan arsip pada komputer yang berbentuk on-line (di dalam memori komputer)
	Keputusan	Menggambarkan keputusan yang harus dibuat dalam proses pengolahan data atau operasi perbandingan logika.
	Garis alir	Menunjukkan arah aliran proses pengolahan data atau sistem.
	Persimpangan Garisalir	Jika dua garis alir bersimpangan, untuk menunjukkan arah masing-masing garis,

		salah satu garis dibuat melengkung.
	Pertemuan garis alir	Digunakan jika terdapat dua garis alir bertemu dan salah satu garis mengikuti garis lainnya.
	Mulai atau berakhir	Menggambarkan awal dan akhir suatu sistem.

2.7 Perangkat Lunak Pendukung Aplikasi

2.7.1 Adobe Flash Professional CS5

Adobe Flash Professional CS.5. Adalah salah satu perangkat lunak animasi vektor yang dibuat oleh *adobe systems inc.* Yang berfungsi untuk membuat animasi serta dapat mengkombinasikan elemen-elemen pada *multimedia*, sehingga dapat menghasilkan media animasi interaktif yang dapat didengar maupun dilihat. Pada *adobe flash professional* dilengkapi dengan fasilitas *script* yang digunakan untuk memberikan fungsi pada animasi dan pemrograman. Pemrograman yang terdapat pada *adobe flash professional cs5* adalah *Action script 2.0* dan *action script 3.0*. Bahasa pemrograman yang digunakan dalam game ini adalah *action script 2.0*. Berikut tampilan *adobe flash Professional cs5*.



Gambar 2.6 Tampilan *Adobe Flash Professional CS.5*.

Pada perangkat lunak *adobe flash professional cs.5* memiliki beberapa operasi dasar dalam penggunaan untuk membuat suatu animasi. Penggunaan operasi dasar pada perangkat lunak *adobe flash professional cs.5* dapat dilihat pada tabel 2.2.

Tabel 2.2 Operasi dasar *adobe flash professional cs.5*

Alat	Fungsi
<i>Arrow Tool</i>	Untuk memilih suatu objek.
<i>SubSelect Tool</i>	Untuk mengolah objek garis.
<i>Free Transform Tool</i>	Untuk mengubah dimensi objek dengan fleksibel.
<i>Gradient Transform Tool</i>	Mengatur gradiasi objek.
<i>Line Tool</i>	Untuk membuat objek garis.
<i>Text Tool</i>	Untuk membuat objek teks.
<i>Oval Tool</i>	Untuk membuat objek lingkaran.
<i>Rectangle Tool</i>	Untuk membuat objek persegi.
<i>Pencil Tool</i>	Untuk membuat objek bebas dengan garis.
<i>Ink Bottle</i>	Untuk memberi warna pada garis luar objek.
<i>Paint Bucket Tool</i>	Untuk memberi warna di dalam objek.

Action script yang terdapat pada *adobe flash professional cs.5* dapat digunakan untuk membuat animasi maupun aplikasi yang interaktif, sehingga pengguna dapat berinteraksi dengan menggunakan kontrol navigasi, memasukkan informasi, menggerakkan objek dengan *keyhoard*, membuat menu dan lain sebagainya. Saat ini terdapat beberapa versi *action script* dalam *adobe flash professional cs5*, antara lain *action script* versi 1.0, versi 2.0 dan versi 3.0.

Action script 2.0 lebih banyak digunakan daripada versi 3.0 karena factor kemudahan dalam pemahaman. Pada penulisan *variabel* maupun *statement* yang terdapat pada *Action script*, juga harus memperhatikan perbedaan antara huruf besar dan kecil, spasi, tanda baca dan penulisan lainnya untuk dapat dibaca dan diproses oleh perangkat lunak *adobe flash professional cs.5*.

BAB III

ANALISA DAN PERANCANGAN

3.1 Analisa Perancangan Sistem

Analisa merupakan tahapan awal didalam membangun sebuah *game*. Analisa adalah tahapan sebelum kepada perancangan. Dimana pada tahap ini, penulis menguraikan pokok permasalahan dan mengumpulkan informasi yang dibutuhkan untuk ke tahap perancangan.

3.1.1 Kebutuhan Fungsional

Kebutuhan fungsional yang digunakan dalam membangun *game* ini meliputi aksi yang dimiliki oleh karakter pemain sepak bola baik dari karakter pemain pada saat melakukan *passing*, *heading*, *shotting*, dan *pinalty*. Kebutuhan fungsional dalam *game* latihan sepak bola, antara lain :

1. Untuk *game passing* dan *heading* terdapat dua macam aksi gerak pemain dalam berjalan mengapai bola, pada *game passing* pemain berjalan secara *vertical*, dan untuk *game heading* pemain akan bergerak secara *horizontal*, dari masing-masing gerak di kontrol oleh tombol arah yang terdapat pada keyboard.
2. Dalam *game passing* dan *heading* pemain harus mengapai objek bola yang di operkan oleh pemain lawan dan melakukan aksi *passing* untuk *game passing* dan melakukan aksi *heading* untuk *game heading*. Jika aksi dapat dilakukan dengan baik maka pemain mendapat nilai.
3. Pada *game shooting* pemain harus menendang bola ke titik sasaran, untuk melakukan tendangan ke titik sasaran tersebut terdapat indikator keakuratan menendang jika pemain melakukan tendangan sesuai dengan indikator keakuratan maka titik objek akan berhasil di tendang oleh pemain.
4. Pada *game pinalty* pertama-tama pemain harus menentukan posisi sasaran pada area gawang, kemudian menentukan posisi titik indikator keakuratan

sasaran sesuai dengan indikator yang ditetapkan. Selanjutnya tekan tombol aksi untuk melakukan tendangan ke gawang untuk dapat mencetak poin.

5. Pada saat pemain dalam *game pinalty* melakukan tendangan maka (kiper) penjaga gawang akan menghalau bola yang mengarah ke gawang agar pemain tidak dapat mencetak poin. Kiper akan bergerak secara otomatis menghalau bola tersebut sesuai arah bola.
6. *Game* kuis dimana pemain harus menjawab beberapa pertanyaan yang terdapat dalam *game* ini dengan benar untuk mendapatkan nilai sebanyak-banyaknya. Soal yang terdapat pada *game* ini muncul secara acak, jumlah soal yang akan di tampilkan secara acak sebanyak 20 soal.

3.1.2 Kebutuhan Non Fungsional

Kebutuhan non fungsional yang digunakan dalam membangun *game* ini meliputi kebutuhan non fungsional *performance* dan kebutuhan non fungsional *interface*. Kebutuhan non fungsional *performance* dalam *game* Latihan Sepak bola ini, antara lain :

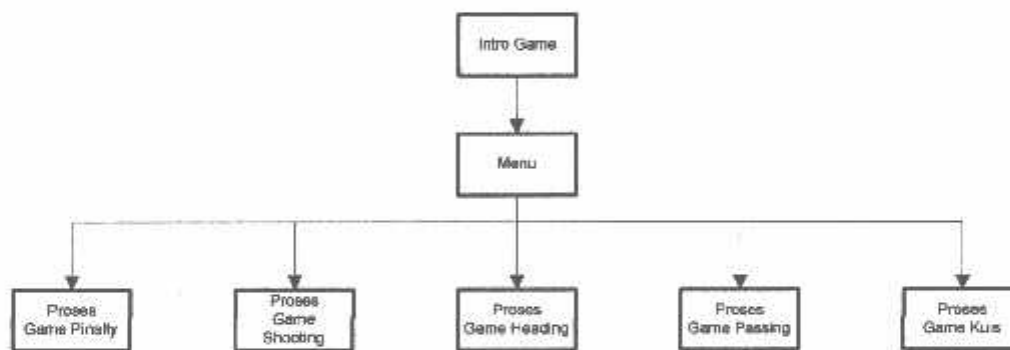
1. Dalam pembuatan *game* latihan sepak bola menggunakan 30.00 *fps* (satuan *frame per second*).
2. Dalam *game* ini terdapat 5 kategori *game*, yakni:
 - a. Kategori *game passing*.
 - b. Kategori *game heading*.
 - c. Kategori *game shotting*.
 - d. Kategori *game pinalty*.
 - e. Kategori *game kuis*.
3. *Game* ini di dukung oleh gambar *vector* pemain dari tiap-tiap pergerakan pemain pada setiap *state* *framenya*.
4. Ukuran *size* lebar tampilan *game* ini 980 x 615 *pixel*.
5. Terdapat *sound* pendukung *game* yang di sesuaikan dengan *game* ini, seperti suara *supporter*, *intro game* dan efek *button*.
6. Terdapat indikator keakuratan untuk *proses* menendang bola.
7. Pemain di nyatakan memenangkan *game* jika berhasil mengumpulkan 4 poin dari 5 kali kesempatan bermain pada masing-masing *game*.

8. Tersedia menu untuk mengulangi permainan yang sama atau untuk kembali memilih permainan yang lainnya.
9. *Game* dimainkan menggunakan *keyboard* arah sebagai kontrol gerak pemain. *mouse* sebagai kontrol untuk menentukan arah target sasaran dalam *game pinalty*, serta *space* sebagai kontrol aksi pemain.

3.2 Perancangan Game

Pada tahap perancangan aplikasi, pembuatan bagan alur dimulai dari pembuatan bagan *intro* serta bagan *menu* dengan menampilkan beberapa menu yang terdiri dari beberapa bagan dari kategori *game* yang ada, di antaranya bagan *game pinalty*, bagan *game shooting*, bagan *game heading*, bagan *game passing*, bagan *game kuis*.

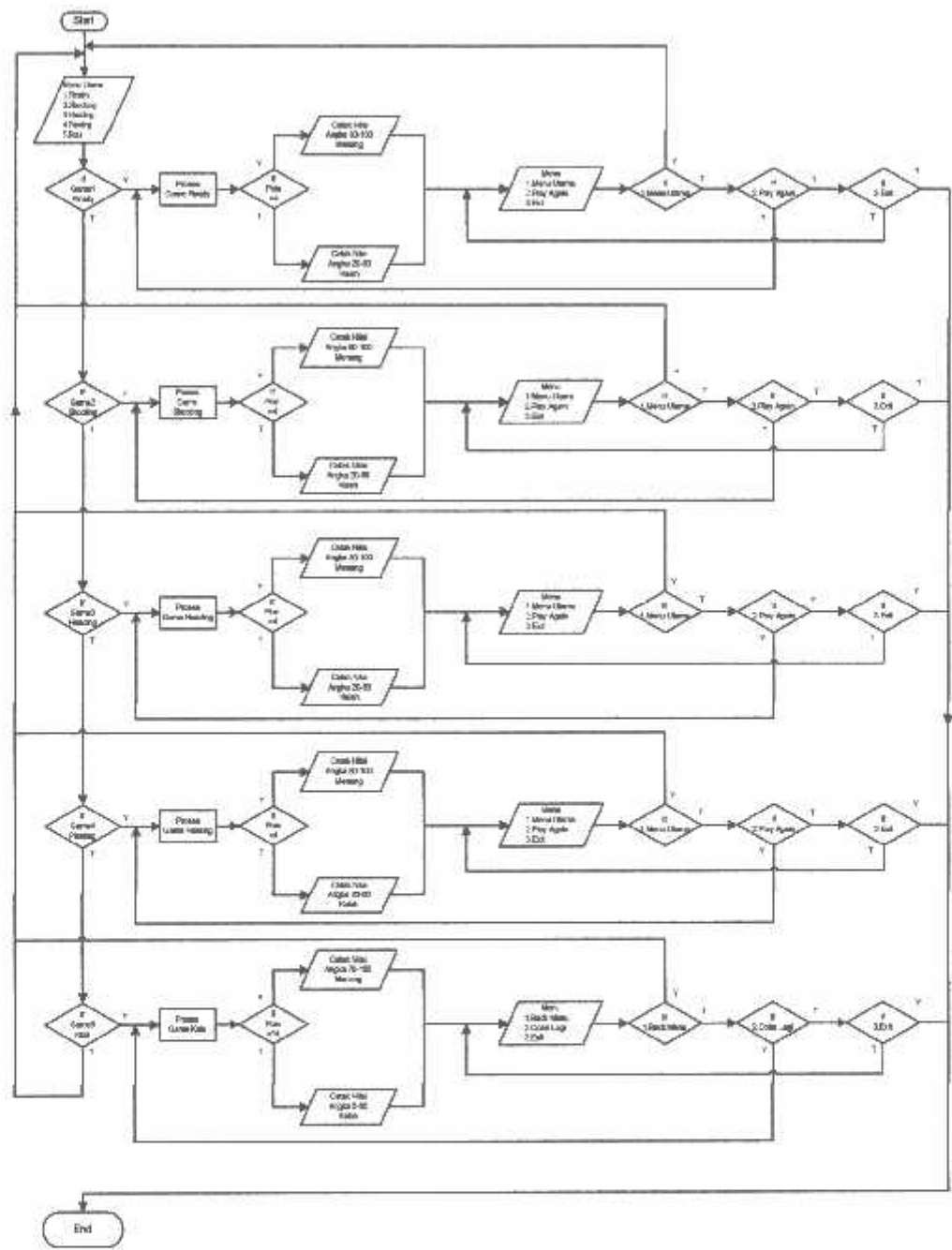
Tampilan dari bagan alur *game* dapat di lihat dpada gambar 3.1. Bagan alur *game*.



Gambar 3.1 Bagan alur *game*.

3.2.1 Alur Permainan Game Latihan Sepak bola.

Pertama kali yang dilakukan dalam membangun *game*, pada *game* latihan sepak bola adalah membuat alur *game* dimana alur sebagai *flowchart* untuk menentukan alur permainan. Seperti yang terlihat pada gambar 3.2 *flowchart* perancangan alur *game*.



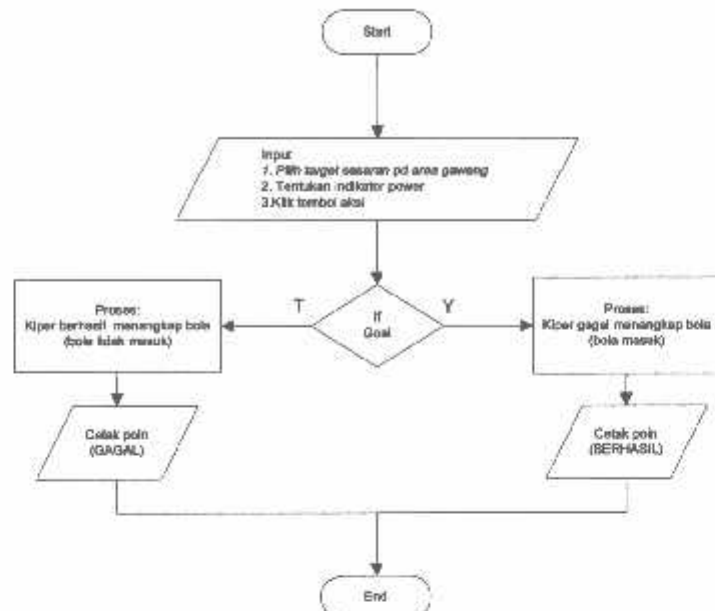
Gambar 3.2 Flowchart perancangan alur game.

Flowchart alur *game* latihan sepak bola dapat dijelaskan sebagai berikut:

1. Ketika *game* ini dijalankan akan muncul menu utama dengan lima buah *button*, yaitu *button game passing*, *button game heading*, *button game shotting*, *button game pinalty* dan *button kuis game*.
 2. Ketika *button game passing* diklik, maka kita akan masuk pada permainan *passing*, dalam permainan ini memiliki 5 kali kesempatan melakukan permainan *passing*. Jika dapat melakukan *passing* sebanyak 4 kali maka memenangkan *game passing*, dan memiliki nilai *score* 80. Jika gagal melakukan *passing* kurang dari 4 kali maka tidak dapat memenangkan *game* / kalah dan memiliki nilai *score* kurang dari 70.
 3. Selanjutnya setelah mengetahui hasil *score*, terdapat menu pilihan dengan tiga *button*, yakni *button back to menu awal*, *button play again*, dan *button exit*. Jika memilih *back to menu* maka akan kembali ke tampilan menu awal *game* yang terdapat lima *button* pilihan *game*. Apabila kita memilih pilihan kedua *play again* maka akan mengulang permainan *passing* dari awal. Jika ingin keluar dari *game* maka pilih pilihan ke tiga *button exit*.
 4. Ketika *button game heading* diklik, maka akan masuk pada permainan *heading*, dalam permainan ini memiliki 5 kali kesempatan melakukan permainan *heading*. jika dapat melakukan *heading* sebanyak 4 kali maka memenangkan *game heading*, dan memiliki nilai *score* 80. Jika tidak dapat memenangkan *game heading*, kurang dari 4 kali maka tidak dapat memenangkan *game* / kalah, dan memiliki nilai *score* kurang dari 70.
 5. Selanjutnya setelah mengetahui hasil *score*, terdapat menu pilihan dengan tiga *button*, yakni *button back to menu awal*, *button play again*, dan *button exit*. Jika memilih *back to menu* maka akan kembali ke tampilan menu awal *game* yang terdapat lima *button* pilihan *game*. Apabila memilih pilihan kedua *play again* maka akan mengulang permainan *heading* dari awal. Jika ingin keluar dari *game* maka pilih pilihan ke tiga *button exit*.
 6. Ketika *button game shooting* diklik, maka akan masuk pada permainan *shooting*, dalam permainan ini memiliki 5 kali kesempatan melakukan permainan *shooting*. Jika dapat melakukan *shooting* sebanyak 4 kali maka memenangkan *game shooting*, dan memiliki nilai *score* 80. Jika tidak dapat melakukan *shooting* kurang dari 4 kali maka tidak dapat memenangkan *game* /kalah,dan memiliki nilai *score* kurang dari 70.
-

7. Selanjutnya setelah mengetahui hasil *score*, terdapat menu pilihan dengan tiga *button*, yakni *button back to menu awal*, *button play again*, dan *button exit*. Jika memilih *back to menu* maka akan kembali ke tampilan menu awal *game* yang terdapat lima *button* pilihan *game*. Apabila memilih pilihan kedua *play again* maka akan mengulang permainan *shooting* dari awal. Jika ingin keluar dari *game* maka pilih pilihan ke tiga *button exit*.
 8. Ketika *button game pinalty* diklik, maka akan masuk pada permainan *pinalty*, dalam permainan ini memiliki 5 kali kesempatan melakukan tendangan *pinalty*. Jika bisa memasukkan bola ke gawang sebanyak 4 kali maka memenangkan *game pinalty*, dan memiliki nilai *score* 80. Jika tidak dapat melakukan tendangan *pinalty* kurang dari 4 kali maka tidak dapat memenangkan *gam* /kalah, dan memiliki nilai *score* kurang dari 70.
 9. Selanjutnya setelah mengetahui hasil *score*, tampil menu pilihan dengan tiga *button*, yakni *button back to menu awal*, *button play again*, dan *button exit*. Jika memilih *back to menu* maka akan kembali ke tampilan menu awal *game* yang terdapat lima *button* pilihan *game*. Apabila memilih pilihan kedua *play again* maka akan mengulang permainan *pinalty* dari awal. Jika ingin keluar dari *game* maka pilih pilihan ke tiga *button exit*.
 10. Ketika *button kuis game* diklik, maka akan masuk pada permainan pengetahuan tentang sepak bola, dalam permainan ini akan di berikan beberapa soal tentang sepak bola yang ditampilkan secara acak, jawab dengan benar semua soal yang ada. Setelah itu penilaian *score* di hitung dari jumlah soal yang benar dikalikan 5 poin dan nilai dihitung menjadi nilai akhir.
 11. Selanjutnya setelah mengetahui hasil *score*, tampil menu pilihan dengan tiga *button*, yakni *button back to menu awal*, *button play again*, dan *button exit*. Jika memilih *back to menu* maka akan kembali ke tampilan menu awal *game* yang terdapat lima *button* pilihan *game*. Apabila memilih pilihan kedua *play again* maka akan mengulang *game kuis* dari awal. Jika ingin keluar dari *game* maka pilih pilihan ke tiga *button exit*.
 12. Jika pada menu awal *game* tidak melakukan pilihan satupun, maka akan tetap pada tampilan menu awal *game*.
-

3.2.2 Alur Proses Pada Game Pinalty.

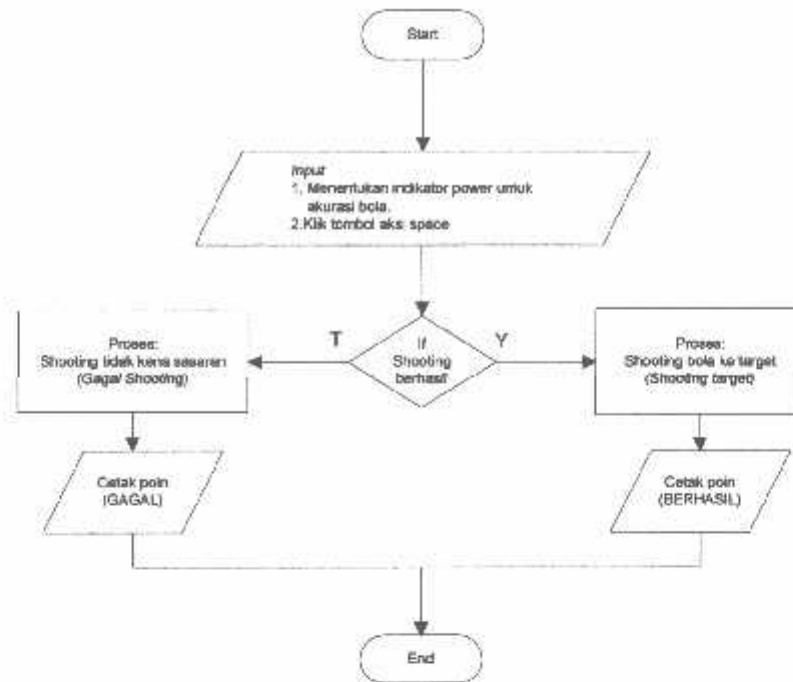


Gambar 3.3 Flowchart proses game pinalty.

Penjelasan untuk flowchart proses pada game pinalty, sebagai berikut:

1. Awal memulai game pinalty pertama menentukan target sasaran pada area gawang.
2. Kedua menentukan kekuatan tendangan dengan menyesuaikan ukuran yang terdapat pada indikator power.
3. Selanjutnya jika sudah yakin dengan posisi indikator power maka dilanjutkan dengan menekan tombol aksi button klik bola untuk melesatkan bola kearah yang sudah ditentukan.
4. Jika bola masuk pada saat melakukan tendangan pinalty maka dapat poin 1 maka akan cetak poin muncul tampilan berhasil.
5. Jika bola tidak masuk pada saat melakukan tendangan pinalty maka tidak dapat poin dan akan muncul tampilan gagal.
6. Proses ini dilakukan sebanyak 5 kali kesempatan dan nilai akan di total pada akhir permainan game pinalty.

3.2.3 Alur Proses Pada Game Shooting

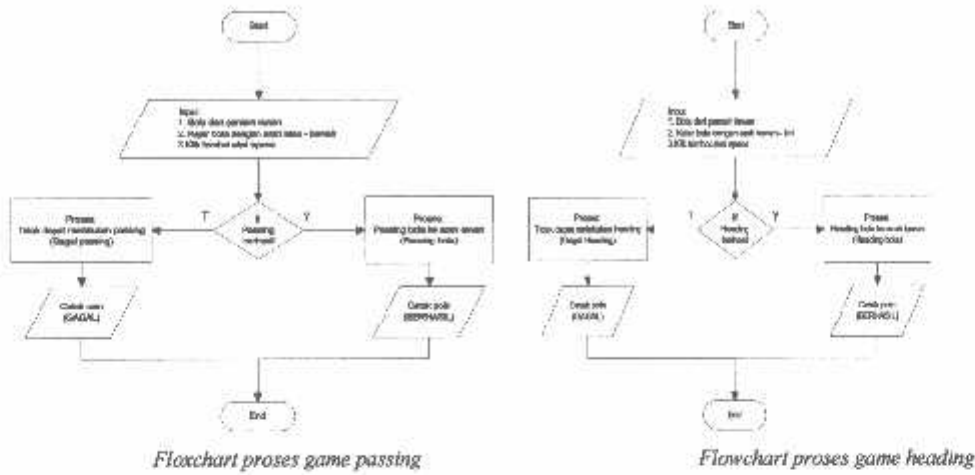


Gambar 3.4 Flowchart alur game shooting.

Penjelasan untuk *flowchart proses* pada *game shooting*, sebagai berikut:

1. Awal memulai *game shooting* pertama menentukan kekuatan tendangan dengan menyesuaikan ukuran yang terdapat pada indikator *power*.
2. Selanjutnya jika sudah yakin dengan posisi indikator *power* maka dilanjutkan dengan menekan tombol aksi *button* klik bola untuk melesatkan bola kearah target sasaran.
3. Jika bola mengenai target sasaran maka dapat poin 1 selanjutnya akan cetak poin muncul tampilan berhasil.
4. Jika bola tidak mengenai target sasaran maka tidak dapat poin dan akan muncul tampilan gagal.
5. Proses ini dilakukan sebanyak 5 kali kesempatan, dan nilai akan di total pada akhir permainan *game shooting*.

3.2.4 Alur Proses Game Passing dan Heading



Gambar 3.5 Flowchart alur game passing dan game heading.

Penjelasan untuk *flowchart proses* pada *game passing* dan *game heading* sebagai berikut:

1. Awal memulai *game passing* dan *game heading* pertama menggerakkan pemain ke target bola yang ditendang oleh pemain lawan (bergerak kearah bola).
2. Selanjutnya jika pemain sudah mendekati bola maka lanjutkan dengan menekan tombol aksi *button* pada *keyboard*, *button space* untuk melesatkan bola kembali kearah pemain lawan.
3. Jika pemain mengenai bola maka dapat poin 1 selanjutnya akan cetak poin dan muncul tampilan berhasil.
4. Jika pemain tidak mengenai bola maka tidak dapat poin dan akan muncul tampilan gagal.
5. Proses ini dilakukan sebanyak 5 kali kesempatan, dan nilai akan di total pada akhir permainan *game passing* dan *game heading*.

3.2.5 Perancangan Kecerdasan Buatan.

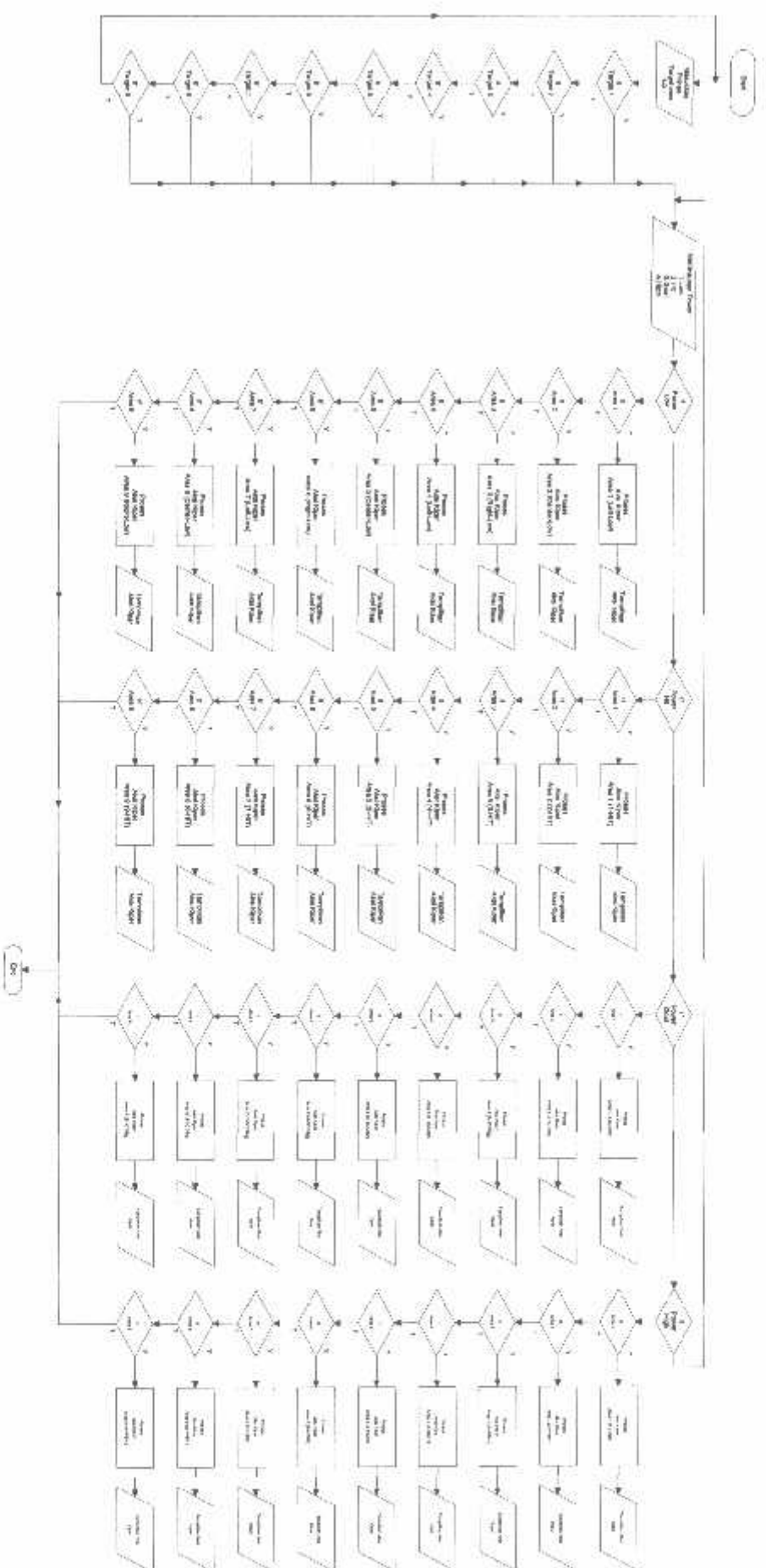
Finite state machine (FSM) adalah metode yang akan diterapkan ke dalam perilaku penjaga gawang (kipper), berfungsi untuk mendefinisikan sekumpulan kondisi yang menentukan kapan suatu *state* (keadaan) harus berubah. Setiap *state* yang sedang dijalankan tersebut menentukan perilaku yang terjadi pada objek yang bersangkutan. Sesuai dengan bentuk implementasi dari *finite state machine* yaitu *naïve approach*. *Naïve approach* sendiri menggunakan *conditional statement* (*if – else* atau *switch – case*).



Gambar 3.6 Perancangan gerak kipper.

3.2.6 Alur Flowchart Metode FSM Pada Game Pinalty.

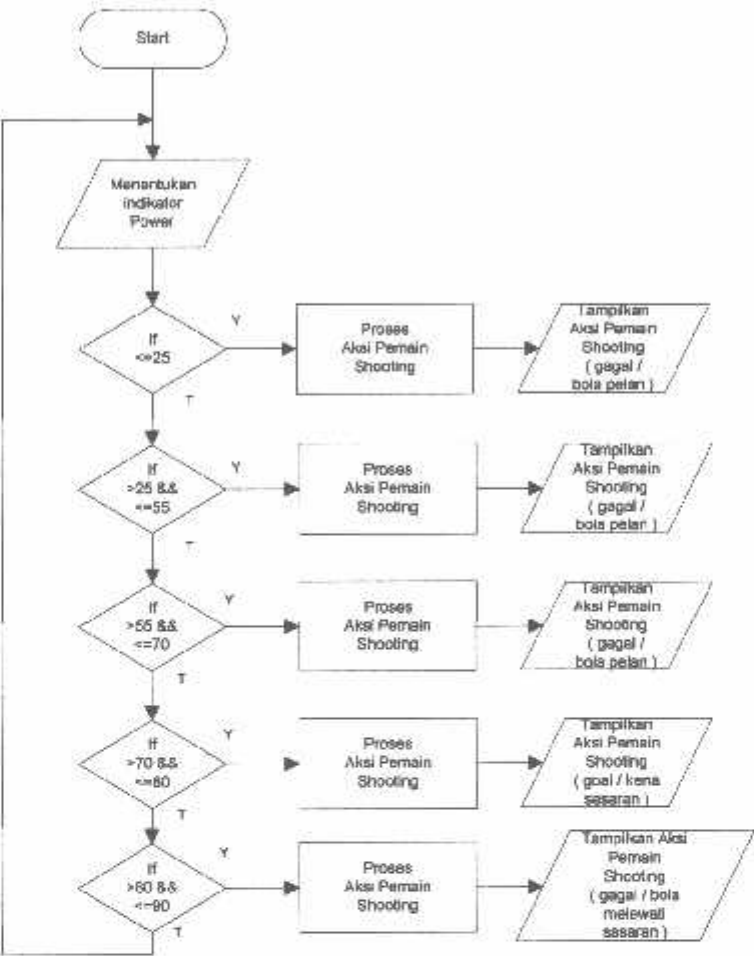
Pada alur *flowchart* ini menjelaskan bagaimana proses kondisi gerak kipper pada saat permainan pada *game penalty* di jalankan. Berikut tampilan *flowchart* metode *finite state machine* pada *game penalty* dapat dilihat pada hal 29.



Gambar 3.7 Flowchart finite state machine pada game penalty

3.2.7 Alur Flowchart Metode FSM Pada Game Shooting

Pada alur *flowchart* ini menjelaskan bagaimana proses kondisi gerak pada saat permainan pada *game shooting* di jalankan. Berikut tampilan *flowchart* metode *finite state machine* pada *game penalty* dapat dilihat pada hal 29.



Gambar 3.8 Flowchart finite state machine pada game shooting

3.3 Properti Game

3.3.1 Perancangan Intro Game

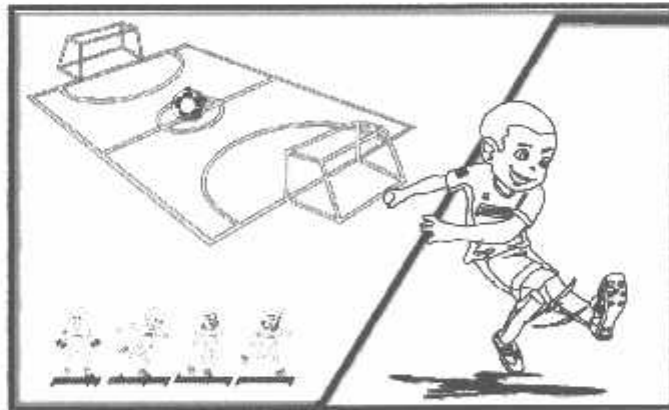
Intro game latihan sepak bola memiliki tampilan dengan animasi gerak *loading* disebelah kiri bawah memiliki durasi loading kira-kira 1-2 menit kemudian akan muncul tombol *play now*.



Gambar 3.9 Perancangan tampilan *intro game*.

3.3.2 Perancangan Menu Game

Perancangan *menu game* ini memiliki animasi gerak pada gambar lapangan dimana animasi ini bergerak membentuk sebuah lapangan dari bagian-bagian garis perframennya, serta terdapat tombol untuk tiap-tiap *game*.



Gambar 3.10 Perancangan tampilan *menu game*.

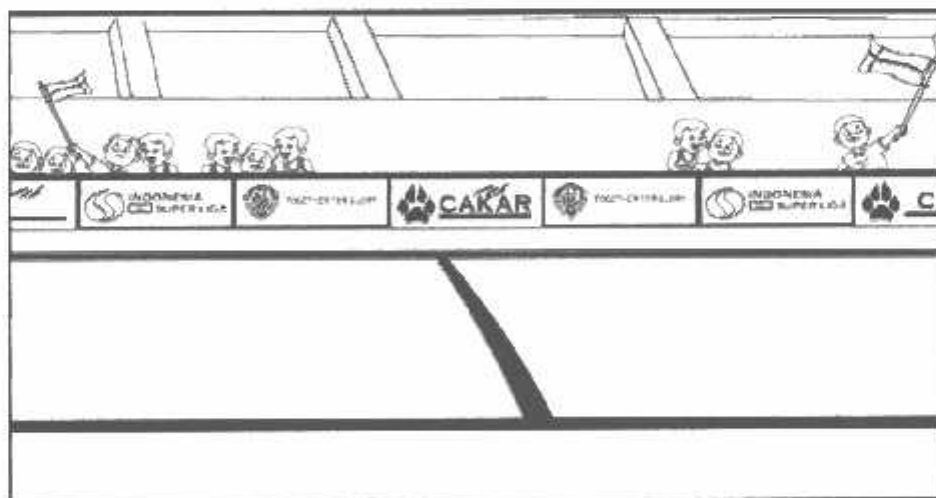
Pada perancangan tampilan *menu* terdapat lima *button* yaitu *button pinalty*, untuk masuk ke *game pinalty*. *Button shooting* untuk *game shooting*, *button heading* untuk masuk ke *game heading*, *button passing*, untuk *game passing* dan *button kuis* untuk masuk ke *game kuis*. *Design button* seperti yang terlihat pada gambar 3.11



Gambar 3.11 Perancangan *button game*.

3.3.3 Perancangan *Arena Game*

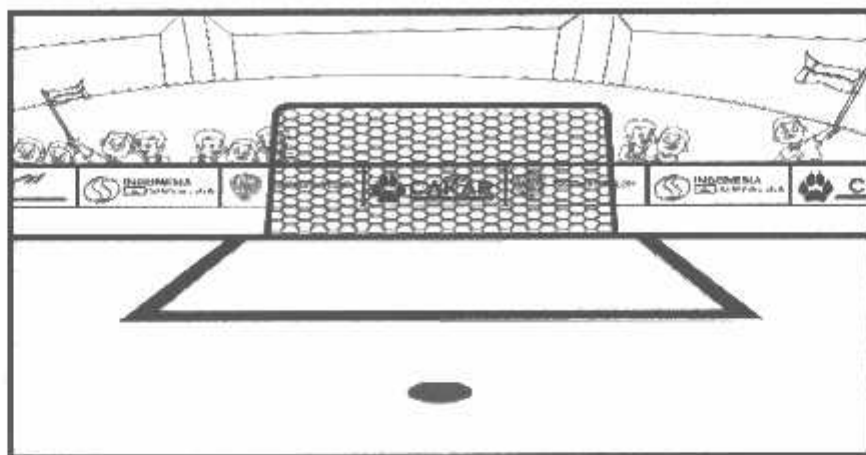
Tampilan *arena* untuk *stage game passing*, *game heading*, dan *game shooting*. *Stage* ini terdiri dari tampilan lapangan, tampilan penonton, tampilan tribun. Seperti pada gambar 3.12 Perancangan tampilan *arena stage game shooting*.



Gambar 3.12 Perancangan *arena game*.

3.3.4 Perancangan Arena Game Pinalty.

Tampilan untuk *stage game pinalty* ini terdiri dari tampilan lapangan, tampilan penonton, tampilan tribun. Seperti pada gambar 3.13. Perancangan tampilan arena *stage game pinalty*.



Gambar 3.13 Perancangan arena untuk *game pinalty*.

3.4 Komposisi Game

3.4.1 Perancangan Karakter Pemain.

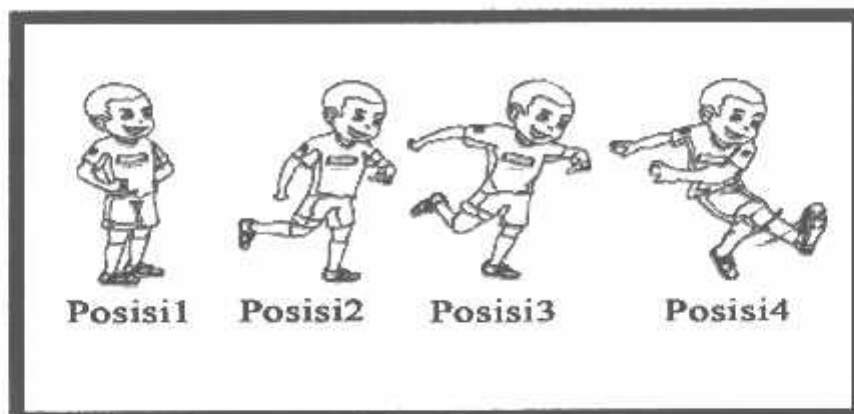
Karakter untuk setiap *game* di buat berbeda-beda pada tiap – tiap *game*. Seperti pada *game1*, *game pinalty* karakter yang di gunakan adalah seorang penjaga gawang (kipcr), kemudian pada *game heading* dan *game passing* karakter yang digunakan adalah pemain tengah, selanjutnya pada *game shooting* karakter yang di tampilkan seorang pemain berposisi sebagai penyerang *striker*.



Gambar 3.14 Perancangan karakter *game*.

3.4.2 Perancangan Gambar Posisi Karakter *Game Shooting*.

Tampilan gerak pemain dalam *game shooting* memiliki beberapa posisi, posisi1 awal sebelum pemain melakukan *shooting* bola ke target. Posisi selanjutnya posisi2 sampai posisi4 pada saat pemain melakukan tendangan *shooting* ke target.



Gambar 3.15 Rancangan posisi karakter *game shooting*.

3.4.3 Perancangan Komponen Game

Indikator *power* berfungsi sebagai pengukur kekuatan tendangan pada saat memainkan *game shooting* dan *game pinalty*. Untuk menentukan titik yang sesuai pada indikator menggunakan tombol aksi.

Tampilan poin ini akan menampilkan poin yang di dapat pada saat kita memainkan *game*.

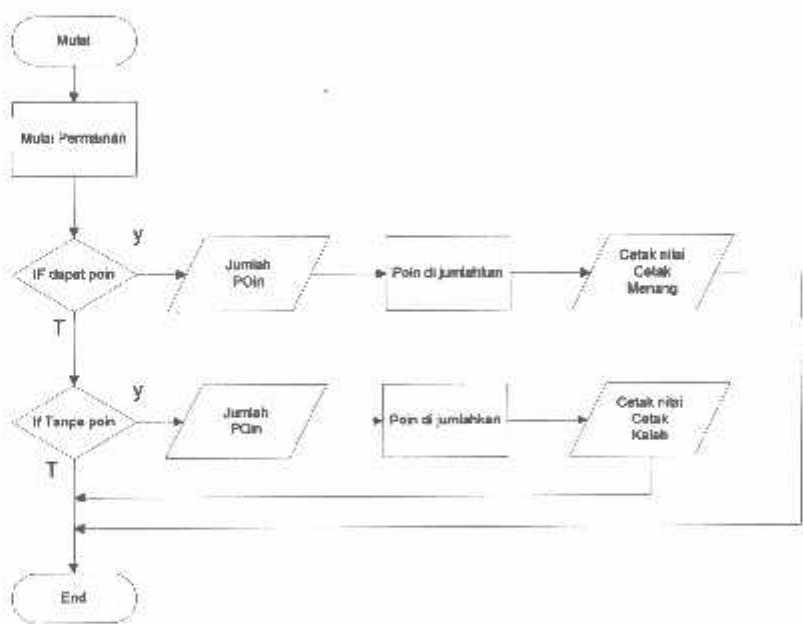


Gambar 3.16 Perancangan komponen *game*.

3.5 Perancangan Scoring.

3.5.1 Perancangan Sistem Penilaian.

Sistem penilaian adalah sebuah sistem yang digunakan pada *game* untuk memberikan penilaian pada *game* setelah selesai dimainkan. Dalam *game* Latihan Sepak bola sistem penilaian tiap *game* berbeda-beda pada *game shooting* sistem penilaian di nilai dari ketepatan pemain menendang bola mengenai target bola yang di gantung. Pada *game passing* dan *heading* sistem penilaian di nilai dari ketepatan pemain menendang bola yang di lesatkan oleh lawan.dan untuk *game pinalty* sistem penilaian di ambil dari banyaknya pemain memasukkan bola ke gawang yang di jaga oleh kiper. Setiap pemain akan mendapatkan nilai sempurna jika dapat mengumpulkan 4 poin dari 5kali kesempatan bermain. Jika nilai kurang dari 4 poin maka nilai tidak sempurna, poin akan bernilai antara 20-60. Tetapi jika dapat mengumpulkan 4 – 5 poin maka nilai sempurna,poin akan bernilai 80-100.



Gambar 3.17 Flowchart alur proses nilai.

3.5.2 Perancangan Poin.

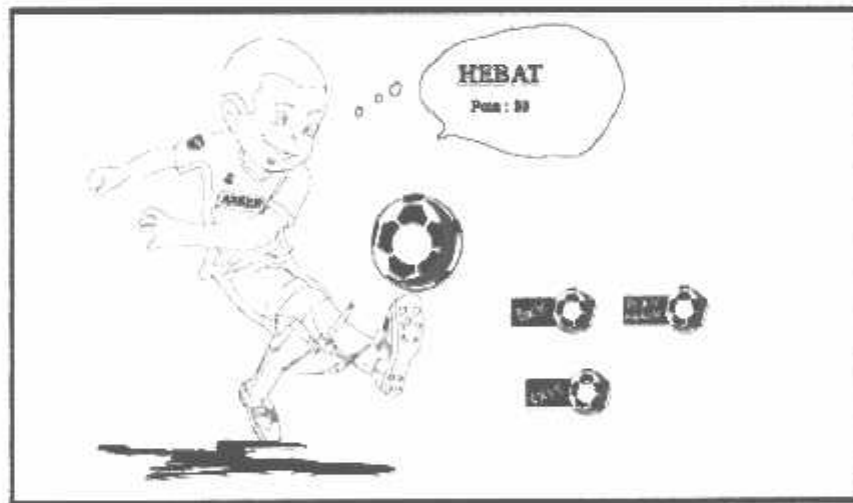
Tampilan hasil pemain jika dapat mengumpulkan nilai poin 4 dari 5 kali kesempatan bermain.



Gambar 3.18 Perancangan tampilan hasil poin.

3.5.3 Perancangan Tampilan Cetak Nilai.

Tampilan ini berfungsi sebagai petunjuk nilai dari hasil permainan *game*. Pada tampilan ini terdapat tampilan nilai, serta terdapat tiga tombol. Tombol pertama *back to menu* untuk kembali ke menu awal *game*. Kedua tombol *play again* untuk mengulang permainan, dan tombol ketiga tombol *exit* untuk keluar dari *game*.



Gambar 3.19 Perancangan tampilan cetak nilai.

3.6. Perancangan Game kuis.

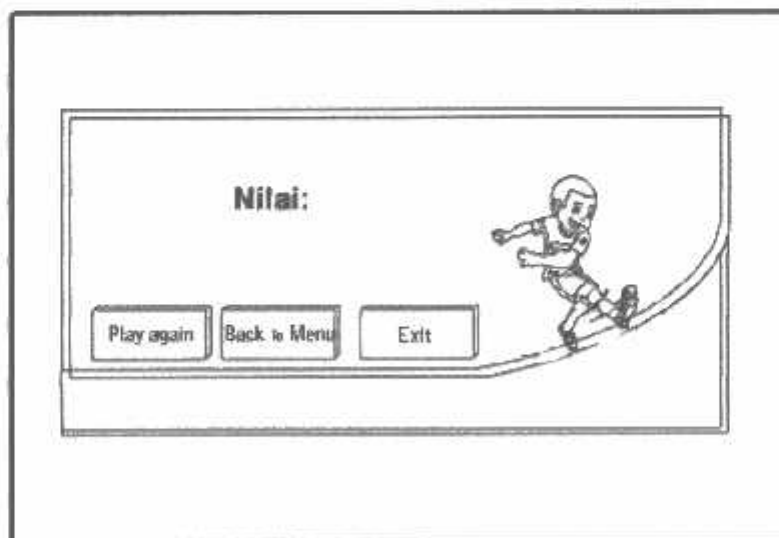
Konsep *game* kuis ini akan menggunakan soal secara *random* dengan jumlah soal sebanyak 20 soal. Serta jawaban pilihan sejumlah 4 pilihan jawaban, dari setiap soal yang akan keluar secara acak.

1. Tampilan *game* kuis.



Gambar 3.20 Perancangan tampilan *game* kuis.

2. Tampilan total nilai, dalam tampilan ini terdapat tombol untuk mengulang soal kuisioner dan tombol back menu untuk kembali ke menu awal.



Gambar 3.21 Perancangan tampilan total nilai.

3.6.1 Perancangan Outro / Exit.

Pada saat mengakhiri permainan pada *game* maka akan tampil *outro exit* seperti pada gambar 3.22. Berikut ini :



Gambar 3.22 Perancangan *outro exit game*.

BAB IV

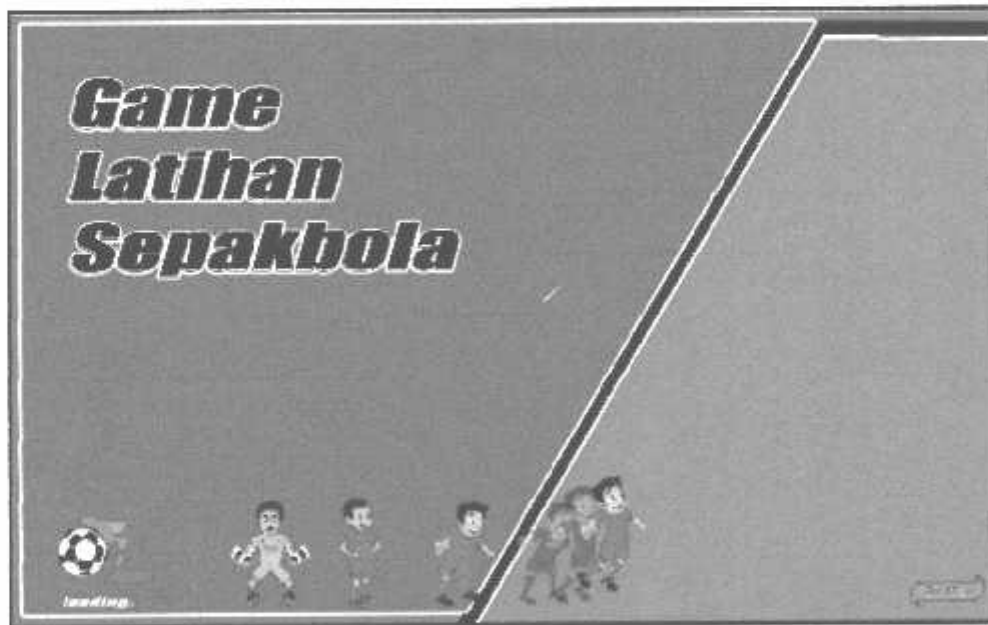
IMPLEMENTASI DAN PENGUJIAN

4.1. Implementasi

Implementasi adalah tahapan dalam membangun *game* latihan sepak bola, pada *proses* implementasi dalam *game* latihan sepak bola ini, membutuhkan beberapa tahapan-tahapan, yang meliputi pembuatan *intro game*, pembuatan halaman *menu* utama, pembuatan *game pinalty*, pembuatan *game shooting*, pembuatan *game heading*, pembuatan *game passing* dan pembuatan *game* tanya jawab / kuis. Serta pembuatan *outro* dan *exit*.

4.1.1 Implementasi Tampilan Intro.

Tampilan *intro* pada *game* latihan sepak bola memiliki animasi *loading* perputaran lingkaran serta *loading* gerak pemain. Sebelum tombol *play now* muncul.



Gambar 4.1 Tampilan *intro loading game*.

Tahapan-tahapan yang diperlukan dalam pembuatan tampilan *intro* :

1. Menjalankan perangkat lunak *Adobe flash professional cs.5*. Membuat *new document* pilih *type action script 2.0* dengan ketentuan ukuran yang akan di gunakan 980 x 615 *pixel* dan ubah ukuran *frame rate* menjadi 30.00 *fps*.
2. *Proses* konfigurasi ukuran lembar kerja, dapat dilakukan pada saat awal membuka *adobe flash professional cs.5*, yakni pilih *file* pilih *new* atau dapat dilakukan dengan *Cntrl + N* lalu oke. *Proses* Konfigurasi tersebut dapat dilihat pada gambar 4.2.



Gambar 4.2 Konfigurasi ukuran lembar kerja.

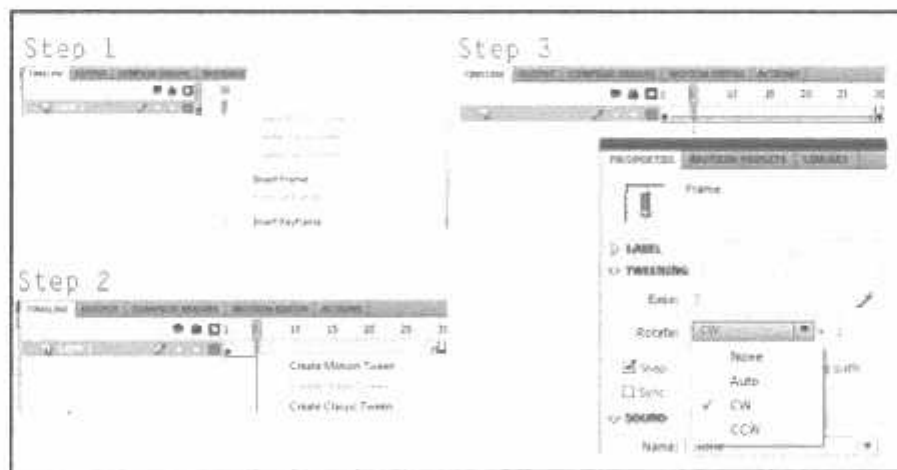
Setelah melakukan konfigurasi pada dokumen yang baru, maka *proses* pembuatan *intro* dapat dilakukan. *Proses* dalam pembuatan halaman *intro* meliputi.

1. Melakukan *import* data ke dalam lembar kerja yang sudah di buat sebelumnya. Dengan cara klik *file* kemudian pada pilihan *import* klik *import to stage*. Setelah itu pilih gambar yang ingin dijadikan *background*. Gambar *background* yang sudah di *import* dapat di lihat pada gambar 4.3



Gambar 4.3 Tampilan *background intro game*.

2. Pada *layer Background* pada *frame 1*, klik kanan pada objek *background* pilih *convert to symbol*, pada halaman *convert to symbol* pilih *movie clip* beri nama area lalu ok.
3. Masuk ke dalam *movie clip* area, buat *layer* baru dengan nama *loading*, membuat animasi gerak untuk *loading*. Pada *layer loading* import gambar bola kedalam lembar kerja caranya sama dengan *import* gambar *background*. Kemudian klik kanan objek bola pilih *convert to symbol* jadikan *movie clip* beri nama *loading*.
4. Masuk ke dalam *movie clip loading* pada *layer 1 frame 1* objek bola untuk membuat bola bergerak maka tentukan durasi gerak dari *frame 1* sampai *frame* ke 30 klik kanan pilih *insert keyframe*, (*step 1*).
5. Kemudian ambil posisi antara *frame 1-30* klik kanan pilih *create classic tween*, (*step 2*).
6. Selanjutnya terdapat garis pada posisi antara *frame 1-30* kemudian klik antara *frame 5*, pada *properties* atur *tweening* ubah *rotate auto* menjadi *cw*. Untuk membuat gerak bola berputar sesuai dengan putaran jarum jam, (*step 3*). Untuk lebih jelas lihat gambar 4.4.

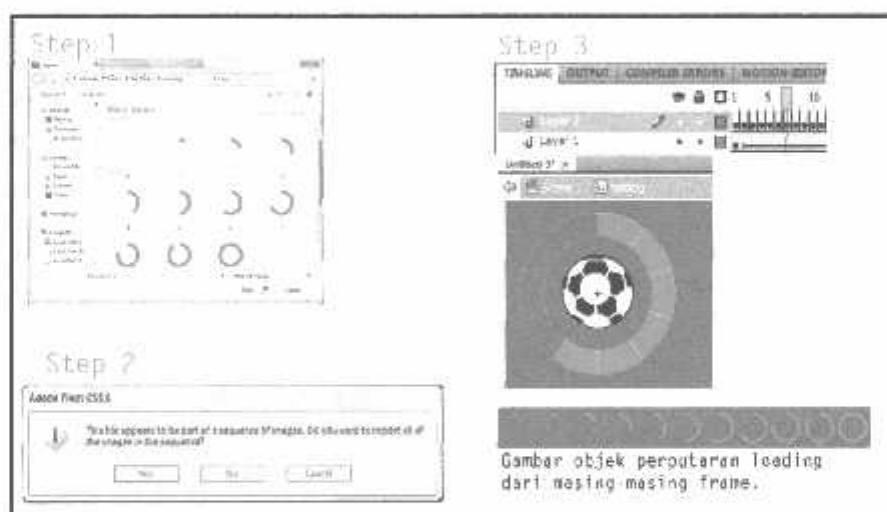


Gambar 4.4 Step membuat gerak bola.

7. Kemudian buat *layer 2* (masih dalam *movie clip loading*) untuk membuat gerak perputaran *loading*. *Import* gambar perputaran *loading* yang sudah

di buat sebelumnya pada *photoshop*. Gambar *loading* yang di buat pada *photoshop* sebelumnya *save* dengan nama 0 - 10 (urut), dengan format *.png*.

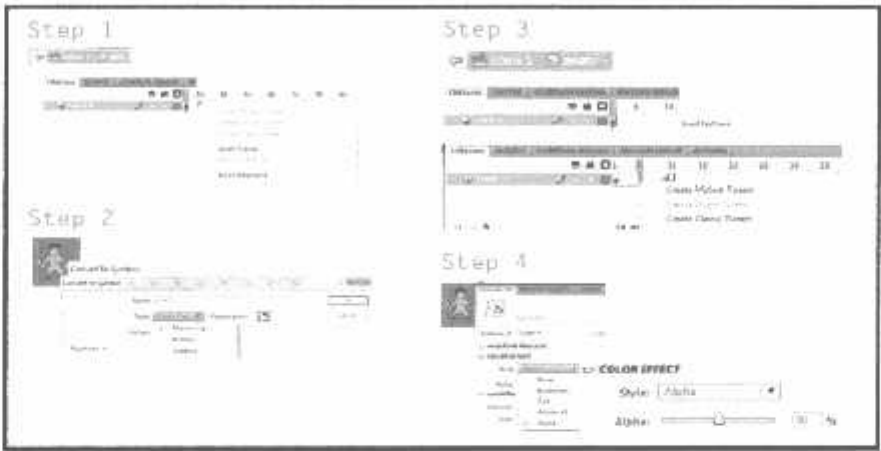
8. Untuk mengimport gambar perputaran *loading* caranya. *File import pilih import to stage*.
9. Muncul Jendela *import data* cari *data* yang akan diimport yang sudah di simpan pada komputer kemudian klik salah satu gambar 01 format *.png* yang akan diimport kemudian klik *open*, (*step 1*).
10. Kemudian akan muncul pesan untuk meng *import* semua gambar klik *ok*, (*step 2*).
11. Pada *layer 2 frame 1 – 11* akan muncul gambar perputaran *loading*, (*step 3*). Lihat gambar 4.5.
12. Untuk membuat perputaran *loading* lebih lama copi kan *frame 1* sampai 11 ke *frame* selanjutnya.



Gambar 4.5 Urutan proses perputaran *loading*.

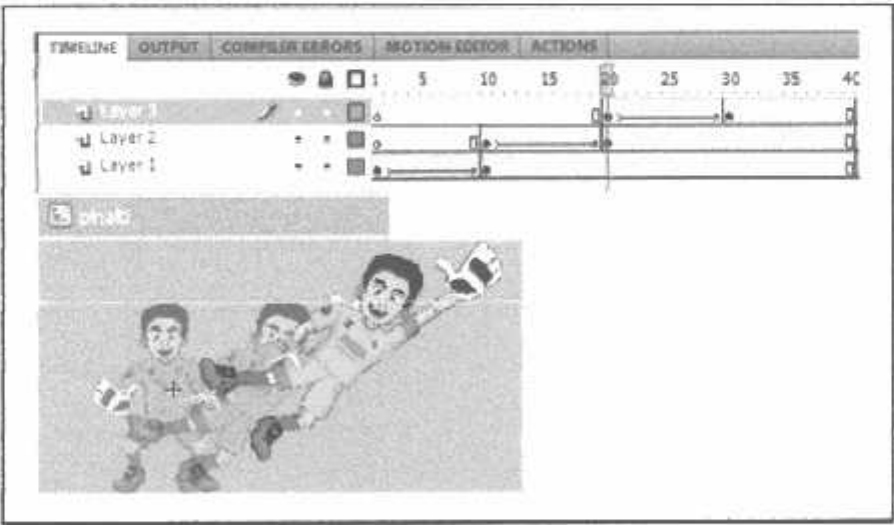
13. Membuat Animasi karakter pemain untuk *loading*. Masuk pada *movie clip area* buat *layer* baru *layer 3*. *Import* gambar pemain yang sudah di buat, pada *layer1 frame1* lalu pada *frame* ke 50. Dengan cara klik kanan pada *layer 50* pilih *insert keyframe*, (*step 1*).
14. Selanjutnya klik kanan pada *frame 1* objek pemain pilih *convert to symbol* pilih *movie clip* beri nama *penalty*, (*step 2*).

15. Pada *movie clip pinalty layer 1 frame 1* objek pemain klik kanan pada *layer 10*. Pilih *insert keyframe* selanjutnya klik kanan antara *frame 1-10* pilih *create classic tween, (step 3)*.
16. Selanjutnya pada *frame 10* geser posisi gerak pemain, klik *frame 10* klik objek pemain lalu atur *properties* pilih *color effect* ubah *style* dari *none* menjadi *alpha* kemudian atur nilai *alpha* menjadi 50%, (*step 4*). Lihat gambar 4.6



Gambar 4.6 Step membuat objek pemain transparan.

17. Selanjutnya lakukan hal yang sama pada *layer 2 frame 10* sampai *frame 20* dan *layer 3 frame 20* sampai *frame 30*. Dengan objek pemain yang berbeda. Lihat gambar 4.7.

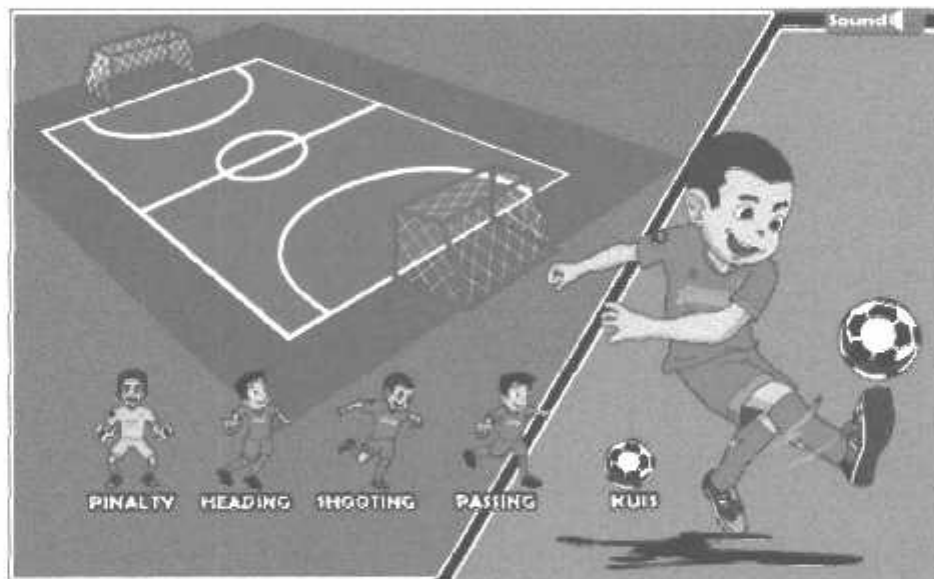


Gambar 4.7 Gerak pemain transparan.

18. Untuk gambar pemain yang lain dapat dilakukan dengan cara yang sama sesuai *proses* pembuatan pemain transparan.

4.1.2 Implementasi Menu Utama.

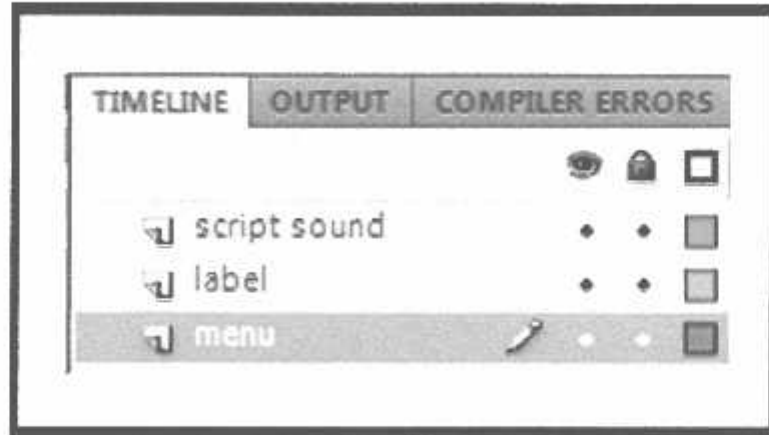
Implementasi pada tampilan *menu* terdapat 5 *button*. *Button game pinalty*, *button game shooting*, *button game heading*, *button game passing*, dan *button kuis game*. *Button - button* ini yang menghubungkan ke halaman permainan, *button game pinalty* menghubungkan ke *game pinalty*, *button game shooting* untuk menghubungkan ke *game shooting*, *button game passing* menghubungkan ke *game passing*, selanjutnya ada *button game heading* yang menghubungkan ke *game heading* dan *button kuis game* menghubungkan ke permainan *kuis game*. *button-button* ini di tampilkan secara sejajar dalam satu tampilan *menu* secara bersamaan . Berikut ini tampilan *menu* utama dapat dilihat pada gambar 4.8 di bawah ini.



Gambar. 4.8 Tampilan menu utama.

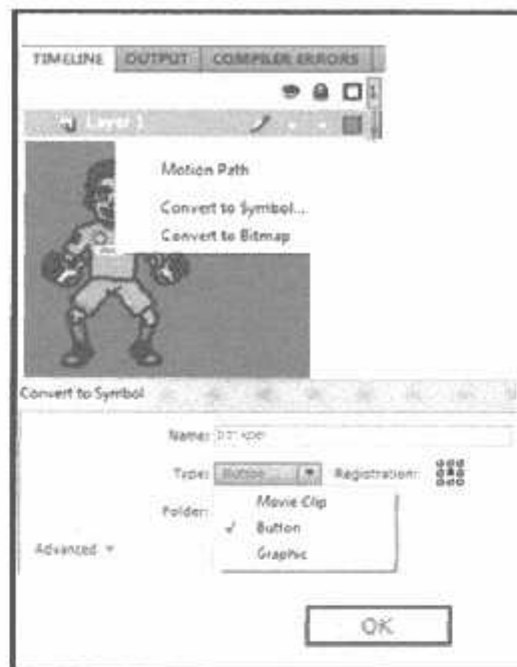
Tahapan-tahapan dalam membuat tampilan menu utama:

1. Pertama buat *new document* dan buat 3 jumlah *layer* yakni *layer1*, *layer2*, *layer3*. Beri nama untuk sctiap *layer*, *layer1* ubah menjadi *menu*, *layer2* ubah menjadi *label*, *layer3* ubah menjadi *sound script*.



Gambar 4.9 Layer untuk pembuatan menu utama.

2. Pilih *layer menu* lalu *import* gambar *background* yang sudah di buat sebelumnya. Klik kanan objek *background* pada *layer menu frame1*, lalu pilih *convert to symbol* pilih *movie clip* beri nama *menu*. Kemudian masuk *movie clip* menu buat *layer 2* dan *import* gambar lapangan. Lalu klik kanan gambar lapangan pilih *convert to symbol* beri nama garis lapangan.
3. Pada *movie clip* garis lapangan buat *layer* baru *layer2* untuk membuat tombol menu tiap-tiap *game*.
4. Pada *layer2 frame1* siapkan gambar untuk tombol lalu klik kanan gambar tersebut pilih *convert to symbol* pilih *button* beri nama *btn kper*.



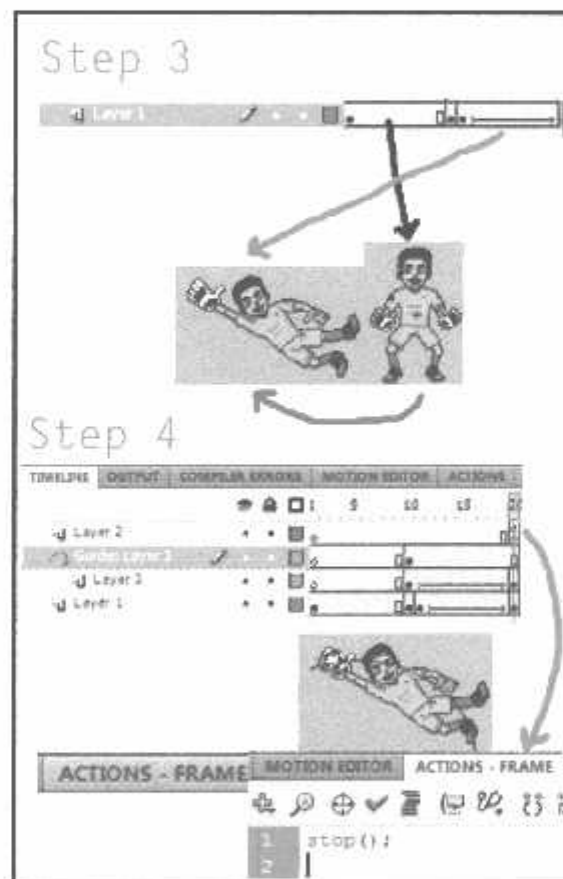
Gambar 4.10 Membuat *button menu game*.

5. Selanjutnya klik 2x pada *button* kiper maka akan masuk dalam area *button* dalam *layer1* terdapat *frame up over down hit* .(*step1*).
6. Klik kanan pada *frame over*, dan *frame down* lalu klik objek kiper pada *frame over* dengan klik kanan pilih *convert to symbol* pilih *movie clip*, beri nama *kp*, (*step 2*).



Gambar 4.11 Membuat *movie clip* pada *button*.

7. Selanjutnya masuk *movie clip* ke pada *layer1* objek kiper klik kanan pada *frame 10* pilih *insert key frame* kemudian pada *frame 11* klik kanan pilih *insert blank keyframe* kemudian *import* gambar untuk gerak ke2. Setelah pada *frame 11* objek sudah muncul klik kanan pada *frame20* ubah posisi gerak pemain, (step 3).
8. Buat *layer* baru untuk objek bola *insert keyframe* pada *layer 10* kemudian *import* gambar bola klik pada *frame 20* klik kanan *insert key frame* kemudian klik kanan pada *frame 15* pilih *create classic tween* selanjutnya klik kanan pada *layer* bola pilih *add classic motion Guide* maka akan muncul *layer* baru dengan nama *guide* pada *frame 1* buat garis untuk jalur bola dan arahkan objek bola pada titik tengah bola. Selanjutnya buat *layer* baru kemudian pada *frame 20* klik *insert keyframe* lalu klik *tools action frame* untuk memasukkan *action script* ketik *action script stop ()*, (step 4).



Gambar 4.12 Membuat efek *motion guide*.

9. Implementasi tampilan keseluruhan *button game* pada menu utama.



Gambar. 4.13 Tampilan keseluruhan *button menu*.

10. *Action script* pada *button menu* untuk melakukan *link* pada tiap-tiap *game*.
Lihat pada table 4.1.

Tabel 4.1 *Action script* untuk tombol-tombol pada menu.

Tombol	Script
<i>Pinalty</i>	<pre>on (press) { loadMovieNum("Game Pinalty.swf", 0); }</pre>
<i>Heading</i>	<pre>on (press) { loadMovieNum("Game Heading.swf", 0); }</pre>
<i>Shooting</i>	<pre>on (press) { loadMovieNum("Game Shooting.swf", 0); }</pre>
<i>Passing</i>	<pre>on (press) { loadMovieNum("Game Passing.swf", 0); }</pre>

Kuis	<pre>on (press) { loadMovieNum("Kuis Soal.swf", 0); }</pre>
------	---

4.1.3 Implementasi Tampilan Petunjuk Permainan.

Tampilan informasi petunjuk permainan dari masing-masing *game* yang ada. Seperti petunjuk permainan untuk *game pinalty*, petunjuk permainan untuk *game heading*, petunjuk permainan untuk *game shooting* dan petunjuk permainan untuk *game passing*, seperti pada gambar 4.14.



Gambar. 4.14 Petunjuk permainan pada tiap-tiap *game*.

4.1.4 Implementasi Komponen Game.

Beberapa fungsi komponen yang terdapat pada *game*.

1. Penunjuk target
Berfungsi sebagai pengganti *mouse* untuk menentukan titik sasaran target pada area gawang pada *game pinalty*.
2. Tombol aksi *button* bola.

Tombol aksi *button* bola berfungsi untuk melakukan aksi pada *game shooting*, *game penalty* pada saat menentukan indikator *power* .

3. Tombol arah.

Tombol ini berfungsi untuk mengerakkan pemain kearah *vertikal / horizontal*, pada *game passing*, dan *game heading*.

4. Indikator *power*.

Indikator *power* berfungsi untuk pemain menentukan kekuatan tendangan pada saat memainkan *game shooting* dan *game pinalty*.

5. Poin

Tampilan poin ini berfungsi untuk menampilkan *score* dari permainan.

6. Tombol aksi *space*

Tombol ini berfungsi pada saat kita memainkan *game heading* dan *game passing*. Untuk melakukan aksi *heading* maupun *passing* bola dari lawan.

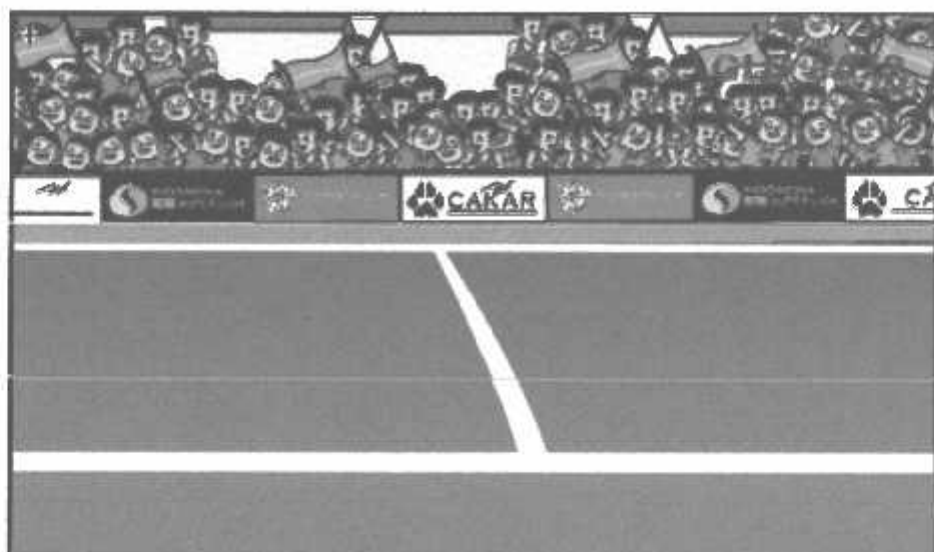


Gambar. 4.15 Tampilan komponen *game*.

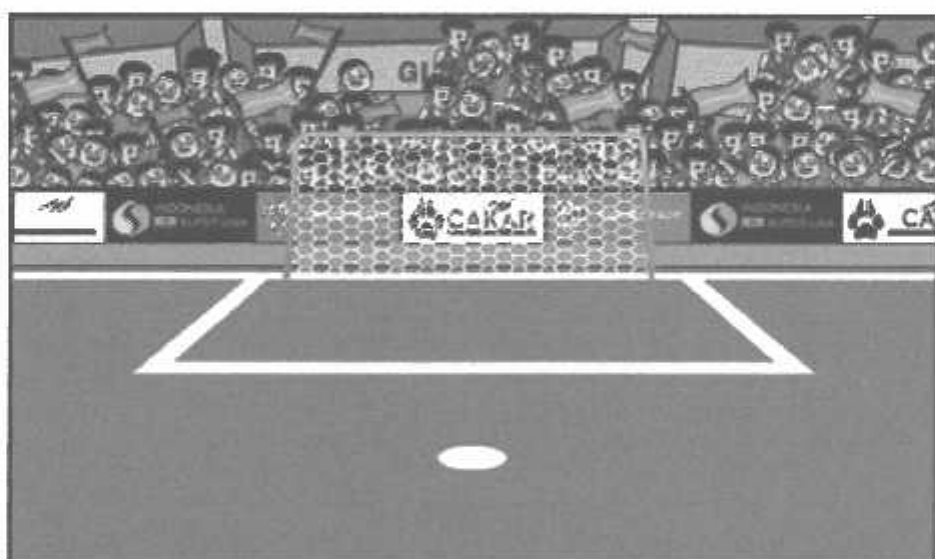
4.1.5 Implementasi Tampilan Arena Game

Terdapat 2 arena *game* pada *game* latihan sepak bola. Pertama arena untuk *game passing*, *game heading* dan *game shooting*. Yang di ditampilkan pada gambar 4.15. Dalam arena tersebut terlihat tampilan arena dari posisi samping lapangan, selanjutnya arena yang ke dua arena untuk *game pinalty* yang di ditampilkan pada gambar 4.16, yang memperlihatkan posisi dari arah

depan gawang. Dari masing masing arena tersebut terdapat tampilan *supporter*, tampilan *tribun*, dan *reklame* dari sisi garis lapangan.



Gambar. 4.16 Tampilan arena 1.

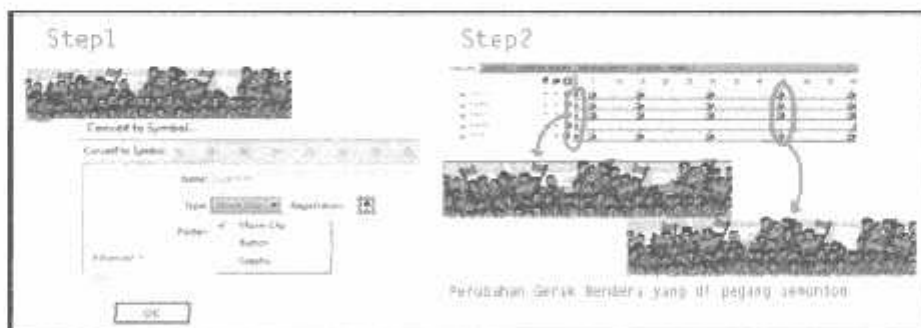


Gambar. 4.17 Tampilan arena 2.

4.1.6 Implementasi Game Pinalty

Implementasi *game pinalty* memiliki beberapa Komponen penting diantaranya *background* lapangan, tampilan *supporter* , area gawang, kursor target, indikator *power*, pemain (kipser) *buton* untuk aksi.

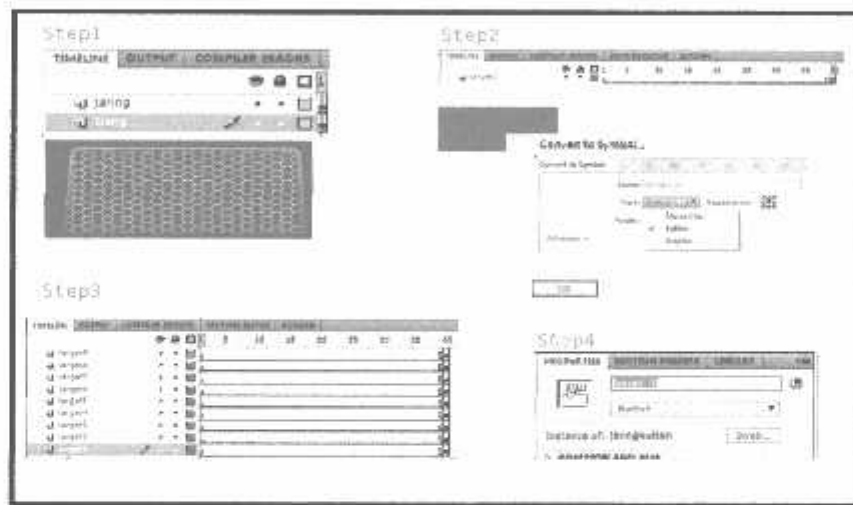
1. Pertama buat *new document* dengan nama *game pinalty*
2. *Import* gambar *background* lapangan yang sudah si buat. Kemudian klik kanan pilih *convert to symbol* pilih *movie clip* beri nama *pinalty game* kemudian buat *layer* baru beri nama *supporter* untuk membuat animasi *supporter*.
3. Pada *layer supporter frame1 insert frame* sampai *frame40*, lalu pada *frame 1* gambar penonton seolah 2 menyaksikan sepak bola kemudian klik kanan pada objek *frame1* pilih *convert to symbol* pilih *movie clip* beri nama *supporter*, (*step 1*).
4. Kemudian buat *layer* baru pada area dalam *movie clip supporter* pada *frame 1 insert keyframe* ke *frame 60*. Kemudian klik *insert key frame* pada *frame 5*, pada gambar objek ubah posisi penonton sesuai gerakan yang diinginkan seperti menggerakkan bendera. Selanjutnya pada *frame 10* klik *insert keyframe* ubah lagi posisi gerak penonton. Lakukan hal ini sampai pada *frame* ke 60. Copi gambar penonton sehingga penonton terlihat banyak, (*step2*).



Gambar 4.18 Membuat *movie clip supporter* dan animasi.

5. Membuat gambar tiang gawang dan jaring sebagai penunjuki luas area sebagai titik sasaran yang bertujuan untuk tujuan gerak bola pada saat bola di jalankan, (*step1*).

6. Pertama buat *layer* baru dengan nama *target1* pada *frame* 40 klik kanan *insert keyframe* lalu buat objek kotak, buat kotak target gawang tersebut menjadi *button* pada *layer target 1 frame*. Pertama klik kanan objek kotak *target area* pilih *convert to symbol* pilih *button* beri nama *jaring button*,(step2).
7. Setelah *button* selesai copi objek kotak untuk target tersebut sebanyak 9x lalu buat *layer* baru sebanyak 8x, dengan nama *target2- target 9*,(step3).
8. Pada objek kotak klik objek kotak beri nama pada *instand name* yang terdapat pada *properties*, (step4).



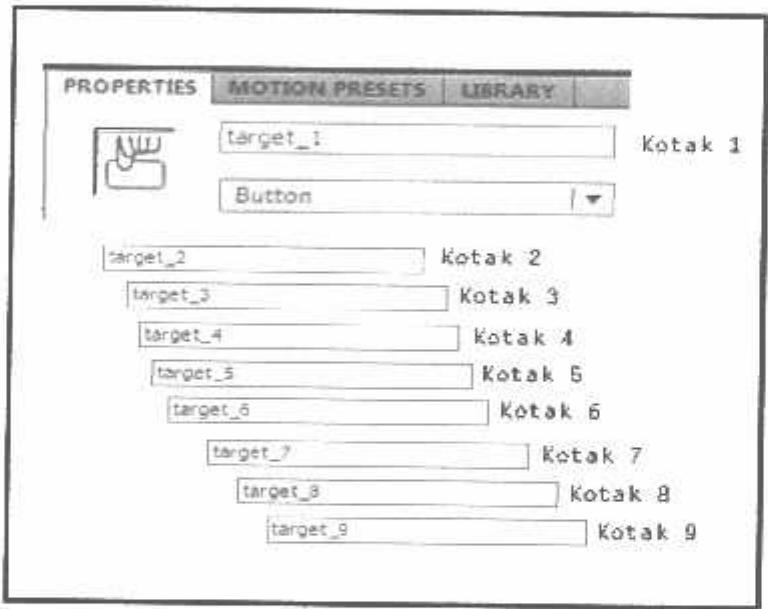
Gambar. 4.19 Membuat area pilihan pada area gawang.

9. Mengatur transparansi objek kotak sehingga objek nantinya tidak terlihat pada saat *game* di jalankan dengan cara klik objek kotak pada *target1* kemudian masuk *tools properties* pilih *syle none* ubah menjadi *alpha* dan atur nilai transparan menjadi 50%, (lakukan hal yang sama pada 8 kotak lainnya).



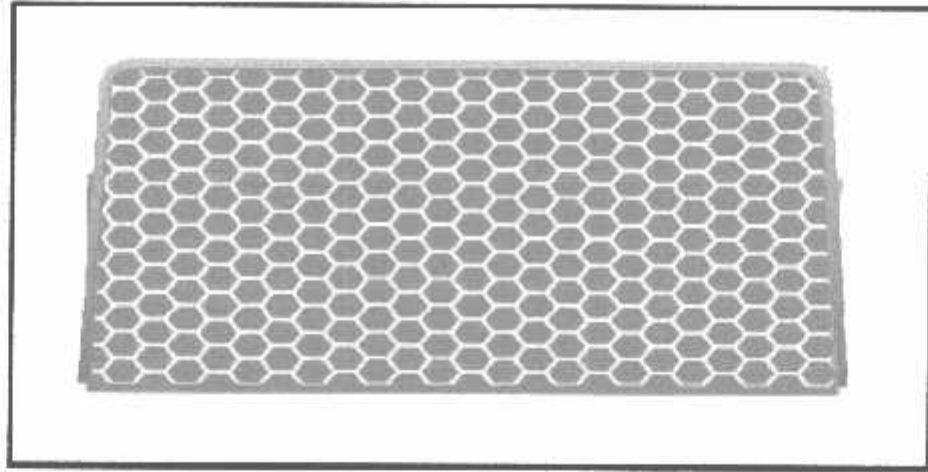
Gambar 4.20 Objek transparan.

10. Setelah objek dicopi sebanyak 9x kemudian dari masing-masing objek beri nama pada *instance name* yang berbeda-beda sesuai urutan nomor target. Seperti target_1,target_2,target_3, sampai target ke 9.



Gambar 4.21 Nama *instance name* pada setiap objek target.

11. Atur posisi kotak pada area gawang sehingga luas gawang tertutupi oleh objek kota tersebut.



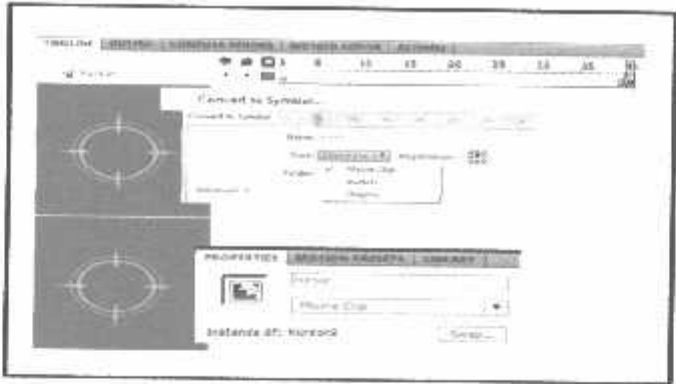
Gambar 4.22 Area kotak yang sudah diatur posisinya.

12. Membuat penjaga gawang. Pertama siapkan semua gambar penjaga gawang yang sudah di buat, kemudian buat *layer* dengan nama pemain klik kanan pada *frame 40 insert keyframe*, kemudian masukkan gambar penjaga gawang dengan posisi awal siap.
13. Klik kanan pada penjaga gawang pilih *convert to symbol* pilih *movie clip* beri nama kiper untuk *instance name KipperMC*.
14. Masuk *movie clip* kiper buat *layer* baru masukkan objek bola. Lihat gambar 4.23.



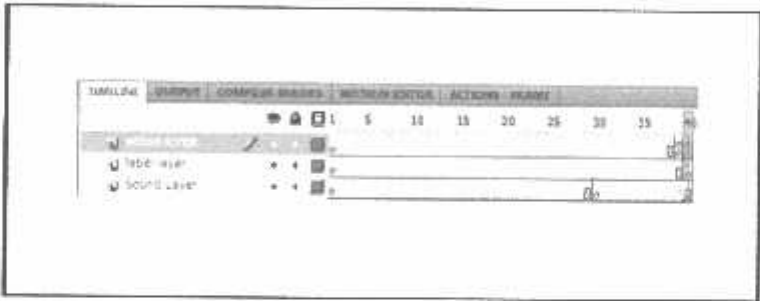
Gambar 4.23 Pemain kiper dalam posisi gawang.

15. Membuat gerak kiper, masuk pada *movie clip* kiper lalu klik pada *frame 10* klik kanan *insert key frame* pilih *convert to symbol* pilih *movie clip* beri



Gambar 4.26 Kursor target.

18. Buat *layer* untuk *action script*, *layer* untuk *sound*, dan *layer* untuk *label name*.



Gambar 4.27 Layer sound, layer script.

19. Kemudian masukkan *script* untuk *game* pada *tools actions frame* atau tekan *f9* untuk membuka jendela *action script frame*.



Gambar 4.28 Tools untuk memasukkan script.

4.1.7 Implementasi Tampilan Score

Tampilan ini menampilkan *score* pemain, seperti pada gambar 4.29. Dalam tampilan *score* ini terdapat *button back menu*, *exit*, *play again*.



Gambar. 4.29 Tampilan nilai *score*.

4.1.8 Implementasi Game Shooting

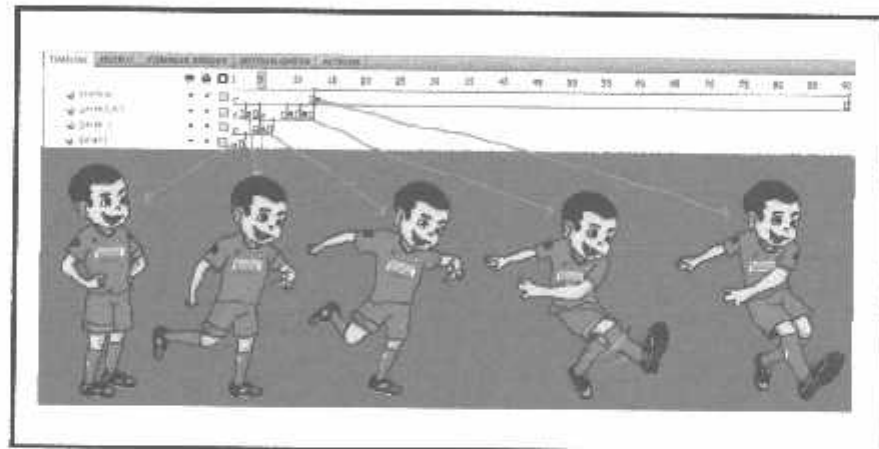
Implementasi *game shooting* memiliki beberapa komponen penting diantaranya *background* lapangan, *supporter*, penendang, objek target, indikator *power*, *buton* untuk aksi.

1. Pertama buat *new document* dengan nama *game shooting*.
2. *Import* gambar *background* lapangan yang sudah si buat. Kemudian klik kanan pilih *convert to symbol* pilih *movie clip* beri nama *shooting game*
3. Selanjutnya buat *layer* baru dengan nama pemain klik kanan pada *frame* 40 lalu buat karakter pemain. Atau *import* gambar pemain yang sudah ada.
4. Pada *layer* pemain objek pemain buat menjadi *movie clip*. Beri nama pemain.



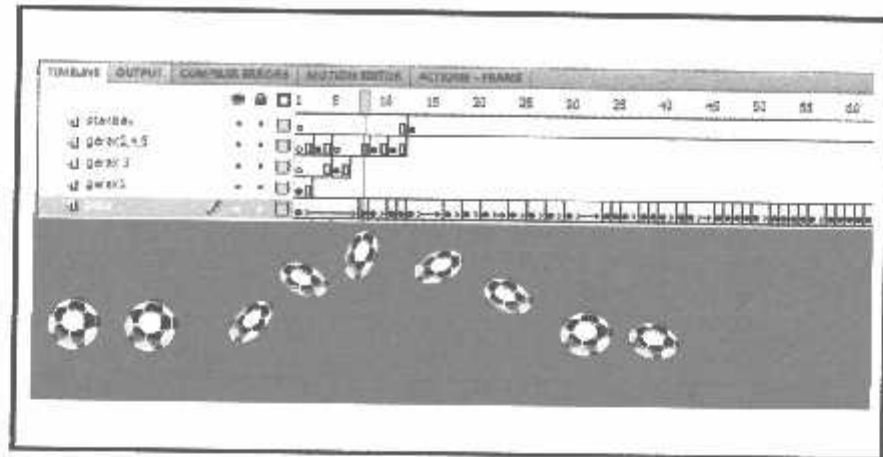
Gambar 4.30 Objek pemain pada *game shooting*.

5. Membuat gerak pemain pada *movie clip* pemain. Masuk *movie clip* kemudian buat *layer* baru untuk gerak pemain dari tiap – tiap gerakan. Sehingga pemain terlihat tampak menendang bola, atur posisi sesuai dengan posisi gerakan dari gambar yang sudah di buat.



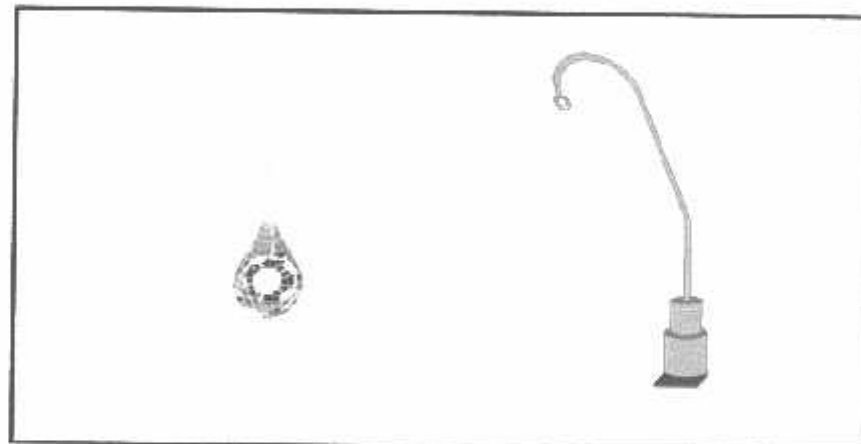
Gambar 4.31 Gerak pemain dalam *game shooting*.

6. Membuat animasi bola bergerak seakan-akan bola di tendang pemain dari yang kita buat tadi.
7. Pertama-tama masuk *movie clip* pemain, buat gambar bola pada *layer* baru dengan nama bola kemudian buat animasinya, pada *frame1* *layer* bola klik kanan ke *frame8*, kemudian klik *insert keyframe* selanjutnya pada *frame 10* klik *insert key frame* dan seterusnya lakukan pembagian *frame* sesuai dengan gerak bola.



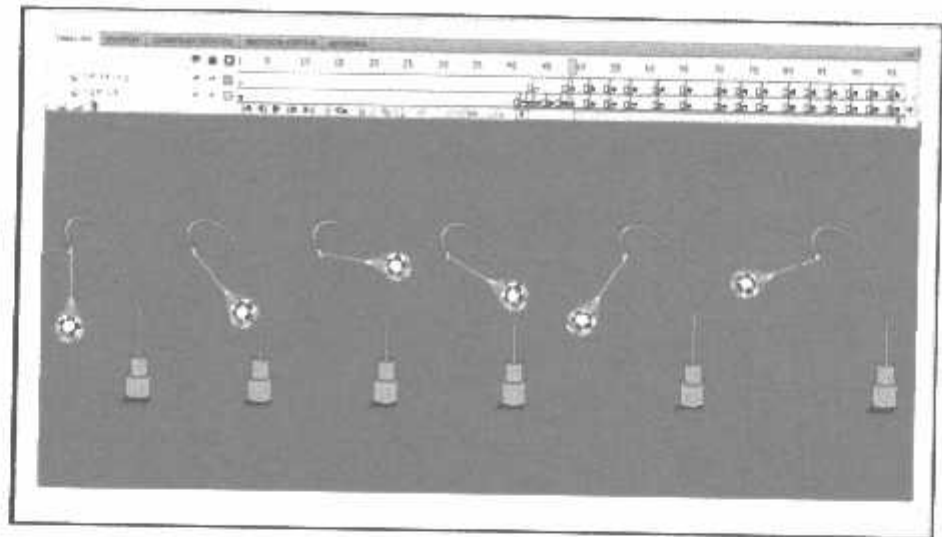
Gambar 4.32 Gerak bola dalam *game shooting*.

8. Membuat objek target (sasaran), pertama buat *frame* target di dalam *movie clip* pemain kemudian pada *frame 1* layer target buat tiang target dan bola gantung seperti gambar di bawah ini.



Gambar 4.33 Tiang target dan objek bola.

9. Membuat animasi gerak bola target pada tiang, jadikan *movie clip* target pada *frame 1* klik kanan *insert key frame* pada *frame 42* selanjutnya buat efek gerak. pertama gabungkan gambar objek bola pada tiang target dengan menempatkan posisi *layer* jaring seakan menempel pada objek bola yang terdapat pada *layer* tiang, seperti pada gambar di bawah ini.

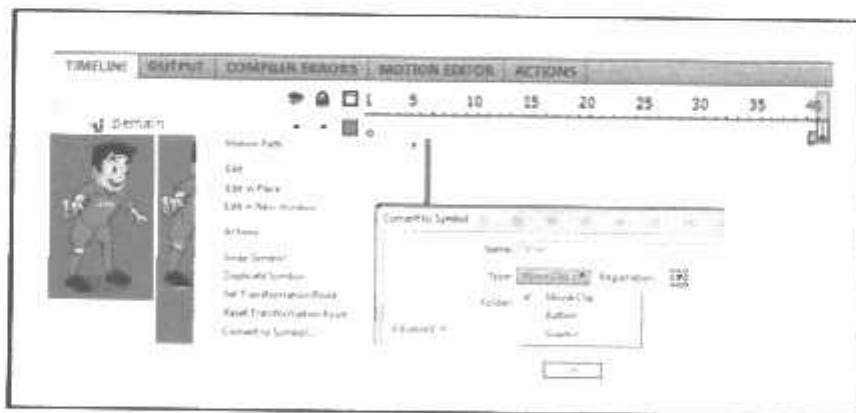


Gambar 4.34 Pergerakan target bola.

4.1.9 Implementasi Game Heading

Implementasi *game heading* memiliki beberapa komponen penting diantaranya *background* lapangan, *supporter*, pemain, penendang (pemain lawan). Tahapan awal membuat *new document* dengan nama *game heading*.

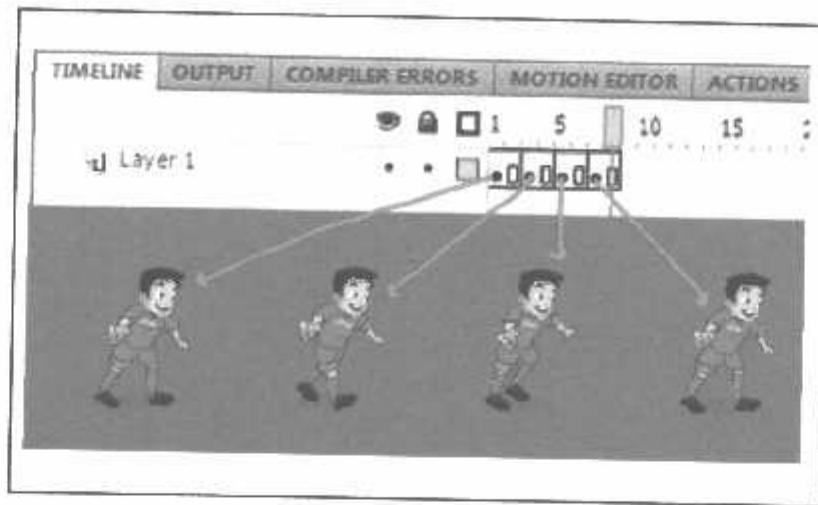
1. Pertama buat *layer* baru dengan nama pemain. Klik kanan pada *frame40* *insert keyframe*, *import* objek pemain kemudian klik kanan pilih *convert to symbol* pilih *movie clip* beri nama pemain lalu oke.



Gambar 4.35 Membuat *movie clip* pemain.

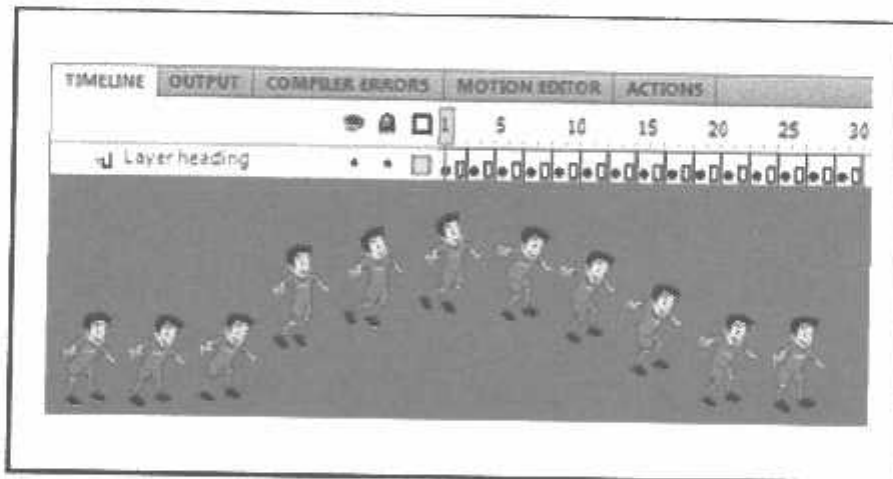
2. Membuat gerak pemain untuk berjalan pertama pada *frame2* klik kanan *insert key frame* kita buat lagi *movie clip* ke 2 kemudian masuk *movie clip* ke 2 kita buat animasi untuk pemain bergerak berlari. untuk membuat

pemain berlari pada *movie clip* ke2 lakukan pembuatan gambar secara bertahap sesuai *frame* yang di butuhkan.



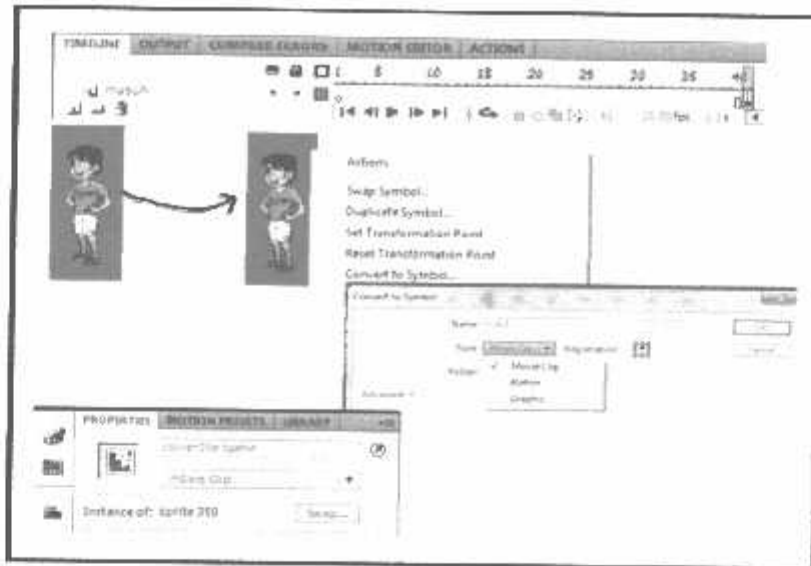
Gambar 4.36 Gerak lari pemain.

- Selanjutnya keluar dari *movie clip* ke2 dan pada *frame* ke 4 buat animasi pemain melakukan *heading*. Pertama buat *movie clip* baru misal nama *head* kemudian pada *frame1* objek pemain pertama, pada *frame* selanjutnya objek pemain pada posisi gerak yang di buat untuk melakukan *heading*.



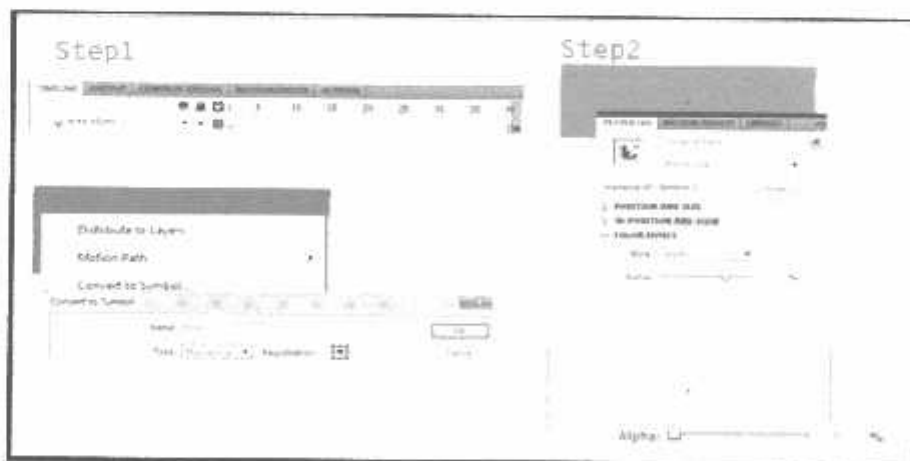
Gambar 4.37 Gerak pemain pada saat akan *heading*.

- Selanjutnya kita akan membuat *layer* musuh untuk pemain lawan yang nantinya berfungsi sebagai pengoper bola yang bergerak secara otomatis.
- Kita buat *layer* musuh selanjutnya klik kanan pada *frame40* lalu *import* gambar pemain lawan pada posisi pertama. Kemudian klik kanan *convert to symbol* buat *movie clip* beri nama musuh.



Gambar 4.38 Membuat *movie clip* pemain lawan.

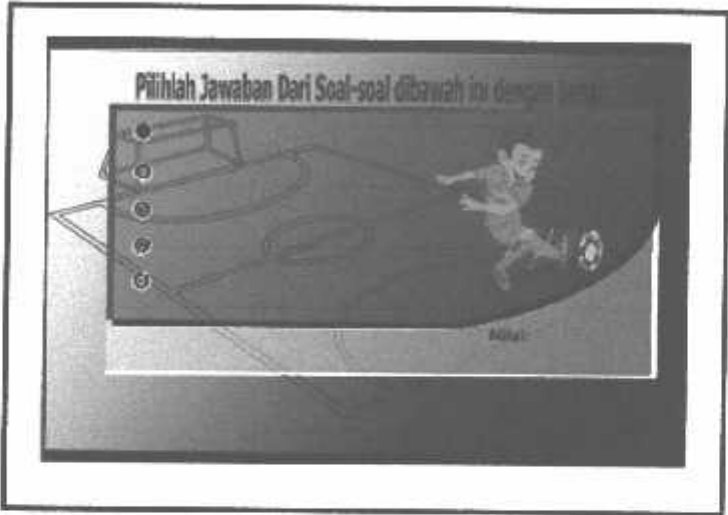
6. Selanjutnya kita akan membuat area lajur bola sehingga bola akan bergerak secara *random* hanya pada area yang sudah di tentukan luas areannya.
7. Kita buat *layer* baru klik pada *frame 40* klik kanan *insert key frame* selanjutnya jadikan objek area tersebut menjadi *movie clip* klik kanan *convert to symbol* pilih *movie clip* beri nama area, (*step1*).
8. Selanjutnya kita jadikan objek tersebut menjadi transparan klik pada objek area tersebut lalu kita lihat pada *tools properties* terdapat pilihan *effect* pilih *color effect* pada *style none* ubah menjadi *alpha* kemudian ubah ukuran dari 100% menjadi 0% untuk kesamaran transparan pada objek, (*step2*).



Gambar 4.39 Membuat objek area menjadi transparan.

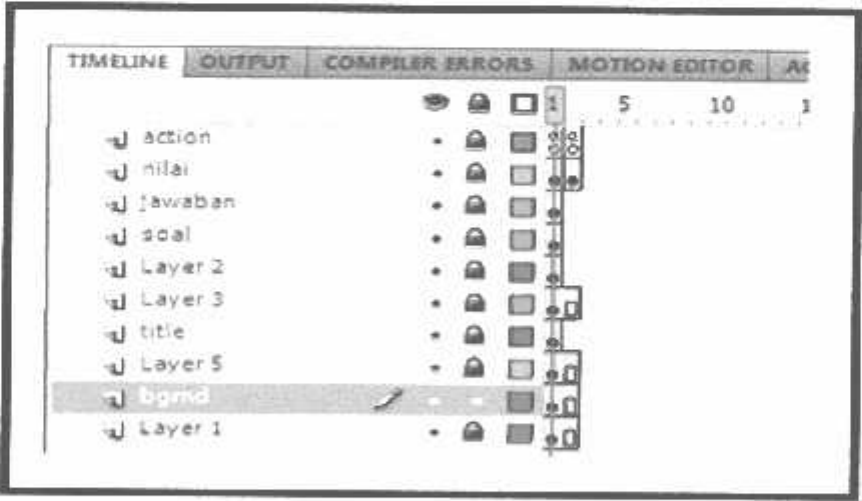
4.1.10 Implementasi Game Kuis

Tampilan untuk *game* kuis dengan soal jawaban secara acak yang akan muncul pada saat *game* di jalankan, pilihan jawaban terdapat 4 jumlah yakni *a,b,c,d*. Dalam kuis ini terhubung dengan lembar soal untuk menuliskan soal-soal yang ada pada *game* yang di dsimpan dalam bentuk *format xml.doc*. Berikut tampilan dari *game* kuis.



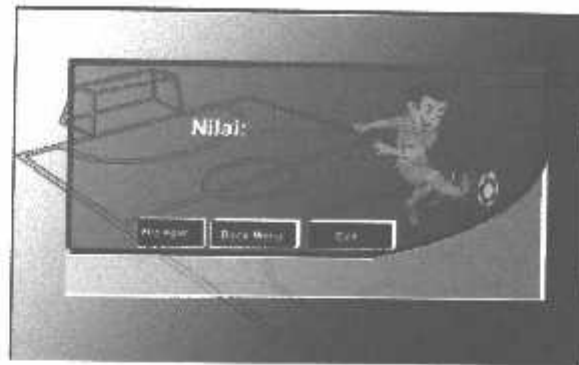
Gambar 4.40 Tampilan *game* kuis.

- 1. Dalam *game* kuis terdapat beberapa *layer* seperti pada gambar di bawah ini gambar 4.41.



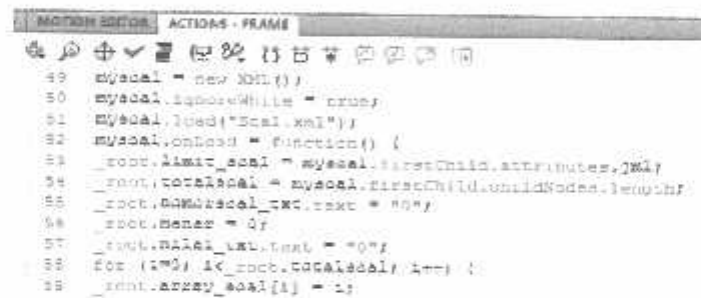
Gambar 4.41 Beberapa *layer* yang ada pada *game* kuis.

2. Tampilan untuk nilai pada *game* kuis nilai keseluruhan, serta terdapat tombol *play again*, *back menu*, *exit*.



Gambar 4.42 Tampilan total nilai.

3. *Script* untuk memanggil *file* soal yang di *save* dengan *format xml.doc* yang terhubung dengan *game kuis*.



Gambar 4.43 *Script link file soal untuk game kuis*

4.1.11 Implementasi Tampilan Exit Game

Tampilan *exit* pada *game* latihan sepak bola.



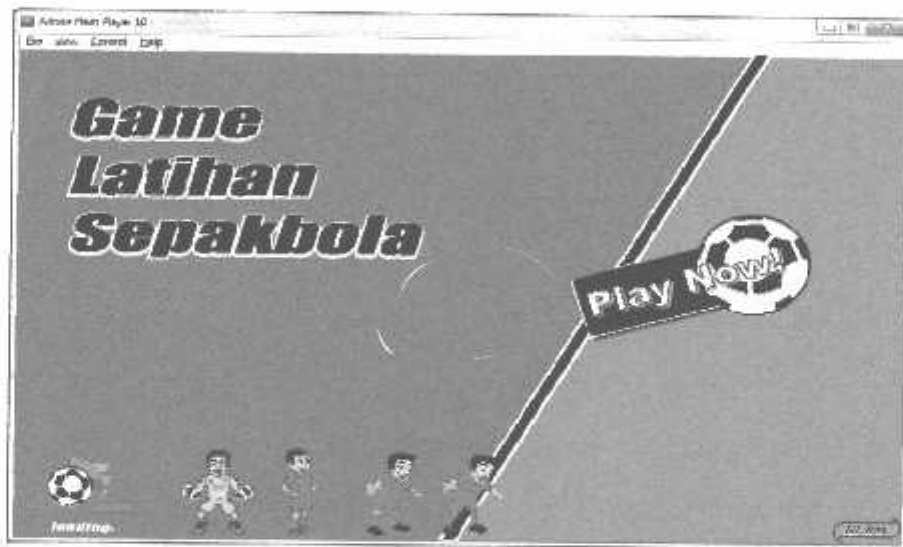
Gambar 4.44 Tampilan *exit game*.

4.2. Pengujian

Pengujian adalah tahapan untuk melihat hasil kesesuaian dari kebutuhan fungsional dan kebutuhan non fungsional pada implementasi *game* latihan sepak bola.

4.2.1 Pengujian Intro Loading

Pengujian *intro loading*, *intro loading* di peruntukkan pada saat awal memulai sebuah *game* di mana *loading* ini untuk menunggu *proses game* yang akan di mainkan. Dalam *intro loading* ini terdapat animasi gerak perputaran *loading* dan gambar grafik gerak serta terdapat animasi gambar pemain melakukan aksi gerak sesuai gaya pemain. Jika *loading* selesai maka akan muncul tombol *play now*.



Gambar. 4.45 Pengujian *intro loading game*.

4.2.2 Pengujian Menu Game

Pengujian menu *game* dalam menu ini terdapat animasi gerak pembuatan arena lapangan sepak bola animasi ini seakan-akan membuat suatu gambar lapangan yang bergerak sampai lapangan sepak bola terlihat utuh. Dalam menu ini terdapat tombol untuk *game* yang akan dimainkan seperti tombol *pinalty* tombol ini untuk menjalankan *game pinalty*, tombol *shooting* untuk memainkan *game shooting*, tombol *heading* tombol ini untuk memainkan

game heading, serta tombol *passing* yang berfungsi untuk memainkan *game passing* dan tombol kuis untuk memainkan *game kuis*.



Gambar 4.46 Tampilan menu *game*.

4.2.3 Pengujian Petunjuk Permainan

Tampilan ini sebagai petunjuk permainan pada saat pemain akan memainkan *game* ini, berikut tampilan petunjuk permainan tiap-tiap *game*.

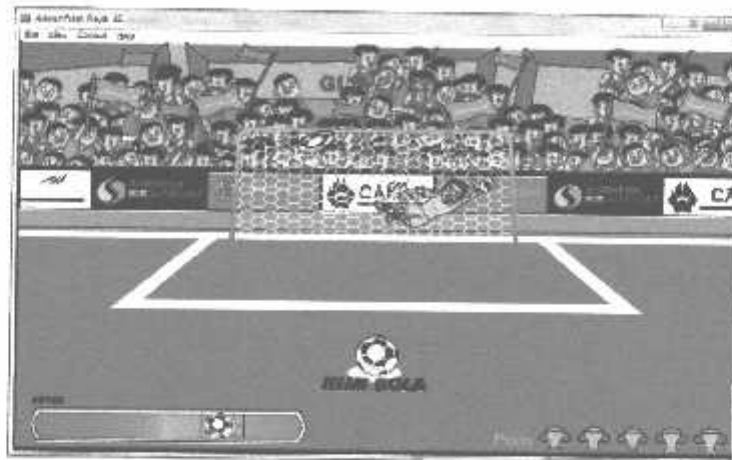


Gambar 4.47 Tampilan Petunjuk permainan.

4.2.4 Pengujian Game Pinalty

Pada pengujian *game pinalty* ini yang di ujikan pertama adalah apakah sasaran yang ditujuk benar benar berfungsi dengan baik. Dan apakah kiper bergerak sesuai dengan lajur bola, *proses* memainkan *game* ini pertama kita menentukan target posisi untuk memasukkan bola ke gawang setelah itu kita

akan menentukan kekuatan tendangan pada indikator *power* setelah menentukan indikator *power*, langkah selanjutnya kita akan menekan tombol aksi untuk menjalankan tendangan ke gawang.



Gambar 4.48 Tampilan pengujian *game penalty*

4.2.5 Pengujian Game Shooting

Tampilan game shooting proses game ini pertama-tama kita akan menentukan indikator *power* jika posisi bola pada indikator pas pada posisi titik tengah maka bola akan tepat mengenai sasaran pada objek bola yang di gantung pada tiang objek, jika posisi tidak tepat pad tengah indikator maka bola tidak akan mengenai sasaran target.



Gambar 4.49 Tampilan pengujian *game shooting*

4.2.6 Pengujian Game Heading

Pada *game heading*, *proses game* ini harus mengenai bola yang di operkan oleh lawan dimana lajur bola ini bergerak secara acak *random*, sehingga pemain di harapkan mengenai sasaran bola yang bergerak secara acak dengan melakukan *heading* bola yang di gerakkan dengan tombol aksi gerak kiri dan kanan dan tombol untuk *heading* menggunakan tombol *space*.



Gambar 4.50 Tampilan pengujian *game heading*.

4.2.7 Pengujian Game Passing

Pada *game passing*, *proses game* ini harus mengenai bola yang di operkan oleh lawan dimana lajur bola ini bergerak secara acak *random*. Sehingga pemain di harapkan mengenai sasaran bola yang bergerak secara acak dengan melakukan *passing* bola yang di gerakkan dengan tombol aksi gerak atas dan bawah dan tombol untuk *passing* menggunakan tombol *space*.



Gambar 4.51 Tampilan Pengujian *game passing*.

4.2.8 Pengujian Game Kuis

Game kuis ini memiliki 20 soal yang di keluarkan secara acak *random* pada saat soal di tampilkan. Cara untuk menjawab dari masing masing soal dilakukan dengan cara memilih tombol a,b,c,d. Setiap soal memiliki nilai 5 poin, dimana nanti poin ini akan di kalkulasi secara keseluruhan. Dalam *game* ini dapat di perbarui soal-soal yang ada dengan mengedit *file* soal yang berformatkan *xml.doc*. *Game* ini juga menampilkan nilai hasil kalkulasi.



Gambar 4.52 Tampilan pengujian *intro game* kuis.



Gambar 4.53 Tampilan pengujian *game* kuis.



Gambar 4.54 Tampilan pengujian total nilai.

4.2.9 Pengujian Tampilan Exit.

Tampilan *exit* merupakan tampilan untuk mengakhiri permainan dari *game* latihan sepak bola,dimana tampilan ini akan muncul pada saat kita akan mengakhiri permainan ini dengan menekan tombol *exit* yang ada pada *game*.Tampilan exit dapat di lihat pada gambar dibawah ini.



Gambar 4.55 Tampilan pengujian *exit game*.

4.3 Pengujian Fungsional

Pengujian fungsional merupakan pengujian mengenai proses fungsional yang terjadi dalam *game*. Hasil dari pengujian dapat dilihat pada tabel 4.2 dibawah ini.

Tabel.4.2 Pengujian Fungsional Game.

No	Fungsi	Hasil
1.	Bagian Intro.	
	a. Apakah animasi intro berjalan dengan baik.	(√)Ya , ()Tidak
	b. Tombol play now apakah berfungsi dengan baik.	(√)Ya , ()Tidak
2.	Bagian Menu.	
	a. Apakah sound pada menu berfungsi dengan baik.	(√)Ya , ()Tidak
	b. Apakah tombo. untuk setiap game berfungsi dengan baik.	(√)Ya , ()Tidak
3.	Game Pinalty.	
	a. Apakah tombol play now berfungsi dengan baik.	(√)Ya , ()Tidak

	b. Apakah tampilan petunjuk permainan berfungsi dengan baik. c. Apakah fungsi kursor untuk menentukan target berfungsi baik. d. Apakah fungsi indicator power berfungsi dengan baik. e. Apakah tombol aksi berfungsi dengan baik. f. Apakah gerak kipper dan gerak bola berjalan dengan baik.	(√)Ya, ()Tidak (√)Ya, ()Tidak (√)Ya, ()Tidak (√)Ya, ()Tidak (√)Ya, ()Tidak
4.	Game Shooting. a. Apakah tombol play now berfungsi dengan baik. b. Apakah tampilan petunjuk permainan berfungsi dengan baik. c. Apakah fungsi indicator power berfungsi dengan baik. c. Apakah tombol aksi berfungsi dengan baik. f. Apakah gerak pemain dan gerak bola berjalan dengan baik.	(√)Ya, ()Tidak (√)Ya, ()Tidak (√)Ya, ()Tidak (√)Ya, ()Tidak (√)Ya, ()Tidak
5.	Game Heading. a. Apakah tombol play now berfungsi dengan baik. b. Apakah tampilan petunjuk permainan berfungsi dengan baik. c. Apakah fungsi ombola rah kanan / kiri berfungsi dengan baik. e. Apakah tombol aksi (space) berfungsi dengan baik. f. Apakah gerak pemain dan gerak bola berjalan dengan baik.	(√)Ya, ()Tidak (√)Ya, ()Tidak (√)Ya, ()Tidak (√)Ya, ()Tidak (√)Ya, ()Tidak
6.	Game Passing. a. Apakah tombol play now berfungsi dengan baik. b. Apakah tampilan petunjuk permainan berfungsi dengan baik. c. Apakah fungsi ombola rah atas / bawah berfungsi dengan baik. c. Apakah tombol aksi (space) berfungsi dengan baik. f. Apakah gerak pemain dan gerak bola berjalan dengan baik.	(√)Ya, ()Tidak (√)Ya, ()Tidak (√)Ya, ()Tidak (√)Ya, ()Tidak (√)Ya, ()Tidak
7.	Game Kuis. a. Apakah Loading kuis berjalan dengan baik. b. Apakah tombol play now berfungsi dengan baik. c. Apakah soal pada game keluar secara acak. d. Apakah tombol jawaban berfungsi dengan baik. e. Apakah tombol pada menu nilai akhir berfungsi dengan baik.	(√)Ya, ()Tidak (√)Ya, ()Tidak (√)Ya, ()Tidak (√)Ya, ()Tidak (√)Ya, ()Tidak

4.3.1 Pengujian Operating Sistem (OS).

Pengujian sistem ini diujikan kepada beberapa jenis OS yang dapat digunakan untuk menjalankan *game* latihan sepak bola .

Berikut beberapa jenis Os yang dapat digunakan untuk menjalankan *game*.

Tabel 4.3. Tabel Pengujian Sistem OS

No	Operating Sistem (O.S)	Hasil	Keterangan
1	Windows Xp	sesuai (√)	Dapat di Gunakan
2	Windows Seven	sesuai (√)	Dapat di Gunakan
3	Windows Vista	sesuai (√)	Dapat di Gunakan

4.3.2 Pengujian Sistem Komputer / Spesifikasi Komputer

Pengujian sistem komputer digunakan sebagai penunjuk spesifikasi system yang di gunakan untuk menjalankan *game* latihan sepak bola. Berikut table pengujian system computer untuk *game* latihan sepak bola.

Tabel 4.4 Pengujian Sistem Komputer /Spesifikasi Komputer

No	Pengujian Sistem Komputer	Hasil	Keterangan
1	Spesifikasi <i>display</i> yang digunakan dalam membangun <i>game</i> , yaitu sebesar 980 x 615 <i>pixel</i> .	sesuai (√)	Game akan dibuat fullscreen sesuai spesifikasi 980 x 615 <i>pixel</i> .
2	Spesifikasi <i>Processor</i> yang digunakan dalam membangun <i>game</i> , yaitu Intel Core i3-350M <i>Processor</i> .	sesuai (√)	Game berjalan dengan normal sesuai fungsional.
3	Spesifikasi <i>RAM</i> yang digunakan dalam membangun <i>game</i> , sebesar 1024 MB.	sesuai (√)	Game berjalan dengan normal sesuai fungsional.
4	Spesifikasi <i>VGA</i> yang digunakan dalam membangun <i>game</i> , yaitu Intel HD Graphics	sesuai (√)	Game berjalan dengan normal sesuai fungsional.

5	<i>Gom Aplikasi</i> yang dapat digunakan sebagai eksekusi /untuk menjalankan game yang berformat <i>file.swf</i>	sesuai (√)	Sesuai format <i>swf</i> dapat dijalankan dengan aplikasi Gom
6	<i>Game</i> dimainkan menggunakan <i>key</i> panah sebagai kontrol pemain. <i>Key Space</i> sebagai Aksi Pemain serta <i>Kursor</i> untuk Penunjuk sasaran.	sesuai (√)	Game berjalan dengan normal sesuai fungsional.
7	<i>Platform</i> yang digunakan dalam membangun <i>game</i> adalah <i>Adobe Flash CS5</i> dengan bahasa pemrograman <i>Action Script 2.0</i> .	sesuai (√)	Pemanfaatan dalam proses perancangan karakter, animasi, dan algoritma <i>FSM</i>
8	<i>Platform</i> yang digunakan dalam <i>editing audio</i> adalah <i>Adobe Premiere Pro CS3</i> .	sesuai (√)	Pemanfaatan dalam proses perancangan audio pada <i>game</i> .
9	Pengujian <i>file.swf</i> . Pengujian <i>file.exe</i>	sesuai (√) sesuai (√)	Game berjalan dengan normal sesuai fungsional pada <i>Adobe Flash Player</i> .

4.3.3 Pengujian User.

Game yang telah dibuat kemudian diujikan kepada *user* untuk dimainkan. Selanjutnya *user* diminta untuk memberikan penilaian kesesuaian, serta komentar *user* terhadap *game* ini. Pengujian dilakukan kepada 10 orang *user* dan penilaian dilakukan dengan cara melakukan pengisian kuesioner oleh masing-masing *user*. Untuk hasil kesesuaian dari setiap *user*, maka akan diberi point 10%. Kuesioner berisi delapan pertanyaan seputar kebutuhan fungsional penilaian sesuai (√) ya, atau tidak sesuai (-) tidak. Menurut 10 orang *user* hasil dari pengujian *user* tersebut dikalkulasikan ke dalam bentuk persentase. Hasil dari pengujian *user* terhadap kebutuhan fungsional secara persentase dapat dilihat pada tabel 4.5

Tabel 4.5 Lembar Pengujian Kuisisioner Untuk User.

No	Kuisisioner	Jawaban		Tanggapan			
1	Apakah game ini sudah memiliki design yang baik dan menarik?	☐ Ya	☐ Tidak	☐ Sangat Menarik	☐ Menarik	☐ Cukup Menarik	☐ Tidak Menarik
2	Apakah game ini mudah digunakan ?	☐ Ya	☐ Tidak	☐ Sangat Menarik	☐ Menarik	☐ Cukup Menarik	☐ Tidak Menarik
3	Menurut Anda bagaimana tingkat kesulitan pada game ini?	☐ Ya	☐ Tidak	☐ Sangat Menarik	☐ Menarik	☐ Cukup Menarik	☐ Tidak Menarik
4	Apakah secara umum game sudah berfungsi dengan baik?	☐ Ya	☐ Tidak	☐ Sangat Menarik	☐ Menarik	☐ Cukup Menarik	☐ Tidak Menarik
5	Apakah game ini mudah dipahami?	☐ Ya	☐ Tidak	☐ Sangat Menarik	☐ Menarik	☐ Cukup Menarik	☐ Tidak Menarik
6	Apakah masing-masing game dapat berfungsi dengan baik?	☐ Ya	☐ Tidak	☐ Sangat Menarik	☐ Menarik	☐ Cukup Menarik	☐ Tidak Menarik
7	Apakah soal-soal dalam game kuis mudah dipahami?	☐ Ya	☐ Tidak	☐ Sangat Menarik	☐ Menarik	☐ Cukup Menarik	☐ Tidak Menarik
8	Saran / Menakikan :						

4.3.4 Hasil Pengujian Kuisisioner

Tabel hasil pengujian kuisisioner digunakan sebagai informasi hasil dari proses pengujian yang dilakukan terhadap 10 user yang melakukan uji pada game latihan sepak bola. Berikut tampilan dari table hasil kuisisioner.

Tabel 4.6 Hasil Kuisisioner User

No	Jawaban	Responden	Prosentase (%)	Tanggapan	Responden	Prosentase (%)
1.	Ya	7	70 %	Sangat Menarik	5	50%
2.	Tidak	3	30%	Menarik	3	30%
3.	-	-	-	Cukup Menarik	2	20%
4.	-	-	-	Tidak Menarik	-	-
Total		10	100%	Total	10	100%

Tabel 4.7 Hasil Kuisisioner Saran / Masukkan.

No	Saran / Masukkan	Responden	Prosentase (%)
1.	Game ini lebih cocok untuk usia pelajar khususnya sekolah dasar (SD).	7	70%
2.	Game khususnya karakter pemain lebih bagus lagi jika di buat 3D.	3	30%
Total		10	100%

BAB V

PENUTUP

5.1. Kesimpulan

Setelah membuat *game* latihan sepak bola ini maka dapat di simpulkan:

1. Berdasarkan hasil pengujian menggunakan metode *finite state machine*, pemain kiper pada *game pinalty* dapat memberikan *respon* terhadap gerak bola.
2. Hasil *publish* pada *game* latihan sepak bola berupa *file.exe* dengan *size* 6,32MB dan *file.swf* secara keseluruhan memiliki *size* 413MB. Semua *file* dari hasil *publish* dapat berjalan dengan normal sesuai fungsional pengujian non fungsional *interface*.
3. Aplikasi *game* ini yang berupa *file.swf* dapat berjalan dengan baik pada saat dijalankan pada aplikasi GOM player.
4. Sistem penilaian dalam *game* ini di tentukan poin dari tiap permainan yang di mainkan setiap poin yang di dapat bernilai 20.
5. Pengujian terhadap *game* latihan sepak bola yang di ujikan pada beberapa kalangan usia pelajar dan mahasiswa menyatakan dari pengujian ini diperoleh hasil *presentase* 70%, secara keseluruhan *game* berjalan dengan baik. Dan *presentase* 50%, *game* dikatakan sangat menarik. Dan didapatkan *presentase* sebesar 30%, *game* ini dikatakan menarik. Serta didapatkan 20% *game* dikatakan cukup menarik. Serta untuk saran dan masukan dari 10 *user* yang mengisi kuisisioner. Terdapat 7 *user* dengan *presentase* 70%, menyatakan *game* ini cocok dimainkan untuk usia pelajar sekolah dasar.

5.2. Saran

Setelah melakukan *proses* pembuatan *game* latihan sepak bola maka pengembangan yang dapat dilakukan :

1. Karakter pemain dalam *game* ini masih menggunakan *design* 2D, diharapkan nantinya dapat di kembangkan menggunakan 3D, akan tampak lebih bagus.
 2. Didalam kategori *game* kuis, jumlah soal hanya terdapat sekitar 20 soal tanya jawab. Diharapkan nantinya jumlah soal yang sudah ada ini bias bertambah lebih banyak lagi soal-soal sebagai pendukung untuk *game* kuis ini.
 3. *Game* ini hanya memiliki 5 kategori yakni *game pinalty*, *game shooting*, *game passing*, *game heading* dan *game* kuis. Pada *game* ini tidak terdapat *game* tendangan bebas, oleh karena itu untuk melengkapi *game* ini diharapkan nantinya ditambahkan kategori *game* baru yakni, *game* tendangan bebas (*free kick*).
-

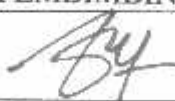
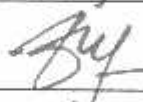
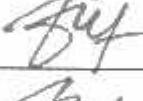
DAFTAR PUSTAKA

- [1] Modul Praktikum Animasi dan Game Teknik Informatika Institut Teknologi Nasional Malang.
- [2] Pengembangan *Finite State Machine* Untuk Memodelkan Agen dan Pergerakannya Pada Olahraga Futsal.
<http://repository.ipb.ac.id/bitstream/handle/123456789/12735/Pendahuluan/G09ffe1.pdf> (diakses pada tanggal 25/05/2013 19:05).
- [3] Dhewiberta H. 2009. *Mahir dalam 7 Hari : Macromedia Flash MX 2004*. Yogyakarta : Andi Offset.
- [4] Radion, Kristo S.ST. 2012. *Easy Game Programming Using Flash and ActionScript 3.0*. Yogyakarta.
- [5] Wandah, 2010. *Dasar Pemrograman Flash Game*, Gava Media, Bandung.
- [6] Widianti, Utami Dewi, 2011. *Pengertian Finite State Machine*.
- [7] Diginovac, 2010, Tip dan Trik Membuat Fitur Game Flash, Elek Media Komputindo, Jakarta.
- [8] Pulung, Nurtantio, Array, Maulana, syafif 2013. *Kreasikan Aksimu Dengan Adobe Flash Dalam Membuat Sistem Multimedia Interaktif*, Penerbit Andy, Yogyakarta.

LAMPIRAN

FORMULIR BIMBINGAN SKRIPSI

Nama : Alexander Taka Logo
NIM : 0918002
Masa Bimbingan : 11 Mei 2013 s/d 11 November 2013
Judul Skripsi : RANCANG BANGUN PEMBUATAN GAME LATIHAN
SEPAK BOLA MENGGUNAKAN METODE FINITE STATE
MACHINE (FSM)

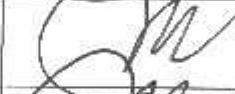
NO	TANGGAL	URAIAN	PARAF PEMBIMBING
1.	05 / 07 / 2013	Demo Game (Revisi Tampilan Font,Pada Menu diganti)	
2.	10 / 07 / 2013	Demo Game (Tampilan Font Menu)	
3.	25 / 07 / 2013	Makalah Seminar Hasil	
4.	31 / 07 / 2013	Demo Game Keseluruhan	
5.	14 / 08 / 2013	Acc Laporan Skripsi Ujian Kompre	
6.			
7.			
8.			
9.			
10.			

Malang, Agustus 2013
Dosen Pembimbing I


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

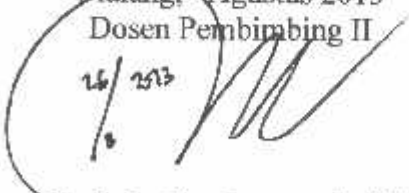
FORMULIR BIMBINGAN SKRIPSI

Nama : Alexander Taka Logo
 NIM : 0918002
 Masa Bimbingan : 11 Mei 2013 s/d 11 November 2013
 Judul Skripsi : RANCANG BANGUN PEMBUATAN GAME LATIHAN
 SEPAK BOLA MENGGUNAKAN METODE FINITE STATE
 MACHINE (FSM)

NO	TANGGAL	URAIAN	PARAF PEMBIMBING
1.	05 / 06 / 2013	Konsultasi Metode Yang digunakan	
2.	12 / 06 / 2013	Bab I, Bab II Revisi	
3.	17 / 06 / 2013	Bab I, Bab II, Acc.	
4.	20 / 06 / 2013	Konsultasi Bab III. (Revisi)	
5.	25 / 06 / 2013	Bab III Acc	
6.	08 / 07 / 2013	Bab IV , Bab V Revisi. (Revisi Pengujian, Flowchart, Kuisisioner)	
7.	15 / 07 / 2013	Bab IV, Bab V Acc	
8.	23 / 07 / 2013	Makalah Revisi	
9.	13/08/2013	Acc.Lap Skripsi Ujian Kompre	
10.			

Malang, Agustus 2013
 Dosen Pembimbing II

16/ 2013


Nurlaily Vendyansyah, ST.



FORMULIR PERBAIKAN UJIAN SKRIPSI

Dalam pelaksanaan Ujian Skripsi Jenjang Strata 1 Jurusan Teknik Informatika, maka perlu adanya perbaikan untuk mahasiswa :

Nama : Alexander Taha Logo
NIM : 09.18.002
Perbaikan Melalui :

Pengujian (Bab IV) : Kelompokkan pengujian dalam
- Pengujian Fungsional
- Pengujian sistem (Hardware / platform)
- Pengujian user

Kesimpulan → didasarkan pada hasil pengujian (rangkuman)

Malang, 16 Agustus 2013


(Michael Ardita)



INSTITUT TEKNOLOGI NASIONAL MALANG
Fakultas Teknologi Industri
Program Studi Teknik Informatika S1

FORMULIR PERBAIKAN UJIAN SKRIPSI

Dalam pelaksanaan Ujian Skripsi Jenjang Strata 1 Jurusan Teknik Informatika, maka perlu adanya perbaikan untuk mahasiswa :

Nama : Aletanor T.L.
NIM : _____
Perbaikan Meliputi : _____

- Indikator power diusahakan lebih tepat ✓
- Flowchart agar shock ✓
- ~~Penyusunan~~ Prinsip ✓
- ~~Penyusunan~~ Met U.S ✓
- ~~Penyusunan~~ Met Identifikasi
- ~~Penyusunan~~ Met Responden diusahakan di komputer ✓
- ~~Penyusunan~~ Met Pengujian sistem komputer ✓
- Abstrak disertai hasil kesimpulan dalam bentuk pengkasan ✓
- Laporan print ulang.

Malang,

(Suryo Adh)



INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
PROGRAM STUDI TEKNIK INFORMATIKA S-1
Jl. Karanglo, KM 2 Malang

FORMULIR PERBAIKAN SKRIPSI

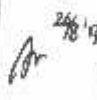
Dalam pelaksanaan ujian skripsi jenjang Strata 1 Program Studi Teknik Informatika, maka perlu adanya perbaikan skripsi untuk mahasiswa :

NAMA : Alexander Taka Logo

NIM : 0918002

JURUSAN : Teknik Informatika S-1

JUDUL : **RANCANG BANGUN PEMBUATAN GAME LATIHAN SEPAK BOLA MENGGUNAKAN METODE FINITE STATE MACHINE (FSM)**

No	Penguji	Tanggal	Uraian	Paraf
1.	Penguji I	16 Agustus 2013	1. Pengujian (Bab IV) Kelompokkan pengujian dalam. Pengujian Fungsional. Pengujian Sistem (Hardware/Platform). Pengujian User. 2. Kesimpulan didasarkan pada hasil pengujian (rangkuman).	 26/8/13
2.	Penguji II	16 Agustus 2013	1. Indikator power diusahakan lebih tepat. 2. Flowchart FSM game penalty. 3. Flowchart FSM game shooting. 4. Pengujian terhadap OS. 5. Pengujian terhadap responden di masukkan ke dalam kesimpulan 6. Pengujian system computer. 7. Abstrak disertai hasil kesimpulan dalam bentuk ringkasan. 8. Laporan print warna	

Anggota Penguji

Dosen Penguji I

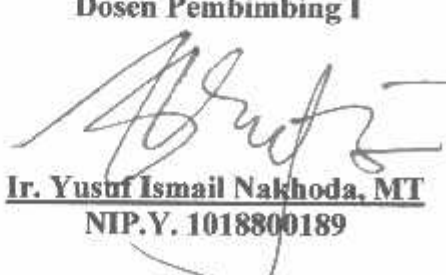

Michael Ardita.ST,MT.
NIP.P. 1031000434

Dosen Penguji II


Suryo Adi Wibowo.ST,MT.
NIP.P. 1031000438

Mengetahui

Dosen Pembimbing I


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

Dosen Pembimbing II


Nurjaily Vandyansyah, ST



INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
PROGRAM STUDI TEKNIK INFORMATIKA S-1
Jl. Karanglo, KM 2 Malang

**BERITA ACARA UJIAN SKRIPSI
FAKULTAS TEKNOLOGI INDUSTRI**

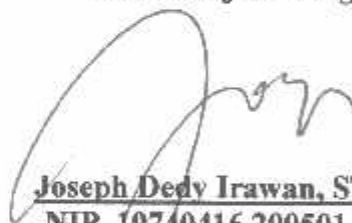
NAMA : Alexander Taka Logo
NIM : 0918002
JURUSAN : Teknik Informatika S-1
JUDUL : **RANCANG BANGUN PEMBUATAN GAME LATIHAN SEPAK
BOLA MENGGUNAKAN METODE FINITE STATE MACHINE
(FSM)**

Dipertahankan dihadapan Majelis Penguji Skripsi Jenjang Strata Satu (S-1) pada :

Hari : Sabtu
Tanggal : 16 Agustus 2013
Nilai : 88.07 (A)

Panitia Ujian Skripsi :

Ketua Majelis Penguji



Joseph Dedy Irawan, ST, MT
NIP. 19740416 200501 1 002

Anggota Penguji :

Dosen Penguji I



Michael Ardita, ST, MT.
NIP.P. 1031000434

Dosen Penguji II



Suryo Adi Wibowo, ST, MT.
NIP.P. 1031000438

- Script kuis:

```
stop();
//---buat function array acakaduk
Array.prototype.acakaduk = function() {
  for (i=0; i<this.length; i++) {
    var tmp = this[i];
    var acakaduk = random(this.length);
    this[i] = this[acakaduk];
    this[acakaduk] = tmp;
  }
};
var array_soal = [];
var array_tanya = [];
var array_pilihan_a = [];
var array_pilihan_b = [];
var array_pilihan_c = [];
var array_pilihan_d = [];
var array_jwb = [];
var jawabku;
var nobaru;
var totalsoal;
var limit_soal;
clearInterval(_root.brs);
//----tombol jawab pilihan dan cek jawaban
_root.tombol_a_btn.onRelease = function() {
  _root.jawabku = "a";
  cek_jwb();
  _root.nobaru++;
  tampil_soal(_root.nobaru);
};
_root.tombol_b_btn.onRelease = function() {
  _root.jawabku = "b";
  cek_jwb();
  _root.nobaru++;
  tampil_soal(_root.nobaru);
};
_root.tombol_c_btn.onRelease = function() {
  _root.jawabku = "c";
  cek_jwb();
  _root.nobaru++;
  tampil_soal(_root.nobaru);
};
_root.tombol_d_btn.onRelease = function() {
  _root.jawabku = "d";
  cek_jwb();
  _root.nobaru++;
  tampil_soal(_root.nobaru);
};
```

```

//---nilai awal(mengambil data dari XML)
mysoal = new XML();
mysoal.ignoreWhite = true;
mysoal.load("Soal.xml");
mysoal.onLoad = function() {
    _root.limit_soal = mysoal.firstChild.attributes.jml;
    _root.totalsoal = mysoal.firstChild.childNodes.length;
    _root.nomorsoal_txt.text = "0";
    _root.benar = 0;
    _root.nilai_txt.text = "0";
    for (i=0; i<_root.totalsoal; i++) {
        _root.array_soal[i] = i;
    }
    _root.array_soal.acakaduk();
    for (j=0; j<_root.limit_soal; j++) {
        _root.array_tanya[j] =
            mysoal.firstChild.childNodes[_root.array_soal[j]].childNodes[0].childNodes[0].toString();
        _root.array_pilihan_a[j] =
            mysoal.firstChild.childNodes[_root.array_soal[j]].childNodes[1].childNodes[0].nodeValue;
        _root.array_pilihan_b[j] =
            mysoal.firstChild.childNodes[_root.array_soal[j]].childNodes[2].childNodes[0].nodeValue;
        _root.array_pilihan_c[j] =
            mysoal.firstChild.childNodes[_root.array_soal[j]].childNodes[3].childNodes[0].nodeValue;
        _root.array_pilihan_d[j] =
            mysoal.firstChild.childNodes[_root.array_soal[j]].childNodes[4].childNodes[0].nodeValue;
        _root.array_jwb[j] =
            mysoal.firstChild.childNodes[_root.array_soal[j]].childNodes[5].childNodes[0].nodeValue;
    }
    tampil_soal(0);

};
//---menampilkan soal dari Array
function tampil_soal(x):Void {
    clearInterval(_root.brs);
    _root.brs = setInterval(counter, 1000);
    if (x<array_tanya.length) {
        _root.soal_txt.text = _root.array_tanya[x];
        _root.pilihanjwb_a_txt.text = _root.array_pilihan_a[x];
        _root.pilihanjwb_b_txt.text = _root.array_pilihan_b[x];
        _root.pilihanjwb_c_txt.text = _root.array_pilihan_c[x];
        _root.pilihanjwb_d_txt.text = _root.array_pilihan_d[x];
        _root.jawaban = _root.array_jwb[x];
        _root.nobaru = x;
        _root.nomorsoal_txt.text = x+1+" ";
    }
}

```

```

    } else {
        hitung_nilai();
        _root.gotoAndStop(2);
    }
}

//----cek jawaban
function cek_jwb():Void {
    //--membandingkan antara jawabku dengan jawaban
    if (_root.jawabku == _root.jawaban) {
        //-- bila benar maka nilai bertambah
        _root.benar += 5;
        _root.nilai_txt.text = _root.benar;
    }
}

//----hitung nilai akhir
function hitung_nilai():Void {
    _root.nilaiakhir = Math.ceil((100/ _root.array_tanya.length)* _root.benar);
}

```

```

2).
stop();
//----kembali ke Soal Latihan
_root.cobalagi_btn.onRelease = function() {
    gotoAndStop(1);
};
//----Tampil nilai akhir
clearInterval(_root.brs);
_root.nilaiakhir_txt.text = _root.nilaiakhir;

```

- Script game penalty

```

function lastStep()
{
    energybarMc = true;
    ballBtn.enabled = true;
    ballMove = true;
    speed = random(11) + 15;
} // End of the function
function Result()
{
    trace ("goal1 = " + goal1);
    trace ("goal2 = " + goal2);
    trace ("goal3 = " + goal3);
    trace ("goal4 = " + goal4);
    trace ("goal5 = " + goal5);
    welldoneMc.gotoAndPlay("over");
    ballpower._visible = false;
} // End of the function

```

```

function playAgain()
{
    pointMc.gotoAndStop(1);
    kipperMc._visible = true;
    ballpower._visible = true;
    firstStep = true;
    goal1 = "";
    goal2 = "";
    goal3 = "";
    goal4 = "";
    goal5 = "";
    totalGoal = 0;
    startGame.gotoAndPlay(2);
}
stop ();
var firstStep = true;
var selectedArea;
var ballCurrent = ballpower._x;
var ballStartX = power._x;
var ballEndX = power._width;
var speed = random(6) + 10;
var ballMove = false;
var LeftRight = "right";
ballBtn.enabled = false;
var onePrecent = int(power._width / 100);
var getStop;
var energybarMc;
var goalOrMiss;
var goal1;
var goal2;
var goal3;
var goal4;
var goal5;
var totalGoal = 0;
var mouseShowHide = true;
onEnterFrame = function ()
{
    if (firstStep == true)
    {
        energybarMc = false;
        if (mouseShowHide == true)
        {
            Mouse.hide();
            kursor._visible = true;
            kursor._x = _root._xmouse;
            kursor._y = _root._ymouse;
        }
        else
        {
            Mouse.show();

```

```

        kursor._visible = false;
        kursor._visible = false;
    } // end else if
    target_1.enabled = true;
    target_2.enabled = true;
    target_3.enabled = true;
    target_4.enabled = true;
    target_5.enabled = true;
    target_6.enabled = true;
    target_7.enabled = true;
    target_8.enabled = true;
    target_9.enabled = true;
    target_1.onRelease = function ()
    {
        selectedArea = "1";
        lastStep();
        firstStep = false;
    };
    target_2.onRelease = function ()
    {
        selectedArea = "2";
        lastStep();
        firstStep = false;
    };
    target_3.onRelease = function ()
    {
        selectedArea = "3";
        lastStep();
        firstStep = false;
    };
    target_4.onRelease = function ()
    {
        selectedArea = "4";
        lastStep();
        firstStep = false;
    };
    target_5.onRelease = function ()
    {
        selectedArea = "5";
        lastStep();
        firstStep = false;
    };
    target_6.onRelease = function ()
    {
        selectedArea = "6";
        lastStep();
        firstStep = false;
    };
    target_7.onRelease = function ()
    {

```

```

        selectedArea = "7";
        lastStep();
        firstStep = false;
    };
    target_8.onRelease = function ()
    {
        selectedArea = "8";
        lastStep();
        firstStep = false;
    };
    target_9.onRelease = function ()
    {
        selectedArea = "9";
        lastStep();
        firstStep = false;
    };
}
else
{
    Mouse.show();
    kursor._visible = false;
    target_1.enabled = false;
    target_2.enabled = false;
    target_3.enabled = false;
    target_4.enabled = false;
    target_5.enabled = false;
    target_6.enabled = false;
    target_7.enabled = false;
    target_8.enabled = false;
    target_9.enabled = false;
} // end else if
if (energybarMc == true)
{
    power._visible = true;
    ballBtn._visible = true;
    ballpower._visible = true;
}
else
{
    power._visible = false;
    ballBtn._visible = false;
    ballpower._visible = false;
} // end else if
if (ballMove == true)
{
    switch (LeftRight)
    {
        case "right":
        {
            if (ballpower._x >= ballEndX)

```

```

        {
            LeftRight = "left";
        } // end if
        ballpower._x = ballpower._x + speed;
        break;
    }
    case "left":
    {
        if (ballpower._x <= ballStartX)
        {
            LeftRight = "right";
        } // end if
        ballpower._x = ballpower._x - speed;
        break;
    }
} // End of switch
} // end if
getStop = int(ballpower._x / onePrecent - 10);
};
ballBtn.onPress = function ()
{
    trace (getStop);
    ballMove = false;
    if (getStop <= 25)
    {
        goalOrMiss = "low";
    } // end if
    if (getStop > 25 && getStop <= 55)
    {
        goalOrMiss = "low";
    } // end if
    if (getStop > 55 && getStop <= 70)
    {
        goalOrMiss = "hit";
    } // end if
    if (getStop > 70 && getStop <= 85)
    {
        goalOrMiss = "goal";
    } // end if
    if (getStop > 85 && getStop <= 90)
    {
        goalOrMiss = "hit";
    } // end if
    if (getStop > 90)
    {
        goalOrMiss = "high";
    } // end if
    switch (goalOrMiss)
    {
        case "goal":

```

```

{
    switch (selectedArea)
    {
        case "1":
        {
            kipperMc.gotoAndStop("1_score");
            break;
        }
        case "2":
        {
            kipperMc.gotoAndStop("2_score");
            break;
        }
        case "3":
        {
            kipperMc.gotoAndStop("3_score");
            break;
        }
        case "4":
        {
            kipperMc.gotoAndStop("4_score");
            break;
        }
        case "5":
        {
            kipperMc.gotoAndStop("5_score");
            break;
        }
        case "6":
        {
            kipperMc.gotoAndStop("6_score");
            break;
        }
        case "7":
        {
            kipperMc.gotoAndStop("7_score");
            break;
        }
        case "8":
        {
            kipperMc.gotoAndStop("8_score");
            break;
        }
        case "9":
        {
            kipperMc.gotoAndStop("9_score");
            break;
        }
    }
    // End of switch
    break;
}

```

```

}
case "low":
{
    switch (selectedArea)
    {
        case "1":
        {
            kipperMc.gotoAndStop("Left_low");
            break;
        }
        case "2":
        {
            kipperMc.gotoAndStop("Center_low");
            break;
        }
        case "3":
        {
            kipperMc.gotoAndStop("Right_low");
            break;
        }
        case "4":
        {
            kipperMc.gotoAndStop("Left_low");
            break;
        }
        case "5":
        {
            kipperMc.gotoAndStop("Center_low");
            break;
        }
        case "6":
        {
            kipperMc.gotoAndStop("Right_low");
            break;
        }
        case "7":
        {
            kipperMc.gotoAndStop("Left_low");
            break;
        }
        case "8":
        {
            kipperMc.gotoAndStop("Center_low");
            break;
        }
        case "9":
        {
            kipperMc.gotoAndStop("Right_low");
            break;
        }
    }
}

```

```

    } // End of switch
    break;
}
case "hit":
{
    switch (selectedArea)
    {
        case "1":
        {
            kipperMc.gotoAndStop("1_hit");
            break;
        }
        case "2":
        {
            kipperMc.gotoAndStop("2_hit");
            break;
        }
        case "3":
        {
            kipperMc.gotoAndStop("3_hit");
            break;
        }
        case "4":
        {
            kipperMc.gotoAndStop("4_hit");
            break;
        }
        case "5":
        {
            kipperMc.gotoAndStop("5_hit");
            break;
        }
        case "6":
        {
            kipperMc.gotoAndStop("6_hit");
            break;
        }
        case "7":
        {
            kipperMc.gotoAndStop("7_hit");
            break;
        }
        case "8":
        {
            kipperMc.gotoAndStop("8_hit");
            break;
        }
        case "9":
        {
            kipperMc.gotoAndStop("9_hit");

```

```

        break;
    }
} // End of switch
break;
}
case "high":
{
    switch (selectedArea)
    {
        case "1":
        {
            kipperMc.gotoAndStop("1_high");
            break;
        }
        case "2":
        {
            kipperMc.gotoAndStop("2_high");
            break;
        }
        case "3":
        {
            kipperMc.gotoAndStop("3_high");
            break;
        }
        case "4":
        {
            kipperMc.gotoAndStop("4_high");
            break;
        }
        case "5":
        {
            kipperMc.gotoAndStop("2_high");
            break;
        }
        case "6":
        {
            kipperMc.gotoAndStop("6_high");
            break;
        }
        case "7":
        {
            kipperMc.gotoAndStop("7_high");
            break;
        }
        case "8":
        {
            kipperMc.gotoAndStop("2_high");
            break;
        }
        case "9":

```

```

        {
            kipperMc.gotoAndStop("9_high");
            break;
        }
    } // End of switch
    break;
}
} // End of switch
ballBtn.enabled = false;
};

```

Game shooting:

- Script Game shooting:

```

function startGame()
{
    ballBtn.enabled = true;
    ballMove = true;
    speed = random(10) + 15;
}
function playAgain()
{
    score1 = "";
    score2 = "";
    score3 = "";
    score4 = "";
    score5 = "";
    scoreOrMiss = "";
    totalScore = 0;
    playerMc._visible = true;
    power._visible = true;
    ballBtn._visible = true;
    ballpower._visible = true;
    welldoneMc.gotoAndStop(1);
    pointMc.gotoAndStop(1);
    startGameMc.gotoAndPlay(2);
}
stop ();
var score1;
var score2;
var score3;
var score4;
var score5;
var scoreOrMiss;
var totalScore = 0;
var ballCurrent = ballpower._x;
var ballStartX = power._x;
var ballEndX = power._width;

```

```

var speed = random(6) + 10;
var ballMove = false;
var LeftRight = "right";
ballBtn.enabled = false;
var onePrecent = int(power._width / 100);
var getStop;
ballpower._x = ballStartX;
score=0;
onEnterFrame = function ()
{
    if (ballMove == true)
    {
        switch (LeftRight)
        {
            case "right":
            {
                if (ballpower._x >= ballEndX)
                {
                    LeftRight = "left";
                }
                ballpower._x = ballpower._x + speed;
                break;
            }
            case "left":
            {
                if (ballpower._x <= ballStartX)
                {
                    LeftRight = "right";
                }
                ballpower._x = ballpower._x - speed;
                break;
            }
        }
    }
    getStop = int(ballpower._x / onePrecent - 10);
};
ballBtn.onPress = function ()
{
    trace (getStop);
    ballMove = false;
    if (getStop <= 25)
    {
        scoreOrMiss = "gagal";
        playerMc.gotoAndStop(2);
    }
    if (getStop > 25 && getStop <= 55)
    {
        scoreOrMiss = "gagal";
        playerMc.gotoAndStop(3);
    }
}

```

```

if (getStop > 55 && getStop <= 70)
{
    scoreOrMiss = "gagal";
    playerMc.gotoAndStop(4);
}
if (getStop > 70 && getStop <= 80)
{
    scoreOrMiss = "score";
    playerMc.gotoAndStop(5);
    //score=score+1;
}
if (getStop > 80 && getStop <= 90)
{
    scoreOrMiss = "gagal";
    playerMc.gotoAndStop(6);
}
if (getStop > 90)
{
    scoreOrMiss = "gagal";
    playerMc.gotoAndStop(7);
}

//if (score>=3){

//}
//else{

//}
ballBtn.enabled = false;
};

```

Game Heading.

- Script game heading:

```

function playAgain()
{
    score1 = "";
    score2 = "";
    score3 = "";
    score4 = "";
    score5 = "";
    scoreOrMiss = "";
    totalScore = 0;
    playerMc._visible = true;
    powerbarMc._visible = true;
    passBtn._visible = true;
    ballMc._visible = true;
    playerStartgame._visible = true;
}

```

```

        welldoneMc.gotoAndStop(1);
        pointMc.gotoAndStop(1);
        startGame.gotoAndPlay(2);
    } // End of the function
stop ();
var ballStartX = ballMc._x;
var ballStartY = ballMc._y;
var score1;
var score2;
var score3;
var score4;
var score5;
var scoreOrMiss;
var totalScore = 0;

```

Game Passing.

- Script game passing.

```

function playAgain()
{
    playerMc._visible = true;
    powerbarMc._visible = true;
    passBtn._visible = true;
    ballMc._visible = true;
    playerStartgame._visible = true;
    pointMc.gotoAndStop(1);
    welldoneMc.gotoAndStop(1);
    score1 = "";
    score2 = "";
    score3 = "";
    score4 = "";
    score5 = "";
    scoreOrMiss = "";
    totalScore = 0;
    startGame.gotoAndPlay(2);
} // End of the function
stop ();
var ballStartX = ballMc._x;
var ballStartY = ballMc._y;
var score1;
var score2;
var score3;
var score4;
var score5;
var scoreOrMiss;
var totalScore = 0;

```

Nama : *Nafa Andia*
 Keterangan : *SKOLAH DATA*

No	Kuis/soal	Jawaban		Tanggapan			
		☐ Ya	☐ Tidak	☐ Sangat Menarik	☐ Menarik	☐ Cukup Menarik	☐ Tidak Menarik
1	Apakah game ini sudah memiliki desain yang baik dan menarik?	☒ Ya	☐ Tidak	☒ Sangat Menarik	☐ Menarik	☐ Cukup Menarik	☐ Tidak Menarik
2	Apakah game ini mudah digunakan?	☒ Ya	☐ Tidak	☒ Sangat Menarik	☐ Menarik	☐ Cukup Menarik	☐ Tidak Menarik
3	Menurut Anda bagaimana tingkat kesulitan pada game ini?	☒ Ya	☐ Tidak	☒ Sangat Menarik	☐ Menarik	☐ Cukup Menarik	☐ Tidak Menarik
4	Apakah secara umum game sudah berfungsi dengan baik?	☒ Ya	☐ Tidak	☐ Sangat Menarik	☒ Menarik	☐ Cukup Menarik	☐ Tidak Menarik
5	Apakah game ini mudah di pahami?	☒ Ya	☐ Tidak	☒ Sangat Menarik	☐ Menarik	☐ Cukup Menarik	☐ Tidak Menarik
6	Apakah masing-masing game dapat berfungsi dengan baik?	☒ Ya	☐ Tidak	☒ Sangat Menarik	☐ Menarik	☐ Cukup Menarik	☐ Tidak Menarik
7	Apakah soal-soal dalam game kuis mudah di pahami?	☒ Ya	☐ Tidak	☒ Sangat Menarik	☐ Menarik	☐ Cukup Menarik	☐ Tidak Menarik
8	Saran / Masukan :	<i>Lucu. Color Bunt Aka</i>					

Name: Wahidul Karim

Keterangan :

No	Kuesioner	Jawaban	Tanggapan
1	Apakah game ini sudah memiliki design yang baik dan menarik?	(☒) Ya	(☒) Sangat Menarik
2	Apakah game ini mudah digunakan?	(☒) Ya	(☒) Sangat Menarik
3	Menurut Anda bagaimana tingkat kesulitan pada game ini?	(☒) Ya	(☒) Sangat Menarik
4	Apakah secara umum game sudah berfungsi dengan baik?	(☒) Ya	(☒) Sangat Menarik
5	Apakah game ini mudah di pahami?	(☒) Ya	(☒) Sangat Menarik
6	Apakah masing-masing game dapat berfungsi dengan baik?	(☒) Ya	(☒) Sangat Menarik
7	Apakah soal-soal dalam game kuis mudah di pahami?	(☒) Ya	(☒) Sangat Menarik
8	Saran / Masukan :	Kurang banyak soal SD, (penting)	

Nama: Mai fimo

Keterangan:

No	Kuisloner	Jawaban	Tanggapan
1	Apakah game ini sudah memiliki desain yang baik dan menarik?	(☒) Ya	(☐) Tidak
2	Apakah game ini mudah dimainkan?	(☒) Ya	(☐) Tidak
3	Menurut Anda bagaimana tingkat kesulitan pada game ini?	(☒) Ya	(☐) Tidak
4	Apakah secara umum game sudah berfungsi dengan baik?	(☒) Ya	(☐) Tidak
5	Apakah game ini mudah di pahami?	(☒) Ya	(☐) Tidak
6	Apakah musik-musik game dapat berfungsi dengan baik?	(☒) Ya	(☐) Tidak
7	Apakah soal-soal dalam game kuis mudah di pahami?	(☒) Ya	(☐) Tidak
8	Saran / Masukan :	Bikin game ini dan Award dan Daring SD	

Nama : *Akhyani*
Keterangan :

No	Kuesioner	Jawaban	Tanggapan
1	Apakah game ini sudah memiliki desain yang baik dan menarik?	☒ Ya	☒ Sangat Menarik
2	Apakah game ini mudah digunakan?	☒ Ya	☒ Sangat Menarik
3	Menurut Anda bagaimana tingkat kesulitan pada game ini?	☐ Ya	☒ Sangat Menarik
4	Apakah secara umum game sudah berfungsi dengan baik?	☒ Ya	☒ Sangat Menarik
5	Apakah game ini mudah di pahami?	☒ Ya	☒ Sangat Menarik
6	Apakah masing-masing game dapat berfungsi dengan baik?	☒ Ya	☒ Sangat Menarik
7	Apakah soal-soal dalam game kuis mudah di pahami?	☐ Ya	☒ Tidak
8	Saran / Masukan :	Balik 30.	

Olif

बोध :

No	Kuesioner	Jawaban	Tanggapan
1	Apakah game ini sudah memiliki design yang baik dan menarik?	(☒) Ya	(☐) Tidak Menarik (☐) Cukup Menarik (☐) Tidak Menarik
2	Apakah game ini mudah digunakan?	(☒) Ya	(☐) Tidak Menarik (☐) Cukup Menarik (☐) Tidak Menarik
3	Menurut Anda bagaimana tingkat kesulitan pada game ini?	(☒) Ya	(☐) Tidak Menarik (☐) Cukup Menarik (☐) Tidak Menarik
4	Apakah secara umum game sudah berfungsi dengan baik?	(☒) Ya	(☐) Tidak Menarik (☐) Cukup Menarik (☐) Tidak Menarik
5	Apakah game ini mudah di pahami?	(☐) Ya	(☒) Tidak Menarik (☐) Cukup Menarik (☐) Tidak Menarik
6	Apakah masing-masing game dapat berfungsi dengan baik?	(☒) Ya	(☐) Tidak Menarik (☐) Cukup Menarik (☐) Tidak Menarik
7	Apakah soal-soal dalam game kuis mudah di pahami?	(☒) Ya	(☐) Tidak Menarik (☐) Cukup Menarik (☐) Tidak Menarik
8	Saran / Masukan :	Mas. Lolat Buai Adek Aka Yg Masih Sekolah SD Kls 5-	

Nama: Bobby

Keterangan:

No	Kuesioner	Jawaban					
		Tanggapan					
1	Apakah game ini sudah memiliki design yang baik dan menarik?	(☒) Ya	(☐) Tidak	(☐) Sangat Menarik	(☒) Menarik	(☐) Cukup Menarik	(☐) Tidak Menarik
2	Apakah game ini mudah digunakan?	(☒) Ya	(☐) Tidak	(☐) Sangat Menarik	(☒) Menarik	(☐) Cukup Menarik	(☐) Tidak Menarik
3	Menurut Anda bagaimana tingkat kesulitan pada game ini?	(☒) Ya	(☐) Tidak	(☐) Sangat Menarik	(☒) Menarik	(☐) Cukup Menarik	(☐) Tidak Menarik
4	Apakah secara umum game sudah berfungsi dengan baik?	(☒) Ya	(☐) Tidak	(☐) Sangat Menarik	(☒) Menarik	(☐) Cukup Menarik	(☐) Tidak Menarik
5	Apakah game ini mudah di pahami?	(☒) Ya	(☐) Tidak	(☐) Sangat Menarik	(☒) Menarik	(☐) Cukup Menarik	(☐) Tidak Menarik
6	Apakah masing-masing game dapat berfungsi dengan baik?	(☒) Ya	(☐) Tidak	(☐) Sangat Menarik	(☒) Menarik	(☐) Cukup Menarik	(☐) Tidak Menarik
7	Apakah soal-soal dalam game kuis mudah di pahami?	(☒) Ya	(☐) Tidak	(☐) Sangat Menarik	(☒) Menarik	(☒) Cukup Menarik	(☐) Tidak Menarik
8	Saran / Masukan :	Game 3D Karakternya baru					

Mama: Silwanfo

Keterangan :

No	KuisKoster	Jawaban	Tanggapan
1	Apakah game ini sudah memiliki design yang baik dan menarik?	☐ Ya ☐ Tidak	☒ Sangat Menarik ☐ Cukup Menarik ☐ Tidak Menarik
2	Apakah game ini mudah digunakan?	☐ Ya ☐ Tidak	☒ Menarik ☐ Cukup Menarik ☐ Tidak Menarik
3	Menurut Anda bagaimana tingkat kesulitan pada game ini?	☐ Ya ☐ Tidak	☒ Menarik ☐ Cukup Menarik ☐ Tidak Menarik
4	Apakah secara umum game sudah berfungsi dengan baik?	☐ Ya ☐ Tidak	☒ Menarik ☐ Cukup Menarik ☐ Tidak Menarik
5	Apakah game ini mudah di pahami?	☐ Ya ☐ Tidak	☒ Menarik ☐ Cukup Menarik ☐ Tidak Menarik
6	Apakah masing-masing game dapat berfungsi dengan baik?	☐ Ya ☐ Tidak	☒ Menarik ☐ Cukup Menarik ☐ Tidak Menarik
7	Apakah soal-soal dalam game kuis mudah di pahami?	☐ Ya ☐ Tidak	☒ Menarik ☐ Cukup Menarik ☐ Tidak Menarik

Saran / Masukan :
Game w/ Mahadine Kuang pas Cricet w/ Anot SD

Nama : *Firman Bayu.*
Keterangan :

No	Kuesioner	Jawaban	Tanggapan				
1	Apakah game ini sudah memiliki design yang baik dan menarik?	(☒) Ya	(☐) Tidak	(☐) Sangat Menarik	(☒) Menarik	(☐) Cukup Menarik	(☐) Tidak Menarik
2	Apakah game ini mudah digunakan?	(☒) Ya	(☐) Tidak	(☐) Sangat Menarik	(☒) Menarik	(☐) Cukup Menarik	(☐) Tidak Menarik
3	Menurut Anda bagaimana tingkat kesulitan pada game ini?	(☒) Ya	(☐) Tidak	(☐) Sangat Menarik	(☒) Menarik	(☐) Cukup Menarik	(☐) Tidak Menarik
4	Apakah secara umum game sudah berfungsi dengan baik?	(☒) Ya	(☐) Tidak	(☐) Sangat Menarik	(☒) Menarik	(☐) Cukup Menarik	(☐) Tidak Menarik
5	Apakah game ini mudah di pahami?	(☒) Ya	(☐) Tidak	(☐) Sangat Menarik	(☒) Menarik	(☐) Cukup Menarik	(☐) Tidak Menarik
6	Apakah masing-masing game dapat berfungsi dengan baik?	(☒) Ya	(☐) Tidak	(☐) Sangat Menarik	(☒) Menarik	(☐) Cukup Menarik	(☐) Tidak Menarik
7	Apakah soal-soal dalam game kuis mudah di pahami?	(☒) Ya	(☐) Tidak	(☐) Sangat Menarik	(☒) Menarik	(☐) Cukup Menarik	(☐) Tidak Menarik
8	Saran / Masukan :	Catat untuk Anak SD. game nya					

Nama : Krisnanto.
Keterangan :

No	Kuisiomer	Jawaban	Tanggapan
1	Apakah game ini sudah memiliki design yang baik dan menarik?	() Ya (x) Tidak	() Sangat Menarik () Menarik (x) Cukup Menarik () Tidak Menarik
2	Apakah game ini mudah digunakan?	(x) Ya () Tidak	() Sangat Menarik () Menarik (x) Cukup Menarik () Tidak Menarik
3	Menurut Anda bagaimana tingkat kesulitan pada game ini?	() Ya (x) Tidak	() Sangat Menarik () Menarik (x) Cukup Menarik () Tidak Menarik
4	Apakah secara umum game sudah berfungsi dengan baik?	() Ya () Tidak	() Sangat Menarik () Menarik () Cukup Menarik () Tidak Menarik
5	Apakah game ini mudah di pahami?	() Ya (x) Tidak	() Sangat Menarik () Menarik (x) Cukup Menarik () Tidak Menarik
6	Apakah masing-masing game dapat berfungsi dengan baik?	() Ya () Tidak	() Sangat Menarik () Menarik () Cukup Menarik () Tidak Menarik
7	Apakah sosial-sosial dalam game kuis mudah di pahami?	() Ya (x) Tidak	() Sangat Menarik () Menarik (x) Cukup Menarik () Tidak Menarik
8	Saran / Masukan :	game y kalanya SD.	