BAB III
ANALISIS DAN PERANCANGAN

III.1. Analisis Game

Analisis *game* merupakan analisis yang dilakukan melalui analisis masalah, analisis game sejenis, analisis game yang akan dikembangkan, analisis algoritma, analisis kebutuhan fungsional, dan analisis kebutuhan non fungsional. Analisis masalah merupakan analisis terhadap masalah-masalah yang muncul dalam penelitian ini. Analisis *game* sejenis dilakukan dengan melakukan observasi langsung terhadap *game-game* sejenis. Analisis game yang akan dikembangkan merupakan analisis terhadap game yang akan dibangun. Analisis kebutuhan non fungsional merupakan analisis yang meliputi perangkat keras dan perangkat lunak yang digunakan. Analisis kebutuhan fungsional merupakan analisis game yang akan dibangun dengan menggunakan pemodelan UML.

III.1.1. Analisis Masalah

Pada tahap analisis masalah, ditemukan berbagai masalah muncul, diantaranya :

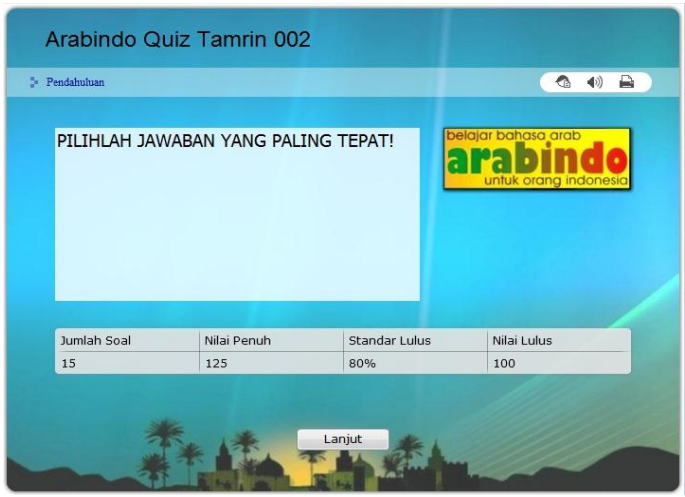
1. Belajar bahasa Arab sebagai bahasa asing

   Walaupun dianggap sebagai bahasa asing oleh bangsa Indonesia bahasa Arab tidak asing di telinga mereka terutama umat Islam. Sayangnya sebagian besar mereka masih beranggapan bahwa bahasa arab hanyalah basa agama sehingga perkembangannya terbatas di lingkungan kaum muslimin yang memperdalam ilmu-ilmu agama. Seperti dituliskan dalam buku metodologi pembelajaran bahasa arab, sikap dan pandangan masyarakat Islam Indonesia yang pasif menyebabkan pendidikan dan pengajaran bahasa Arab di tanah air berjalan sangat lambat dan tidak banyak mengalami perubahan yang mendasar. Meskipun usaha-usaha pengembangannya bukan masalah baru, namun metode dan sistem yang digunakan kebanyakan masih sangat tradisiona, dengan pola-pola yang digunakan dimasa lalu [HYPERLINK \l "Cha11" 4].
2. *Game* sebagai metode pembelajaran angka dan huruf hijaiyah

Setiap metode memiliki latar belakang, karakteristik, dan dasar pikiran yang berbeda, bahkan bisa jadi suatu metode muncul karena menolak metode sebelumnya. Jika metode-metode tersebut digabungkan, maka akan menjadi sebuah kolaborasi yang saling menyempurnakan. Kegiatan belajar mengajar akan menjadi sangat bervariatif, tidak terfokus pada satu kegiatan. Maka penggabungan berupa media animasi dan suara dalam game diharapkan akan membuat pembelajaran angka dan huruf hijaiyah ini memacu motivasi belajar pada anak-anak 4].

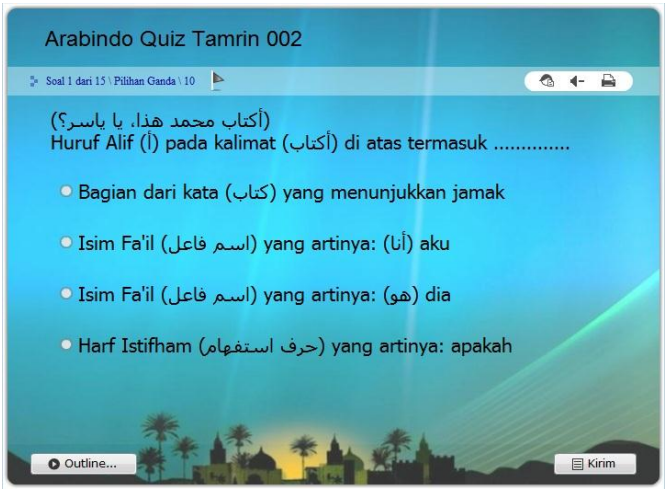
III.1.2. Analisis *Game* Sejenis

Berdasarkan hasil penelitian *Game* edukasi bertema bahasa Arab merupakan *game* edukasi 2 dimensi yang bertujuan untuk memberikan informasi-informasi dan mengenalkan bahasa Arab. Contohnya *game* edukasi Arabindo Quiz Tamrin [HYPERLINK \l "muh12" 3]. *Game* ini berbasis kuis dengan pertanyaan yang berkaitan dengan pembelajaran bahasa Arab. Pertanyaan pada *game* ini menggunakan huruf arab dan tidak menggunakan tanda baca, terdapat 4 pilihan jawaban yang dapat dipilih dari setiap pertanyaan. Ada 15 soal yang diberikan dan setiap soal yang dijawab dengan benar akan mendapatkan point. Pada awal permainan pemain yang baru bermain akan langsung masuk ke soal yang pertama, tetapi apabila pemain telah selesai menjawab semua soal maka akan ada pilihan tinjau untuk melihat kembali hasil pilihan jawaban yang pemain telah pilih. Tampilan awal *game* adalah sebagai berikut :



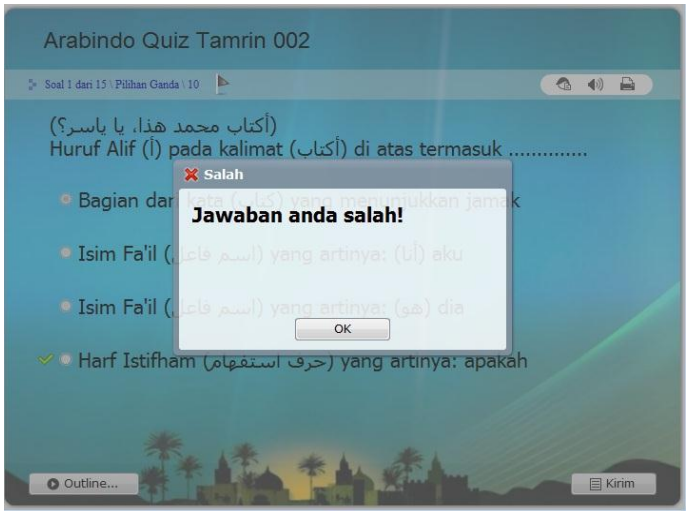
Gambar III. 1 Tampilan awal *game* Arabindo quiz tamrin

Tampilan setiap soal yang diberikan pada *game* ini adalah sebagai berikut :



Gambar III. 2 Tampilan soal *game* Arabindo quiz tamrin

Setiap soal ada 4 pilihan jawaban yang bisa dipilih, ketika pemain memilih salah satu jawaban dan mengklik button kirim maka akan akan muncul hasil dari jawaban tersebut benar atau salah. Seperti pada gambar berikut ini :



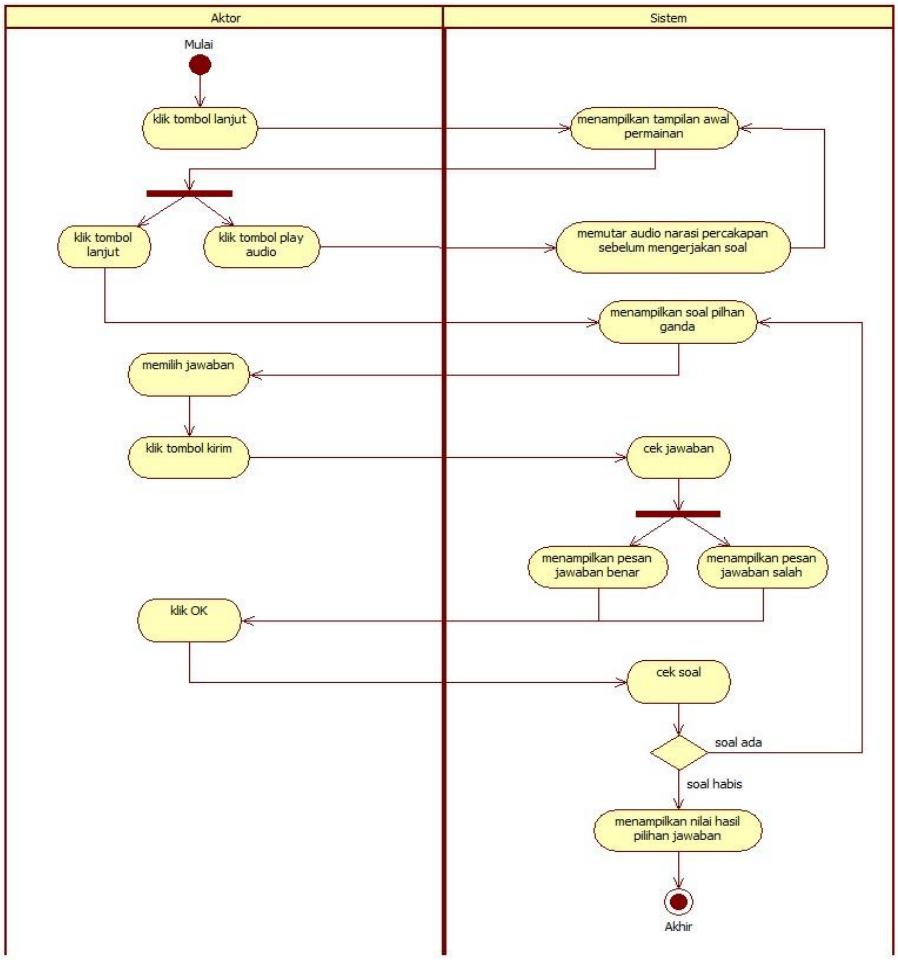
Gambar III. 3 Tampilan hasil pilihan jawaban *game* Arabindo quiz tamrin

Ketika pemain telah menjawab seluruh soal yang diberikan, maka akan muncul nilai yang berhasil di dapatkan oleh pemain. Seperti pada gambar berikut ini :



Gambar III. 4 Nilai hasil jawaban *game* Arabindo quiz tamrin

Alur permainan pada *game* Arabindo quiz tamrin dijelaskan pada activity diagram berikut ini :



Gambar III. 5 Activity diagram *game* Arabindo quiz tamrin

Perbandingan *game* Arabindo quiz tamrin dan *game* yang akan dikembangkan dijelaskan pada table berikut ini :

Tabel III. 1 Perbandingan *game* sejenis dengan *game* yang akan dikembangkan

No.	Perbandingan	*Game* Arabindo quiz tamrin	*Game* yang akan dikembangkan
1.	Jenis *game*	Quiz *game*	Memory *game*, puzzle *game*, Quiz *game*
2.	Tema *game*	Bahasa Arab	Bahasa Arab
3.	Karakter	Tidak ada karakter	2 karakter utama
4.	Tampilan antarmuka	2 dimensi	2 dimensi
5.	Tingkat kesulitan	1 level	3 level
6.	Platform	Desktop	Mobile
7.	*Player*	1 orang pengguna atau *single player*	1 orang pengguna atau *single player*

III.1.3. Analisis *Game* yang Akan Dikembangkan

Analisis *game* yang akan dikembangkan merupakan bagian yang mendeskripsikan *game* yang akan dikembangkan. Pada bagian ini terdiri dari pengenalan, *story line*, tingkat kesulitan, dan *gameplay*.

III.1.3.1. Pengenalan

Game yang akan dibuat berjudul Petualangan Anak Soleh dengan menggunakan model *game* edukasi yang dirancang untuk pengayaan pembelajaran menggunakan teknologi multimedia interaktif. *Game* ini merupakan *game* edukasi untuk mempelajari angka dan huruf hijaiyah. Deskripsi penerapan dalam *game* adalah sebagai berikut :

1. *Game* dimainkan dengan cara Tapping yaitu teknik interaksi dengan menyentuh layar dengan satu jari lalu melepaskannya kembali. Teknik ini sebagai pengganti fungsi click pada mouse.

2. Pada awal permainan player diminta untuk memilih karakter yang akan di mainkan. Ada 2 karakter utama yang dapat dipilih yaitu seorang anak laki-laki muslim dan seorang anak perempuan muslim.
3. *Game* terdiri dari 3 stage, player harus melewati setiap stage untuk menuju stage 3 dan menyelesaikan permainan.
4. Pembelajaran terdiri dari pembelajaran angka dan huruf hijaiyah.
5. Setiap stage akan di beri pertanyaan yang berbeda dan tingkat kesulitan yang berbeda.
6. Setelah menyelesaikan setiap stage player akan di beri pilihan untuk memilih arah jalan yang akan dilalui menuju stage berikutnya. Setiap jalan yang di pilih oleh player akan berpengaruh terhadap akhir cerita atau ending story.

III.1.3.2. Story Line

Game Petualangan Anak Soleh menceritakan perjalanan seorang anak laki-laki yang bernama Ahmad atau seorang anak perempuan yang bernama Siti. Karakter yang akan dimainkan tergantung karakter yang dipilih oleh *player*. Mereka bertujuan untuk belajar tentang ilmu-ilmu islam di pesantren bernama pesantren anak soleh. Pesantren Anak Soleh merupakan pesantren yang berada di tengah hutan. Setiap tahun pesantren ini menerima santri baru untuk diajarkan ilmu-ilmu tentang agama islam. Untuk dapat diterima menjadi santri di pesantren tersebut, *player* harus melewati 3 test yang berbeda dari 3 orang penguji test.

Pada awal cerita player akan bertemu dengan penjaga stage 1 yaitu ustad anwar. Pada stage 1 ini player akan belajar mengenai angka hijaiyah. untuk bisa melanjutkan ke stage berikutnya player harus bermain game dan menyelesaikannya. Ketika player berhasil menyelesaikan game pada stage 1 player akan memilih jalan yang akan dilaluinya, setiap arah jalan yang dipilih akan menentukan akhir cerita dari game. Ada dua arah jalan yang dapat dipilih oleh player yaitu :

1. Arah jalan ke kanan

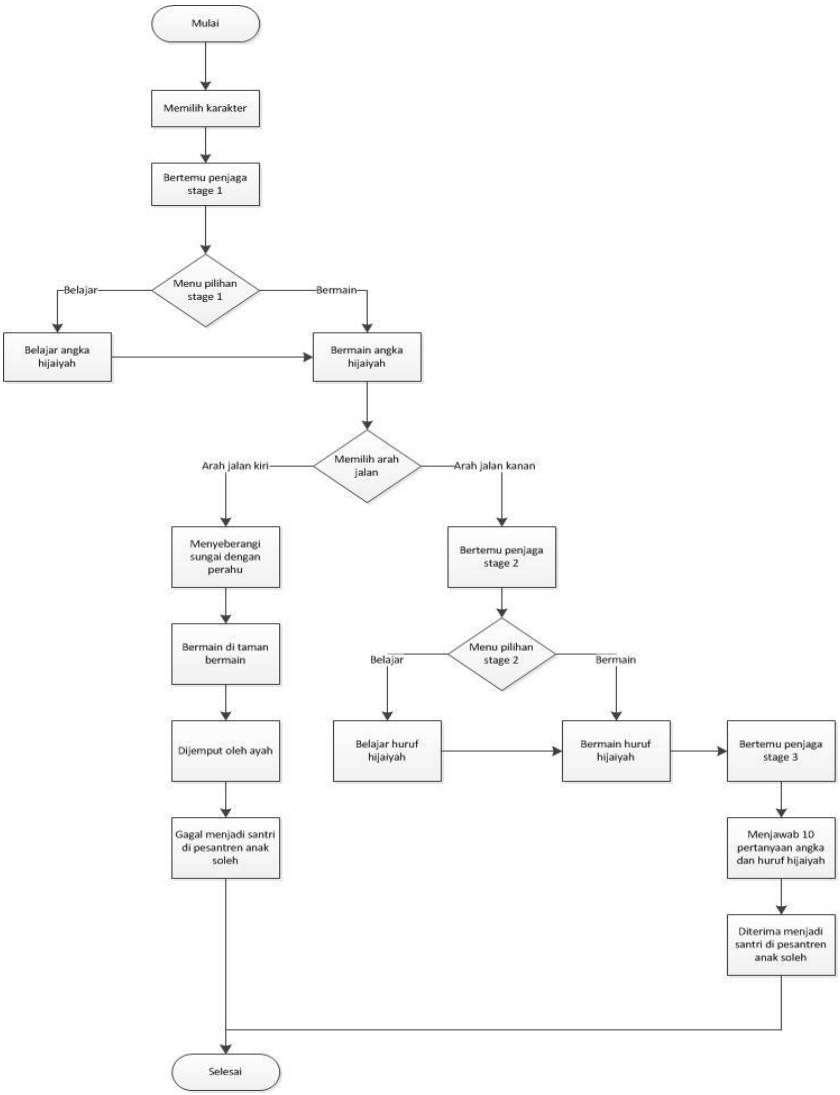
   Jika player memilih arah jalan ke kanan maka player akan bertemu dengan ustadzah nur yaitu penjaga pada stage ke 2. Pada stage ke 2 ini player akan belajar mengenai huruf hijaiyah. untuk melanjutkan ke stage ke 3 player

harus bermain game puzzle terlebih dahulu. Apabila player berhasil menyelesaikan game puzzle player akan bertemu dengan pa ustad imron yaitu penjaga pada stage ke 3. Pada stage ke 3 ini player akan diberikan 10 pertanyaan tentang angka dan hururf hijaiyah. ketika player telah selesai menjawab 10 pertanyaan maka *player* berhasil menjadi santri di Pesantren anak soleh.

2. Arah jalan ke kiri

   Jika player memilih arah jalan ke kiri player akan melihat sungai yang harus disebrangi dengan menggunakan perahu. Ketika berhasil menyebrangi sungai maka player akan melihat sebuah taman bermain. ketika sedang asik bermain tiba-tiba muncul ayah dari player yang menjemputnya untuk pulang. Dan player gagal untuk menjadi santri di pesantren anak soleh.

Untuk lebih jelas mengenai alur cerita akan digambarkan pada flowchar berikut ini :



Gambar III. 6 Flowchart alur cerita

III.1.3.3. Tingkat Kesulitan

Game ini memiliki tingkat kesulitan yang berbeda pada setiap *stage*. Tingkat kesulitan yang akan dihadapi oleh *player* adalah sebagai berikut:

1. Stage 1

Pada tingkat kesulitan stage 1 player harus mencocokkan angka hijaiyah dengan benar, di awal permainan player memilih terlebih dahulu jumlah matriks yang akan dimainkan, player bisa memilih matriks 4 x 4 atau 5 x 4. Pada matriks 4 x 4 player harus mencocokkan gambar angka yang sama dari angka 1 sampai angka 8 yang posisi gambar diacak menggunakan fungsi random. Contoh untuk pencocokkan angka hijaiyah dengan matriks 4 x 4 adalah sebagai berikut:

Tabel III. 2 Contoh pencocokkan angka hijaiyah dengan matriks 4 x 4

Gambar angka 1	Gambar angka 2	Gambar angka 3	Gambar angka 4
Gambar angka 5	Gambar angka 6	Gambar angka 7	Gambar angka 8
Gambar angka 1	Gambar angka 2	Gambar angka 3	Gambar angka 4
Gambar angka 5	Gambar angka 6	Gambar angka 7	Gambar angka 8

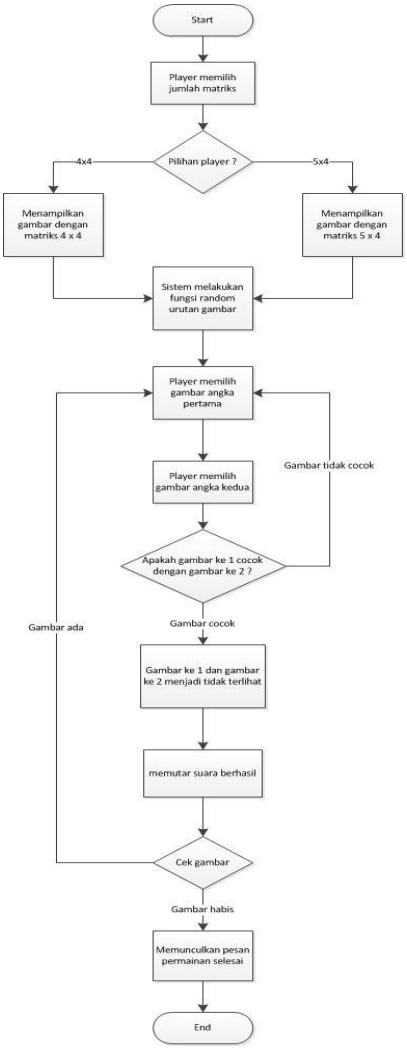
Langkah pertama *player* dapat memilih salah satu gambar angka secara acak, misalkan player memilih gambar angka 1 kemudian player dapat memilih gambar yang kedua, misalkan gambar kedua yang dipilih player adalah gambar angka 1. Maka jawaban *player* dinyatakan benar, tetapi jika gamabr pertama dan gambar kedua yang dipilih oleh *player* berbeda maka jawaban dinyatakan salah. Begitupun seterusnya hingga gambar angka 1 sampai 8 berhasil dicocokkan. Untuk matriks 5 x 4 player harus mencocokkan gambar angka yang sama dari angka 1 sampai angka 10 yang posisi gambar diacak

menggunakan fungsi random. Contoh untuk pencocokkan angka hijaiyah dengan matriks 5 x 4 adalah sebagai berikut:

Tabel III. 3 Contoh pencocokkan angka hijaiyah dengan matriks 5 x 4

Gambar angka 1	Gambar angka 2	Gambar angka 3	Gambar angka 4	Gambar angka 5
Gambar angka 6	Gambar angka 7	Gambar angka 8	Gambar angka 9	Gambar angka 10
Gambar angka 1	Gambar angka 2	Gambar angka 3	Gambar angka 4	Gambar angka 5
Gambar angka 6	Gambar angka 7	Gambar angka 8	Gambar angka 9	Gambar angka 10

Berikut ini gambar flowchart untuk pencocokkan angka hijaiyah:



Gambar III. 7 Flowchart pencocokkan angka hijaiyah

2. Stage 2

Pada tingkat kesulitan stage 2 player harus menyusun huruf hijiyah dengan benar, di awal permainan akan ditampilkan gambar susunan huruf yang harus disusun oleh player. Kemudian pada saat player menekan tombol mulai posisi gambar akan di acak menggunakan fungsi random. Contoh penyusunan huruf hijaiyah adalah sebagai berikut:

Tabel III. 4 Contoh penyusunan huruf hijaiyah

Gambar 1	Gambar 2	Gambar 3	Gambar 4
Gambar 5	Gambar 6	Gambar 7	Gambar 8

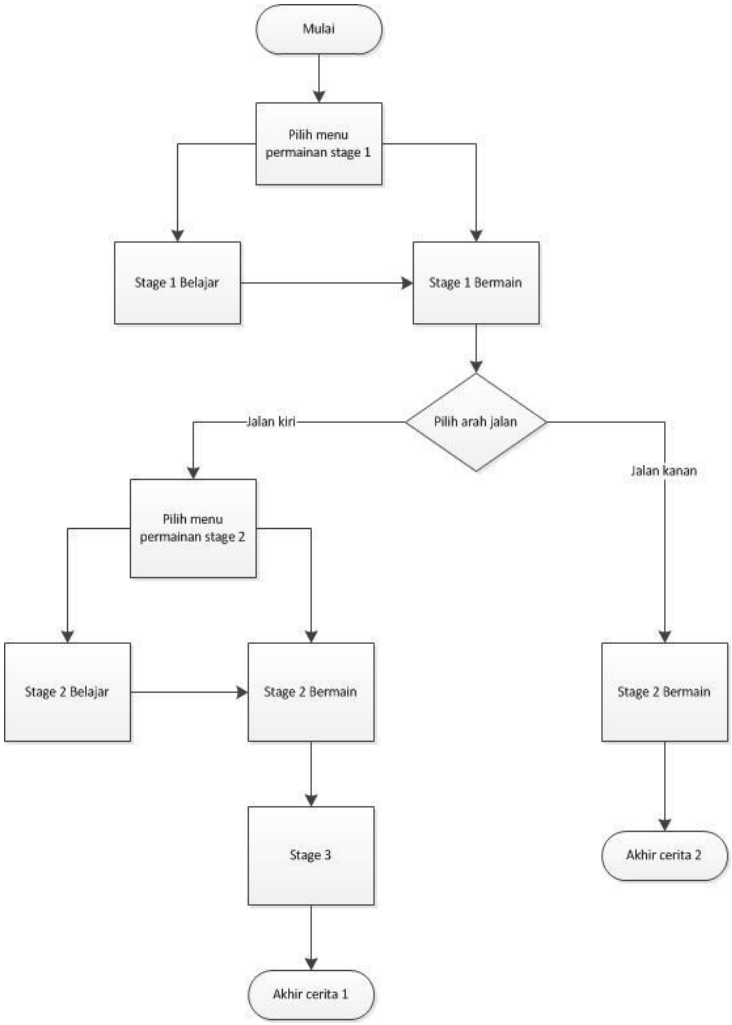
Gambar 9	Gambar 10	Gambar 11	Gambar 12
Gambar 13	Gambar 14	Gambar 15	

Karena jumlah huruf hijaiyah ada 30 maka ketika player berhasil menyusun huruf dari huruf 1 sampai huruf ke-15 akan dilanjutkan dengan menyusun huruf ke-16 sampai huruf ke-30.

III.1.3.4. *Game*play

Dalam aplikasi *game* ini, player memiliki tugas untuk menjawab dengan benar setiap pertanyaan yang diberikan oleh penjaga pada setiap stage untuk menuju stage berikutnya. Bagi pemula, stage yang dapat dimainkan berawal dari stage 1. Sebelum masuk ke permainan *player* dapat memilih menu belajar yaitu pengenalan mengenai angka dan hururf hijaiyah. Pada menu belajar stage 1 player dapat mempelajari angka hijaiyah mulai dari bentuk angka dan cara pengucapannya.

Ketika *player* selesai belajar angka maka *player* bisa langsung memilih menu bermain untuk langsung bermain di stage 1. Apabila player dapat menyelesaikan stage 1 akan di berikan pilihan arah ke kiri atau ke kanan untuk menuju ke stage 2. Sebelum masuk ke stage 2 player bisa memilih menu belajar huruf hijaiyah. Pada menu pembelajaran huruf hijaiyah player akan mulai belajar dari bentuk dan cara pengucapannya. Setiap arah yang dipilih oleh player akan mempengaruhi kepada ending dari cerita. Ada dua akhir cerita yang kemungkinan dapat dipilih oleh player. Setiap *game* dimainkan dengan cara Tapping yaitu teknik interaksi dengan menyentuh layar dengan satu jari lalu melepaskannya kembali.



Gambar III. 8 Alur Permainan

III.1.4. Analisis Penggunaan Algoritma terhadap *game* puzzle

Game puzzle adalah permainan tradisional yang diterapkan dalam sebuah papan n x n, dimana setiap blok papan diberi angka dari 1 sampai ($n^2 - 1$) dan satu buah papan kosong (*blank*), yang terurut dari sisi kiri atas papan sampai kanan bawah. Satu papan kosong (diisi paling kanan) digunakan untuk memindahkan papan-papan disekelilingnya. Pemindahan bisa dilakukan dengan memindahkan papan kosong tersebut ke atas, ke bawah, ke kanan, atau ke kiri. Papan yang kosong harus terus dipindah-pindahkan agar solusi akhir bisa didapat. Solusi akhir dari permainan ini adalah papan-papan tersebut terurut dari nilai yang paling kecil hingga nilai yang paling besar seperti pada gambar dibawah ini :

1	2	3
4	5	6
7	8	

Gambar III. 9 Susunan dari *game* puzzle yang sudah terurut

Aturan-aturan yang ada dalam menyusun *game* puzzle adalah sebagai berikut :

Tabel III. 5 Aturan pergerakan dalam *game* puzzle

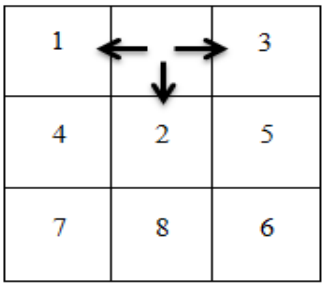
Aturan ke-	kondisi	Pergerakan papan kosong
1	Jika papan kosong berada di tengah	Atas, bawah, kiri, kanan
2	Jika papan kosong berada di paling kanan	Atas, bawah, kiri
3	Jika papan kosong berada di paling kiri	Atas, bawah, kanan
4	Jika papan kosong berada di paling atas	Kiri, kanan, bawah
5	Jika papan kosong berada di paling bawah	Kiri, kanan, atas
6	Jika papan kosong berada di sudut kanan atas	Kiri, bawah
7	Jika papan kosong berada di sudut kiri atas	Kanan, bawah
8	Jika papan kosong berada di sudut kanan bawah	Kiri, atas
9	Jika papan kosong berada di sudut kiri bawah	Kanan, atas

Algoritma yang diterapkan dalam *game* puzzle yaitu untuk mengecek dengan benar susunan posisi puzzle dari nilai yang paling kecil hingga nilai yang paling besar dengan menggunakan metode pengurutan seleksi (*Selection Sort*). Setiap papan diberi nilai dari 1 sampai 15 dan 1 papan kosong (*null*). Kemudian dilakukan fungsi acak agar posisi papan tidak terurut pada awal permainan.

1		3
4	2	5
7	8	6

Gambar III. 10 Contoh acak posisi pada awal permainan

Langkah selanjutnya yaitu memindahkan blok papan yg berisi nilai dengan blok papan yang kosong. Arah langkah untuk memindahkan papan kosong tergantung pada kondisi papan kosong itu berada. Misalkan jika papan kosong berada di tengah maka papan kosong dapat dipindahkan keatas, kebawah, kekiri, dan kekanan.



Gambar III. 11 Pergerakan posisi papan kosong berada ditengah

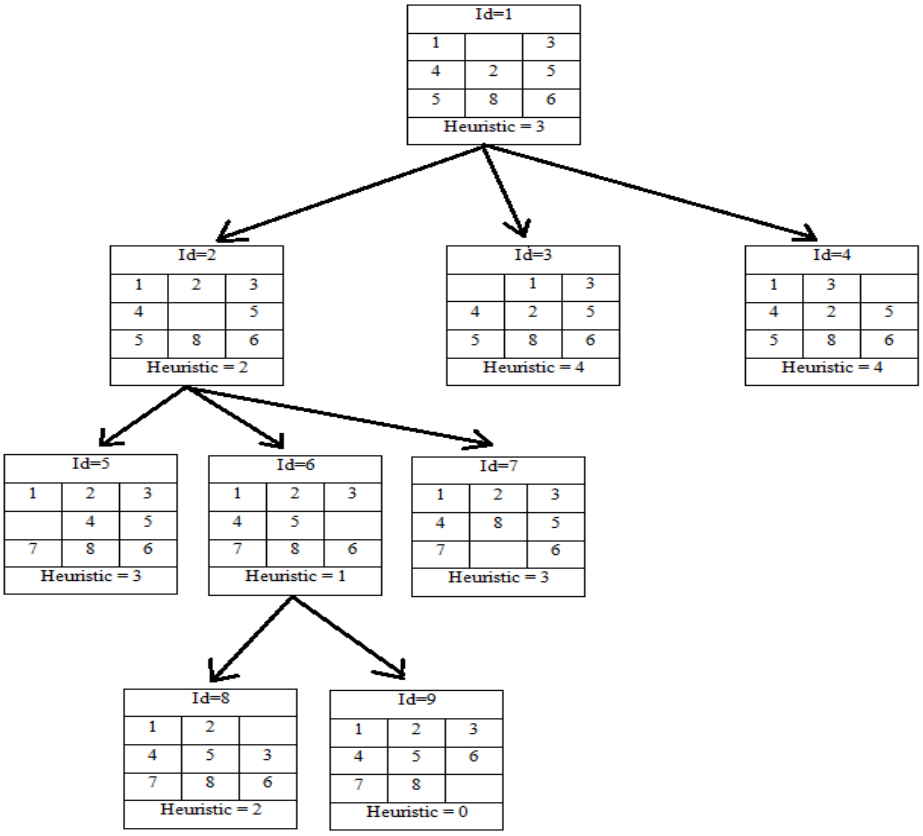
Untuk permasalahan *game puzzle* akan digunakan pohon *m-ary* yaitu pohon yang memiliki anak paling banyak sejumlah *m.*

Algoritma untuk menyelesaikan *game puzzle* yaitu sebagai berikut:

1. Tentukan *heuristic* semua daun pada pohon
2. Cari *heuristic* paling kecil pada semua daun
3. Evaluasi daun yang memiliki heuristic paling kecil yaitu menentukan anak-anaknya

4. Ulangi langkah 1 sampai 3 hingga ditemukan daun yang *heuristic*-nya sama dengan 0.

Sesuai dengan algoritma diatas, terlebih dahulu dicari *heuristic* semua daun yang ada pada pohon.



Gambar III. 12 Solusi *game puzzle* dengan menggunakan *heuristic*

Pada gambar diatas kondisi awal puzzle yaitu pada id=1. Posisi yang salah pada puzzle dengan id=1 yaitu angka 2, 5, dan 6. Jarak angka 2 ke posisi sebenarnya adalah 1 kotak. Jarak angka 6 ke posisi sebenarnya adalah 1 kotak. Jadi nilai *heuristic* id=1 dalah 1+1+1=3. Berdasarkan algoritma diatas, terlebih dahulu dicari *heuristic* semua daun yang ada pada pohon. Sekarang, hanya terdapat satu daun pada pohon, yaitu id=1. Nilai *heuristic* id=1 adalah 3, sesuai hasil pada gambar III.11. kemudian dicari nilai *heuristic* yang paling kecil. Karena hanya terdapat satu daun, maka id=1 sekaligus menjadi daun yang terpilih untuk dievaluasi yaitu mencari anak-anaknya.

Anak-anak dari id=1 yaitu id=2, id=3, dan id=4. Sekarang terdapat 3 daun yaitu id=2, id=3, dan id=4. Id=1 bukan daun lagi karena telah memiliki anak. Sekarang cari *heuristic* semua daun. Sesuai dengan cara sebelumnya, diperoleh *heuristic* id=2 yaitu 2, id=3 yaitu 4, dan id=4 yaitu 4. *Heuristic* paling kecil terdapat pada id=2. Sekarang id=2 akan dievaluasi lagi karena merupakan elemen daun yang memiliki nilai *heuristic* paling kecil. Setelah dievaluasi, id=2 memiliki anak yaitu id=5, id=6, dan id=7. Elemen daun pada pohon sekarang yaitu, id=3, id=4, dan id=5, id=6 dan id=7. Nilai-nilai *heuristic* paling kecil terdapat pada daun id=6.

Hal yang sama tetap dilakukan yaitu melakukan evaluasi pada daun yang memiliki *heuristic* paling kecil yaitu id=6. Id=6 memiliki anak id=8 dan id=9. Id=8 memiliki *heuristic* =2 dan id=9 memiliki *heuristic* =0. Evaluasi selesai dilakukan karena telah ditemukan daun yang memiliki heuristic = 0.

III.1.5. Analisis Kebutuhan Non Fungsional

Analisis kebutuhan non fungsional dilakukan untuk mengetahui spesifikasi kebutuhan untuk sistem. Spesifikasi kebutuhan melibatkan analisis perangkat keras / *hardware*, analisis perangkat lunak / *software.*

III.1.5.1. Analisis Kebutuhan Perangkat Lunak

Analisis kebutuhan perangkat lunak pada *game* ini terbagi menjadi dua bagian yaitu analisis kebutuhan perangkat lunak untuk penggembang dan analisis kebutuhan perangkat lunak untuk pengguna.

1. Analisis kebutuhan perangkat lunak untuk pengembang

   Perangkat lunak yang digunakan dalam pengembangan aplikasi *game* ini adalah sebagai berikut :

   a. Sistem Operasi Windows 7 64 bit

   b. Eclipse

   c. Android SDK

   d. Adobe Photoshop

2. Analisis kebutuhan perangkat lunak untuk pengguna

   Perangkat lunak yang dibutuhkan pengguna dalam memainkan aplikasi *game* ini adalah sebagai berikut :

a. Sistem Operasi Android 2.3.3 ke atas

b. *flash Player*

III.1.5.2. Analisis Kebutuhan Perangkat Keras

Analisis kebutuhan perangkat keras pada *game* ini terbagi menjadi dua bagian yaitu analisis kebutuhan perangkat keras untuk pengguna dan analisis kebutuhan perangkat keras untuk pengembang.

1. Analisis kebutuhan perangkat keras untuk pengembang

   Pada aplikasi ini, analisis perangkat perangkat keras yang digunakan pengembang adalah sebagai berikut :

   a. Processor 2.10GHz

   b. RAM 4 GB

   c. VGA 512 MB

   d. Harddisk 320 GB

   e. Monitor

   f. Mouse dan Keyboard

2. Analisis Kebutuhan Perangkat Keras Untuk Pengguna

   Perangkat keras yang dibutuhkan pengguna dalam memainkan aplikasi *game* adalah smartphone. Spesifikasi smartphone yang diperlukan untuk memainkan *game* ini adalah sebagai berikut

   a. Sistem Operasi Android 2.3.3 keatas

   b. Processor 1,00 Ghz

   c. Kapasitas RAM 128 MB

   d. Kapasitas penyimpanan 5 MB

III.1.5.3. Analisis dan Kebutuhan Perangkat Pikir

Analisis kebutuhan perangkat pikir dilakukan untuk mengetahui kemampuan *player* (pemain) yang dapat memainkan *game* ini. Analisis kebutuhan perangkat piker digambarkan dalam table kebutuhan *player*. Kebutuhan *player* dijabarkan pada tabel III-6.

Tabel III. 6 Tabel Kebutuhan Player

Karakteristik Player	Keterangan
Computer Literacy	Dapat menggunakan *smartphone* berbasis

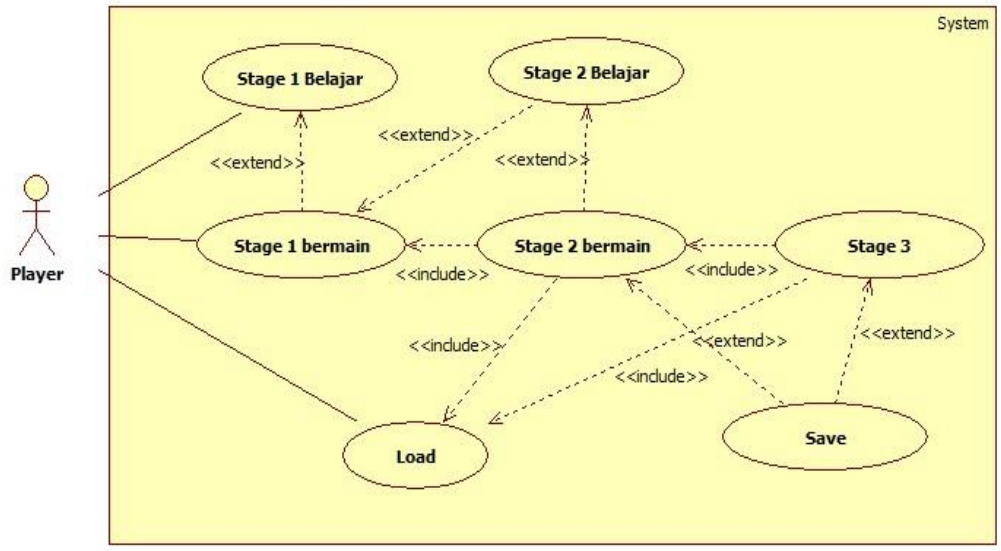
	Android
Age	6 tahun keatas [12]
Application Experience	Tidak dibutuhkan
Gender	Pria dan Wanita
Disabilities	Buta total dan buta warna

III.1.6. Analisis Kebutuhan Fungsional

Tahap pertama yang dilakukan dalam melakukan analisis berorientasi objek menggunakan UML adalah menentukan aktor atau pengguna sistem. Kata aktor dalam konteks UML, menampilkan peran (*rules*) yang pengguna (atau sesuatu di luar sistem yang dikembangkan yang dapat berupa perangkat keras, *end player*, sistem yang lain, dan sebagainya).

III.1.6.1. Use Case Diagram

Use case merupakan gambaran skenario dari interaksi antara *player* dengan sistem. Sebuah diagram use case menggambarkan hubungan antara actor dan kegiatan yang dapat dilakukan terhadap aplikasi.



Gambar III. 13 Use Case Diagram

III.1.6.2. Definisi Aktor

Definisi aktor berfungsi menjelaskan peran setiap aktor yang terdapat pada use case.

Tabel III. 7 Definisi Aktor

No.	Aktor	Deskripsi
1	Player	Player / Pengguna yang memainkan *game* ini

III.1.6.3. Definisi Use Case

Definisi Use Case berfungsi untuk menjelaskan proses yang terdapat pada setiap use case.

Tabel III. 8 Definisi Use Case

No.	Use Case	Deskripsi
1	*Stage 1* bermain	Proses permainan pada *stage* 1
2	*Stage 1* belajar	Proses pembelajaran pada *stage* 1
3	*Stage 2* bermain	Proses permainan pada *stage 2*
4	Stage 2 belajar	Proses pembelajaran pada *stage* 2
5	*Stage 3*	Proses permainan pada *stage* 3
6	*Load Game*	Proses memulai permainan dari save point
7	*Save Game*	Proses menyimpan permainan

III.1.6.4. Skenario Use Case

Skenario setiap bagian pada use case menunjukkan proses apa yang terjadi pada setiap bagian didalam use case tersebut, dimana player memberikan perintah pada setiap bagian dan respon apa yang diberikan oleh sistem kepada player setelah player memberikan perintah pada setiap bagian-bagian use case.

Tabel III. 9 Skenario Use Case Stage 1 bermain

Identifikasi	
Nomor	1
Nama	*Stage* 1 bermain
Tujuan	Memainkan *stage* 1
Deskripsi	Proses permainan pada *stage* 1
Aktor	*Player*

Skenario Utama		
Kondisi Awal	Aktor berada di dalam permainan pada *stage* 1	
Aksi Aktor		Reaksi Sistem
1. Memulai *stage* 1		2. Menampilkan form *stage* 1
3. Memilih jumlah matriks		4. Menampilkan form jumlah matriks yang dipilih
6. Memilih gambar ke-1		5. Mengacak urutan gambar
7. Memilih gambar ke-2		8. Cek gambar ke-1 sama dengan gambar ke-2
		9. *Play* suara berhasil
		10. Update gambar
11. Tap lanjut		12. Menampilkan stage 2
Skenario Tambahan		
Kondisi Awal	Aktor berada di pesan keluar dari permainan	
1. Menekan tombol Ya		2. Menampilkan menu utama
3. Menekan tombol Tidak		4. Menampilkan form *stage* 1

Tabel III. 10 Skenario Use Case Stage 1 Belajar

Identifikasi	
Nomor	2
Nama	*Stage* 1 belajar
Tujuan	Belajar angka hijaiyah
Deskripsi	Proses pembelajaran pada *stage* 1
Aktor	*Player*
Skenario Utama	
Kondisi Awal	Aktor berada di dalam permainan pada *stage* 1 belajar
Aksi Aktor	Reaksi Sistem

1. Memulai *stage* 1 belajar	2. Menampilkan form *stage* 1 belajar
3. Tap pilihan angka	4. Menampilkan form pilihan angka
5. Tap sound	6. *Play* sound
7. Tap back	8. Menampilkan form *stage* 1 belajar
9. Tap home	10. Menampilkan form menu utama
11. Tap Bermain	12. Menampilkan form stage 1 bermain
Skenario Tambahan	
Kondisi Awal	Aktor berada di pesan kembali ke home
1. Menekan tombol Ya	2. Menampilkan menu utama
3. Menekan tombol Tidak	4. Menampilkan form *stage* 1 belajar

Tabel III. 11 Skenario Use Case Stage 2 bermain

Identifikasi	
Nomor	2
Nama	*Stage* 2 bermain
Tujuan	Memainkan *stage* 2
Deskripsi	Proses permainan pada *stage* 2
Aktor	*Player*
Skenario Utama	
Kondisi Awal	Aktor berada di dalam permainan pada *stage* 2
Aksi Aktor	Reaksi Sistem
1. Memulai *stage* 2	2. Menampilkan form *stage* 2
4. Menyusun gambar secara berurutan	3. Melakukan acak gambar
	5. Gambar sudah terurut dari urutan 1 sampai 15
7. Tap tombol lanjut	6. Menampilkan pesan untuk menyusun gambar berikutnya
	8. Menampilkan form susunan gambar ke

	2
10. Menyusun gambar sesuai urutan	9. melakukan acak gambar
	11. Gambar sudah terurut dari urutan 16 sampai 30
13. Tap tombol lanjut	12. menampilkan pesan berhasil
	14. menampilkan stage 3
Skenario Tambahan	
Kondisi Awal	Aktor berada di pesan keluar dari permainan
1. Menekan tombol Ya	2. Menampilkan menu utama
3. Menekan tombol Tidak	4. Menampilkan form *stage* 2

Tabel III. 12 Skenario Use Case Stage 2 Belajar

Identifikasi	
Nomor	2
Nama	*Stage* 2 belajar
Tujuan	Belajar huruf hijaiyah
Deskripsi	Proses pembelajaran pada *stage* 2
Aktor	*Player*
Skenario Utama	
Kondisi Awal	Aktor berada di dalam permainan pada *stage* 1 belajar
Aksi Aktor	Reaksi Sistem
1. Memulai *stage* 2 belajar	2. Menampilkan form *stage* 2 belajar
3. Tap pilihan huruf	4. Menampilkan form pilihan huruf
5. Tap sound	6. *Play* sound
7. Tap back	8. Menampilkan form stage 2 belajar
9. Tap home	10. Menu utama
11. Tap Bermain	12. Menampilkan form stage 2
Skenario Tambahan	

Kondisi Awal	Aktor berada di pesan kembali ke home
1. Menekan tombol Ya	2. Menampilkan menu utama
3. Menekan tombol Tidak	4. Menampilkan form *stage* 2 belajar

Tabel III. 13 Skenario Use Case *Stage* 3

Identifikasi	
Nomor	3
Nama	*Stage* 3
Tujuan	Memainkan *stage* 3
Deskripsi	Proses permainan pada *stage* 3
Aktor	*Player*
Skenario Utama	
Kondisi Awal	Aktor berada di dalam permainan pada *stage* 3
Aksi Aktor	Reaksi Sistem
1. Memulai *stage* 3	2. Menampilkan form *stage* 3
3. Memilih tampilkan soal	4. Acak soal
6. Memilih jawaban	5. Menampilkan soal
8. Tekan tombol lanjut	7. Cek jawaban
	9. Cek soal
	10. Menampilkan akhir cerita permainan
Skenario Tambahan	
Kondisi Awal	Aktor berada di pesan keluar dari permainan
1. Menekan tombol Ya	2. Menampilkan menu utama
3. Menekan tombol Tidak	4. Menampilkan form *stage* 3

Tabel III. 14 Skenario Use Case *Load Game*

Identifikasi	
Nomor	4

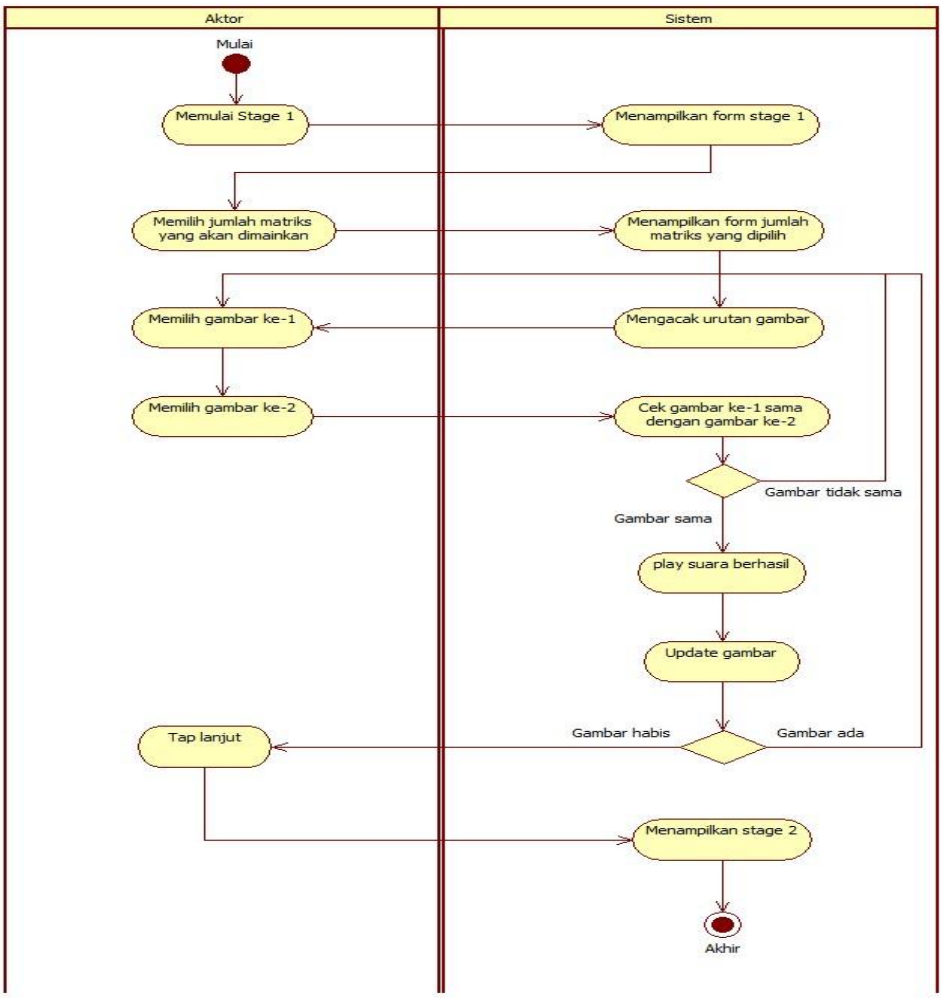
Nama	*Load Game*	
Tujuan	Memulai permainan dari *save point*	
Deskripsi	Proses memulai permainan dari *save point*	
Aktor	*Player*	
Skenario Utama		
Kondisi Awal	Aktor berada di dalam form *Load game*	
Aksi Aktor		Reaksi Sistem
1. Memilih *Load Game*		2. Menampilkan form *Load Game*
3. Memilih *save point*		4. Menampilkan level permainan sesuai *save point*

Tabel III. 15 Skenario Use Case *Save Game*

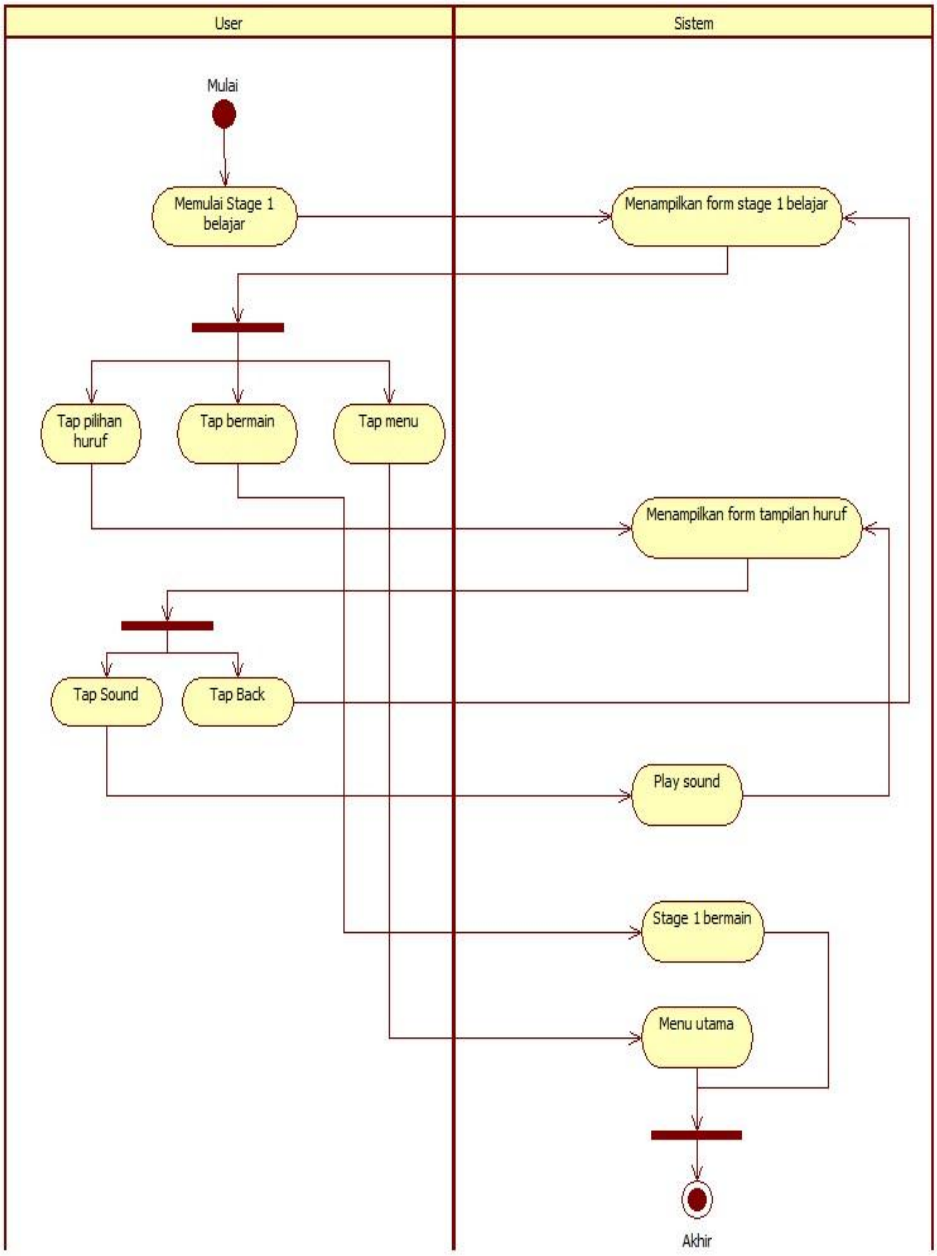
Identifikasi		
Nomor	5	
Nama	*Save Game*	
Tujuan	Menyimpan permainan	
Deskripsi	Proses menyimpan permainan	
Aktor	*Player*	
Skenario Utama		
Kondisi Awal	Aktor berada di dalam form *save point*	
Aksi Aktor		Reaksi Sistem
1. Memilih *Save Game*		2. Menampilkan form *Save game*
		3. Menyimpan permainan ke storage
Skenario Tambahan		
Kondisi Awal	Aktor berada di pesan *Save Game*	
1. Menekan tombol Ya		2. Menampilkan pesan permainan telah di *save*
3. Menekan tombol Tidak		4. Menampilkan kembali form permainan

III.1.6.5. Activity Diagram

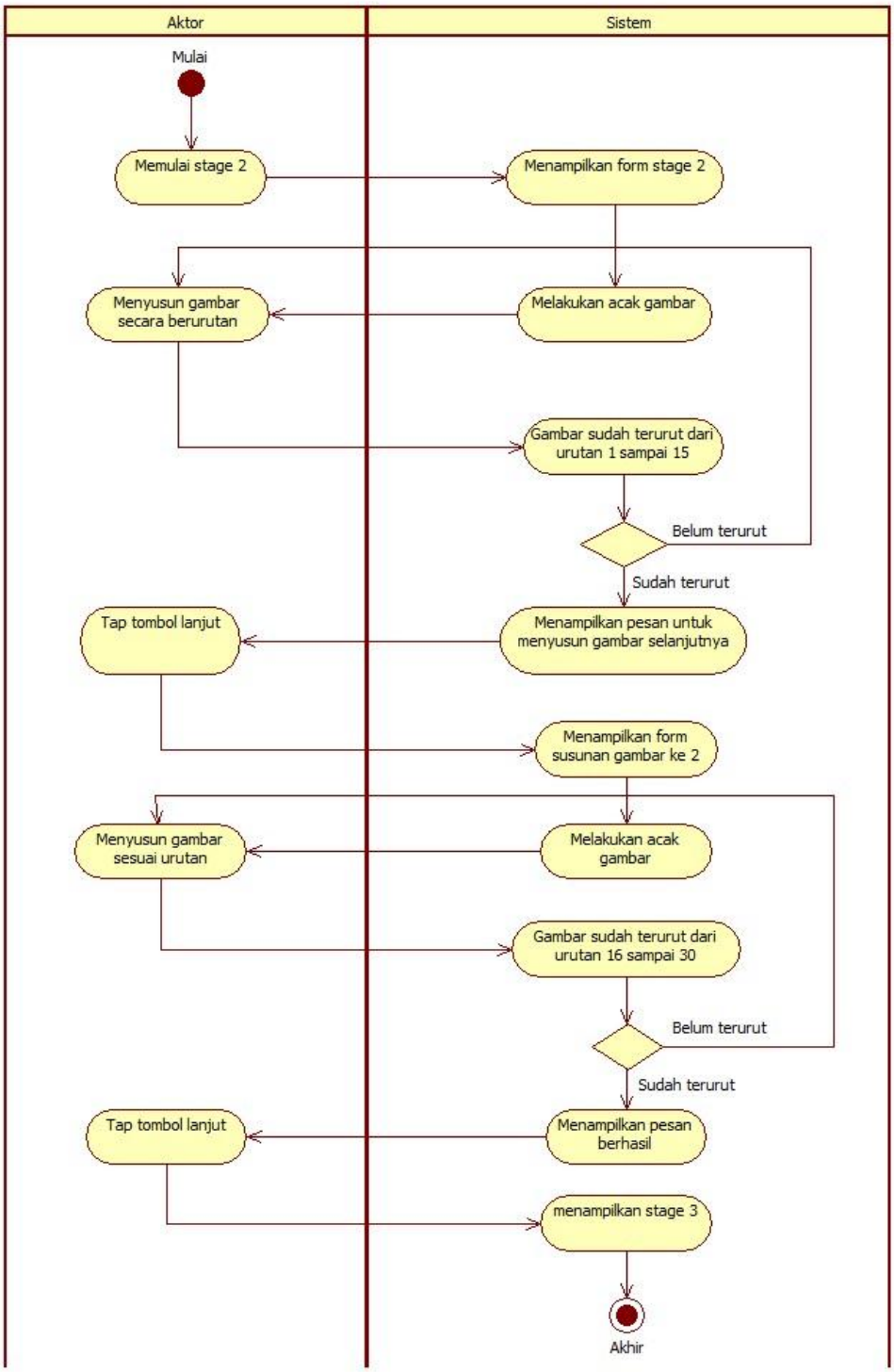
Penggambaran alir aktivitas pada *game* digambarkan menggunakan *Activity Diagram.* *Activity Diagram* menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum. Sebuah aktivitas dapat direalsasikan oleh satu *Use Case* atau lebih. Aktivitas menggambarkan proses yang berjalan, sementara *Use Case* menggambarkan bagaimana actor menggunakan sisitem untuk melakukan aktivitas



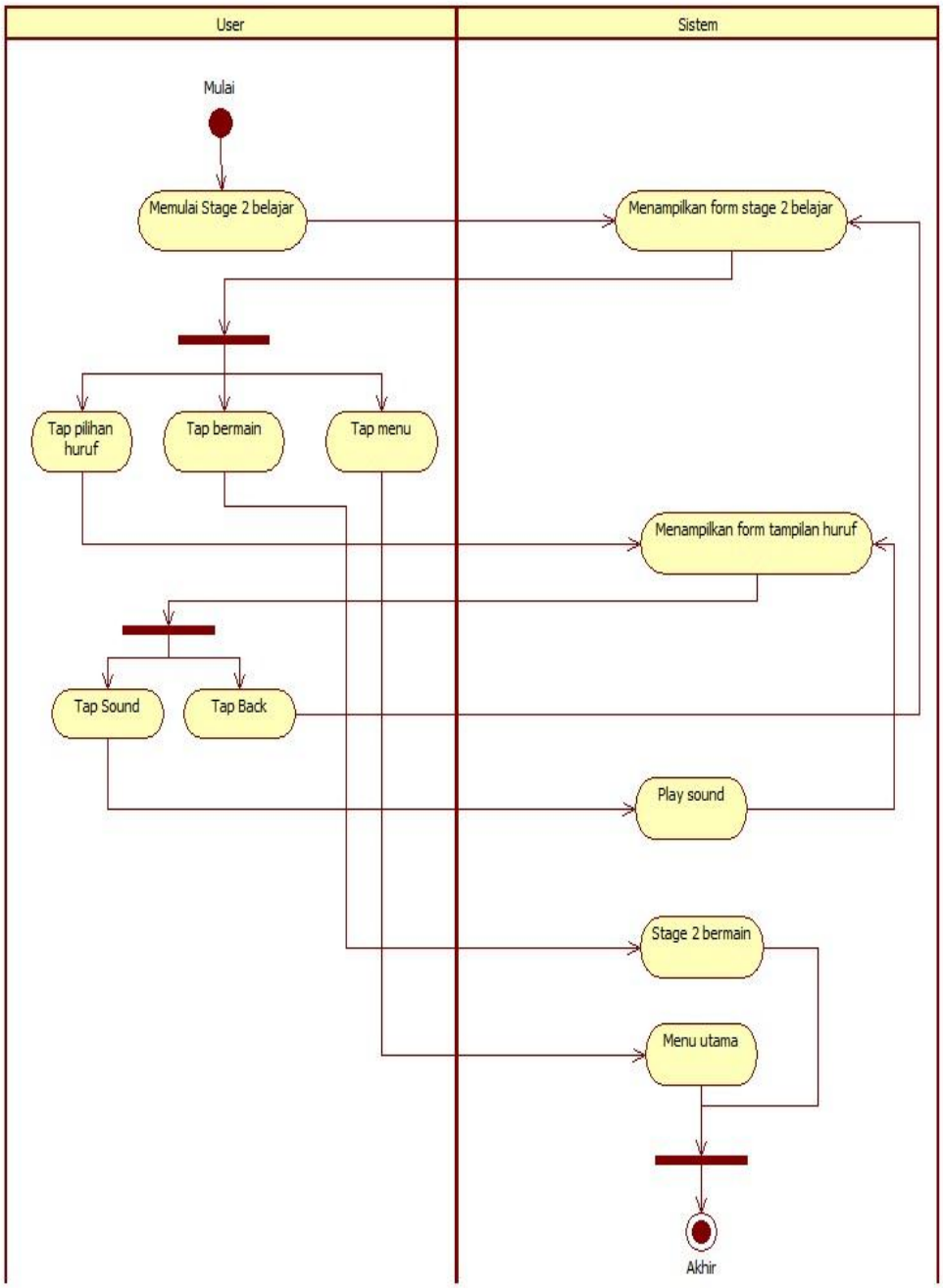
Gambar III. 14 *Activity Diagram Stage* 1 bermain



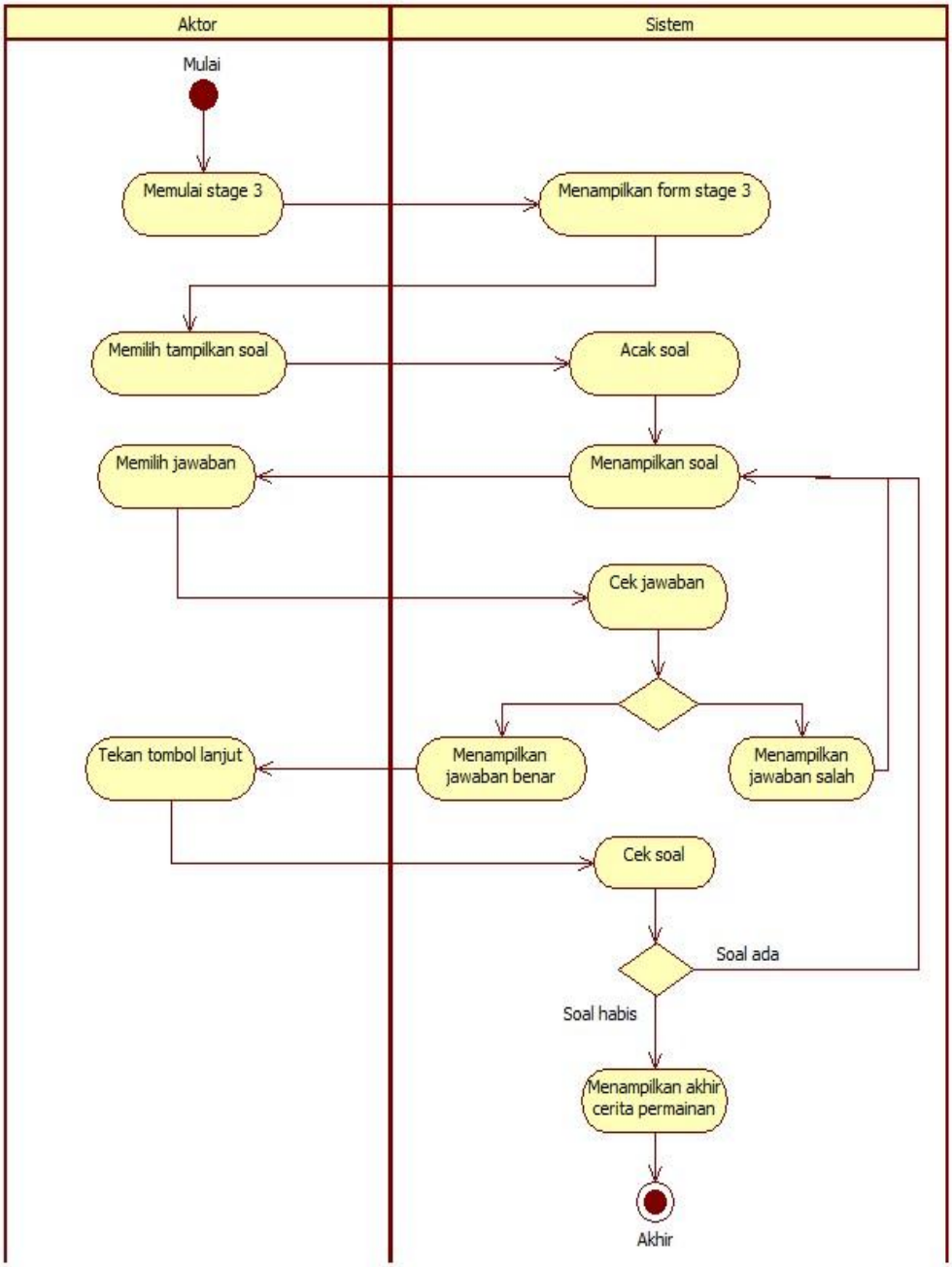
Gambar III. 15 *Activity Diagram Stage* 1 Belajar



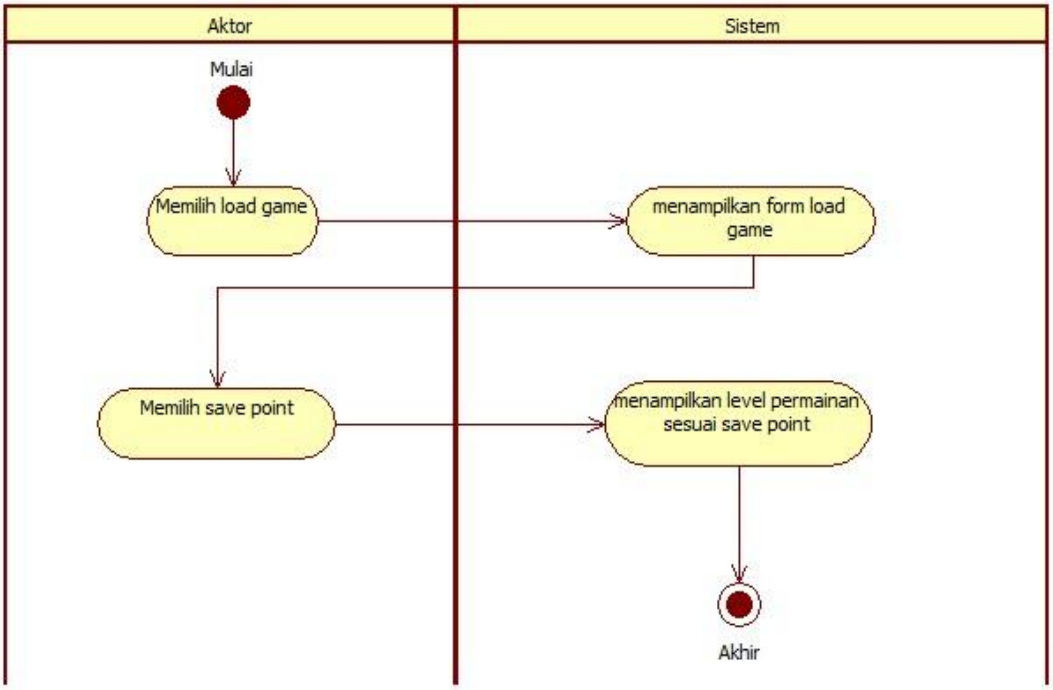
Gambar III. 16 *Activity Diagram Stage* 2 bermain



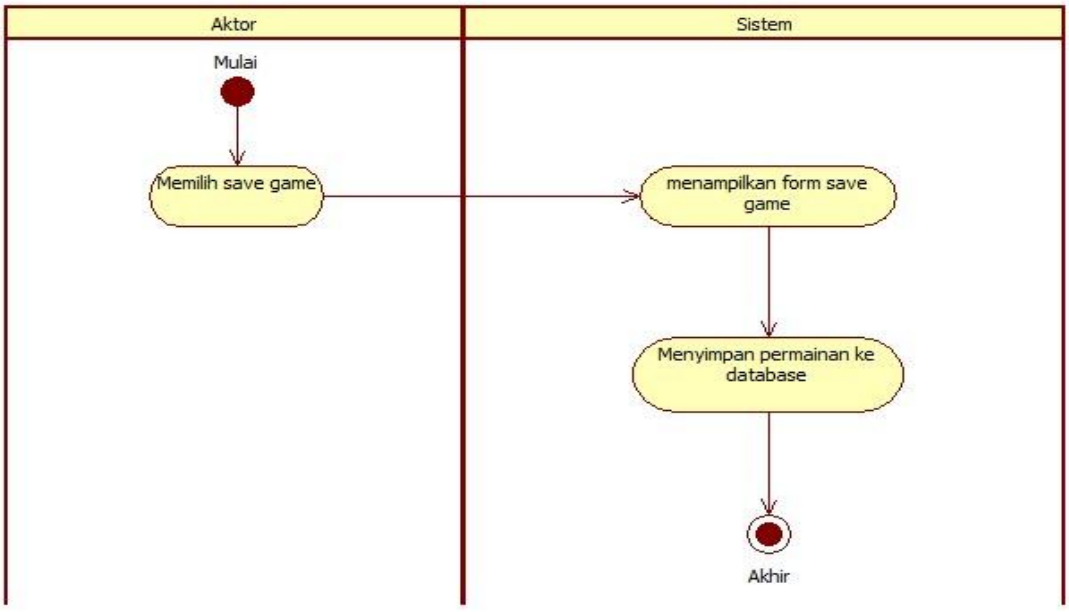
Gambar III. 17 *Activity Diagram Stage* 2 Belajar



Gambar III. 18 *Activity Diagram Stage* 3



Gambar III. 19 *Activity Diagram* Load *Game*



Gambar III. 20 Activity Diagram Save *Game*

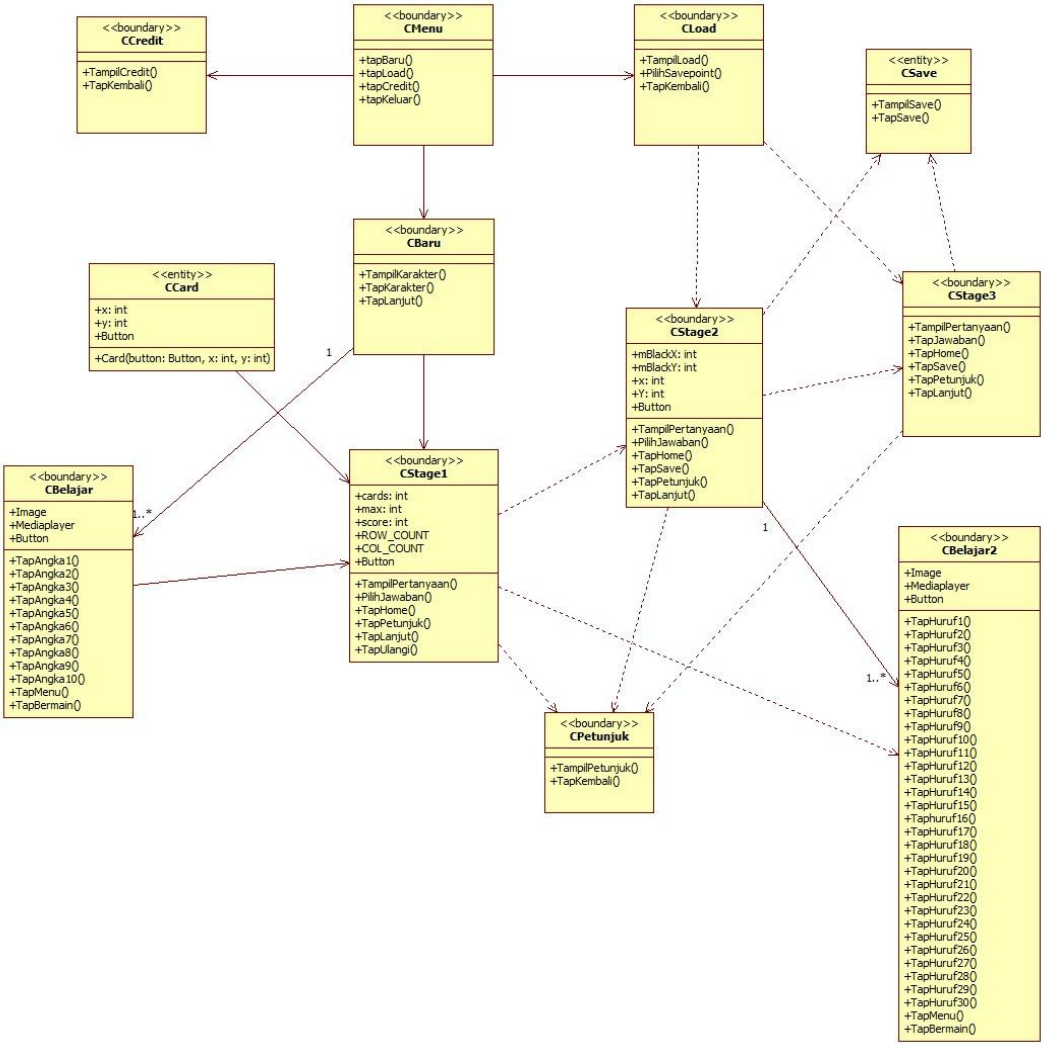
III.1.6.6. Class Diagram

Kelas digambarkan dengan *class diagram*. Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang

akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi. *Class diagram* ditunjukan pada gambar III.19. Pendefinisian kelas ditnjukan pada Tabel III.16.

Tabel III. 16 Definisi Kelas

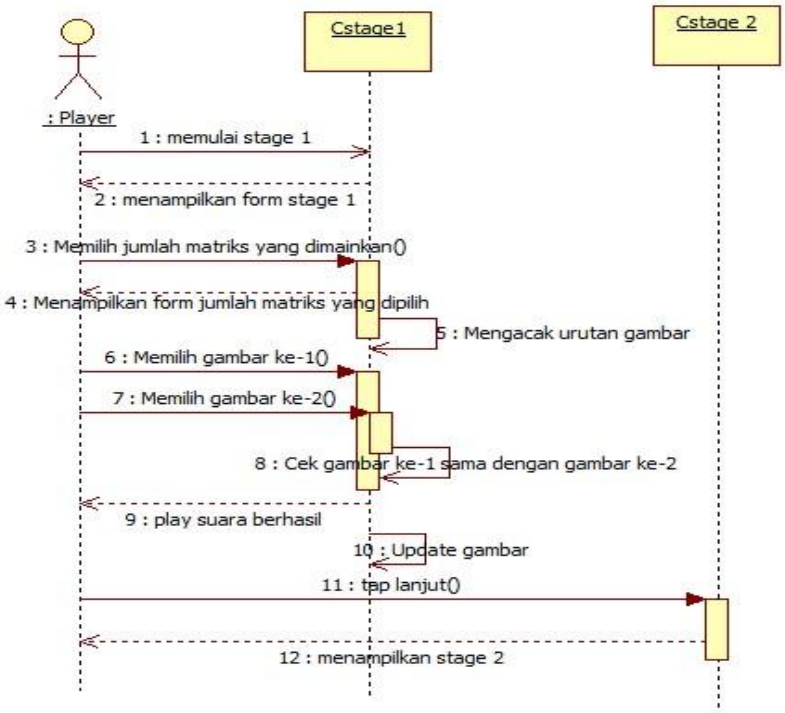
Jenis Kelas	Nama Kelas
Boundary	Cmenu
	Cbaru
	Cload
	Ccredit
	Cstage1
	Cstage2
	Cstage3
	Cpetunjuk
Entity	Ccard
	Csave



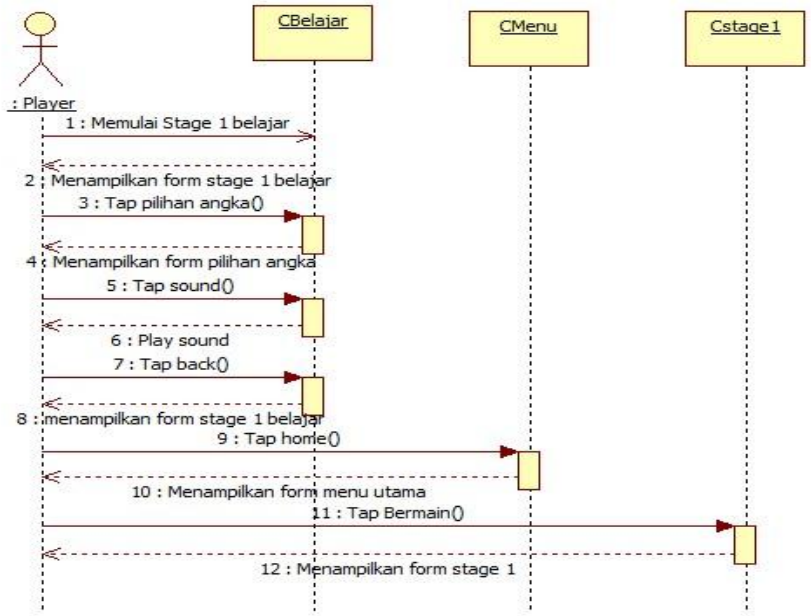
Gambar III. 21 *Class Diagram*

III.1.6.7. Penggambaran Interaksi Anter Objek Pada *Game*

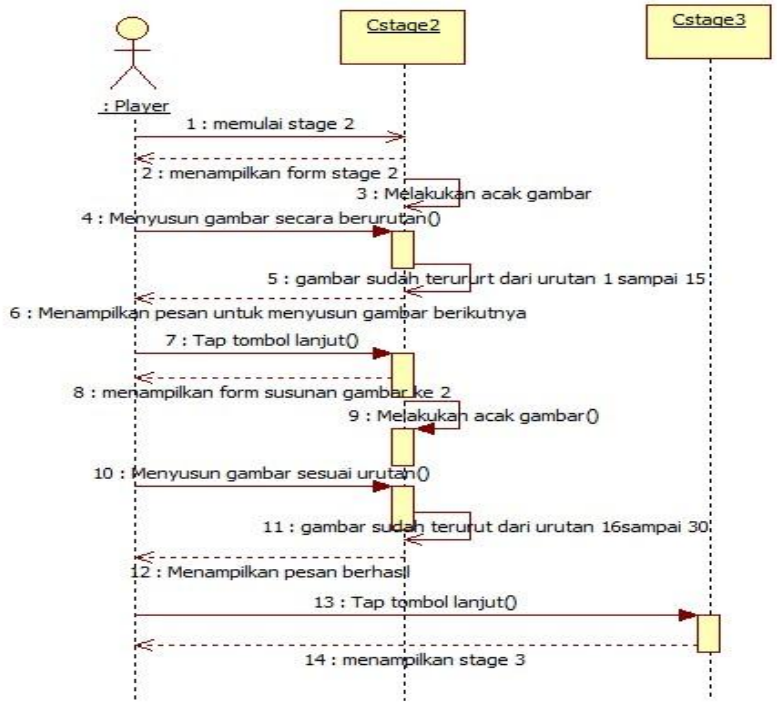
Penggambaran interaksi antar objek menggambarkan interaksi antar objek yang terjadi di dalam dan di luar sistem pada *game* (termasuk *player*, *display* dan sebagainya). Penggambaran interaksi antar objek pada *game* menggunakan *Sequence Diagram. Sequence Diagram* terdiri atas dimensi *vertical* (waktu) dan dimensi *horizontal* (objek-objek yang terkait). Pada *sequence diagram* ini menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan pada sistem sebagai respon dari sebuah *event* untuk menghasilkan *output* tertentu.



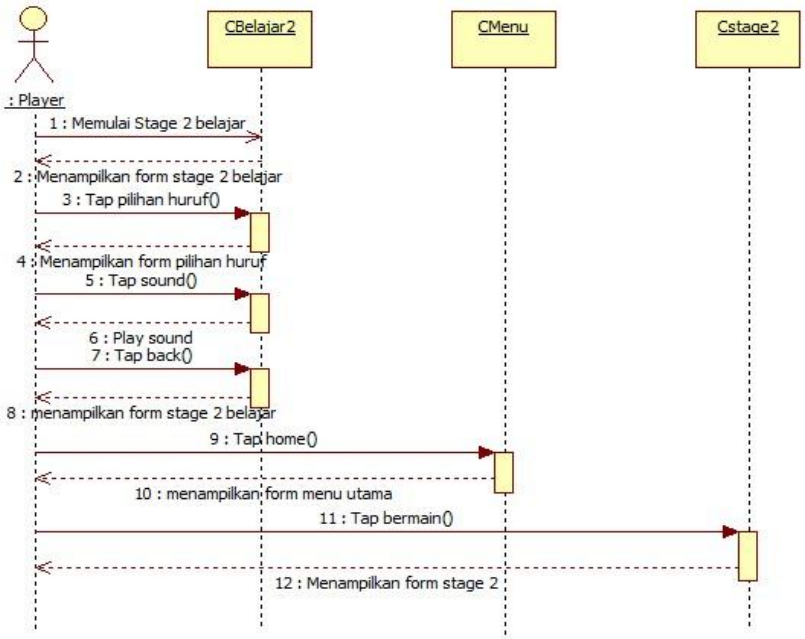
Gambar III. 22 *Sequence Diagram Stage* 1



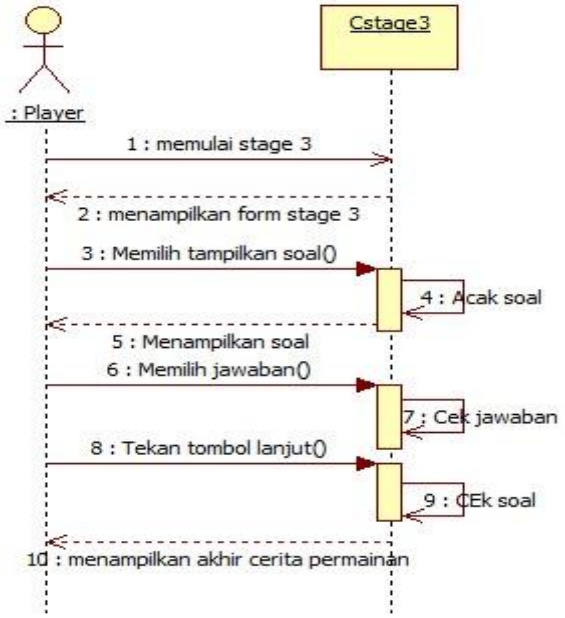
Gambar III. 23 *Sequence Diagram Stage* 1 Belajar



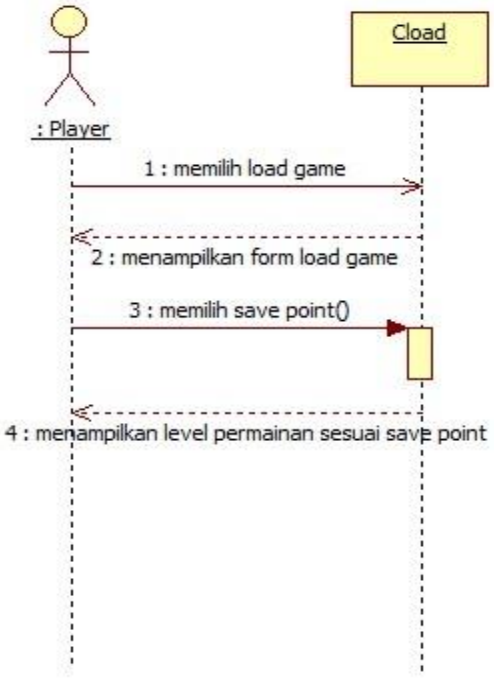
Gambar III. 24 *Sequence Diagram Stage* 2 bermain



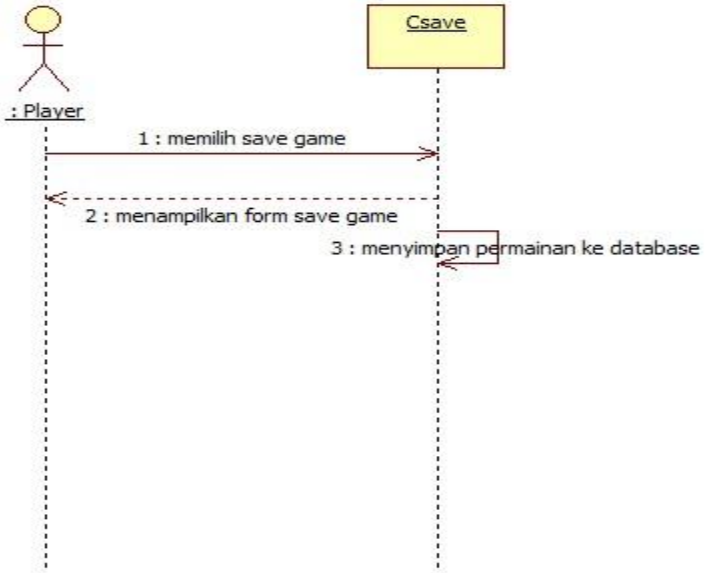
Gambar III. 25 *Sequence Diagram Stage* 2 Belajar



Gambar III. 26 *Sequence Diagram Stage* 3



Gambar III. 27 Sequence Diagram Load *Game*



Gambar III. 28 Sequence Diagram Save *Game*

III.2. Perancangan Sistem

Tahap perancangan merupakan kelanjutan dari proses analisis dimana dilakukan perubahan-perubahan terhadap sistem yang sedang berjalan. Hal ini dilakukan untuk mengatasi kekurangan yang ada, memudahkan pekerjaan yang dilakukan oleh orang yang terlibat dan menghemat waktu pekerjaan.

Perancangan *game* ini meliputi perancangan struktur menu, perancangan antarmuka, perancangan pesan, jaringan semantik, perancangan method. Semua perancangan yang telah disebutkan diatas harus saling berhubungan agar menciptakan suatu sistem yang baik.

III.2.1. Perancangan Komponen Permainan

Perancangan komponen permainan merupakan bagian pendeskripsian setiap komponen yang ada pada *game* baik karakter, bangunan, dan *story board.* Perancangan komponen permainan terdiri dari perancangan karakter dan perancangan *story board.*

III.2.1.1. Perancangan Karakter

Perancangan karakter adalah bagian dimana karakter pada *game* dijelaskan ciri-cirinya baik itu secara fungsionalitas peran maupun secara desain.

Tabel III. 17 Perancangan Karakter

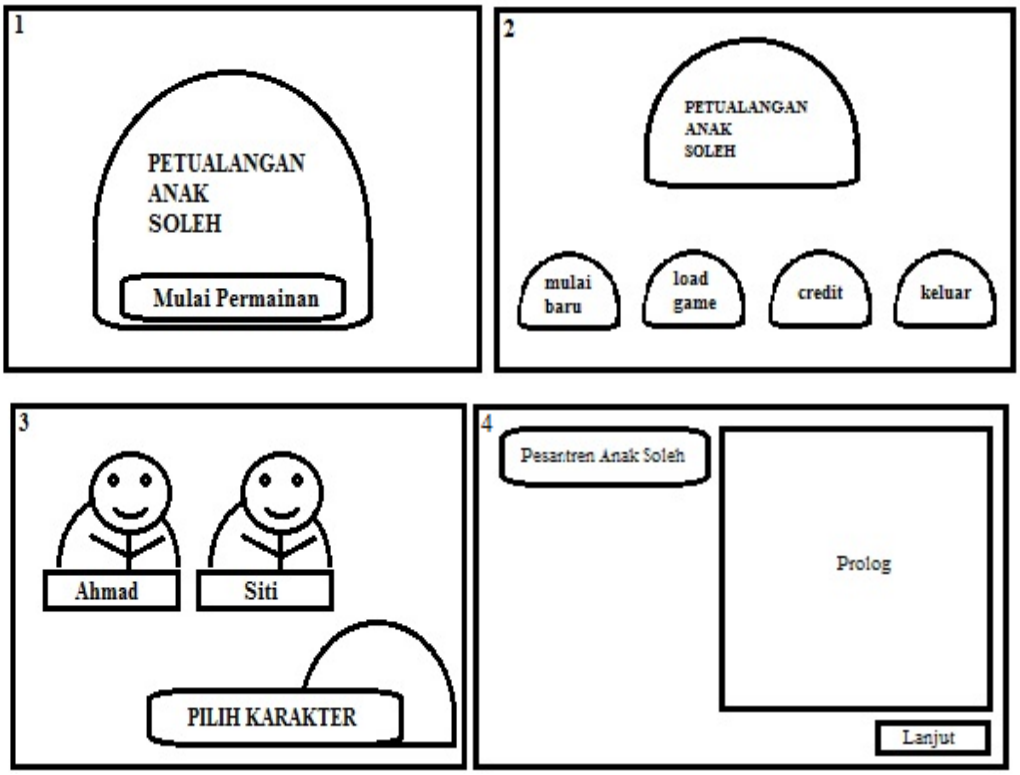
No.	Nama Karakter	Desain Karakter	Keterangan
1.	Ahmad		a) Karakter utama pada permainan b) Mempunyai sifat pemberani c) Mempunyai tujuan Ingin menjadi santri di pesantren anak soleh
2.	Siti		a) Karakter utama pada permainan b) Mempunyai sifat yang lemah lembut c) Mempunyai tujuan menjadi santri di pesantren anak soleh
3.	Ustad Anwar		a) Penjaga pada stage 1 b) Ustad paling muda di pesantren anak soleh
4.	Ustadzah Nur		a) Penjaga pada stage 2 b) Ustadzah satu-satunya di pesantren anak soleh

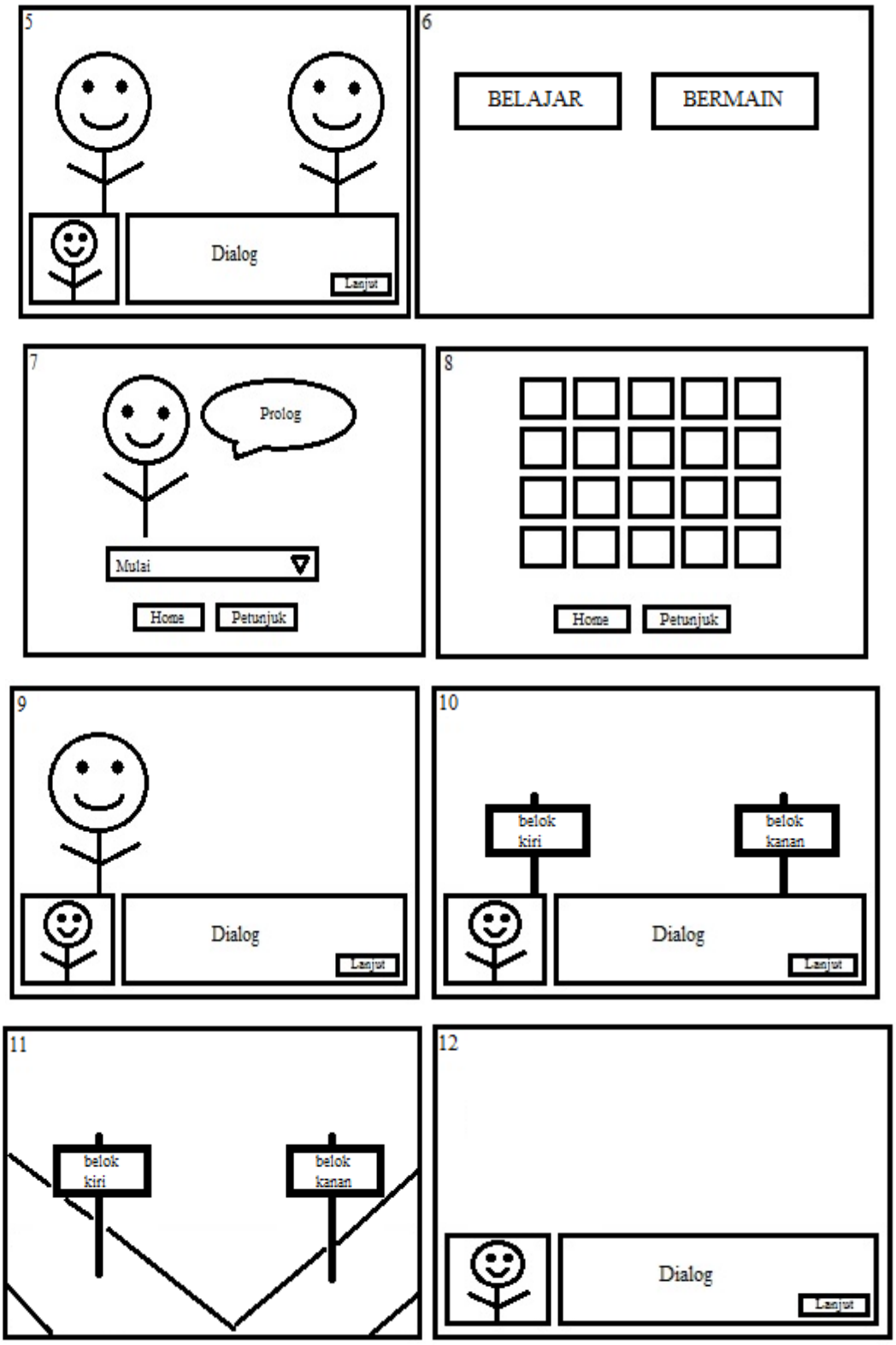
5	Haji Imron		a) Penjaga pada stage 3 b) Pemilik pesantren anak soleh

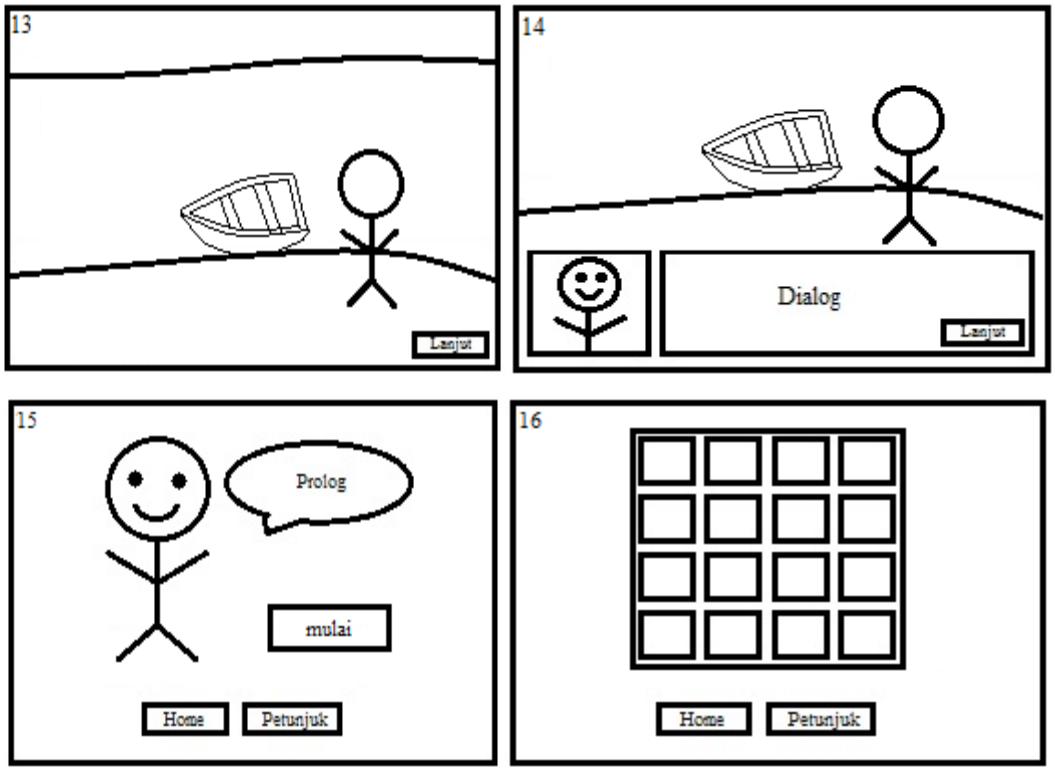
III.2.1.2. Story Board

Storyboard merupakan sketsa gambar yang disusun secara berurutan berdasarkan naskah untuk menggambarkan lingkungan pada *game*. Terdapat 4 *storyboard* pada *game* anak soleh, yaitu *storyboard* bermain arah jalan kiri, *storyboard* bermain arah jalan kanan, *storyboard* belajar angka, dan *storyboard* belajar huruf. Storyboard tersebut dapat dilihat pada gambar dibawah ini:

III.2.1.2.1. Storyboard bermain arah jalan kiri




5
Dialog
Lanjut
6
BELAJAR
BERMAIN
7
Prolog
Mulai
Home
Petunjuk
8
Home
Petunjuk
9
Dialog
Lanjut
10
belok kiri
belok kanan
Dialog
Lanjut
11
belok kiri
belok kanan
12
Dialog
Lanjut



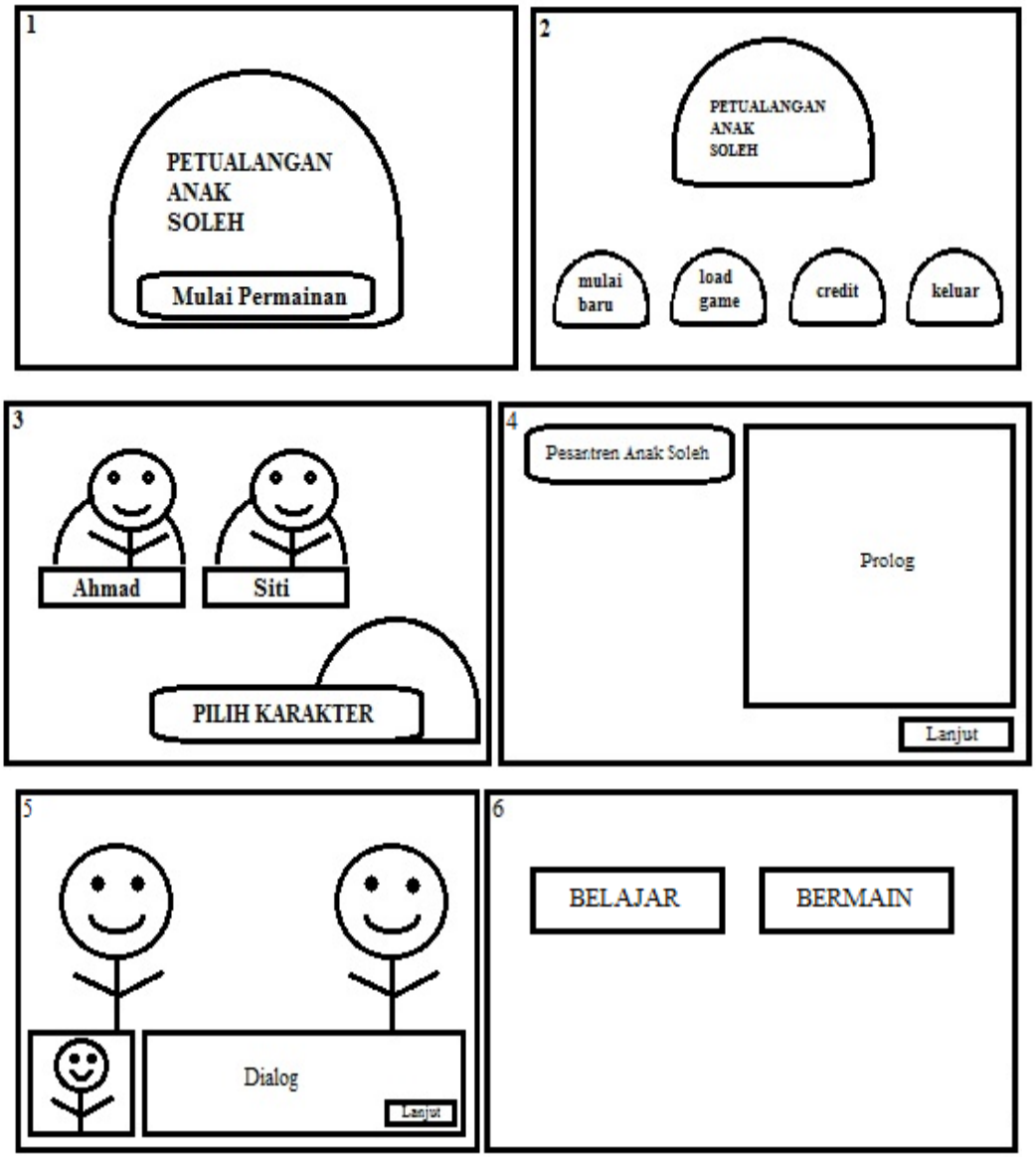
Gambar III. 29 *Storyboard* bermain arah jalan kiri

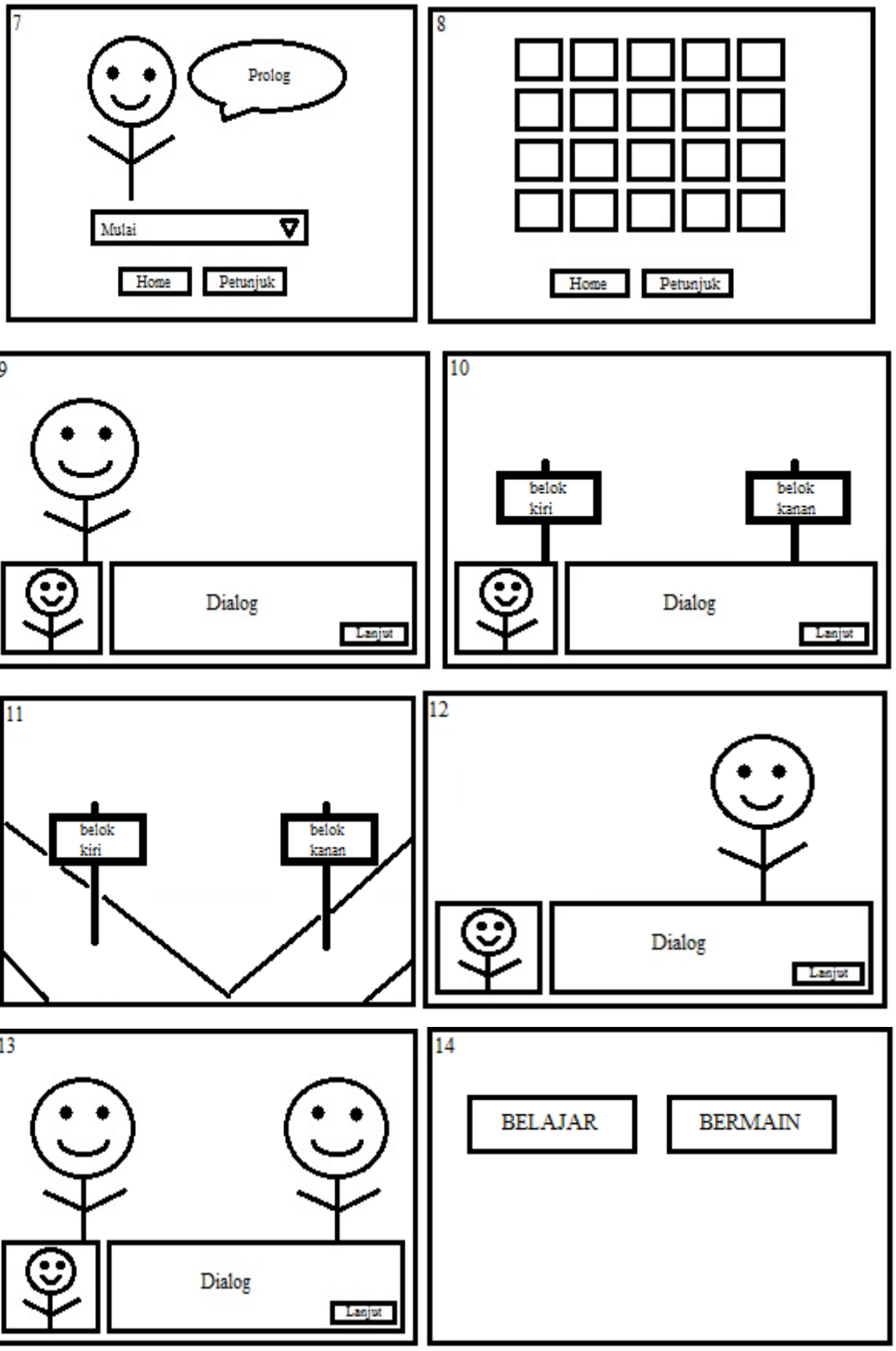
Penjelasan gambar dari storyboard arah jalan kiri adalah sebagai berikut :

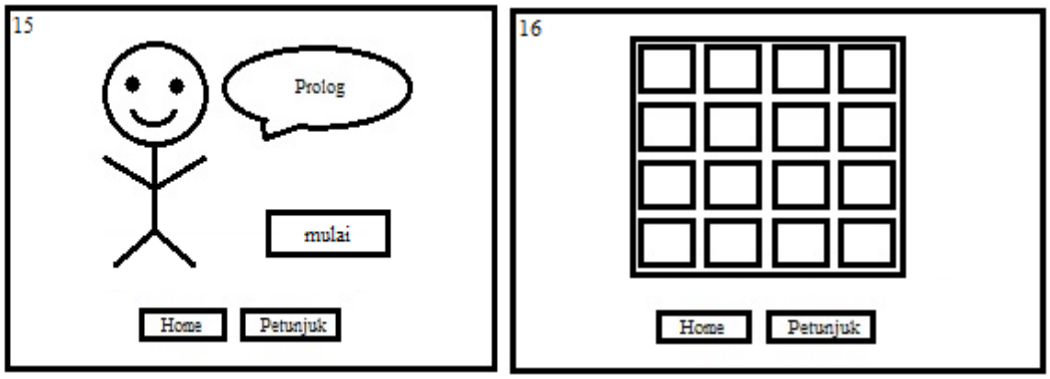
1. Gambar menu awal permainan petualangan anak soleh.
2. Gambar menu utama menampilkan menu mulai baru, load *game*, credit, dan keluar.
3. Gambar menu pilih karakter, untuk memilih karakter yang akan dimainkan. Ada dua karakter yang dapat dipilih yaitu ahmad dan siti.
4. Gambar prolog dari pesantren anak soleh.
5. Gambar dialog antara karakter dan penjaga stage pertama.
6. Gambar menu pilihan belajar dan bermain. Player memilih menu bermain
7. Gambar prolog penjelasan tentang cara bermain di stage 1.
8. Gambar permainan dari stage 1
9. Gambar prolog setetlah permainan stage 1 selesai
10. Gambar dialog untuk memilih jalan yang akan dilewati
11. Gambar memilih jalan yang akan dilewati, terdapat dua pilihan yaitu memilih jalan ke kiri atau jalan ke kanan. Player memilih jalan ke kiri

12. Gambar dialog setelah memilih jalan ke kiri
13. Gambar karakter menemukan sebuah perahu di sungai
14. Gambar dialog karakter setelah menemukan perahu di sungai
15. Gambar prolog penjelasan tentang cara bermain di stage 2
16. Gambar permainan dari stage 2.

III.2.1.2.2. Storyboard bermain arah jalan kanan




7
Prolog
Mulai
Home
Petunjuk
8
Home
Petunjuk
9
Dialog
Lanjut
10
belok kiri
belok kanan
Dialog
Lanjut
11
belok kiri
belok kanan
12
Dialog
Lanjut
13
Dialog
Lanjut
14
BELAJAR
BERMAIN

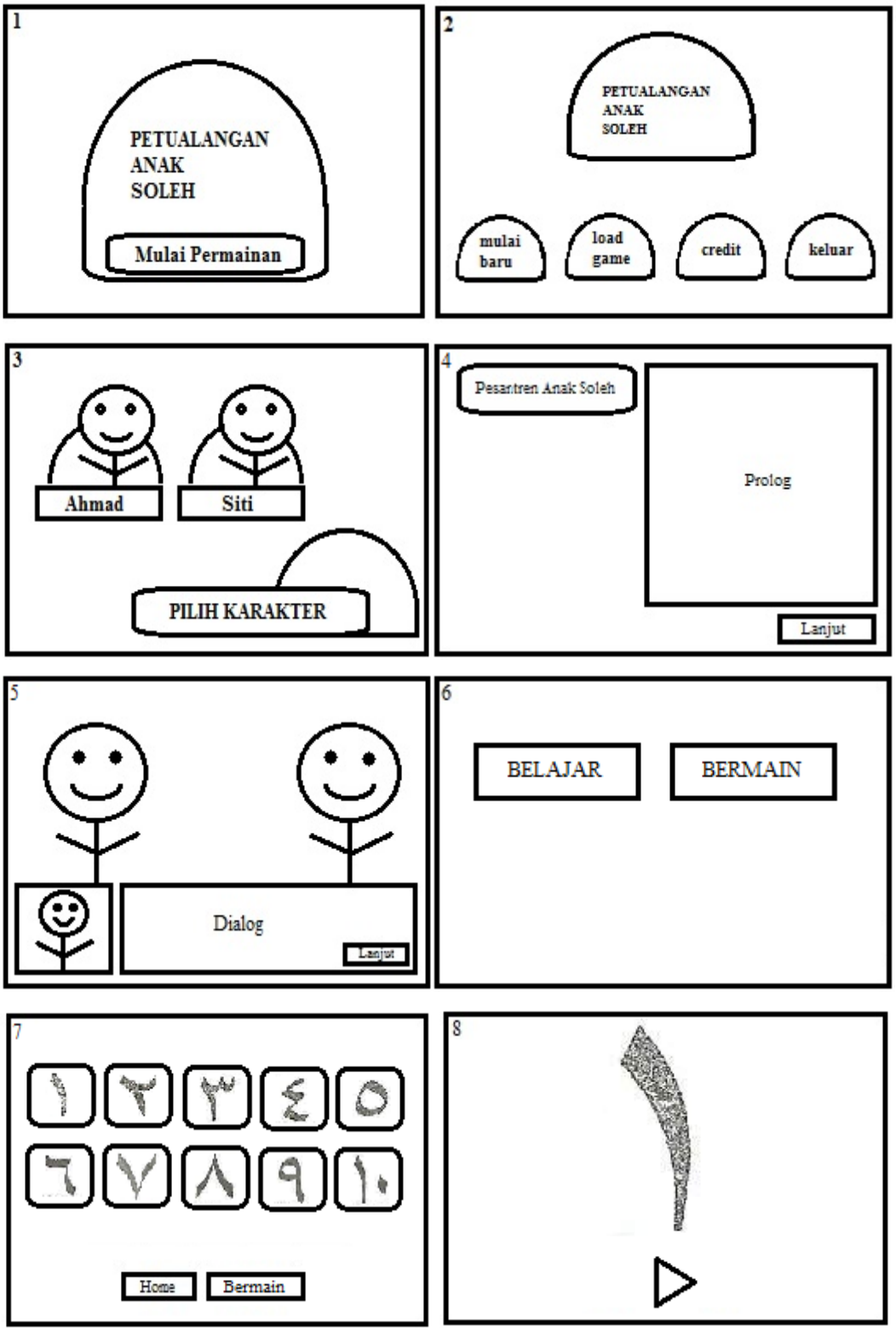


Gambar III. 30 *Storyboard* bermain arah jalan kanan

Penjelasan gambar dari storyboard arah jalan kanan adalah sebagai berikut :

1. Gambar menu awal permainan petualangan anak soleh.
2. Gambar menu utama menampilkan menu mulai baru, load *game*, credit, dan keluar.
3. Gambar menu pilih karakter, untuk memilih karakter yang akan dimainkan. Ada dua karakter yang dapat dipilih yaitu ahmad dan siti.
4. Gambar prolog dari pesantren anak soleh.
5. Gambar dialog antara karakter dan penjaga stage pertama.
6. Gambar menu pilihan belajar dan bermain. Player memilih pilihan bermain
7. Gambar prolog penjelasan tentang cara bermain di stage 1.
8. Gambar permainan dari stage 1
9. Gambar prolog setetlah permainan stage 1 selesai
10. Gambar dialog untuk memilih jalan yang akan dilewati
11. Gambar memilih jalan yang akan dilewati, terdapat dua pilihan yaitu memilih jalan ke kiri atau jalan ke kanan. Player memilih jalan ke kanan
12. Gambar dialog karakter setelah memilih arah jalan ke kanan
13. Gambar dialog antara karakter dengan penjaga stage 2
14. Gambar menu pilihan belajar dan bermain. Player memilih menu bermain
15. Gambar prolog penjelasan tentang cara bermain di stage 2
16. Gambar permainan dari stage 2

III.2.1.2.3. Storyboard belajar angka

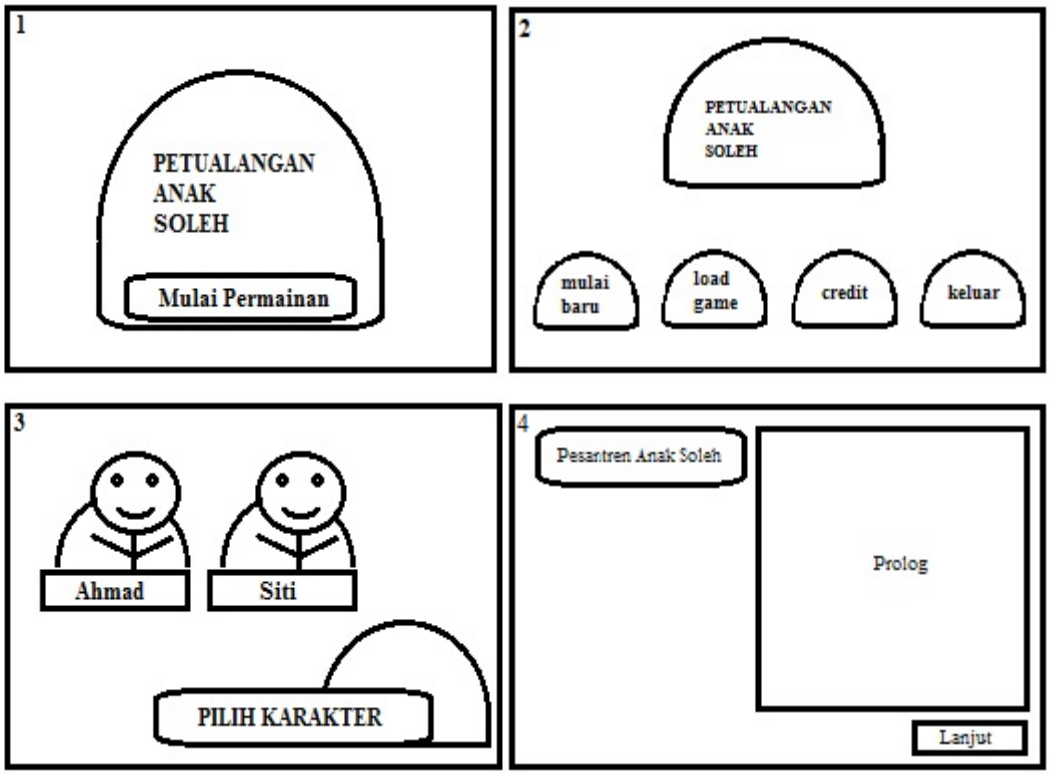


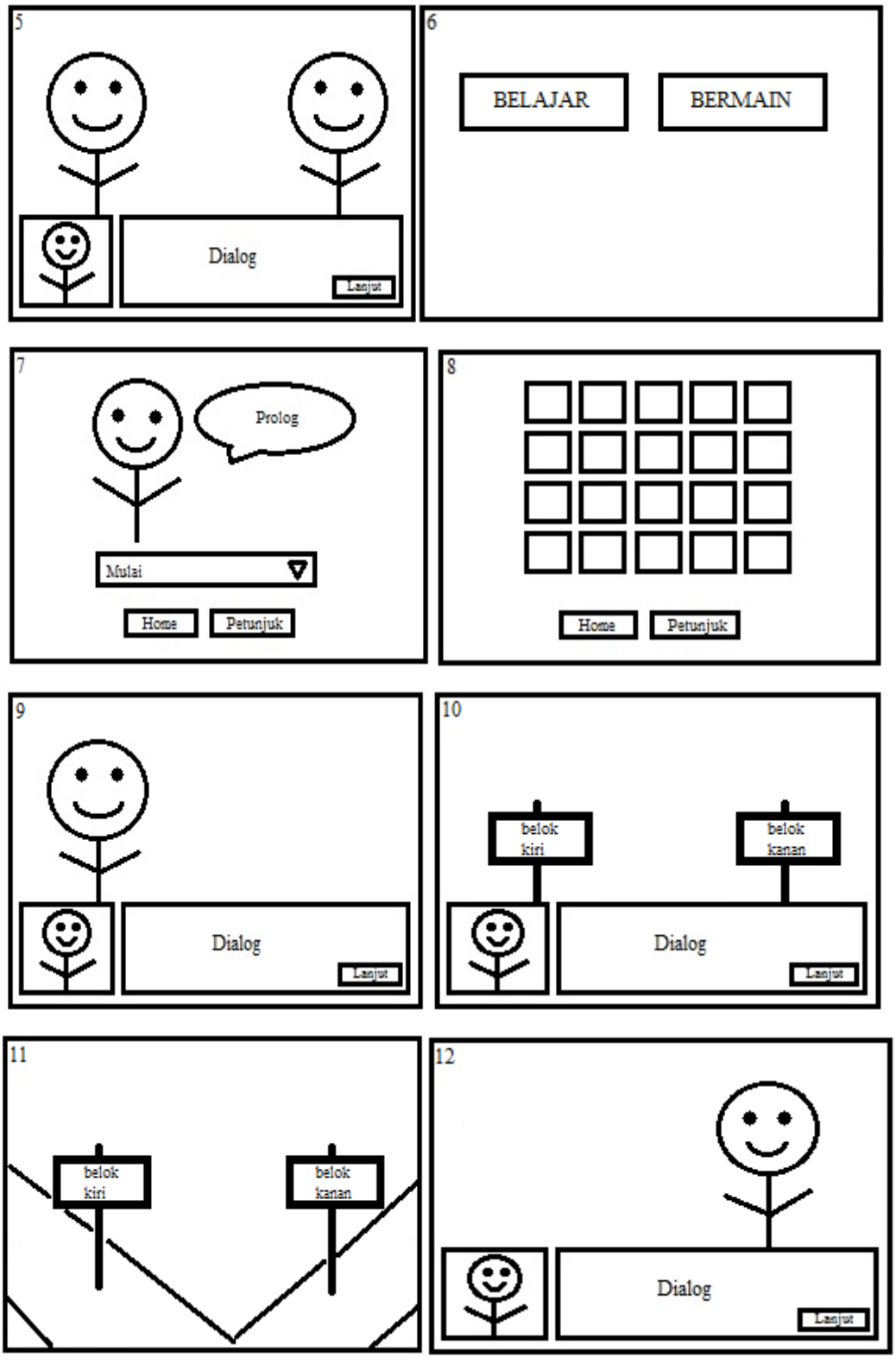
Gambar III. 31 Storyboard belajar angka

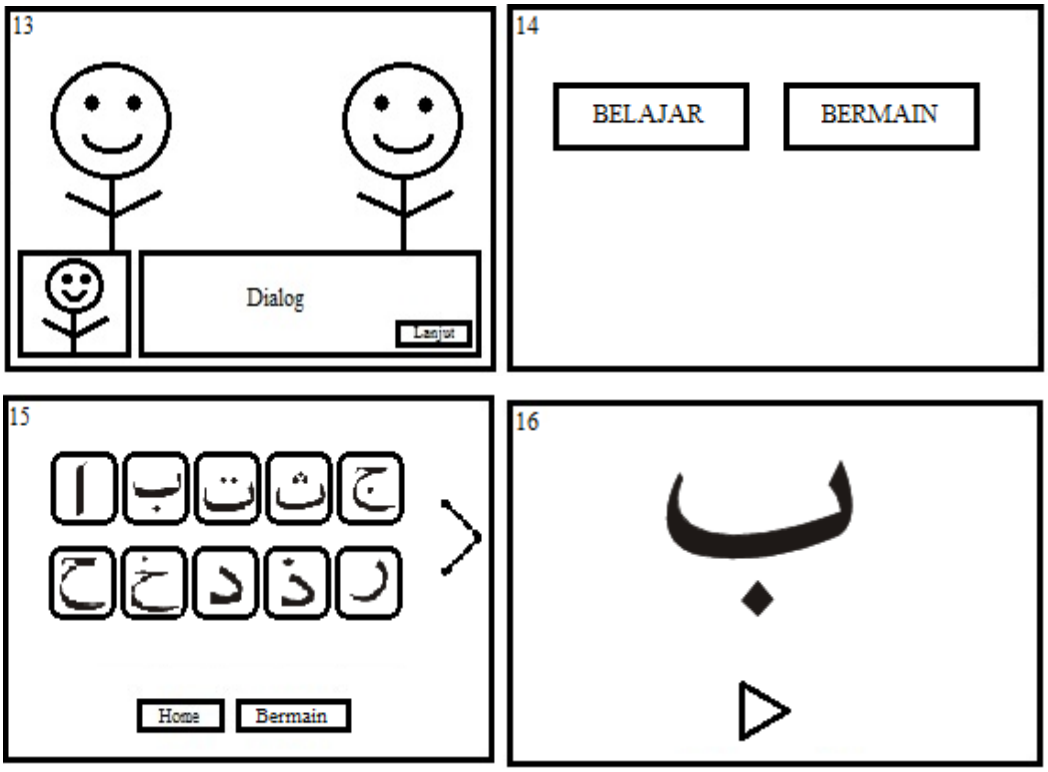
Penjelasan gambar dari storyboard belajar angka adalah sebagai berikut :

1. Gambar menu awal permainan petualangan anak soleh.
2. Gambar menu utama menampilkan menu mulai baru, load *game*, credit, dan keluar.
3. Gambar menu pilih karakter, untuk memilih karakter yang akan dimainkan. Ada dua karakter yang dapat dipilih yaitu ahmad dan siti.
4. Gambar prolog dari pesantren anak soleh.
5. Gambar dialog antara karakter dan penjaga stage pertama.
6. Gambar menu pilihan belajar dan bermain. Player memilih menu belajar
7. Gambar pilihan angka dari angka 1 sampai 10.
8. Gambar angka hasil pilihan player

III.2.1.2.4. Storyboard belajar huruf




5
Dialog
Lanjut
6
BELAJAR
BERMAIN
7
Prolog
Mulai
Home
Petunjuk
8
Home
Petunjuk
9
Dialog
Lanjut
10
belok kiri
belok kanan
Dialog
Lanjut
11
belok kiri
belok kanan
12
Dialog
Lanjut



Gambar III. 32 Storyboard belajar huruf

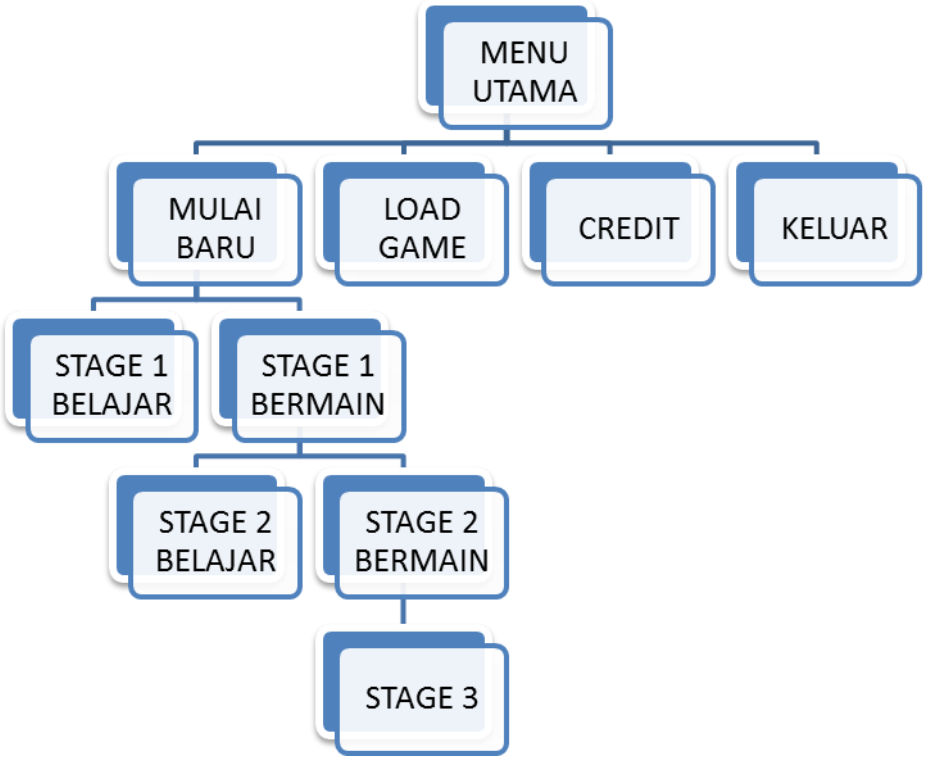
Penjelasan gambar dari storyboard belajar huruf adalah sebagai berikut :

1. Gambar menu awal permainan petualangan anak soleh.
2. Gambar menu utama menampilkan menu mulai baru, load *game*, credit, dan keluar.
3. Gambar menu pilih karakter, untuk memilih karakter yang akan dimainkan. Ada dua karakter yang dapat dipilih yaitu ahmad dan siti.
4. Gambar prolog dari pesantren anak soleh.
5. Gambar dialog antara karakter dan penjaga stage pertama.
6. Gambar menu pilihan belajar dan bermain. Player memilih pilihan bermain
7. Gambar prolog penjelasan tentang cara bermain di stage 1.
8. Gambar permainan dari stage 1
9. Gambar prolog setetlah permainan stage 1 selesai
10. Gambar dialog untuk memilih jalan yang akan dilewati
11. Gambar memilih jalan yang akan dilewati, terdapat dua pilihan yaitu memilih jalan ke kiri atau jalan ke kanan. Player memilih jalan ke kanan

12. Gambar dialog karakter setelah memilih arah jalan ke kanan
13. Gambar dialog antara karakter dengan penjaga stage 2
14. Gambar menu pilihan belajar dan bermain. Player memilih menu belajar
15. Gambar pilihan huruf dari huruf pertama sampai huruf terakhir
16. Gambar huruf hasil pilihan player

III.2.2. Perancangan Struktur Menu

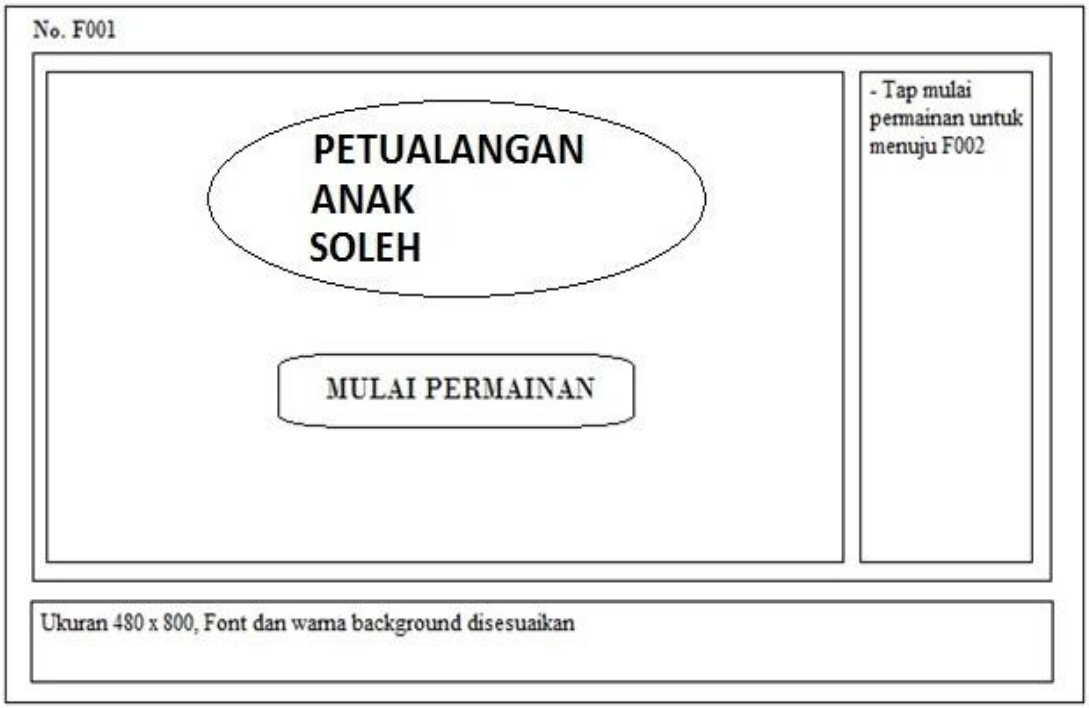
Dalam pembangunan aplikasi diperlukan sebuah struktur menu untuk menunjukan keterkaitan dan tingkat antara menu utama dan sub menu, sehingga dapat lebih mudah dimengerti. Berikut struktur menu dari *game* yang akan dibangun :



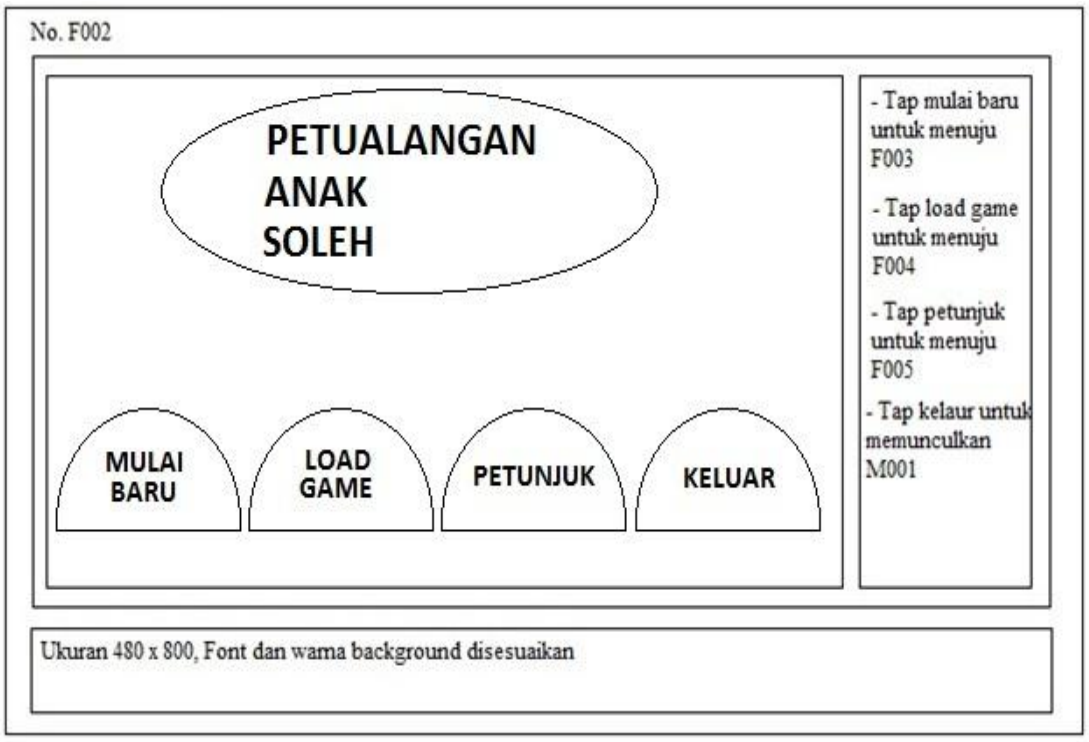
Gambar III. 33 Struktur Menu

III.2.3. Perancangan Antar Muka

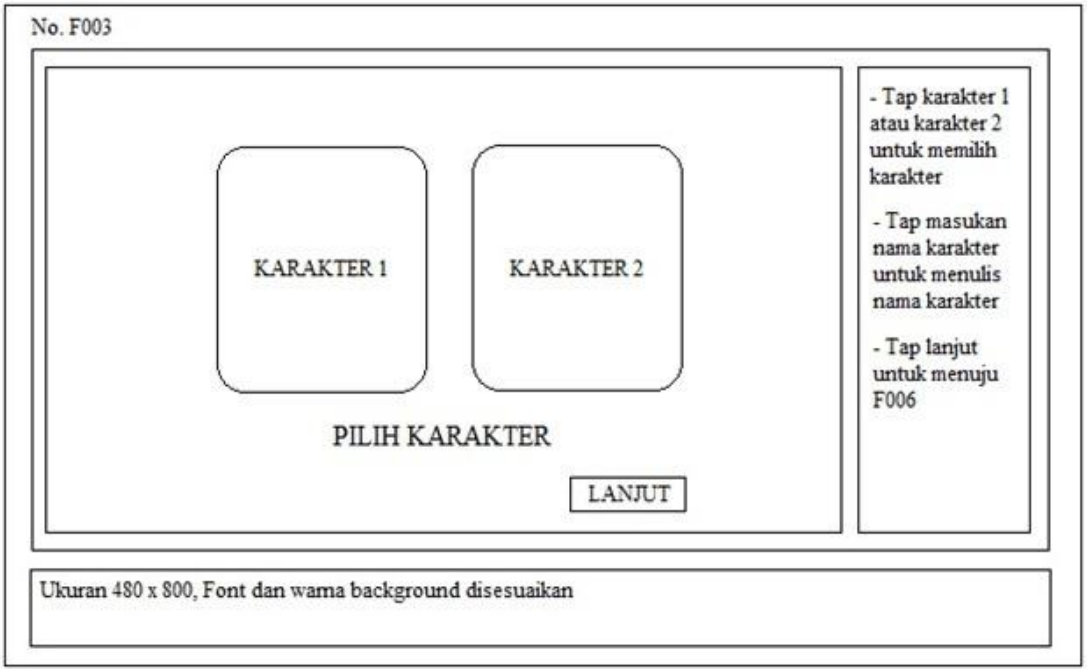
Perancangan antarmuka merupakan salah satu bagian penting dalam perancangan sistem karena nantinya antarmuka tersebut akan menjadi fasilitas yang menjembatani interaksi manusia dengan siistem. Perancangan antarmuka yang akan dibangun pada *game* ini adalah sebagai berikut:



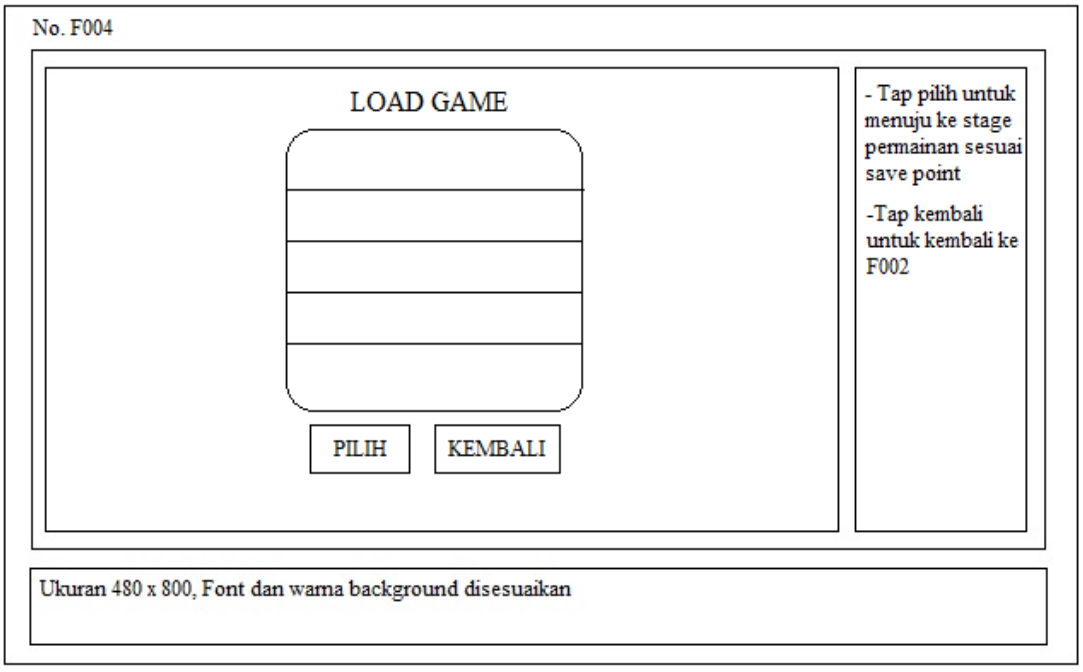
Gambar III. 34 Tampilan Mulai Permainan



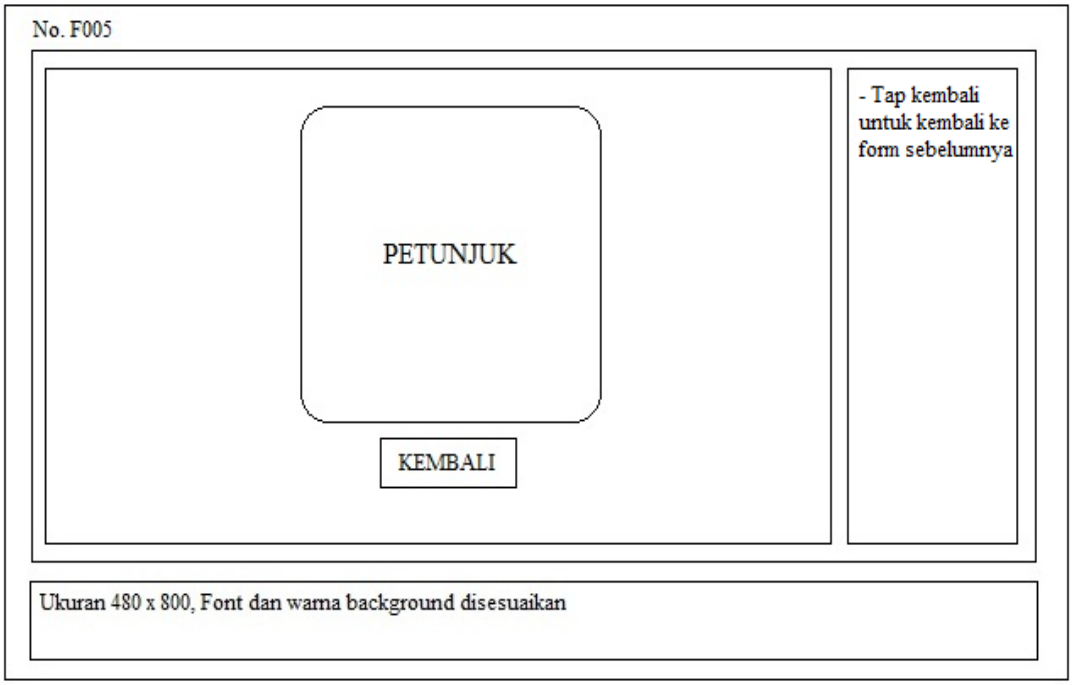
Gambar III. 35 Tampilan Menu



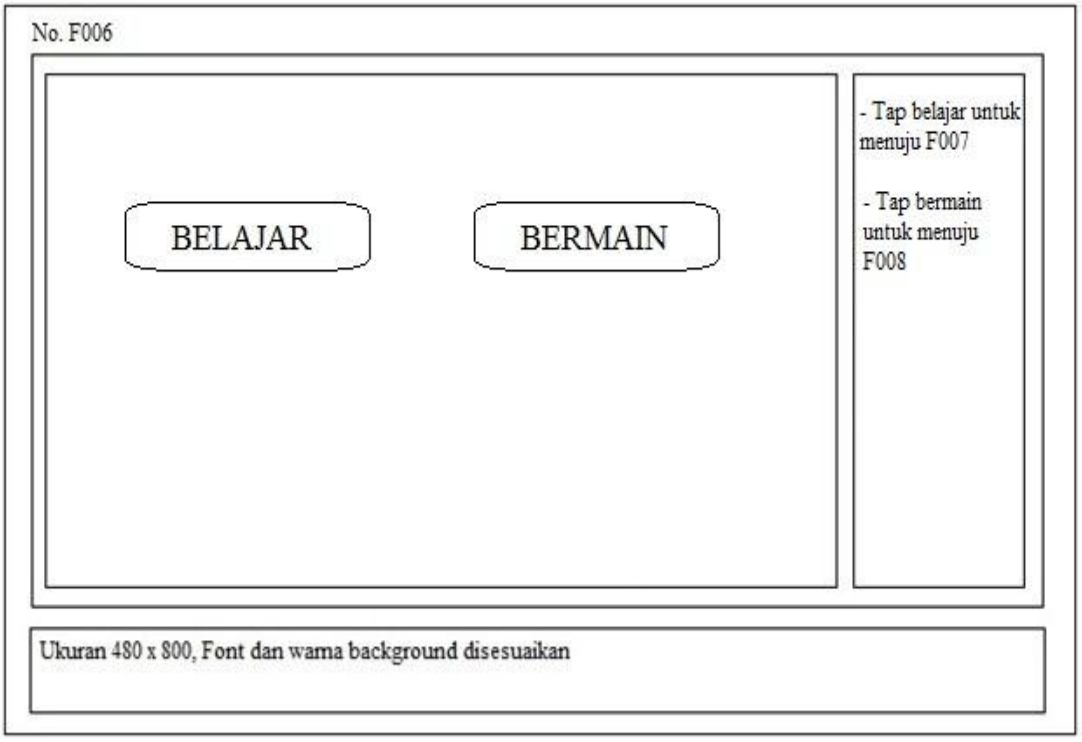
Gambar III. 36 Tampilan Pilih Karakter



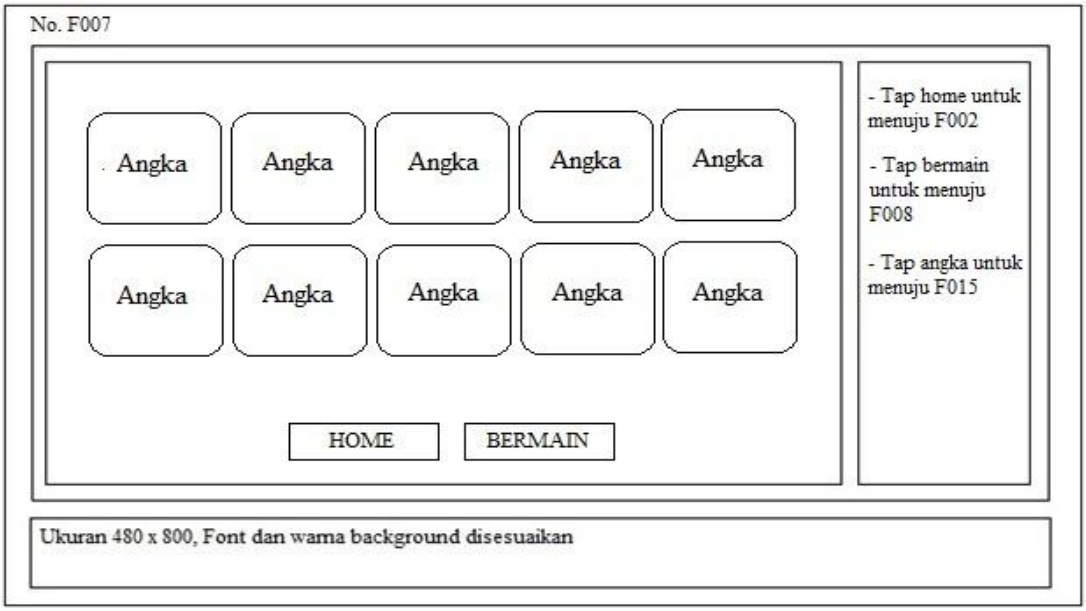
Gambar III. 37 Tampilan Load *Game*



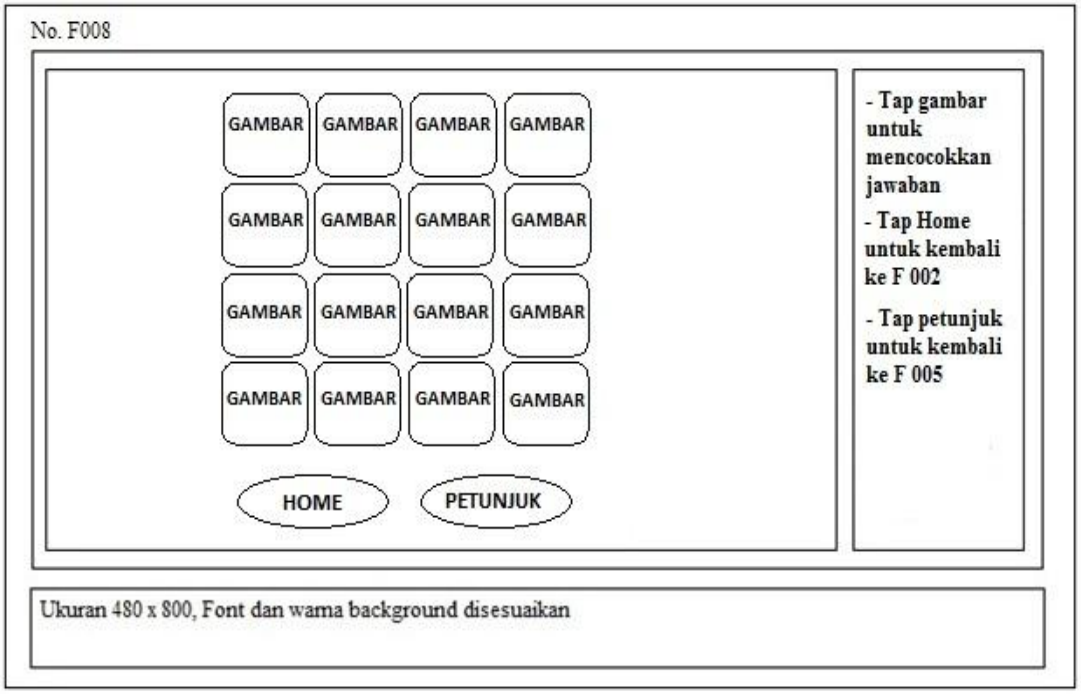
Gambar III. 38 Tampilan Petunjuk



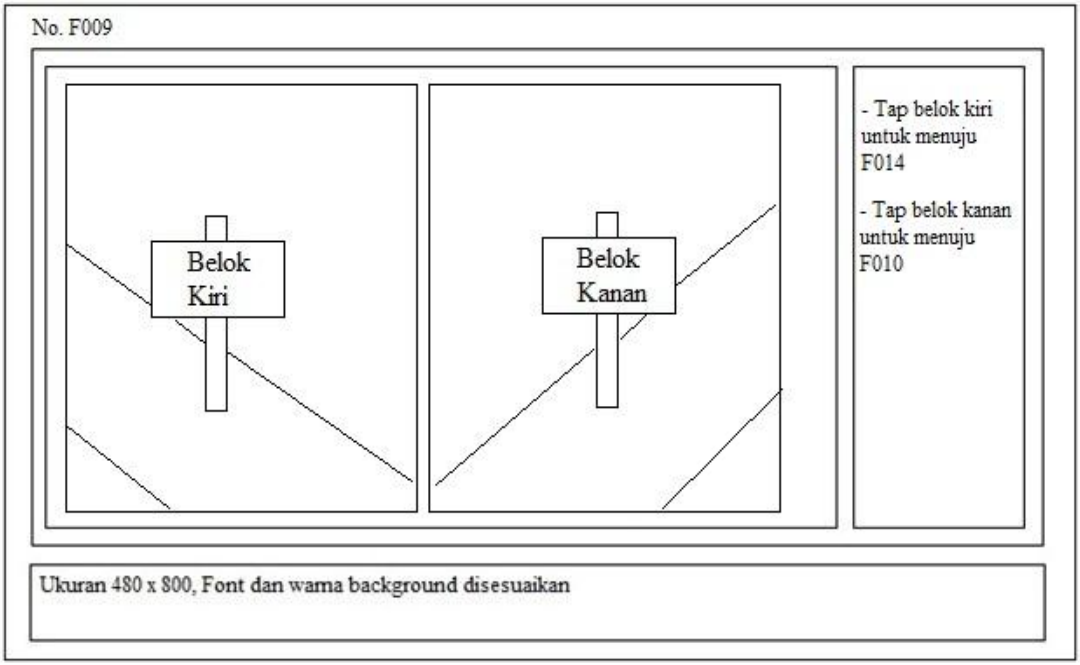
Gambar III. 39 Tampilan menu pilihan belajar dan bermain stage 1



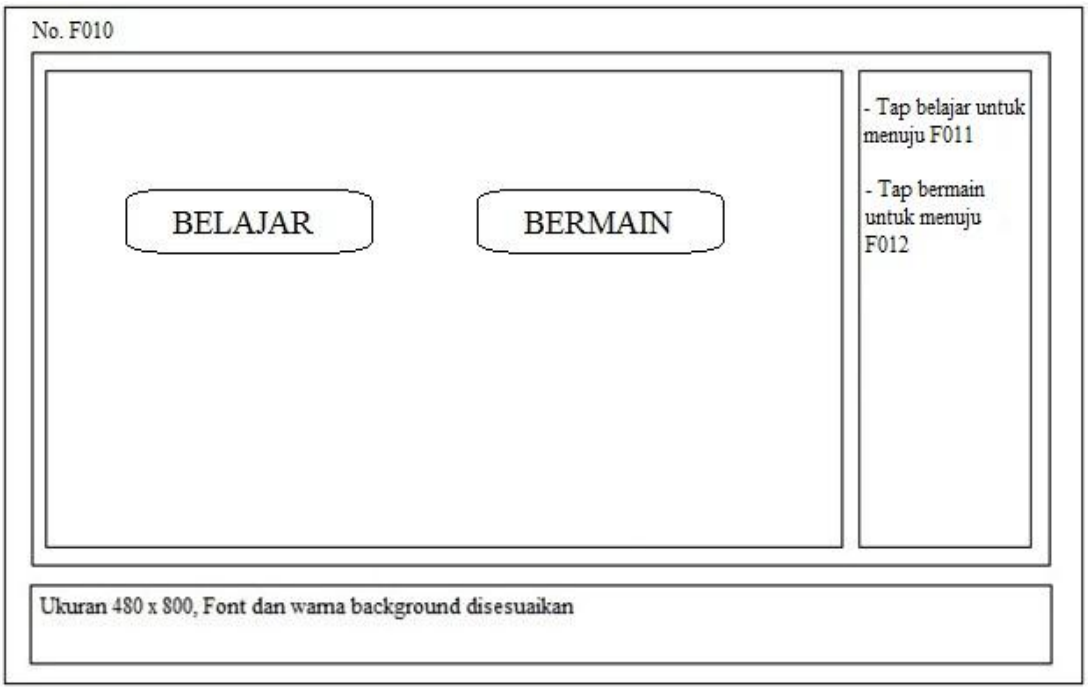
Gambar III. 40 Tampilan belajar stage 1



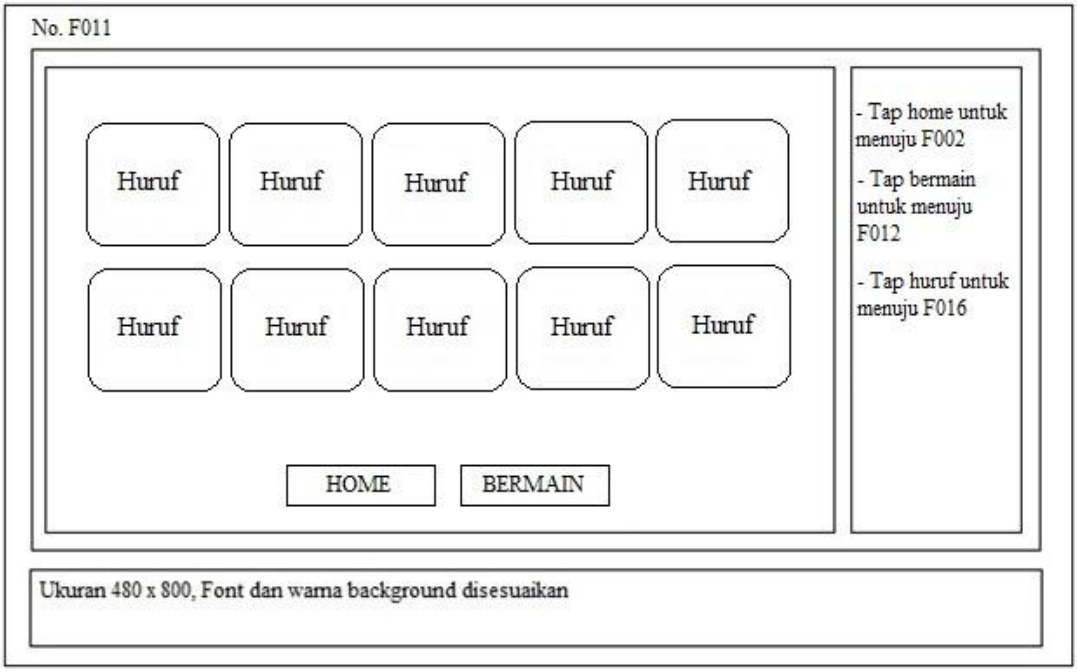
Gambar III. 41 Tampilan bermain stage 1



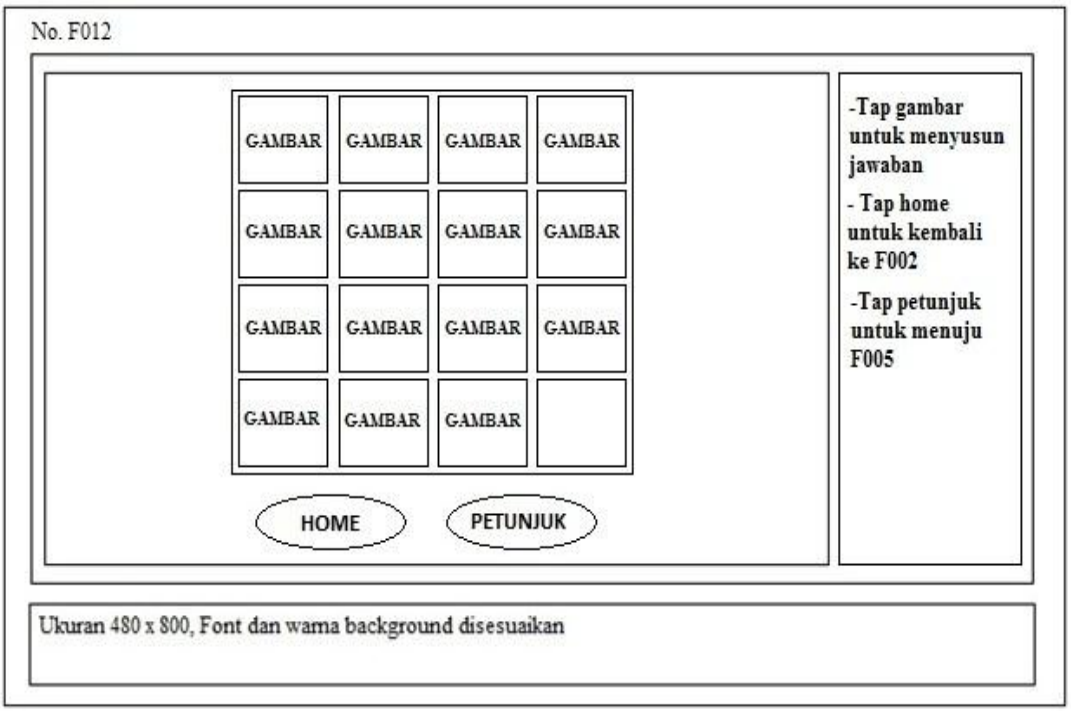
Gambar III. 42 Tampilan pilih arah jalan



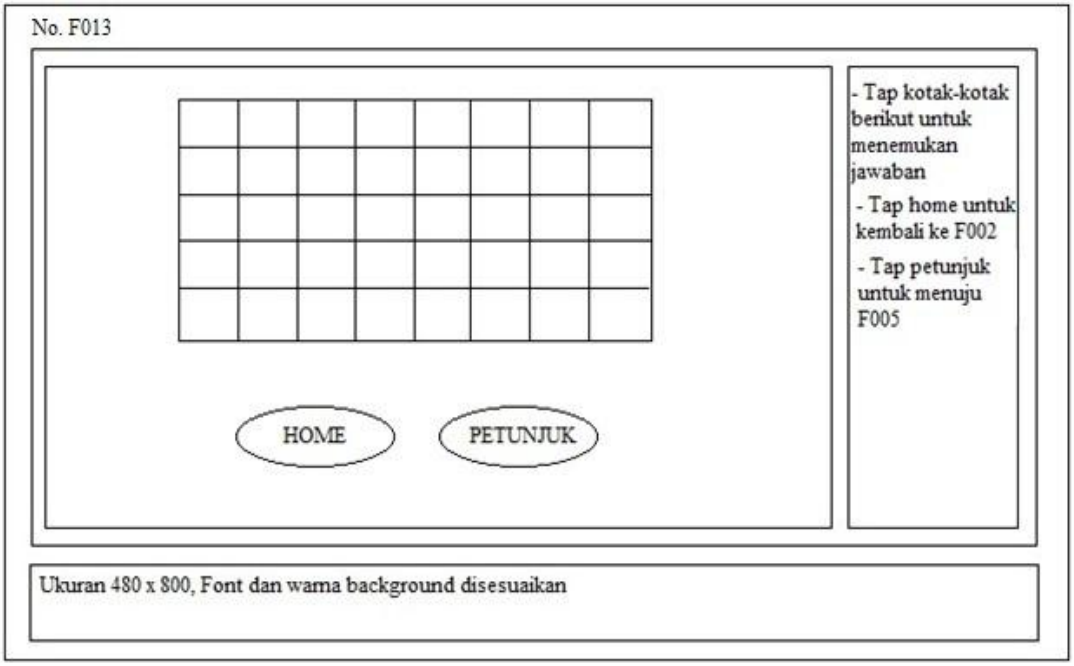
Gambar III. 43 Tampilan menu pilihan belajar dan bermain stage 2



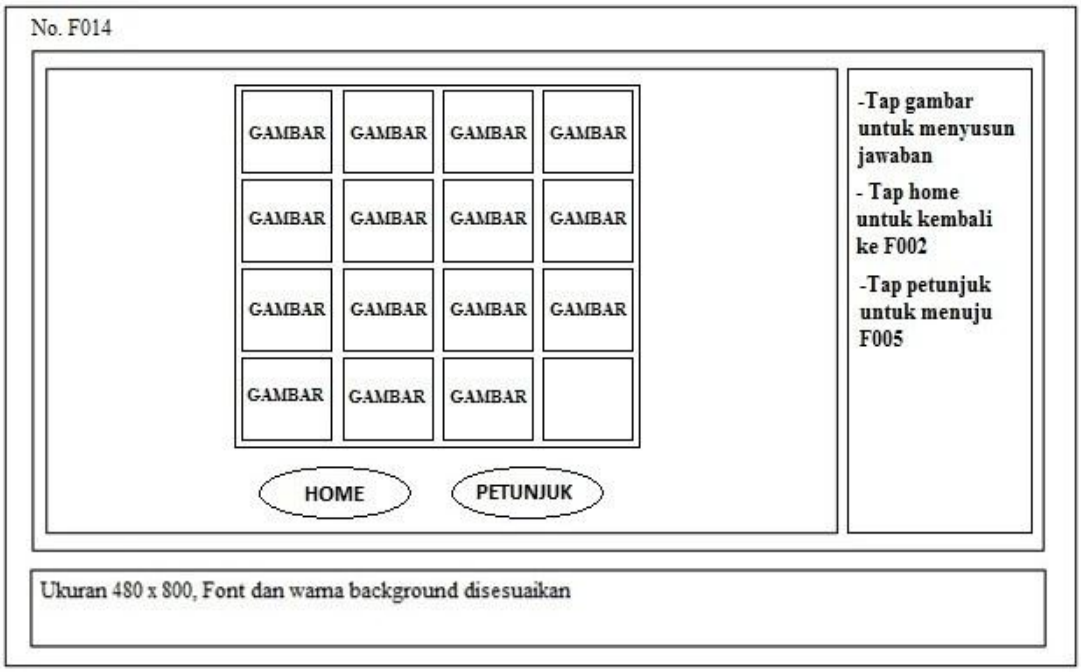
Gambar III. 44 Tampilan belajar stage 2



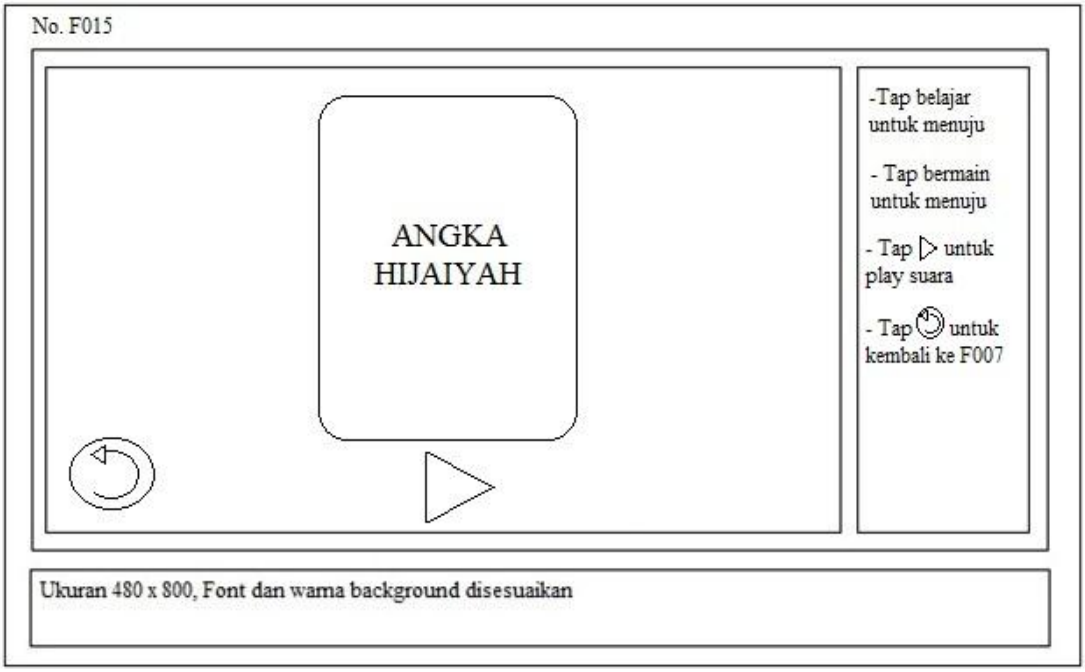
Gambar III. 45 Tampilan bermain stage 2 arah jalan kanan



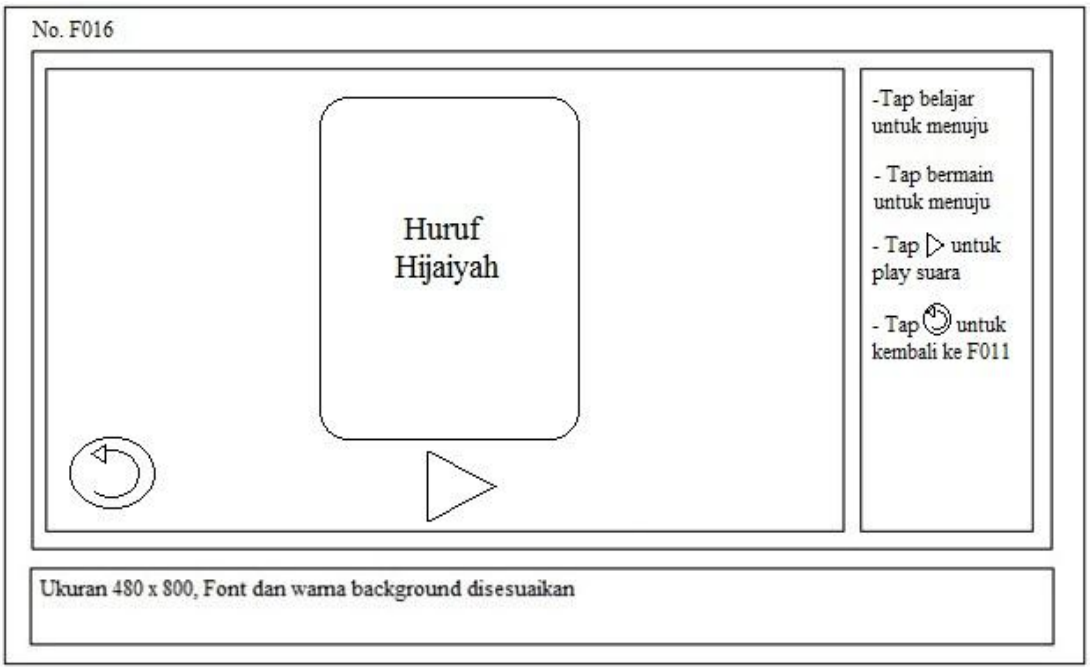
Gambar III. 46 Tampilan bermain stage 3



Gambar III. 47 Tampilan bermain stage 2 arah jalan kiri



Gambar III. 48 Tampilan belajar stage 1 pilihan player

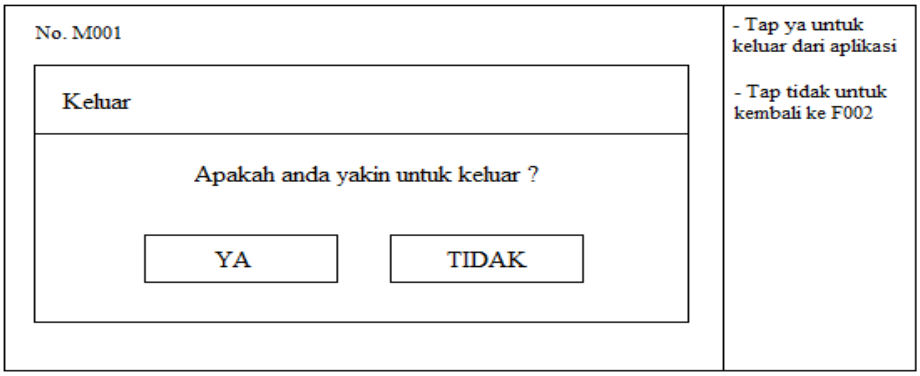


Gambar III. 49 Tampilan belajar stage 2 pilihan player

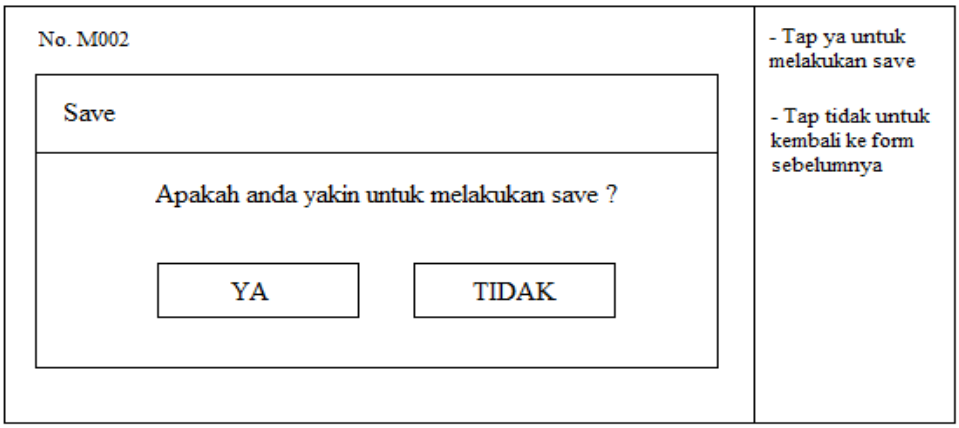
III.2.4. Perancangan Pesan

Perancangan pesan adalah rancangan antarmuka dari pesan yang akan ditampilkan apabila terjadi kesalahan dan pemberitahuan dalam proses yang

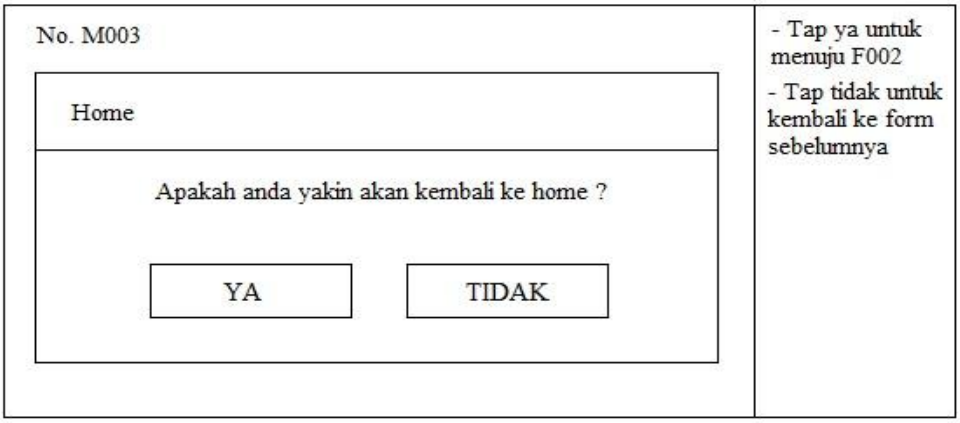
dilakukan. Rancangan pesan dalam dari *game* yang akan dibangun adalah sebagai berikut:



Gambar III. 50 Tampilan Pesan Keluar Aplikasi



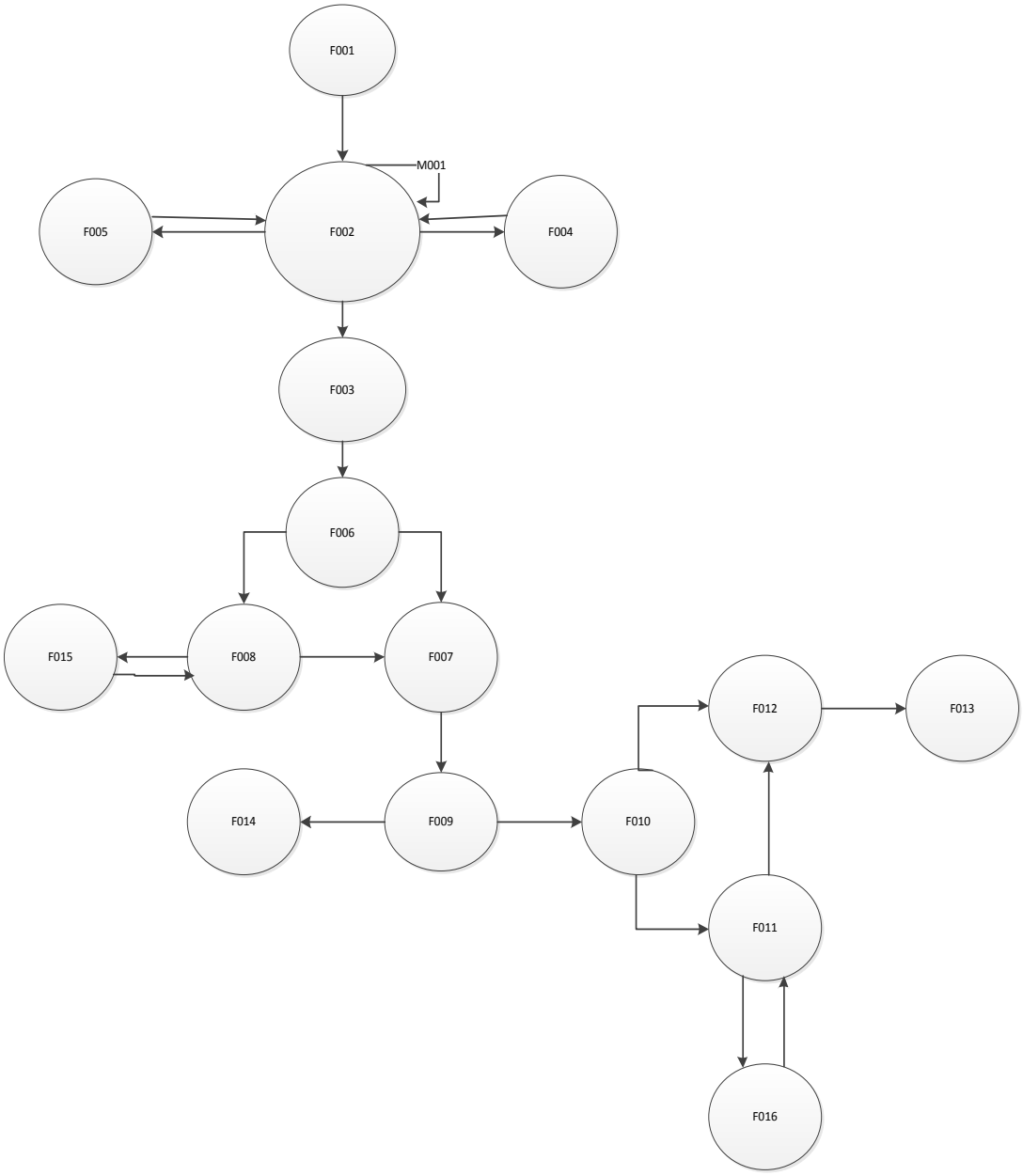
Gambar III. 51 Tampilan Pesan Save Data Aplikasi



Gambar III. 52 Tampilan Pesan Kembali ke home

III.2.5. Jaringan Semantik

Jaringan semnatik merupakan alat efektif untuk mempresentasikan pemetaan data, yang bertujuan mencegah terjadinya duplikasi data. Berikut adalah jaringan semantic pada *game* ini:



Gambar III. 53 Jaringan Semantik

III.2.6. Perancangan Method

Perancangan method merupakan pemaparan proses pada method-method yang digunakan pada setiap kelas. Perancanagn method digambarkan dengan menggunakan Flow Chart.

1. Method CStage1

```
/* Method diawali dengan melakukan inisialisasi untuk menghitung
jumlah baris, jumlah kolom, nomor kolom dan nomor baris */
int ROW_COUNT = -1
int COL_COUNT = -1
int [] [] cards
int turns
int x
int y

/*Method dilanjutkan dengan melakukan input dari player berupa
pilihan untuk menentukan jumlah kolom dan baris*/

Spinner itemSelected [x,y]
Case 1
x = 4; y = 4
Case 2
x = 5; y = 4

/* proses penampilan kartu dengan acak sesuai jumlah kolom dan
baris yang dipilih oleh player*/
New Game(int c, int r)
ROW_COUNT = r
COL_COUNT = c
(int y = 0; y < ROW_COUNT; y++) {
mainTable.addView(createRow(y));
}

Update CardHandler   //cek ketersediaan gambar pada matriks
CheckCard()

/*dilakukan pencocokkan, apakah gambar yang pertama sama dengan
gambar kedua yang dipilih player atau tidak*/
if (cards[seconedCard.x][seconedCard.y] ==
cards[firstCard.x][firstCard.y]) {

/*Jika kedua gambar cocok, maka tampilan gambar yang dipilih akan
menjadi transparan*/
firstCard.button.setVisibility(View.INVISIBLE);
seconedCard.button.setVisibility(View.INVISIBLE);

/* jika kedua gambar tidak tidak cocok, maka tampilan gambar akan
kembali ke awal*/
else {
seconedCard.button.setBackgroundDrawable(backImage);
firstCard.button.setBackgroundDrawable(backImage);
}
firstCard = null;
```

```
48  seconedCard = null;
```

Gambar III. 54 Method Stage1

2. Method CStage2

```
/* Lakukan inisialisasi berupa pengecekan apakah game sudah
terselesaikan atau belum dan inisialisasi jumlah kolom dan
baris */

integer [] mPuzzleSolved
int mBlack x
int mBlack y

/* Dibuat status untuk board apakah sudah selesai
atau belum, kemudian dihitung ukuran board dengan
mengkalikan jumlah kolom */

Get boardstatus()
Int size = mNumCols * mNumCols
mPuzzleSolve = new integer[size]


Shuffle()   // Susunan kartu diacak menggunakan prosedur
shuffle
int NumSwap = (int)
(math.random()*100) + 15


CheckifSolvable()  // pengecekan setiap perubahan posisi
kartu
{
    var n = 0;
    for (var i = 1; i <= 16; i++)
    {
        if (!(ContentPanel.Children[i] is StackPanel))
continue;
        var num1 = FindItemValueByPosition(i);
        var num2 = FindItemValueByPosition(i - 1);
        if (num1 > num2)
        {
            n++;
        }
    }
     var emptyPos = FindEmptyItemPosition();
    return n % 2 == (emptyPos + emptyPos / 4) % 2 ? true :
false;
}
/* Bila kartu selesai disusun tampilkan dialog yang
menyatakan player memenangkan permainan*/
didWin()
if(game1.didWin(0)
ShowDialog(DIALOG_YOUWIN_ID)
```

Gambar III. 55 Method Stage2