BAB 3
ANALISIS DAN PERANCANGAN SISTEM

3.1. Analisis Sistem

Analisis sistem merupakan penguraian dari suatu sistem yang utuh ke dalam bagian-bagian komponennya dengan maksud untuk mengidentifikasi dan mengevaluasi kebutuhan-kebutuhan permasalahan yang diharapkan sehingga dapat diusulkan perbaikan-perbaikanya. Analisis bertujuan untuk mendapatkan pemahaman secara keseluruhan tentang sistem yang akan dibuat.

3.1.1. Analisis Masalah

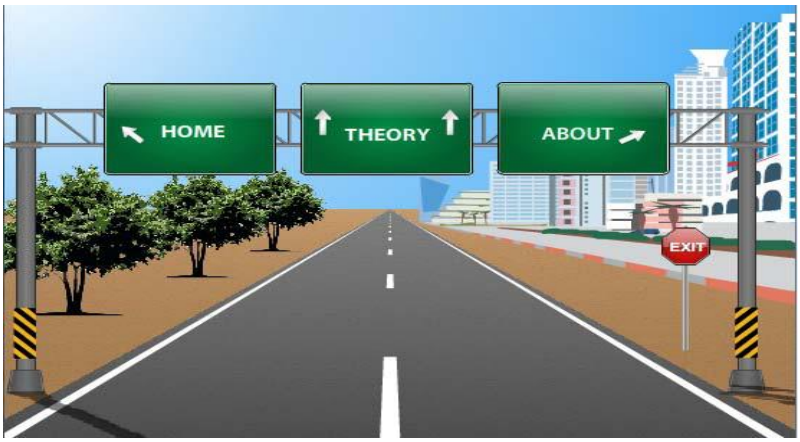
Analisis masalah yang dilakukan dengan cara mengamati dan melakukan evaluasi pada *game* yang akan dibangun oleh penulis. Menganalisis permasalahan yang ada di jalan sangat erat kaitannya untuk keselamatan dalam berlalu lintas karena akan berdampak kecelakaan yang menimbulkan kerugian ataupun dapat terjadi kematian, upaya preventif dalam menjaga keamanan dan keselamatan di jalan harus menjadi prioritas yang diutamakan. Peraturan-peraturan yang sudah diterapkan oleh kepolisian, tetapi sikap pengendara masih banyak yang tidak mentaati peraturan-peraturan yang berlaku di jalan raya, sehingga menimbulkan terjadi kemacetan-kemacetan lalu lintas ataupun tejadinya kecelakaan lalu lintas. Hal ini disebabkan karena lemahnya tingkat kedisiplinan, juga lemah dalam kepatuhan, lemah dalam sikap dan perilaku berlalu lintas. Upaya yang dapat dilakukan dalam meningkatkan agar lebih disiplin untuk mentaati tata tertib lalu lintas dan mengutamakan keselamatan berkendara, sebagaimana untuk pembekalan diri dalam mengantisipasi mencegah terjadinya kecelakaan dengan sebuah aplikasi *game*.

3.1.2. Analisis *Game* Sejenis

Tahap ini analisis *game* sejenis bertujuannya untuk mendapatkan konsep permainan dan perbandingan dari jenis *game* yang sudah ada. *Game* sejenis yang akan dibahas adalah..

3.1.2.1. Pengenalan Aplikasi Multimedia Cara Berkendara Yang Baik

Aplikasi multimedia cara berkendara yang baik ini merupakan media pembelajaran dan evaluasi. Aplikasi ini memvisualisasikan cara berkendara yang baik materi yang disajikan tentang rambu-rambu lalu lintas yang ada di Indonesia dan cara berkendara yang baik. Beberapa menu utama dari *game* sejenis dapat dilihat pada gambar 3.1.



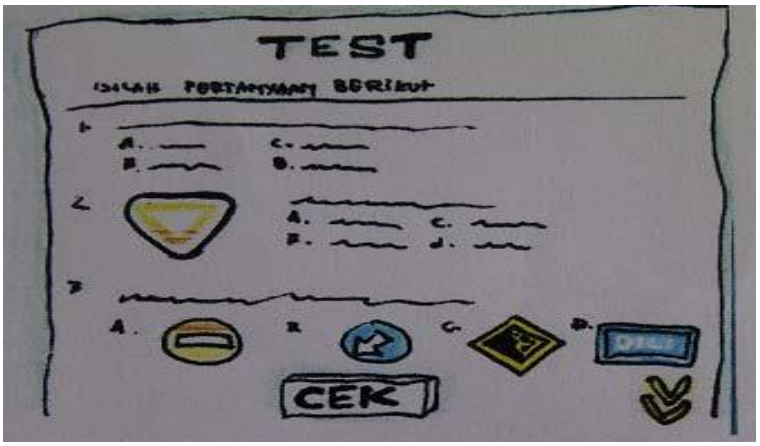
Gambar 3. 1 Tampilan Menu Utama *Game* Sejenis [2].

Menu utama yang terdapat 3 buah tombol yaitu *home*, teori, dan *about*. Apabila menekan tombol teori ditekan akan secara otomatis pemain akan masuk ke menu teori. Pada menu ini terdapat 3 tombol yang dapat dipilih pemakai yaitu: tombol rambu, cara berkendara, dan simulasi. Beberapa dari menu teori dapat dilihat pada gambar 3.2.



Gambar 3. 2 Tampilan Menu Teori *Game* Sejenis [2].

Menu rambu, apabila pemain memilih rambu terdapat 3 pilihan yaitu rambu peringatan, rambu perintah dan rambu larangan. Pemakai juga mengetahui arti maksud rambu tersebut dan juga fasilitas evaluasi. Adapun evaluasinya dapat dilihat pada gambar 3.3.



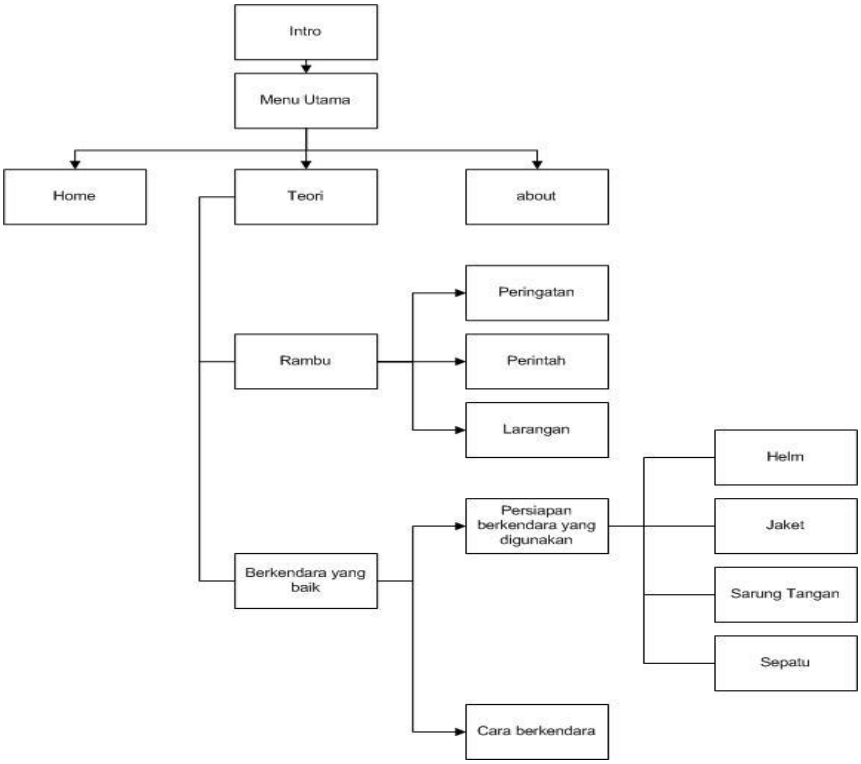
Gambar 3. 3 Tampilan Evaluasi *Game* Sejenis [2].

Pada menu cara berkendara, pemain akan diajarkan cara untuk mengenal atribut perlengkapan berkendara yaitu helm, sarung tangan, dan sepatu. Lihat pada gambar 3.4.



Gambar 3. 4 Tampilan Cara Berkendara Yang Baik *Game* Sejenis [2].

Pada menu simulasi, menampilkan cara berkendara yang baik untuk berkendara. Berikut ini struktur menu *game* sejenis dapat dilihat pada gambar 3.5.



Gambar 3. 5 Struktur *Game* Sejenis Aplikasi Multimedia Berkendara yang Baik

Berikut kesimpulan analisis komponen-komponen yang ada dalam *game* yang menjadi acuan perbandingan dalam pembangunan *game* akan dibangun. Lihat pada tabel 3.1.

Tabel 3. 1 Perbandingan *Game* Sejenis

	Aplikasi multimedia cara berkendara yang baik	*Game* edukasi *safety riding* untuk menerapkan keselamatan berlalu lintas berbasis desktop
Karakter	Ada	Ada
Level	Tidak ada	Ada
Suara	Ada	Ada
Musik	Tidak ada	Ada
Waktu	Tidak ada	Ada
Penerapan AI	Tidak ada	Ada

	Aplikasi multimedia cara berkendara yang baik	*Game* edukasi *safety riding* untuk menerapkan keselamatan berlalu lintas berbasis dekstop
Materi yang dibahas	Rambu lalu lintas terdiri rambu peringatan, larangan, dan perintah	a. Rambu lalu lintas terdiri rambu peringatan, larangan, perintah, dan petunjuk b. Persiapan untuk pengendara, c. Kenali kondisi keamanan sepeda motor sebelum digunakan, dan d. Teknik dasar berkendara yang baik dan benar.
Multiplayer	Tidak ada	Tidak ada
Animasi	Ada	Ada
Soal quiz	Ada	Ada
Simulasi	Ada	Ada

3.1.2.2. Pengenalan Road Pop

Game Road Pop adalah permainaan menguji pengetahuan anak-anak mengenai rambu berlalu lintas dengan cara yang lebih menyenangkan tanpa meninggalkan unsur edukasi perihal berlalu lintas yang baik dan benar. *Game Road Safety* terdiri dari 4 *level* dengan 3 sub *level* disetiap tingkatnya [17]. Lihat *game* sejenis pada gambar 3.6.



Gambar 3. 6 Tampilan Utama *Game* Sejenis *Game Road Pop* [17]



Gambar 3. 7 Tampilan Area Permainan *Game* Sejenis *Road Pop* [17]

Berikut ini penjelasan *gameplay road pop* rambu lalu lintas, yaitu :

a. *Game* ini, ada 2 tokoh bersaudara, yakni Tisa dan Aksa. Percakapan keduanya berkaitan dengan lalu lintas.
b. Permainan ini terdiri dari tahapan tertib berlalu lintas mulai dari pejalan kaki, pesepeda, pengendara motor dan mobil.
c. Permainan ini pemain harus menyamakan minimal 3 ikon rambu-rambu lalu lintas atau perangkat keselamatan berkendara dalam satu garis.
d. Pasangkan 3 ikon tadi baik vertikal maupun horisontal agar tetap mendapat poin sampai batas waktu yang sudah ditentukan.
e. Pemain mengumpulkan ikon-ikon yang diminta pada *level* yang dimainkan untuk mendapatkan poin setinggi mungkin.

3.1.3. Analisis *Game* Edukasi *Safety Riding* untuk Menerapkan Keselamatan Berlalu Lintas

Game yang akan dibuat adalah pembangunan *game* edukasi *safety riding* untuk menerapkan keselamatan berlalu lintas, *game* ini dengan menggunakan berbasis 2D. *Game* ini juga merupakan sistem *single player* dengan menggunakan kontrol *mouse* dan tombol *keyboard*.

Aplikasi *game* edukasi *safety riding* yang dibangun tersedia juga untuk menyajikan materi pembelajaran, yaitu :

1. Berikut ini materi pengenalan rambu-rambu lalu lintas pada *game* ini adalah.
   a. Rambu Peringatan
   b. Rambu Perintah
   c. Rambu Larangan
   d. Rambu Petunjuk
2. Berikut ini materi pengenalan keselamatan berkendara s*afety riding* pada *game* ini adalah.
   a. Persiapan untuk pengendara
   b. Kenali kondisi keamanan sepeda motor sebelum berjalan
   c. Teknik dasar berkendara yang baik.
3. Batasan yang terdapat pada *game* edukasi *safety riding* ini adalah sebagai berikut :
   a. Sistem *game* ini *single player*.
   b. Grafik 2 Dimensi.
   c. *Game* edukasi *safety riding* untuk menerapkan keselamatan berlalu lintas ini dijalankan secara *offline*.
   d. Penyajian materi pembelajaran yang dibahas pada *game* ini hanya mengenai tentang pengenalan rambu lalu lintas, persiapan untuk pengendara, kenali kondisi keamanan sepeda motor sebelum digunakan, dan teknik dasar berkendara yang baik.

3.1.3.1. Gameplay

Game edukasi *safety riding* ini masing-masing memiliki *gameplay* yang berbeda. Berikut penjelasan tentang *gameplay* dari masing-masing *game* edukasi *safety riding* untuk menerapkan keselamatan berlalu lintas adalah sebagai berikut:

1. *Gameplay* Periksa Kendaraan

G*ameplay* periksa kendaraan yang dibangun pada *game* ini memberikan pembelajaran bagi pengendara motor, supaya lebih memperhatikan untuk

memeriksa kondisi kendaraannya dan mengutamakan keamanan kondisi fisik sepeda motor sebelum digunakan.

Aturan *gameplay* pada *game* periksa kendaraan adalah sebagai berikut.

a. Pemain harus memeriksa dan memperbaiki 7 kondisi bagian kendaraannya yang ditampilkan di dalam permainan
b. Kontrol *game* ini dengan menggunakan mouse.
c. Permainan ini setiap bagian kondisi kendaraan yang berhasil diperbaiki, maka pemain akan mendapatkan skor 100.
d. Permainan ini hanya memiliki batas waktu 30 detik untuk memeriksa kondisi kendaraan setiap melakukan perbaikan
e. Apabila pemain batas waktu telah habis maka pemain dinyatakan gagal.
f. Permainan ini memiliki kesempatan kesalahan hanya 3 kali saja setiap pemeriksaan.
g. Apabila pemain menemukan kondisi bagian kendaraan yang harus diperbaiki, Jika ya, maka perbaiki kendaraan sehingga nilai skor akan bertambah 100 dan jika tidak, maka pemain mengklik kondisi bagian kendaraan yang salah, maka pemain akan mendapat pengurangan skor 25 setiap pemeriksaan yang salah. Sedangkan jika pemain yang masih mempunyai skor 0 ketika mengklik bagian kondisi kendaraan yang salah tidak dikurangi dan tetap 0.
h. Diakhir permainan, apabila pemain mendapatkan nilai skor >=500 dan skor <=600 maka pemain mendapatkan prestasi yang “sangat baik”, jika pemain mendapatkan nilai skor >=400 dan skor <=475 maka pemain mendapatkan prestasi yang “baik”, dan jika pemain mendapatkan nilai skor <=375 maka pemain mendapatkan prestasi yang “cukup”.

2. *Gameplay* Tebak Arti Rambu

Gameplay game tebak arti rambu-rambu ini pada permainan ini memberikan pembelajaran untuk anak-anak menjadi lebih tahu dan mengerti tentang jenis-jenis rambu lalu lintas dan fungsi arti simbol rambu lalu lintas.

Aturan *gameplay* pada *game* tebak arti rambu adalah sebagai berikut.

a. Pemain akan menjawab pertanyaan yang ditampilkan di dalam permainan.
b. Untuk menjawab pertanyaan *game* tebak arti rambu, pemain harus menebak huruf apa saja yang menjadi bagian dari jawaban pada pertanyaan.
c. Pemain harus menjawab pertanyaan sebanyak 10 soal.
d. Permainan ini kesempatan salah dalam menjawab hanya 3 kali
e. Setiap pertanyaan yang benar akan mendapat skor 100.
f. Kesalahan pemain jika salah menjawab menebak huruf maka nilai skor yang diperoleh pemain akan mendapat pengurangan skor 20 setiap salah menebak huruf.
g. Permainan ini hanya memiliki batas waktu 30 detik setiap soal
h. Apabila pemain batas waktu telah habis maka pemain dinyatakan gagal
i. Soal pada *game* tebak arti rambu akan diacak, agar soal mendapat urutan soal yang tidak sama.
j. Diakhir permainan setiap babak, apabila pemain mendapatkan nilai skor >=750 dan skor <=1000 maka pemain mendapatkan prestasi yang "sangat memuaskan", jika pemain mendapatkan nilai skor >=550 dan skor <750 maka pemain mendapatkan prestasi yang "memuaskan", dan jika pemain mendapatkan nilai skor<550 maka pemain mendapatkan prestasi yang "kurang memuaskan".

3. *Gameplay* Simulasi Lalu Lintas

Memvisualisasikan prilaku berkendara yang baik, dengan mensimulasikan lalu lintas pada *game* ini, sehingga pemain atau pengendara merasakan keadaan nyatanya menjadi pengendara yang lebih berdisplin dalam mematuhi peraturan lalu lintas yang ada pada saat berlalu lintas.

Aturan *gameplay* pada *game* simulasi lalu lintas adalah sebagai berikut.

a. Pemain harus menyelesaikan misi yang diminta ibu untuk membeli buah-buahan ke pasar, dan berhati-hatilah saat berkendara patuhilah peraturan rambu lalu lintas yang ada.
b. Jika melanggar maka akan tampil informasi larangan atau tilang.

c. Jika pemain menemukan rambu lalu lintas dilarang parkir maka akan mendapat informasi perintah yang harus dilakukan. dan jika menemukan rambu lalu lintas tempat parkir maka misi berhasil diselesaikan.

3.1.4. Analisis Algoritma *Greedy*

Game tebak arti rambu algoritma yang digunakan untuk menyelesaikan *game* tebak arti rambu adalah algoritma *greedy*. Implementasi Algoritma *Greedy* digunakan untuk mengacak pertanyaan dan jawaban, agar pertanyaannya tidak sama. Adapun penjelasan dari pengimplementasian algoritma *greedy* untuk menyelesaikan *game* tebak arti rambu adalah sebagai berikut.

Pseudocode Algoritma Greedy lihat gambar 3.8.

Pseudo-code algoritma *greedy* adalah sebagai berikut:

```
procedure greedy(input C: himpunan_kandidat;
                 output S : himpunan_solusi)
{ menentukan solusi optimum dari persoalan optimasi dengan algoritma greedy
  Masukan: himpunan kandidat C
  Keluaran: himpunan solusi S
}
Deklarasi
   x : kandidat;
Algoritma
   S¬{}              { inisialisasi S dengan kosong }
   while (belum SOLUSI(S)) and (C ≠ {} ) do
      x¬SELEKSI(C);   { pilih sebuah kandidat dari C}
      C¬ C - {x}      { elemen himpunan kandidat berkurang satu }
     if LAYAK(S È {x}) then
        S¬S È {x}
      endif
    endwhile
{SOLUSI(S) sudah diperoleh or C = {} }
```

Gambar 3. 8 *Psoudocode* Algoritma Greedy

Elemen–elemen yang digunakan dalam penerapan algoritma *greedy* antara lain :

1. Himpunan Kandidat = C
   C ={A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P,Q,R,S,T,U,V,W,X,Y,Z}
2. Himpunan Solusi = S
   Himpunan Solusi (S) adalah jawaban dari pertanyaan yang ditampilkan.
   S ={A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P,Q,R,S,T,U,V,W,X,Y,Z}
3. Fungsi Seleksi
   Mengambil sebuah kandidat dari Himpunan Kandidat (C).
4. Fungsi Kelayakan
   Memeriksa apakah ada beberapa huruf berdasarkan kandidat yang dipilih ada dalam barisan huruf dari jawaban. Dilakukan dengan cara mencocokkan pertanyaan yang ditampilkan dengan pertanyaaan dari daftar soal, jika sudah ketemu bandingkan kandidat dengan jawaban tersebut. Jika ada tampilkan semua kandidat yang ada dalam jawaban, jika tidak ada maka buang kandidat tersebut dan jangan dipilih lagi.
5. Fungsi Objektif
   Batas maksimal kesalahan dalam memilih kandidat adalah tiga (3) kali.

Algoritma *greedy* dapat dituliskan sebagai berikut :

a. Inisialisasi S dengan kosong.
b. Pilih sebuah kandidat dari C dengan fungsi seleksi.
c. Kurangi C dengan kandidat yang telah terpilih.
d. Periksa apakah kandidat yang dipilih tersebut, membentuk solusi atau tidak. Jika ya masukkan kandidat ke S, jika tidak buang kandidat tersebut dan tidak usah dihiraukan lagi serta tambahkan satu untuk variabel kesalahan.
e. Periksa apakah S yang sudah terbentuk telah memberikan solusi yang lengkap. Jika ya, berhenti. Jika tidak, cek kesalahan memilih kandidat apakah sudah tiga (3) kali sebelum mengulangi dari langkah dua (2).

Misalkan : Ada sebuah soal tebak arti rambu. Lihat pada gambar 3.9.



Gambar 3. 9 Soal Gambar Tebak Arti Rambu

Berikut langkah untuk menyelesaikan soal tersebut dengan menggunakan algoritma *greedy* adalah sebagai berikut.

1. Menentukan himpunan kandidat (C).



2. Menentukan himpunan solusi(S), untuk mendapatkan solusi diharuskan membandingkan pertanyaan dengan pertanyaan yang ada didaftar soal, setelah ditemukan hitung panjang huruf dari jawaban untuk pertanyaan tersebut. Sediakan kotak sebanyak panjang huruf dari jawaban. Hasil akhir dari solusi jawaban yang benar adalah **LAMPU LALULINTAS**.

S= _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _

3. Berilah nilai nol (0) untuk variable kesalahan(fungsi objektif), Jika salah pilih huruf maka tambahkan variable kesalahan=1.

   Kesalahan = 0

4. Lakukan langkah no 2, ambil sebuah kandidat dari C.

   Kandidat yang diambil = L

5. Lakukan langkah no 3, kurangi C dengan kandidat yang diambil.



6. Lakukan langkah ke 4 dan 5, periksa apakah kandidat membentuk solusi jika ya tampilkan atau masukkan dalam S (himpunan solusi), jika tidak tambahkan 1 untuk variabel kesalahan.

S= | L _ _ _ _ L _ L _ L _ _ _ _ _ |

   Kesalahan = 0

7. Ulangi langkah no 2, ambil sebuah kandidat dari C

   Kandidat yang diambil = A

8. Lakukan langkah 3, kurangi C dengan kandidat yang diambil.



9. Ulangi langkah ke 4 dan 5, periksa apakah kandidat membentuk solusi jika ya tampilkan atau masukkan dalam S (himpunan solusi), jika tidak tambahkan 1 untuk variabel kesalahan.

S= | L A _ _ _ L A L _ L _ _ _ A _ |

   Kesalahan = 0

10. Ulangi Langkah no 2, ambil sebuah kandidat dari C

    Kandidat yang ambil = U

11. Lakukan langkah 3, kurangi C dengan kandidat yang diambil.



12. Ulangi langkah ke 4 dan 5, periksa apakah kandidat membentuk solusi jika ya tampilkan atau masukkan dalam S (himpunan solusi), jika tidak tambahkan 1 untuk variabel kesalahan.

S= | L A _ _ U L A L U L _ _ A _ |

Kesalahan = 0

13. Ulangi Langkah no 2, ambil sebuah kandidat dari C

Kandidat yang ambil = I

14. Lakukan langkah 3, kurangi C dengan kandidat yang diambil.



15. Ulangi langkah ke 4 dan 5, periksa apakah kandidat membentuk solusi jika ya tampilkan atau masukkan dalam S (himpunan solusi), jika tidak tambahkan 1 untuk variabel kesalahan.

S= | L A _ _ U L A L U L I _ A _ |

Kesalahan = 0

16. Ulangi Langkah no 2, ambil sebuah kandidat dari C

Kandidat yang ambil = N

17. Lakukan langkah 3, kurangi C dengan kandidat yang diambil.



18. Ulangi langkah ke 4 dan 5, periksa apakah kandidat membentuk solusi jika ya tampilkan atau masukkan dalam S (himpunan solusi) jika tidak tambahkan 1 untuk variabel kesalahan.

S= | L A _ _ U L A L U L I N _ A _ |

Kesalahan = 0

19. Ulangi Langkah no 2, ambil sebuah kandidat dari C

Kandidat yang ambil = S

20. Lakukan langkah 3, kurangi C dengan kandidat yang diambil.



21. Ulangi langkah ke 4 dan 5, periksa apakah kandidat membentuk solusi jika ya tampilkan atau masukkan dalam S (himpunan solusi), jika tidak tambahkan 1 untuk variabel kesalahan.

S= | L A _ _ U L A L U L I N _ A S |

Kesalahan=0

22. Ulangi Langkah no 2, ambil sebuah kandidat dari C

Kandidat yang ambil = M

23. Lakukan langkah 3, kurangi C dengan kandidat yang diambil.



Kesalahan=0

24. Ulangi langkah ke 4 dan 5, periksa apakah kandidat membentuk solusi jika ya tampilkan atau masukkan dalam S (himpunan solusi), jika tidak tambahkan 1 untuk variabel kesalahan.

S= | L A M _ U L A L U L I N _ A S |

Kesalahan=0

25. Ulangi Langkah no 2, ambil sebuah kandidat dari C

Kandidat yang ambil = P

26. Lakukan langkah 3, kurangi C dengan kandidat yang diambil.



27. Ulangi langkah ke 4 dan 5, periksa apakah kandidat membentuk solusi jika ya tampilkan atau masukkan dalam S (himpunan solusi), jika tidak tambahkan 1 untuk variabel kesalahan.

S= L A M P U L A L U L I N _ A S

Kesalahan=0

28. Ulangi Langkah no 2, ambil sebuah kandidat dari C

Kandidat yang ambil = T

29. Lakukan langkah 3, kurangi C dengan kandidat yang diambil.



Kesalahan=0

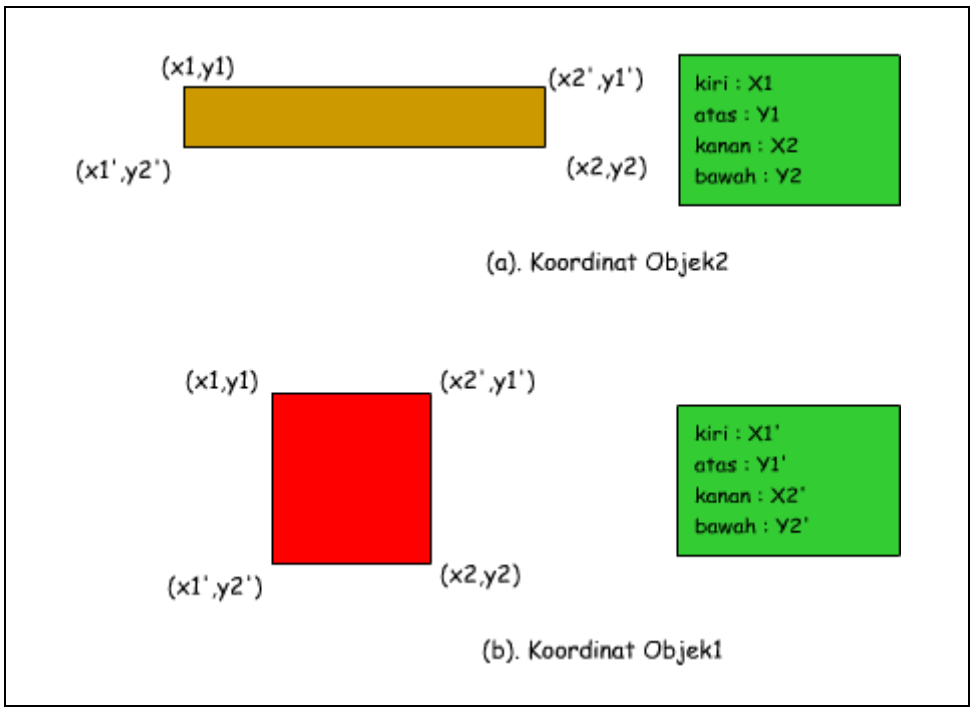
30. Ulangi langkah ke 4 dan 5, periksa apakah kandidat membentuk solusi jika ya tampilkan atau masukkan dalam S (himpunan solusi), jika tidak tambahkan 1 untuk variabel kesalahan.

S= L A M P U L A L U L I N T A S

3.1.5. Analisis Algoritma *Collision Detection*

Collision detection yang digunakan pada *game* ini adalah *rectangles Collision Detection*. Teknik ini merupakan pendeteksian tabrakan menggunakan *bound* berbentuk persegi panjang, dimana setiap objek yang dibuat dalam *game* memiliki *bound* masing-masing. Cara ini cukup mudah karena hanya perlu membandingkan apakah pada kedua objek pada koordinat tertentu.

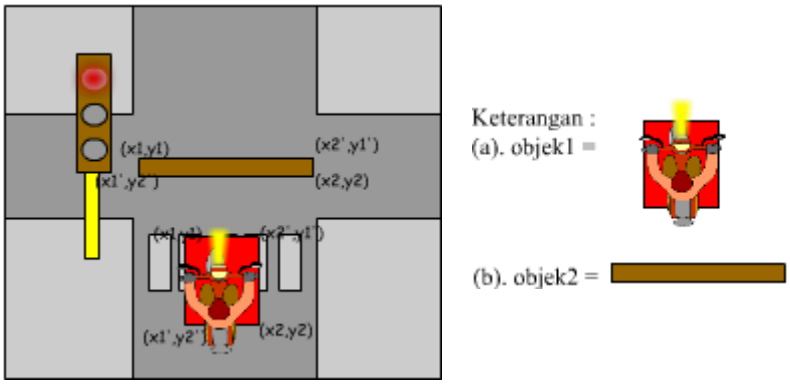
Cara mengetahui bagaimana tabrakan terjadi, pertama adalah menentukan titik koordinat dari masing-masing objek. Contoh, ada dua buah persegi panjang dengan nama objek1 dan objek. Objek2 mempunyai koordinat: (X_1,Y_1), (X_1,Y_2), (Y_2,Y_1), (X_2,Y_2). Sedangkan koordinat objek1: (X_1',Y_1'), (X_1',Y_2'), (X_2',Y_1'), (X_2',Y_2'). Lihat pada gambar 3.10.



Gambar 3. 10 Koordinat *Rectangles Collision Detection*

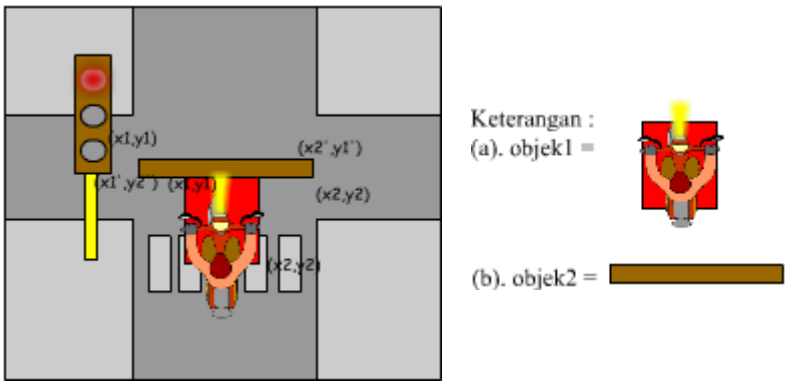
Berikut ini skenario terjadinya *collison* dan tidak terjadinya *collison* diantara kedua objek tersebut adalah sebagai berikut.

1. Memeriksa tepi bawah objek1, tabrakan jika $Y_1 > Y_1' > Y_2$. Tidak ada tabrakan dalam hal ini karena $Y_1' > Y_2$.
2. Memeriksa tepi atas objek1, tabrakan jika $Y_1 > Y_2' > Y_2$. Tidak ada tabrakan dalam hal ini karena $Y_1' < Y_2$. Lihat pada gambar 3.11.



Gambar 3. 11 Kondisi Dimana Kedua Objek Tidak Terjadi *Collision*

3. Mirip dengan no. (1), tepi kali ini $Y_1 > Y_1' > Y_2$ sehingga ada tabrakan. Seperti no.(2), tepi $Y_1 > Y_2' > Y_2$ sehingga ada tabrakan. Lihat pada gambar 3.12.



Gambar 3. 12 Kondisi Dimana Kedua Objek Terjadi *Collision*

3.1.6. Definisi Kebutuhan Perangkat

Definisi kebutuhan perangkat lunak berisikan deskripsi dari kebutuhan perangkat lunak yang akan dibangun, baik kebutuhan fungsional maupun kebutuhan non-fungsional.

3.1.6.1. Definisi Kebutuhan Perangkat Lunak Fungsional

Definisi kebutuhan perangkat lunak fungsional berisi pernyataan dari layanan sistem yang harus disediakan, bagaimana sistem harus bereaksi terhadap *input* tertentu dan bagaimana sistem harus berperilaku dalam situasi tertentu dapat dilihat pada tabel 3.2.

Tabel 3. 2 Kebutuhan Perangkat Lunak Fungsional

Nomor	Kebutuhan fungsional
SKPL-F-01	Sistem dapat menampilkan menu materi pembelajaran
SKPL-F-02	Sistem dapat menampilkan pilihan menu permainan
SKPL-F-03	Sistem dapat menampilkan menu petunjuk permainan
SKPL-F-04	Sistem dapat menampilkan menu pembuat
SKPL-F-05	Sistem dapat menampilkan informasi tentang tata tertib berlalu lintas.
SKPL-F-06	Sistem dapat menampilkan soal secara acak

3.1.6.2. Definisi Kebutuhan Perangkat Lunak Non-Fungsional

Definisi kebutuhan perangkat lunak non-fungsional adalah deskripsi mengenai batasan-batasan dari layanan-layanan dan fungsi-fungsi dari sebuah sistem yang akan dibangun, seperti batasan waktu, batasan dari pengembangan proses, dan batasan pengguna dapat dilihat pada tabel 3.3.

Tabel 3. 3 Kebutuhan Perangkat Lunak Non-Fungsional

Nomor	Kebutuhan Non-Fungsional
SKPL-NF-01	Sistem yang dibangun dalam bentuk tampilan 2 dimensi
SKPL-NF-02	Sistem tidak menangani bentuk permainan *online*
SKPL-NF-03	Sistem tidak menangani permainan melebihi satu orang pemain saja.
SKPL-NF-04	Sistem yang dibangun tidak disertai dengan adanya sistem penyimpanan .
SKPL-NF-05	Sistem yang dibangun membutuhkan sistem operasi windows
SKPL-NF-06	Sistem yang dibangun tidak memiliki pilihan pengaturan suara

3.1.7. Analisis Kebutuhan Non-Fungsional

Analisis kebutuhan non-fungsional menggambarkan kebutuhan luar sistem yang diperlukan untuk menjalankan aplikasi yang dibangun. Kebutuhan non-fungsional pada *game* edukasi s*afety riding* untuk menerapkan keselamatan berlalu lintas adalah meliputi kebutuhan perangkat keras, kebutuhan perangkat lunak dan pengguna sistem yang akan memakai aplikasi, yaitu:

3.1.7.1. Analisis Kebutuhan Penguna

Analisis kebutuhan pengguna merupakan analisis terhadap *user* yang akan menggunakan sistem yang dibangun untuk kebutuhan pembelajaran terutama untuk anak-anak usia 7 tahun ke atas, hal ini dikarenakan pada *game* yang dibangun selain untuk bermain juga terdapat unsur pembelajaran yang dapat menambah pengetahuan bagi anak-anak dan remaja ataupun orang dewasa. Analisis pengguna ini juga mencakup analisis beberapa parameter terhadap calon pengguna dari aplikasi, diantaranya :

3.1.7.1.1. Pengetahuan dan Pengalaman Pengguna (*User Knowledge and Experience)*

Pengetahuan dan pengalaman merupakan salah satu faktor penting yang dapat dijadikan acuan pengguna dalam penggunaan aplikasi yang akan dibangun. Klasifikasi *knowledge and experience* dari pengguna aplikasi. Lihat pada tabel 3.4.

Tabel 3. 4 Analisis Klasifikasi *Knowledge And Experience*

	Keterangan
Educational level	*Game* yang akan dibangun ini ditujukan terutama untuk anak-anak berusia 7 tahun ke atas.
Reading and Write level	*Game* yang akan dibangun ditujukan untuk pengguna yang sudah dapat menulis dan membaca dengan baik.
Typing Skill	*Game* yang akan dibangun ini untuk pengguna yang memiliki kemampuan dalam mengetik yang baik pada sistem operasi komputer
Computer Literacy	*Game* ini dibangun ditujukan untuk pengguna yang mampu atau pernah menggunakan sistem operasi komputer.
Native laungage	*Game* yang akan dibangun menggunakan bahasa indonesia.
System Experience	*Game* yang akan dibangun ditujukan untuk pengguna yang sudah mempunyai pengalaman dan mengerti menjalankan sistem operasi komputer.

3.1.7.1.2. Karakteristis Fisik Pengguna (*User Physical Characteristic*)

Keadaan fisik seseorang mungkin akan berpengaruh pada penggunaan aplikasi *game* ini. Ada hal-hal yang harus diperhatikan juga terhadap *user* dari karakteristik fisiknya untuk dapat menggunakan aplikasi ini yaitu, *usia*, *gender Handednes*, dan *Color Blind* seperti dapat terlihat pada tabel 3.5.

Tabel 3. 5 Analisis Karakteristis Fisik Pengguna

Age	7 tahun ke atas
Gender	Laki-laki dan perempuan
Handednes	Kanan dan kiri
Color blind	*User* yang tidak bisa membedakan warna yang satu dengan yang lainnya (buta warna) masih mampu menggunakan aplikasi ini, karena tidak ada indikator warna-warna khusus yang membedakan antara

	fungsional yang satu dengan fungsional yang lainnya. Akan tetapi penggunaannya tidak akan optimal karena dalam *game* ini terdapat banyak sekali perbedaan warna yang menunjang interaksi dan ketertarikan dalam permainan

3.1.7.2. Analisis Kebutuhan Perangkat Lunak

Perangkat lunak digunakan dalam sebuah sistem merupakan perintah perintah yang diberikan kepada perangkat keras. Adapun perangkat lunak yang digunakan dalam membangun aplikasi *game* edukasi ini. Lihat pada tabel 3.6.

Tabel 3. 6 Analisis Kebutuhan *Software (Develover)*

Nama Perangkat	Jenis
Sistem Operasi	Windows 8
Adobe Flash CS 6	DDR2 3072 Mb
Desain	CorelDraw X4
Adobe Flash Player	9,10,11
Sound	WavePad Sound Editor

Spesifikasi minimum perangkat lunak yang dapat dipergunakan untuk menjalankan aplikasi *game* edukasi ini. Lihat pada tabel 3.7.

Tabel 3. 7 Analisis Kebutuhan *Software (user)*

Nama Perangkat	Jenis
Sistem Operasi	Windows XP
Adobe Flash CS 3	DDR2 512 Mb

3.1.7.3. Analisis Kebutuhan Perangkat Keras

Analisis perangkat keras merupakan salah satu hal yang penting karena tanpa *hardware* yang memenuhi syarat program yang akan dibuat tidak akan dapat berjalan. Spesifikasi perangkat keras yang dibutuhkan dan digunakan dalam mengembangkan aplikasi *game* edukasi ini. Lihat pada tabel 3.8.

Tabel 3. 8 Analisis Kebutuhan *Hardware (Develover)*

Nama Perangkat	Spesifikasi
Processor	Core 2 Duo T5750 CPU 2.00 Ghz
Memori	DDR2 3072 Mb
Harddisk	160 GB

Nama Perangkat	Spesifikasi
VGA Card	358 Mb Resolusi 1366 x 768
Monitor	17 inch
Speaker	1 unit
Mouse dan *Keyboard*	1 unit

Spesifikasi minimum perangkat keras yang dapat dipergunakan untuk menjalankan aplikasi *game* edukasi *safety ridng* ini. Lihat pada tabel 3.9.

Tabel 3. 9 Analisis Kebutuhan *Hardware (User)*

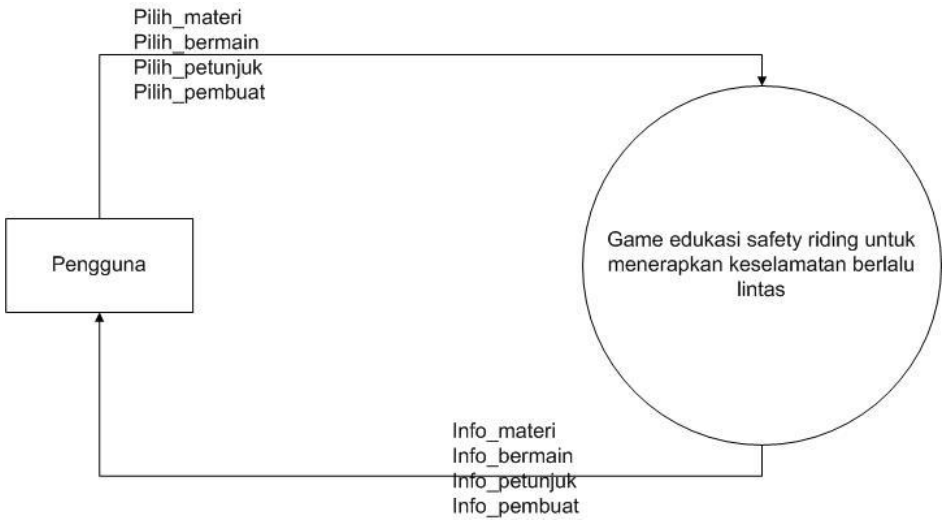
Nama Perangkat	Spesifikasi
Processor	CPU 1.8 Ghz
Memori	DDR2 2048 Mb
Harddisk	80 GB
VGA Card	256 Mb Resolusi 800 x 600
Monitor	15 inch
Speaker	1 unit
Mouse dan *Keyboard*	1 unit

3.1.8. Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional menggambarkan kebutuhan sistem yang akan dibangun pada *game* edukasi *safety riding* untuk menerapkan keselamatan berlalu lintas. Adapun kebutuhan fungsional pada *game* yang akan dibangun ini dengan pemodelan prosedural.

3.1.8.1. Diagram Konteks

Diagram konteks adalah diagram yang menggambarkan *input*, *process* dan *output* pada sistem perangkat lunak yang akan dibangun. Diagram konteks pada *game* edukasi *safety riding* dapat dilihat pada gambar 3.12.



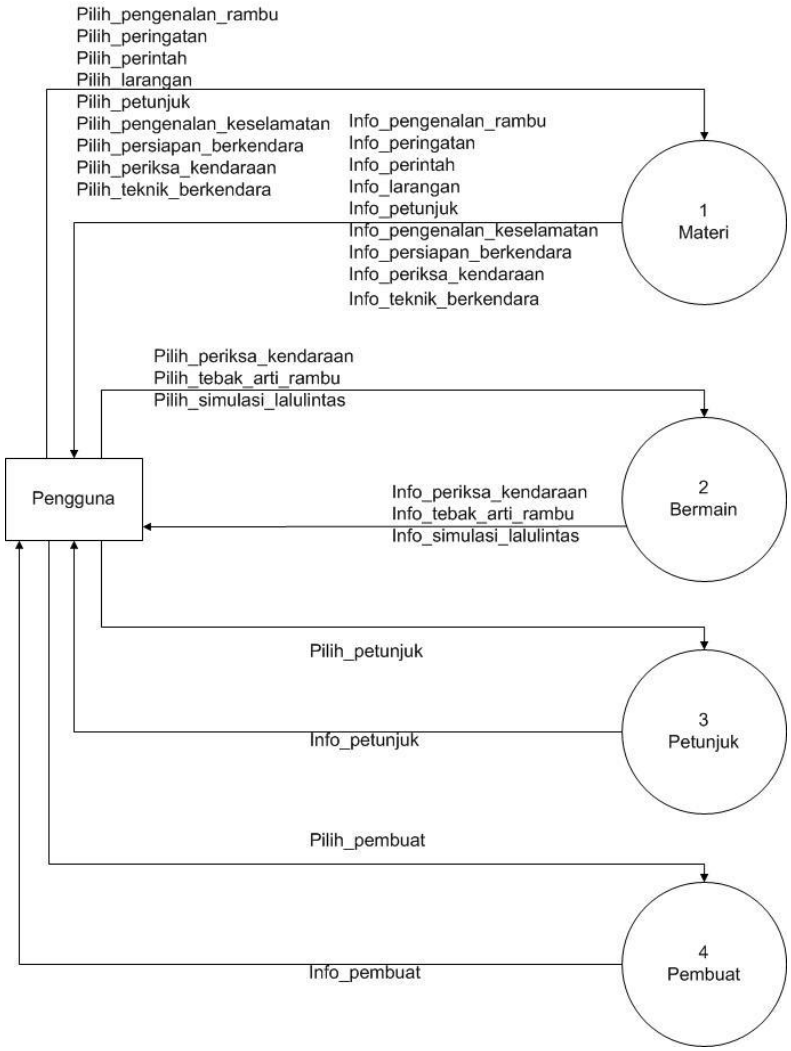
Gambar 3. 13 Diagram Konteks *Game* Edukasi *Safety Riding*

3.1.8.2. DFD *(Data Flow Diagram)*

Data *Flow Diagram* merupakan suatu media yang digunakan untuk menggambarkan aliran data yang mengalir pada suatu sistem informasi. Berikut DFD (*Data Flow Diagram)* dari *game* edukasi *safety riding* untuk menerapkan keselamatan berlalu lintas adalah sebagai berikut.

1. DFD *Level* 1

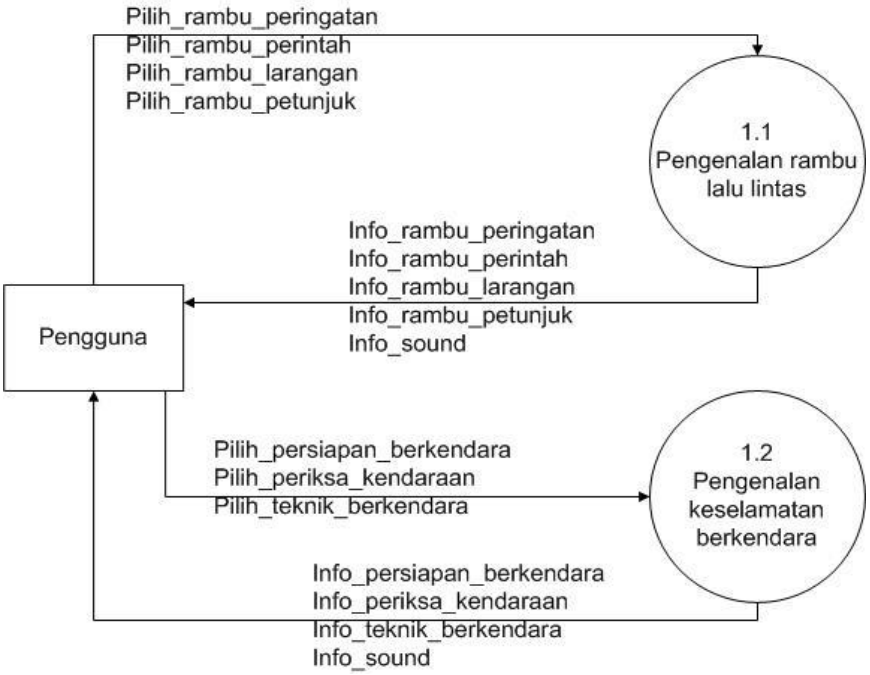
DFD *level* 1 dibuat jika pada diagram konteks masih terdapat proses yang harus dijelaskan secara lebih rinci. DFD *level* 1 terdapat proses-proses, yaitu proses menampilkan materi, proses mulai bermain, menampilkan petunjuk, dan menampilkan pembuat. Lihat pada gambar 3.14.



Gambar 3. 14 DFD *Level* 1 *Game* Edukasi *Safety Riding*

2. DFD *Level* 2 Proses 1 Penyampaian Materi

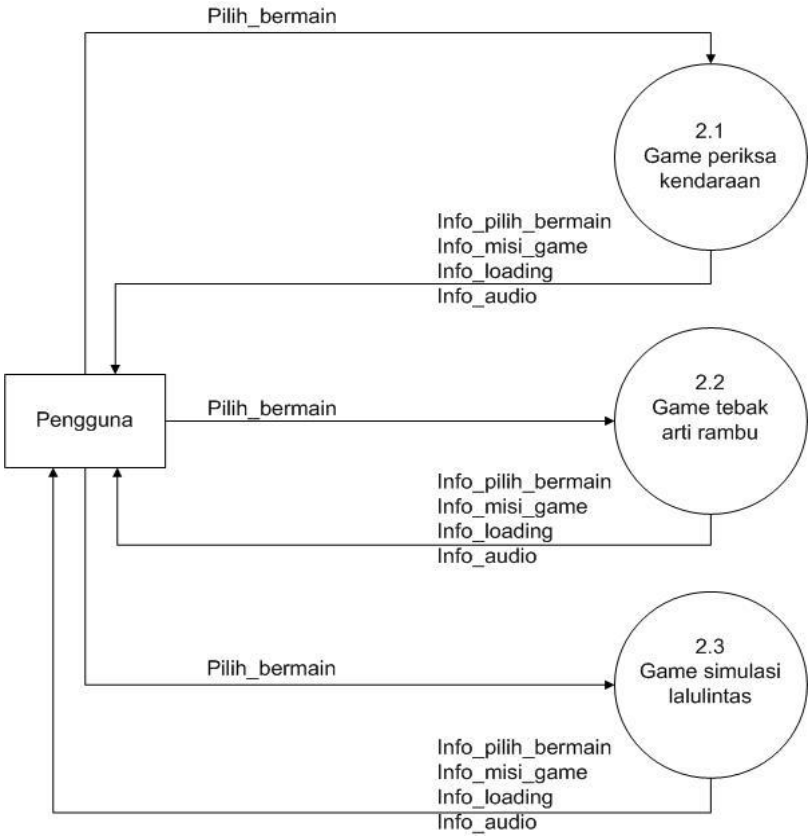
Proses yang terdapat pada DFD *level* 2 proses 1 adalah proses penyampaian materi yang terdiri atas proses 1.1 pengenalan rambu lalu lintas, dan proses 1.2 pengenalan keselamatan berkendara. DFD *level* 2 proses 1 penyampaian materi dapat dilihat pada gambar 3.15.



Gambar 3. 15 DFD *Level* 2 Proses 1 Penyampaian Materi

3. DFD *Level* 2 Proses 2 Mulai Bermain

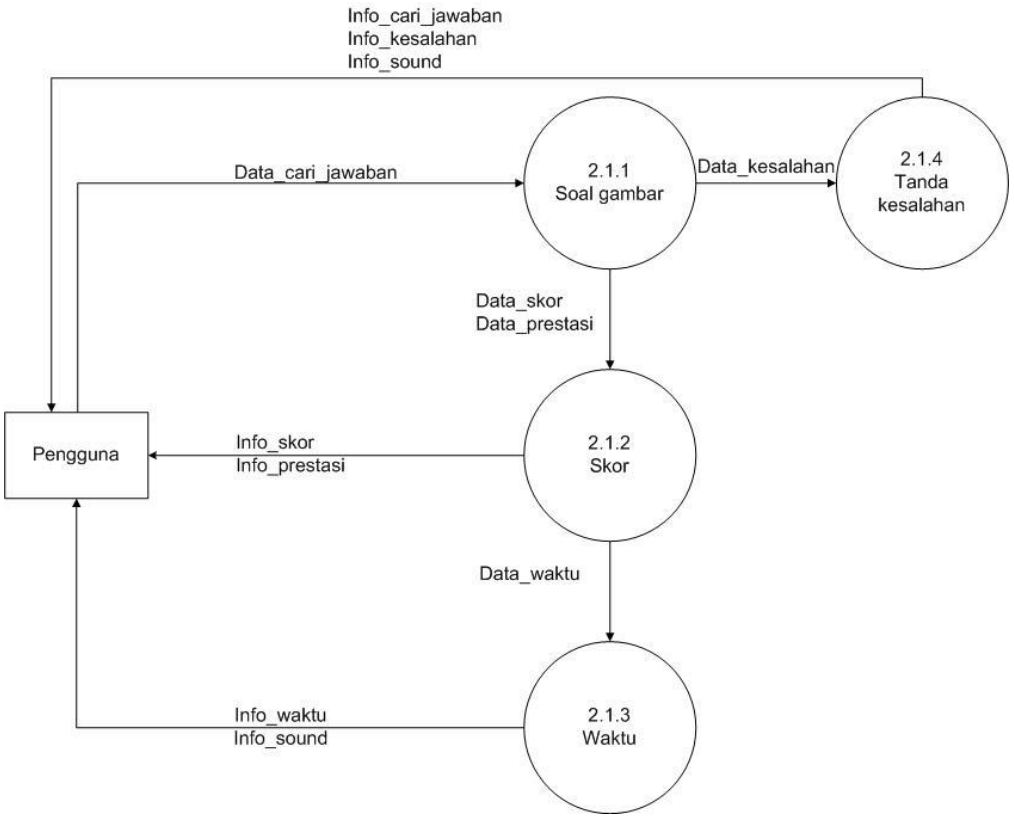
Proses yang terdapat pada DFD *level* 2 proses 2 bermain adalah proses bermain yang terdiri atas proses 2.1 permainan periksa kendaraan, proses 2.2 permainan tebak arti rambu, dan proses 2.3 permainan simulasi lalu lintas. DFD *level* 2 proses 2 muali bermain dapat dilihat pada gambar 3.16.



Gambar 3. 16 DFD *Level* 2 Proses 2 Mulai Bermain

4. DFD *Level* 3 Proses 2.1 *Game* Periksa Kendaraan

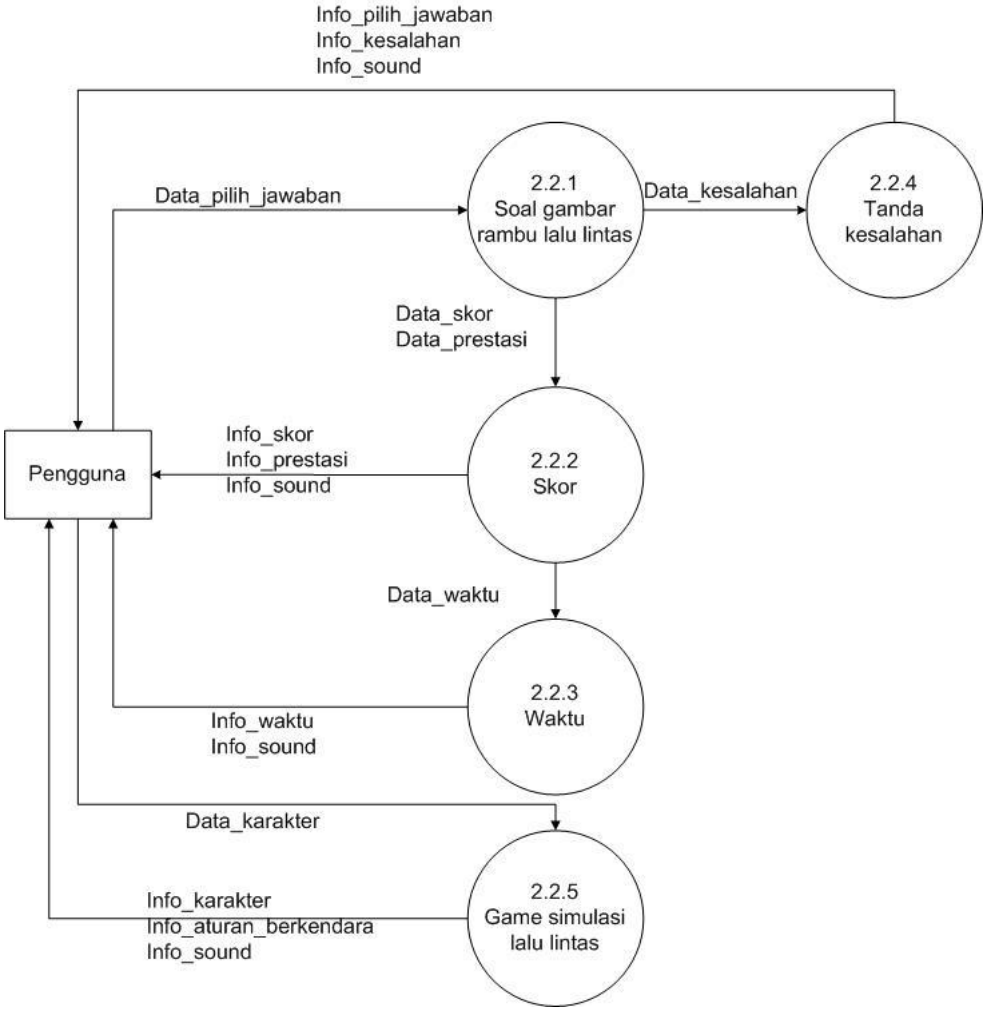
Proses yang terdapat pada DFD *level* 3 proses 2.1 adalah mulai *game* periksa kendaraan yang terdiri atas proses 2.1.1 menampilkan perbaikan, proses 2.1.2 menampilkan skor, proses 2.1.3 menampilkan waktu, dan proses 2.1.4 menampilkan tanda kesalahan. DFD *level* 2 proses 2.1 periksa kendaraan dapat dilihat pada gambar 3.17.



Gambar 3. 17 DFD *Level* 3 Proses 2.1 *Game* Periksa Kendaraan

5. DFD *Level* 3 Proses 2.2 Permainan Tebak Arti Rambu

Proses yang terdapat pada DFD *level* 3 proses 2.2 adalah mulai *game* tebak arti rambu yang terdiri atas proses 2.2.1 mengacak soal rambu lalu lintas, proses 2.2.2 menampilkan skor, proses 2.2.3 menampilkan waktu, proses 2.2.4 menampilkan tanda kesalahan, dan proses 2.2.5 simulasi lalu lintas. DFD *level* 3 proses 2.2 *game* tebak arti rambu dapat dilihat pada gambar 3.18.



Gambar 3.18 DFD *Level* 3 Proses 2.2 *Game* Tebak Arti Rambu

3.1.8.3. Spesifikasi Proses

Spesifikasi proses adalah deskripsi dari setiap elemen proses yang terdapat dalam program, yang meliputi nama *process*, *input*, *output* dan keterangan dari proses. Spesifikasi proses dari aplikasi *game* edukasi *safety riding* ini ditujukan pada tabel 3.10.

Tabel 3. 10 Spesifikasi Proses

No	Proses	Keterangan
1	No Proses	1.1
	Nama	Pengenalan rambu lalu lintas
	Source (Sumber)	Pengguna
	Deskripsi	Sistem akan memproses dan menampilkan materi belajar pengenalan rambu lalu lintas
	Input	Pilih materi
	Output	Info materi, info sound
	Logika proses	1. Pengguna memilih menu materi pada menu utama 2. Sistem akan memproses dan menampilkan pilihan materi proses 1.1 dan materi proses 1.2. 3. Sistem akan memproses pilih materi 4. Sistem akan menampilkan informasi materi untuk belajar dan info sound
No	Proses	Keterangan
2	No Proses	1.2
	Nama	Pengenalan keselamatan berkendara
	Source (Sumber)	Pengguna
	Deskripsi	Sistem akan memproses dan menampilkan materi belajar pengenalan keselamatan berkendara
	Input	Pilih materi
	Output	Info materi, info *sound*
	Logika proses	1. Pengguna memilih menu materi pada menu utama 2. Sistem akan memproses dan menampilkan pilihan materi proses 1.2. 3. Sistem akan memproses pilih materi 4. Sistem akan menampilkan informasi materi untuk belajar dan info sound
No	Proses	Keterangan
3	No Proses	2.1
	Nama	*Game* Periksa Kendaraan
	Sumber	Pengguna
	Deskripsi	Sistem akan memproses dan menampilkan permainan periksa kendaraan
	Input	Pilih bermain

	Output	Info pilih bermain, info misi game, info loading, info audio, dan info suara
	Logika proses	1. Pengguna memilih bermain periksa kendaraan dengan menggunakan tombol *mouse* kiri 2. Sistem menampilkan informasi misi *game dan* informasi *loading game* periksa kendaraan 3. Sistem menampilkan informasi audio, informasi suara 4. Sistem akan memproses permainan periksa kendaraan dan 5. Sistem akan menampilkan permainan periksa kendaraan
No	Proses	Keterangan
4	No Proses	2.2
	Nama	*Game* Tebak Arti Rambu
	Sumber	Pengguna
	Deskripsi	Sistem akan memproses dan menampilkan informasi permainan tebak arti rambu
	Input	Pilih bermain
	Output	Info pilih bermain, info misi game, info loading, info audio, dan info suara
	Logika proses	1. Pengguna memilih bermain tebak arti rambu dengan menggunakan tombol *mouse* kiri 2. Sistem menampilkan informasi misi *game dan* informasi *loading game* tebak arti rambu 3. Sistem menampilkan informasi audio, informasi suara 4. Sistem akan memproses permainan tebak arti rambu dan 5. Sistem akan menampilkan permainan tebak arti rambu
No	Proses	Keterangan
5	No Proses	2.3
	Nama	*Game* simulasi lalu lintas
	Sumber	Pengguna
	Deskripsi	Menampilkan berupa informasi simulasi lalu lintas
	Input	Pilih bermain
	Output	Info pilih bermain, info misi *game*, info loading, info audio, dan info suara
	Logika proses	1. Pengguna memilih bermain simulasi lalulintas dengan menggunakan tombol *mouse* kiri 2. Sistem menampilkan informasi misi *game dan* informasi

		loading game simulasi lalu lintas 3. Sistem menampilkan informasi audio, dan informasi suara 4. Sistem akan memproses permainan simulasi lalu lintas dan 5. Sistem akan menampilkan permainan simulasi lalu lintas
No	Proses	Keterangan
6	No Proses	2.1.1
	Nama	Soal gambar
	Sumber	Pengguna dan sistem
	Deskripsi	Sistem akan menampilkan gambar perbaikan kendaraan ke layar permainan
	Input	Data cari jawaban
	Output	Info cari jawaban
	Logika proses	1. Pengguna memilih proses 2.1 2. Sistem akan menampilkan soal gambar ke layar permainan 3. Pengguna mencari jawaban gambar menggunakan tombol *mouse* kiri 4. Sistem akan mengecek benar atau salah jawaban 5. Sistem akan menampilkan info cari jawaban
No	Proses	Keterangan
7	No Proses	2.1.2
	Nama	Skor
	Sumber	Sistem
	Deskripsi	Sistem menampilkan skor permainan
	Input	Data skor, dan data prestasi
	Output	Info skor, dan info prestasi, dan info *sound*
	Logika proses	1. Sistem akan melakukan proses 2.1.1 2. Sistem akan menginisialisasikan skor awal=0, skor awal tidak dapat mengurangi skor. 3. Jika ya maka sistem akan memproses data skor dan menambahkan skor 100, dan jika tidak maka sistem akan mengurangi skor 25. 4. Sistem akan menampilkan infomasi skor , informasi prestasi dan informasi *sound*
No	Proses	Keterangan
8	No Proses	2.1.3
	Nama	waktu

	Sumber	Sistem
	Deskripsi	Sistem akan menampilkan waktu selama permainan berlangsung
	Input	Data waktu
	Output	Info waktu, dan info *sound*
	Logika proses	1. Sistem akan melakukan proses 2.1.1 dan proses 2.2.1 2. Sistem akan menginisialisasikan waktu awal=30 detik selama permainan 3. Sistem akan memproses data waktu 4. Sistem akan menampilkan infomasi waktu dan informasi *sound*
No	Proses	Keterangan
9	No Proses	2.1.4
	Nama	Tanda kesalahan
	Sumber	Sistem
	Deskripsi	Sistem menampilkan data tanda kesalahan
	Input	Data kesalahan dan data cari jawaban
	Output	info kesalahan, info cari jawaban dan infomasi *sound*
	Logika proses	1. Sistem akan melakukan proses 2.1.1 2. Sistem akan menginisialisasikan kesalahan awal=0 3. Jika ya, maka sistem akan memproses data kesalahan dan menampilkan informasi kesalahan dan informasi *sound*, jika tidak, maka sistem akan memproses dan menampilkan informasi cari jawaban
No	Proses	Keterangan
10	No Proses	2.2.1
	Nama	Soal gambar rambu lalulintas
	Sumber	Pengguna dan Sistem
	Deskripsi	Sistem akan mengacak soal gambar
	Input	Data pilih jawaban
	Output	Info pilih jawaban
	Logika proses	1. Pengguna memilih proses 2.2. 2. Sistem akan menampilkan soal gambar yang sudah diacak 3. Sistem akan menginisialisasikan kesalahan=0,dan skor=0 4. Pengguna menjawab soal gambar dengan menggunakan tombol *mouse* kiri dan memilih isi jawaban huruf yang ada disebuah himpunan kandidat (C). C={A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P,Q,R,S,T,U,V,W,X,Y,Z}

		5. Setelah di hitung maka sistem akan menyeleksi huruf (fungsi seleksi) yang dipilih oleh pengguna dari yang ada dihimpunan kandidat (C). 6. Sistem akan memproses dan memeriksa data huruf jawaban yang dipilih oleh pengguna dan membandingkan isi huruf jawaban yang ada disistem sebuah himpunan solusi (S). S={A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P,Q,R,S,T,U,V,W,X,Y,Z} 7. Sistem akan memeriksa fungsi kelayakan apakah jawaban huruf sudah terbentuk atau tidak dengan cara : a. Jika ya huruf tersedia atau sudah terbentuk pada isi jawaban dihimpunan solusi yang dipilih oleh pengguna, maka sistem akan menghitung panjang huruf dari jawaban dan menyediakan panjang huruf untuk menampilkan slot kosong sesuai panjang huruf jawaban dan huruf jawaban yang dipilih akan ditampilkan ke layar oleh sistem dan huruf yang dipilih tersebut dihimpunan kandidat (C) akan membuangnya agar tidak dipilih kembali. b. Jika tidak huruf tersedia atau belum terbentuk pada isi jawaban dihimpunan solusi, maka jawaban huruf disebuah himpunan kandidat(C) akan dibuang dan dikurangi oleh sistem agar tidak dipilih kembali serta menambahkan variabel kesalahan=1 kemudian menampilkan simbol tanda kesalahan ke layar. 8. Begitu seterusnya pengguna melakukan langkah(4) dan sistem akan mengulangi langkah (5), langkah (6), langkah (7) dan langkah (8) sampai hasil akhir solusi optimal. Apabila : a. Jika benar maka jawaban huruf sudah terbentuk jawaban huruf yang solusi yang lengkap. b. Jika tidak maka jawaban huruf belum terbentuk solusi yang lengkap.
No	Proses	Keterangan
11	No Proses	2.2.2
	Nama	Skor
	Sumber	Sistem
	Deskripsi	Sistem akan menampilkan skor permainan
	Input	Data skor, data prestasi
	Output	Info skor , info prestasi dan info *sound*

	Logika proses	1. Sistem akan melakukan proses 2.2.1 2. Sistem akan menginisialisasikan skor awal=0 3. Sistem akan memproses data skor 4. Sistem akan menampilkan infomasi skor
No	Proses	Keterangan
12	No Proses	2.2.3
	Nama	Menampilkan waktu
	Sumber	Sistem
	Deskripsi	Menampilkan waktu permainan
	Input	Data jawaban, Data waktu
	Output	Info jawaban, Info waktu
	Logika proses	1. Sistem akan menginisialisasikan waktu awal=30 detik 2. Sistem akan melakukan proses 2.2.1 3. Sistem akan memproses data waktu 4. Sistem akan menampilkan infomasi waktu
No	Proses	Keterangan
13	No Proses	2.2.4
	Nama	Menampilkan kesalahan
	Sumber	Sistem
	Deskripsi	Menampilkan data kesalahan
	Input	Data jawaban, data kesalahan
	Output	Info jawaban, info kesalahan,
	Logika proses	1. Sistem akan menginisialisasikan kesalahan awal=0 2. Sistem akan melakukan proses 2.2.1 3. Sistem akan memproses data kesalahan 4. Sistem akan mengecek benar atau salah isi jawaban 5. Sistem akan menampilkan infomasi kesalahan
No	Proses	Keterangan
14	No Proses	2.2.5
	Nama	Simulasi Lalu Lintas
	Sumber	Pengguna, sistem
	Deskripsi	Proses menggerakan karakter
	Input	Data karakter
	Output	Info karakter
	Logika proses	1. Pengguna menekan panah kiri, panah kanan, panah atas, dan *space* pada *keyboard* dan klik kiri pada *mouse*

		2. Karakter bergerak sesuai dengan yang diperintahkan 3. Jika pengguna menekan tombol panah atas karakter akan bergerak ke atas 4. Jika pengguna menekan tombol *space* pada *keyboard* karakter akan membunyikan suara klakson 5. Jika pengguna menekan tombol panah kiri karakter akan bergerak ke kiri 6. Jika pengguna menekan tombol panah kanan karakter akan bergerak ke kanan 7. Apabila antara player1 terjadi *collision* objek2 maka a. Jika *true*, maka sistem akan menampilkan informasi pelanggaran lalu lintas. b. Jika *false*, maka sistem tidak menampilkan informasi pelanggaran lalu lintas. 8. Jika pengguna mengklik kiri pada *mouse* untuk keluar dari simulasi lalu lintas

3.1.8.4 Kamus Data

Kamus data atau *data dictionary* adalah katalog fakta tentang data dan kebutuhan-kebutuhan informasi dari suatu sistem. Kamus data untuk diagram arus data pada *game* edukasi *safety riding* ini adalah dapat dilihat pada tabel 3.11.

Tabel 3. 11 Kamus Data

No	Detail	Keterangan
1.	Nama	Pilih materi
	Deskripsi	Berisi data mengenai materi yang berkaitan dengan seputar rambu lalu lintas dan keselamatan berkendara
2.	Nama	Data permainan
	Deskripsi	Berisi tentang menu level permainan
3.	Nama	Data pilih jawaban
	Deskripsi	Berisi tentang data pilih jawaban
	Struktur data	Isi jawaban
	Isi Jawaban	*Type String* [*A..Z*] atau [*a..z*]
4.	Nama	Skor
	Deskripsi	Berisi data skor
	Struktur data	Nilai skor
	Nilai skor	*Type numeric* (*integer*) [0..*9*]

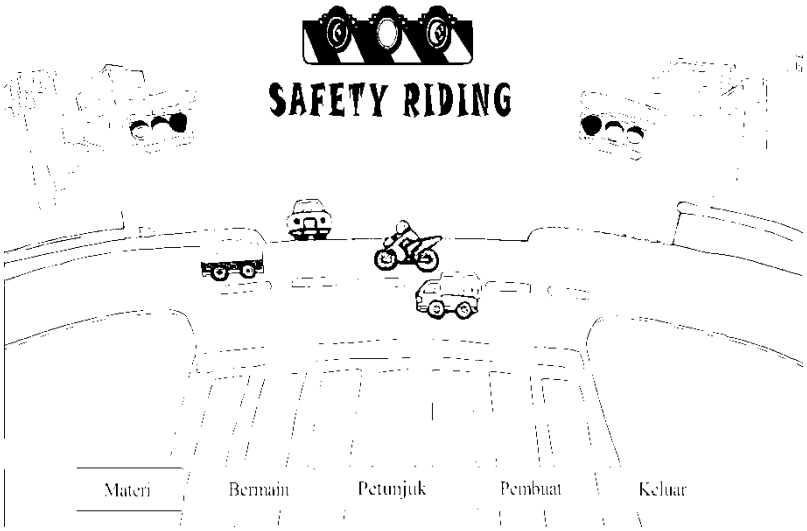
5.	Nama	Waktu
	Deskripsi	Berisi data waktu
	Struktur data	waktu
	waktu	*Type numeric* (*integer*) [0..*9*]

3.2. Perancangan Sistem

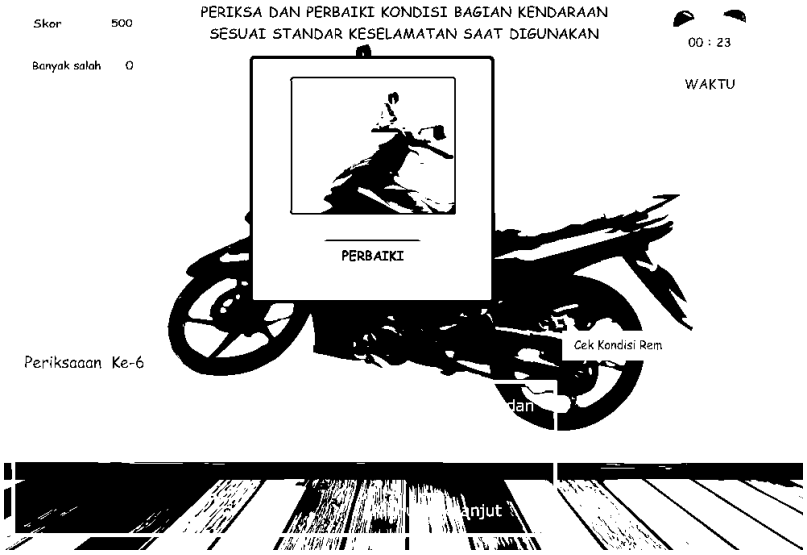
Perancangan adalah suatu bagian dari metodologi pengembangan pembangunan suatu perangkat lunak yang dilakukan setelah tahapan untuk memberikan gambaran secara terperinci.

3.2.1. Perancangan *Storyboard*

Storyboard merupakan gambar sketsa yang disusun berurutan sesuai dengan naskah, dengan *storyboard* dapat menyampaikan ide cerita kepada orang lain dengan lebih mudah. Berikut merupakan *storyboard* dari *game* edukasi *safety riding* untuk menarapkan keselamatan berlalu lintas adalah. Lihat gambar 3.19.



Gambar 3. 19 *Stroryboard* Tampilan Menu Utama



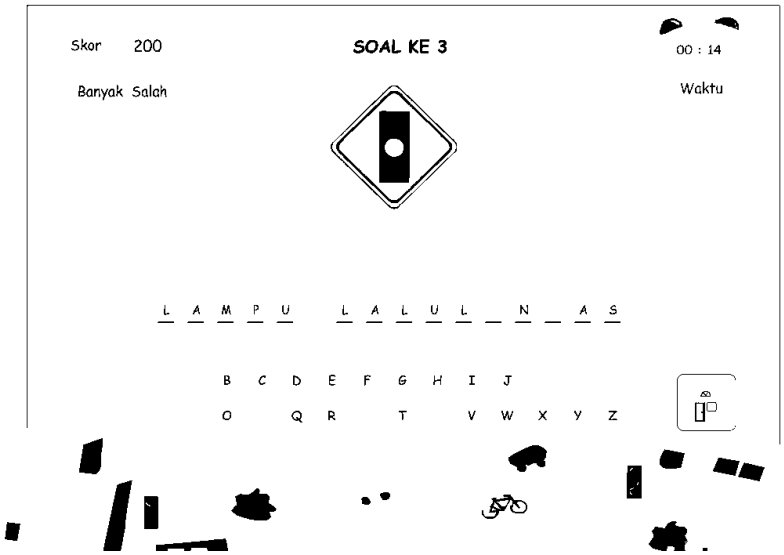
Gambar 3. 20 *Stroryboard Game* Periksa Kendaraan



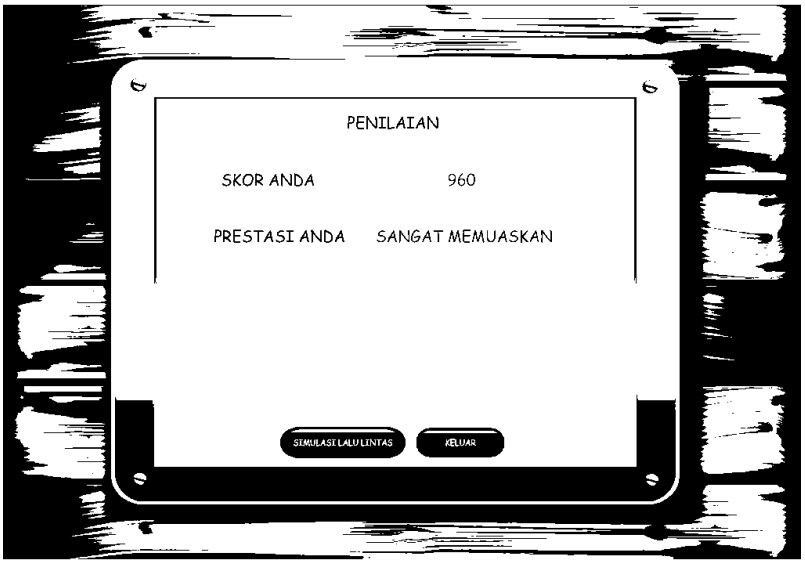
Gambar 3. 21 *Stroryboard* Pesan Konfirmasi Gagal



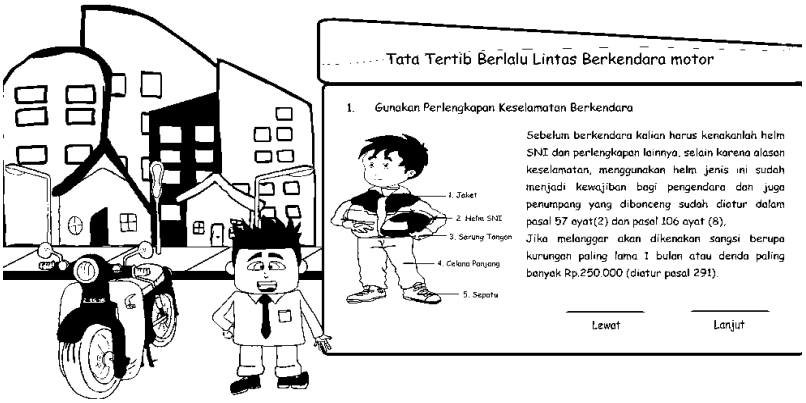
Gambar 3. 22 *Storyboard* Pesan Konfirmasi Berhasil



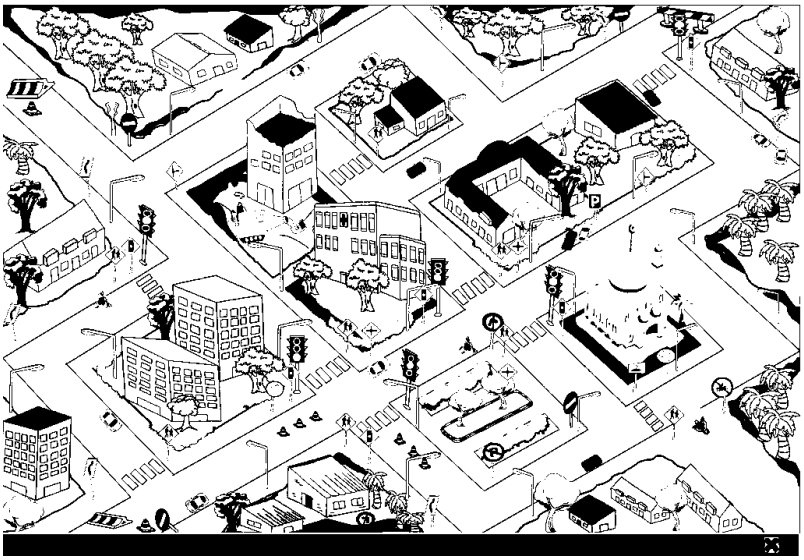
Gambar 3. 23 *Stroryboard Game* Tebak Arti Rambu



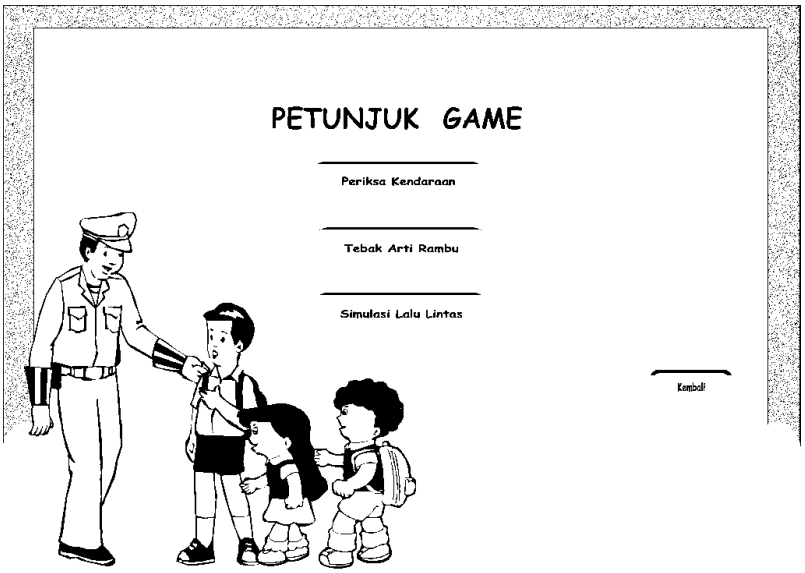
Gambar 3. 24 *Storyboard* Pesan Konfirmasi Penilaian



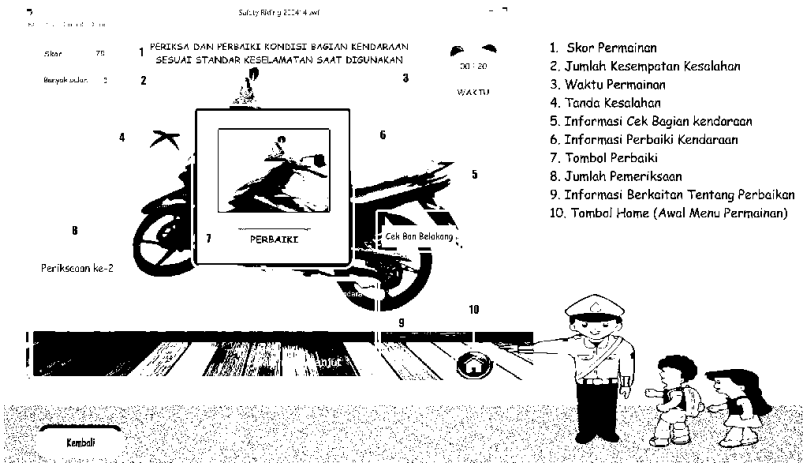
Gambar 3. 25 *Storyboard* Simulasi Lalu Lintas



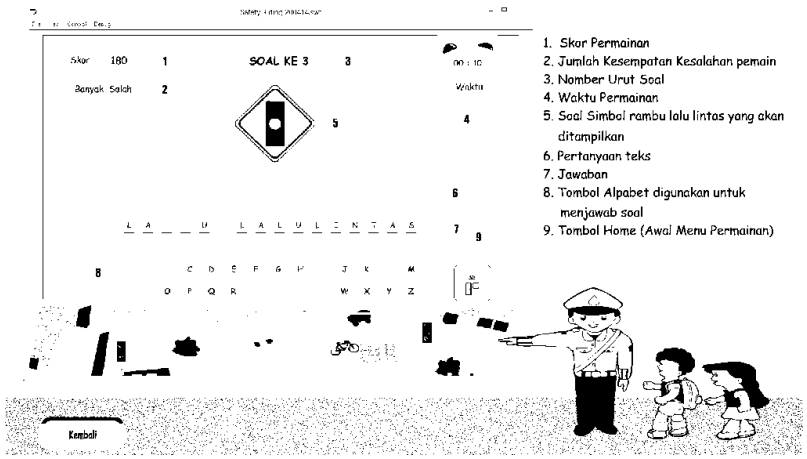
Gambar 3. 26 *Storyboard* Simulasi Lalu Lintas



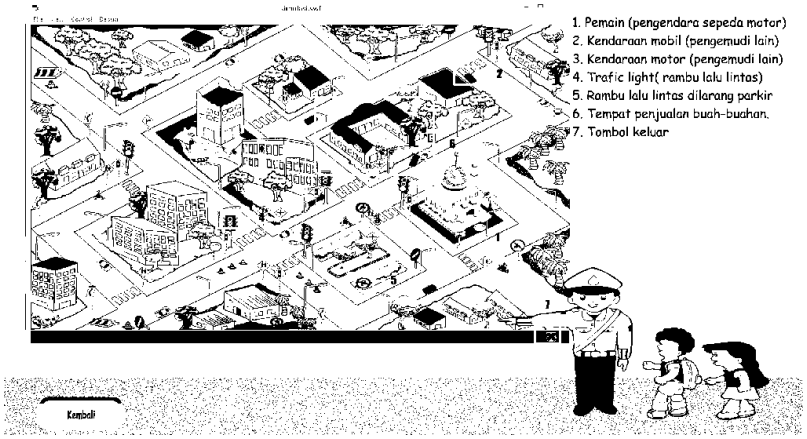
Gambar 3. 27 *Storyboard* Menu Petunjuk *Game*



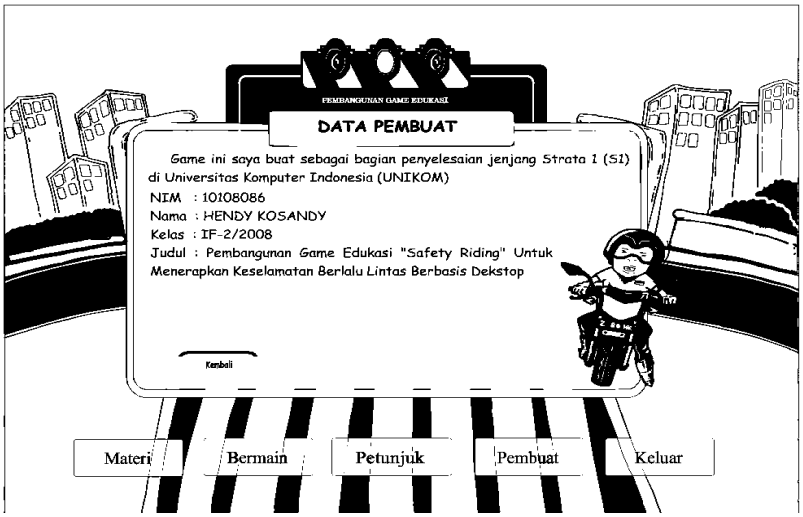
Gambar 3. 28 *Storyboard* Menu Petunjuk Periksa Kendaraan



Gambar 3. 29 *Storyboard* Menu Petunjuk Tebak Arti Rambu



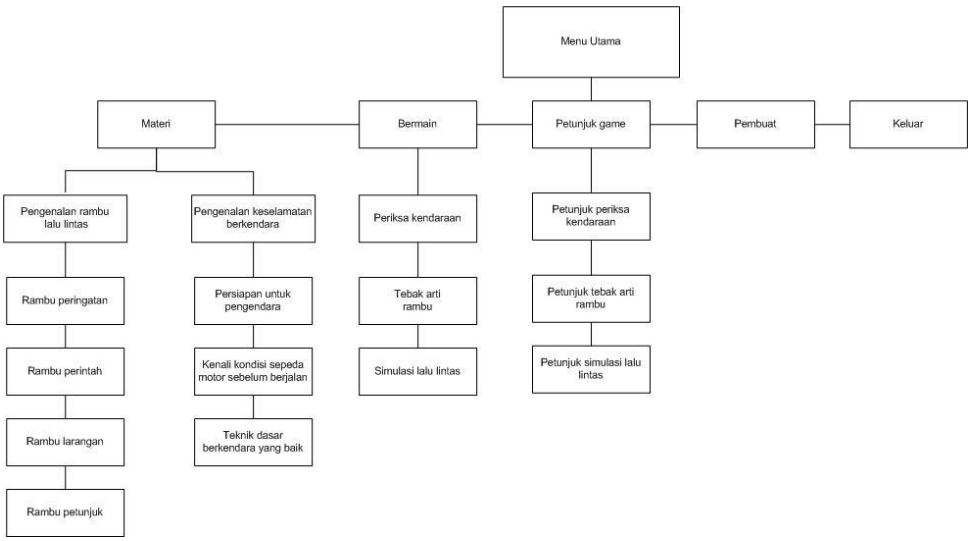
Gambar 3. 30 *Storyboard* Menu Petunjuk Simulasi Lalu Lintas



Gambar 3. 31 *Storyboard* Pembuat

3.2.2. Perancangan Struktur Menu

Perancangan struktur menu program ini membantu dalam merancang bagian-bagian dari sistem yang sebenarnya dan untuk mengetahui bagian mana yang nantinya akan diakses terlebih dahulu setelah program selesai dibuat. Perancangan arsitektur menu pada *game* edukasi *safety riding* untuk menerapkan keselamatan berlalu lintas dapat dilihat pada gambar 3.32.



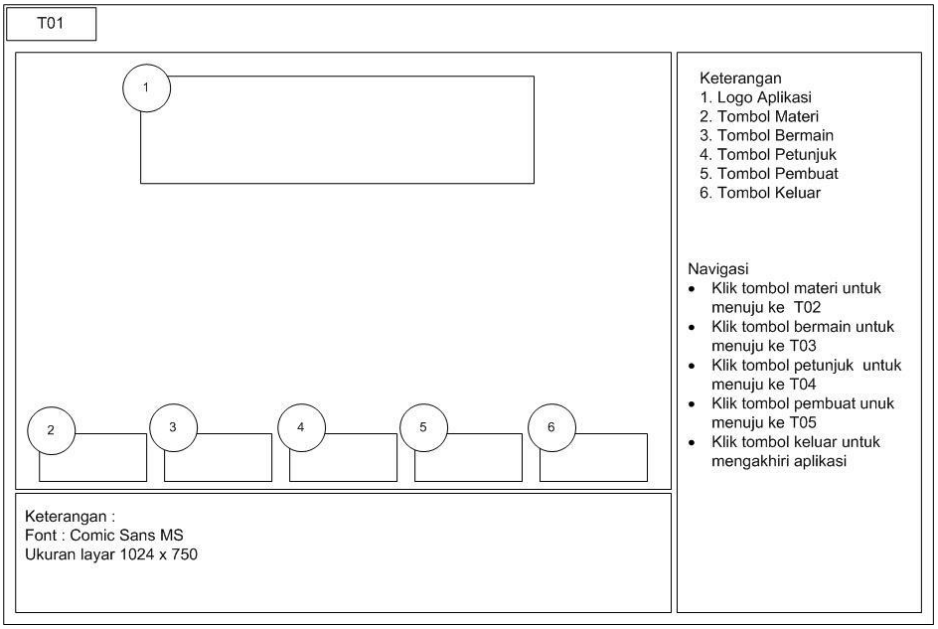
Gambar 3. 32 Struktur Menu *Game* Edukasi *Safety Riding* untuk Menerapkan Keselamatan Berlalu Lintas

3.2.3. Perancangan Antarmuka

Perancangan antarmuka dibutuhkan untuk mawakili keadaan sebenarnya dari aplikasi yang akan dibangun. Berikut ini contoh perancangan antarmuka yang ada dari aplikasi yang akan dibangun yaitu.

3.2.3.1. Perancangan Antarmuka Menu Utama

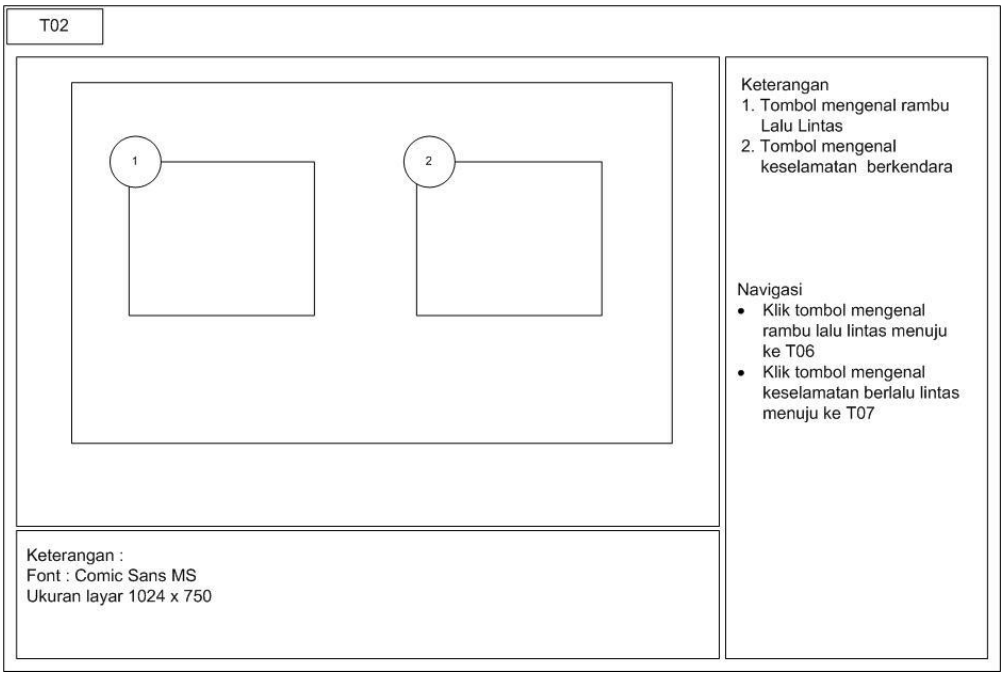
Perancangan antarmuka menu utama pada *game* yang dibangun dapat dilihat pada gambar 3.33.



Gambar 3. 33 Antarmuka Menu Utama

3.2.3.2. Perancangan Antarmuka Menu Kategori Materi

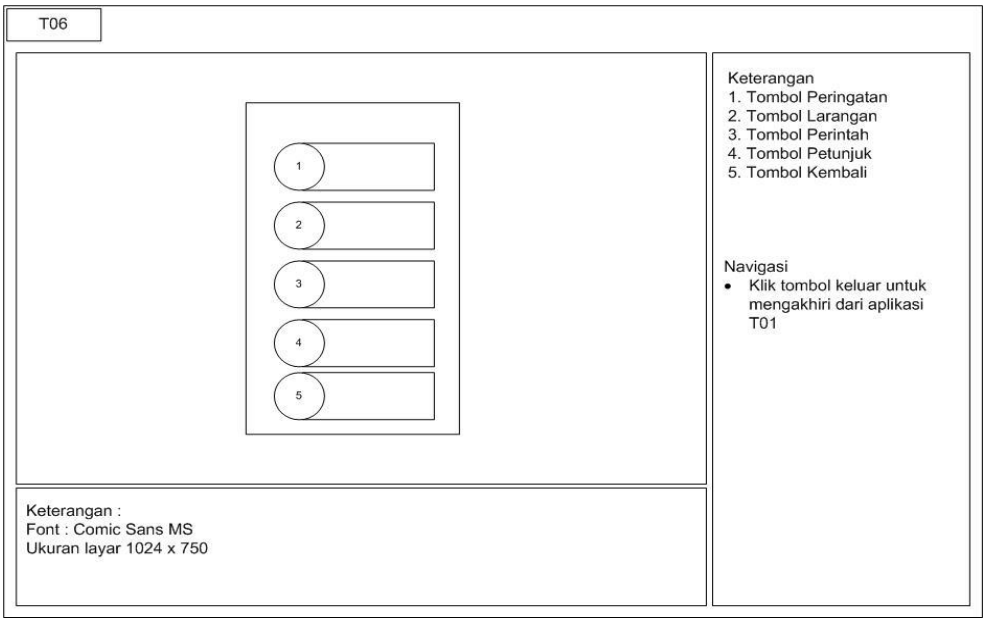
Perancangan antarmuka menu kategori materi dapat dilihat pada gambar 3.34.



Gambar 3. 34 Antarmuka Menu Kategori Materi

3.2.3.3. Perancangan Antarmuka Menu Materi Rambu

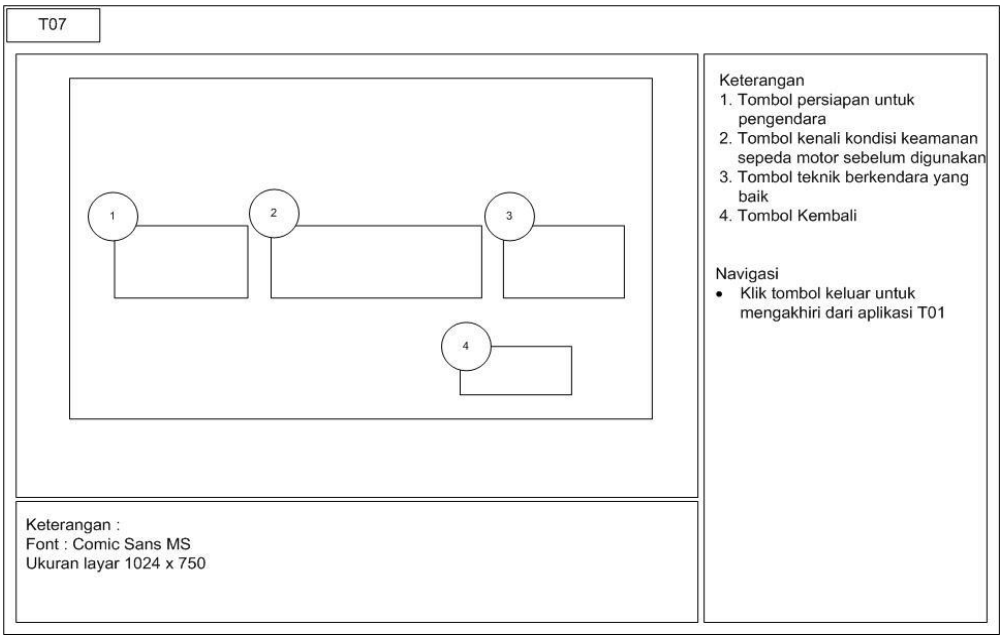
Perancangan antarmuka menu menu materi rambu. Lihat gambar 3.35.



Gambar 3. 35 Antarmuka Menu Materi Rambu

3.2.3.4. Perancangan Antarmuka Menu Keselamatan Berkendara

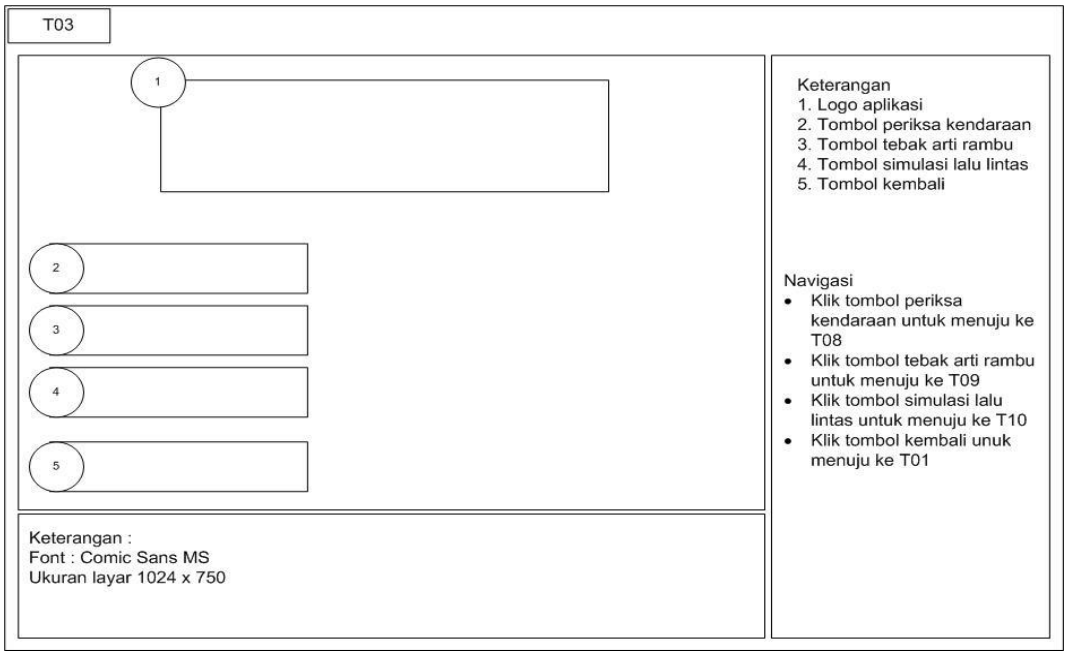
Perancangan antarmuka menu keselamatan berkendara dapat dilihat pada gambar 3.36.



Gambar 3. 36 Antarmuka Menu Materi Keselamatan Berkendara

3.2.3.5. Perancangan Antarmuka Menu Bermain

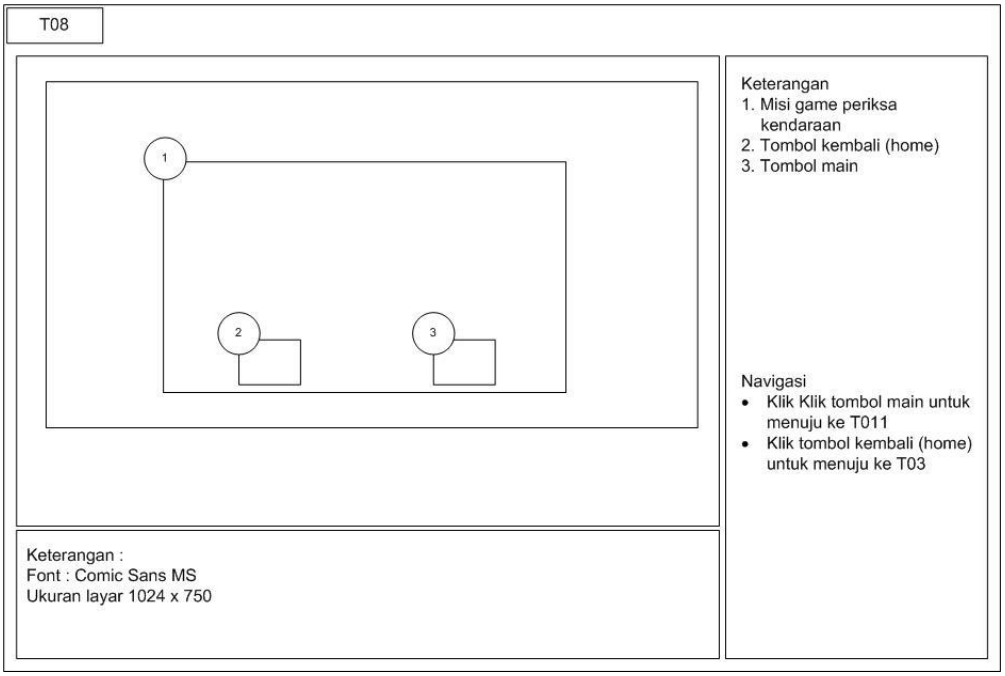
Perancangan antarmuka Menu bermain dapat dilihat pada gambar 3.37.



Gambar 3. 37 Antarmuka Menu Bermain

3.2.3.6. Perancangan Antarmuka Misi Game Periksa Kendaraan

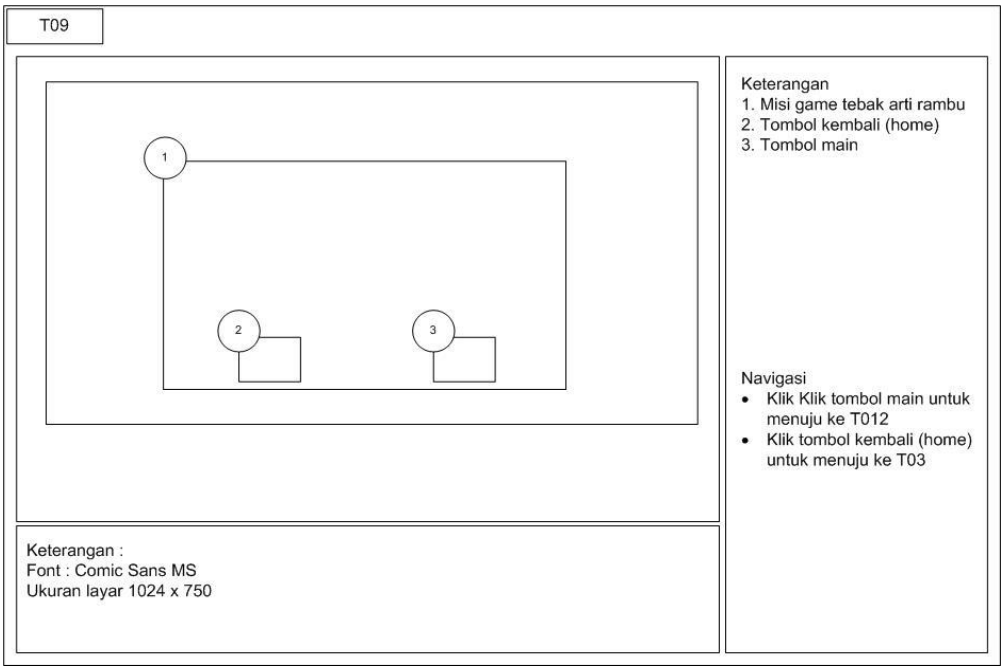
Perancangan antarmuka misi *game* periksa kendaraan.Lihat gambar 3.38.



Gambar 3. 38 Antarmuka Misi *Game* Tebak Arti Rambu

3.2.3.7. Perancangan Antarmuka Misi *Game* Perbaiki Kendaraan

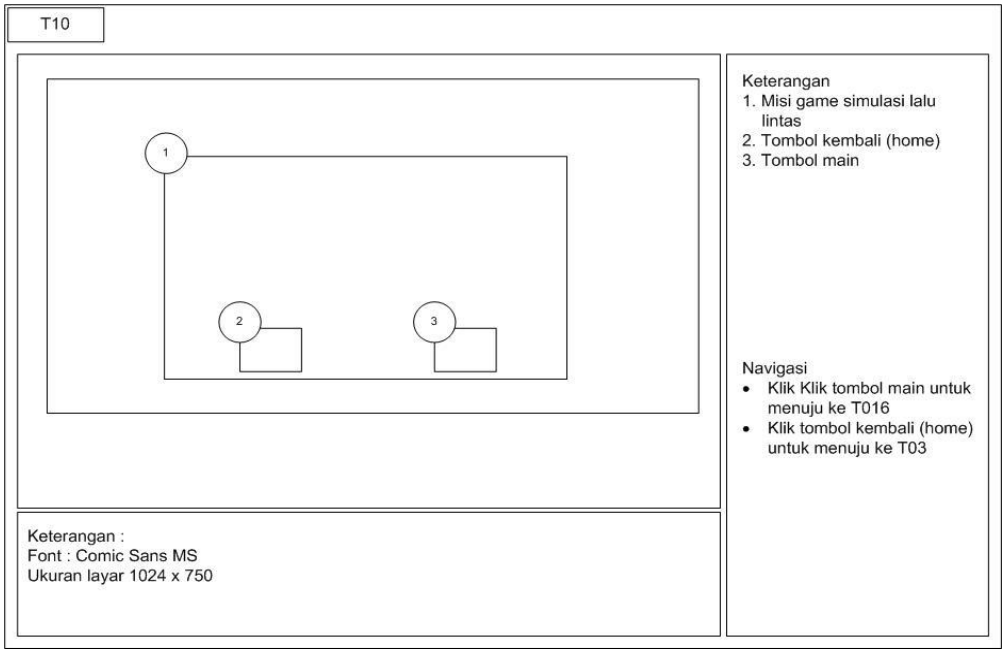
Perancangan antarmuka misi *game* perbaiki kendaraan lihat gambar 3.39.



Gambar 3. 39 Antarmuka Misi *Game* Periksa Kendaraan

3.2.3.8. Perancangan Antarmuka Misi *Game* Simulasi Lalu Lintas

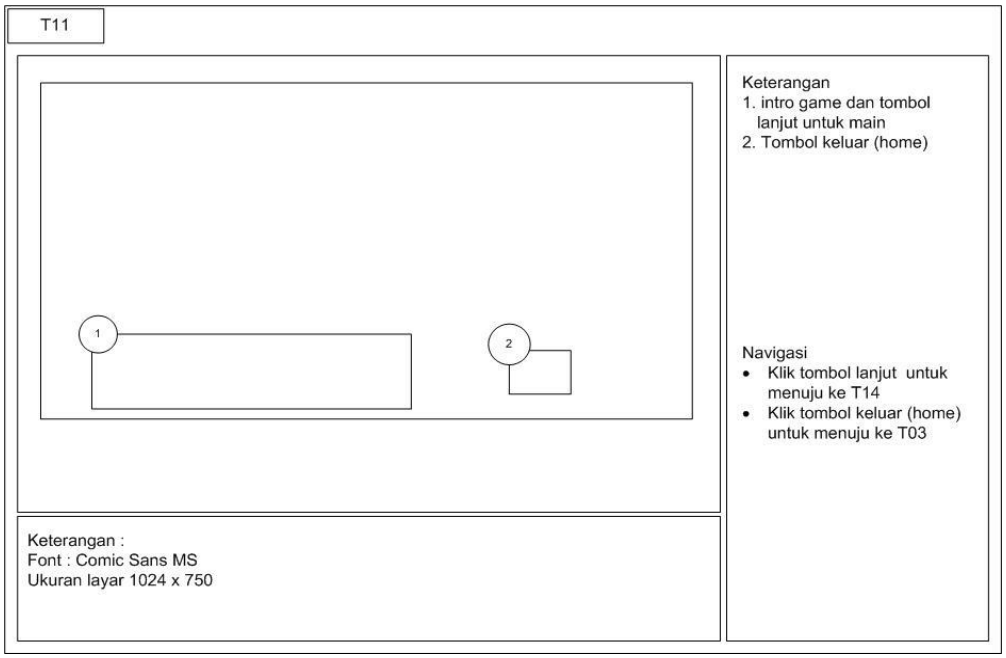
Perancangan antarmuka misi *game* simulasi lalu lintas lihat gambar 3.40.



Gambar 3. 40 Antarmuka *Game* Simulasi Lalu Lintas

3.2.3.9. Perancangan Antarmuka *Storyline Game* Periksa Kendaraan

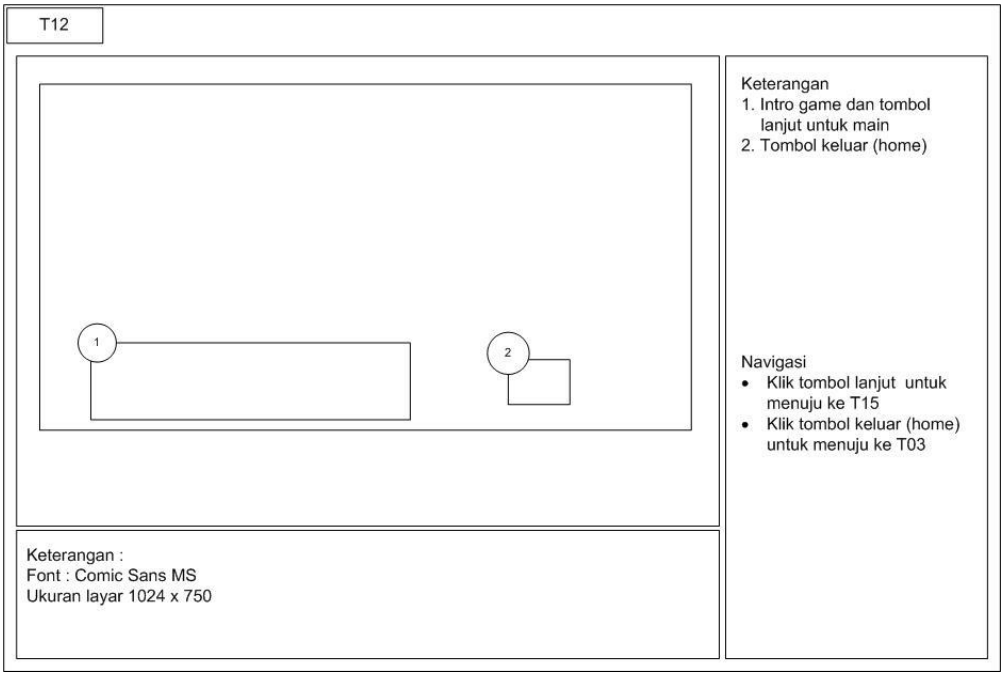
Perancangan antarmuka intro *game* periksa kendaraan lihat gambar 3.41.



Gambar 3. 41 Antarmuka *Storyline Game* Periksa Kendaraan

3.2.3.10. Perancangan Antarmuka *Storyline* Game Tebak Arti Rambu

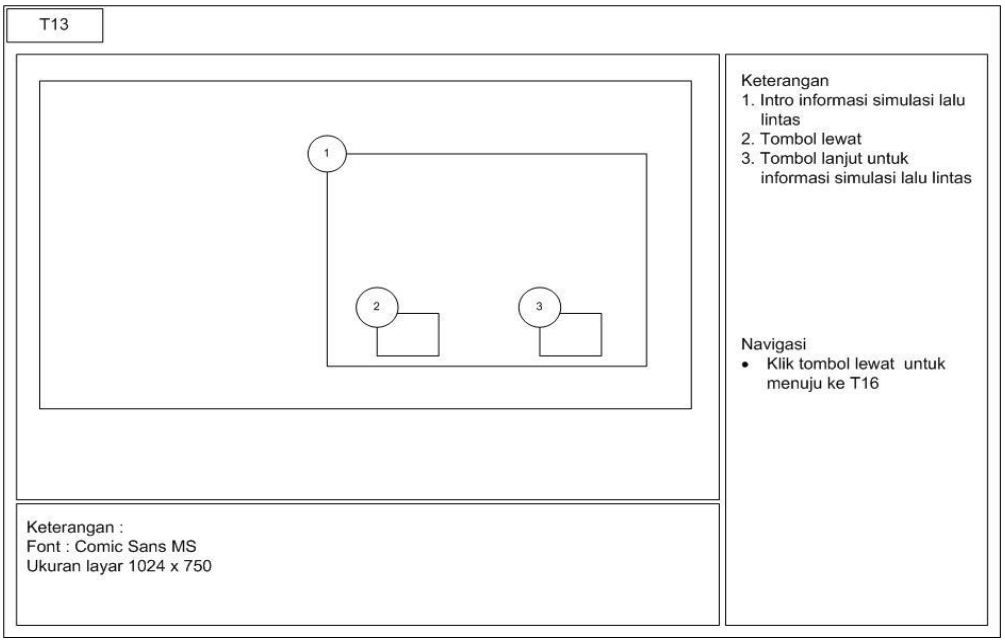
Perancangan antarmuka intro *game* tebak arti rambu lihat gambar 3.42.



Gambar 3. 42 Antarmuka *Storyline Game* Tebak Arti Rambu

3.2.3.11. Perancangan Antarmuka *Storyline* Game Simulasi Lalu Lintas

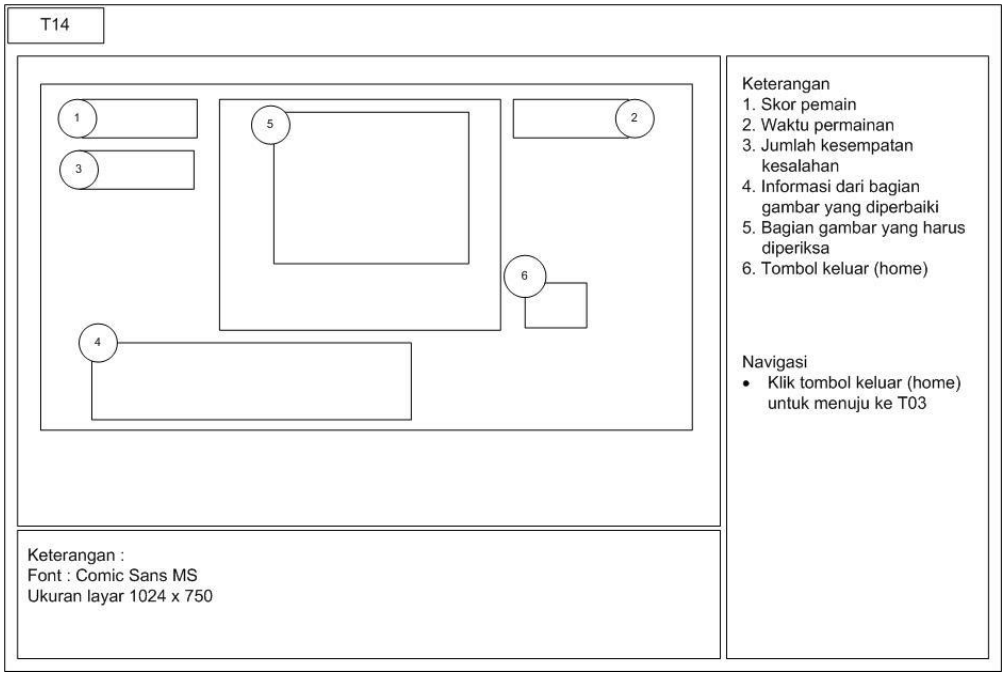
Perancangan antarmuka intro *game* simulasi lalu lintas lihat gambar 3.43.



Gambar 3. 43 Antarmuka *Storyline* Game Simulasi Lalu Lintas

3.2.3.12. Perancangan Antarmuka *Game* Periksa Kendaraan

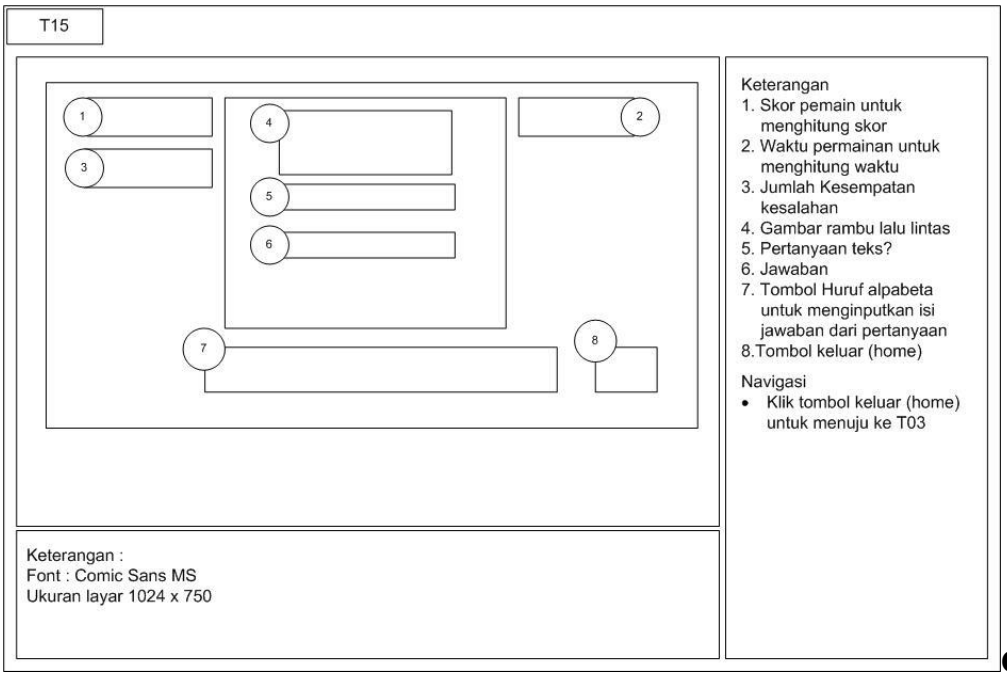
Perancangan antarmuka *game* periksa kendaran. Lihat gambar 3.44.



Gambar 3. 44 Antarmuka *Game* Periksa Kendaraan

3.2.3.13. Perancangan Antarmuka *Game* Tebak Arti Rambu

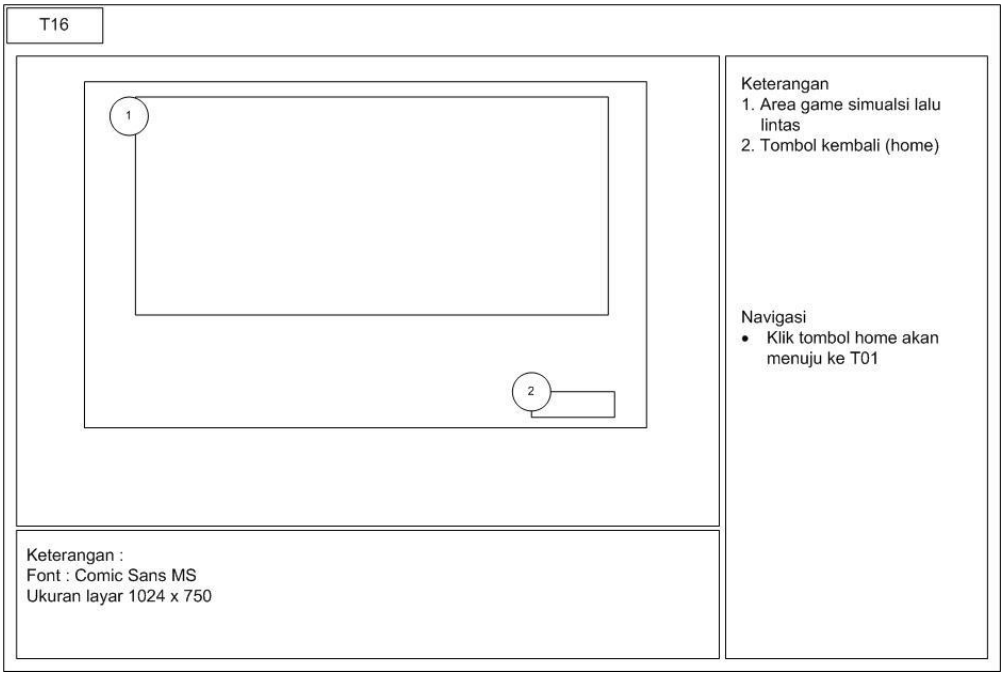
Perancangan antarmuka *game* tebak arti rambu lihat gambar 3.45.



Gambar 3. 45 Antarmuka *Game* Tebak Arti Rambu

3.2.3.14. Perancangan Antarmuka *Game* Simulasi Lalu Lintas

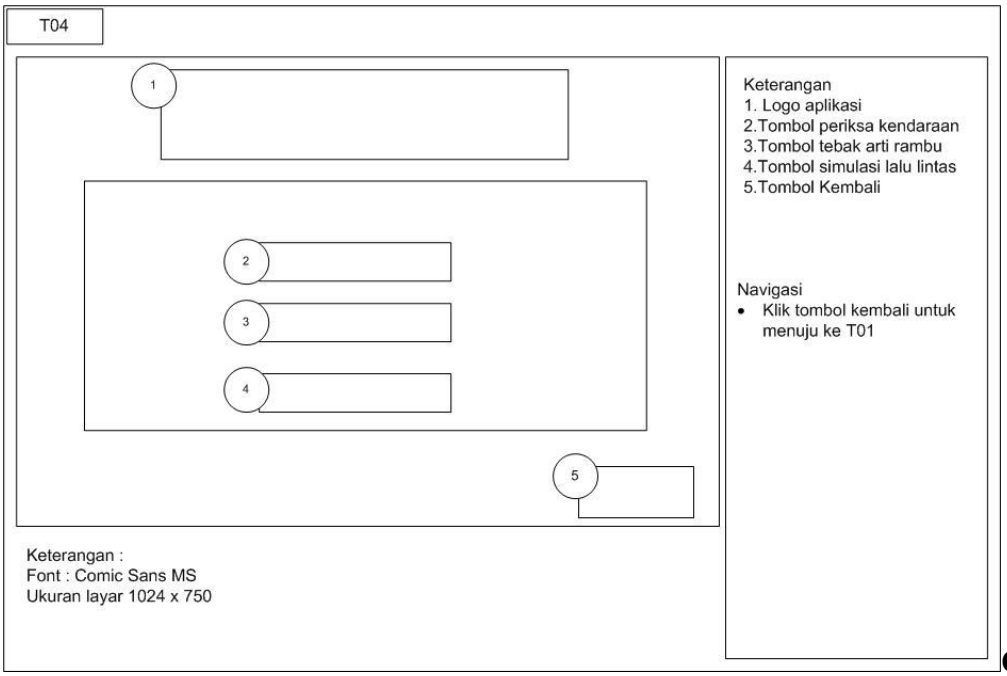
Perancangan antarmuka *game* simulasi lalu lintas lihat gambar 3.46.



Gambar 3. 46 Antarmuka *Game* Simulasi Lalu lintas

3.2.3.15. Perancangan Antarmuka Menu Petunjuk *Game*

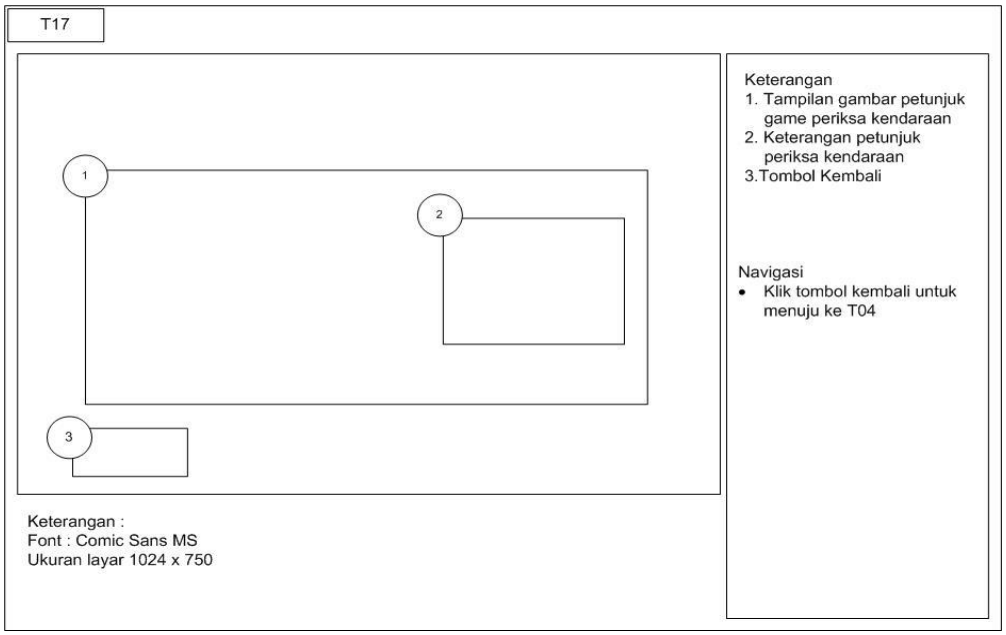
Perancangan antarmuka menu petunjuk *game* gambar 3.47.



Gambar 3. 47 Antarmuka Menu Petunjuk *Game*

3.2.3.16. Perancangan Antarmuka Petunjuk *Game* Periksa Kendaraan

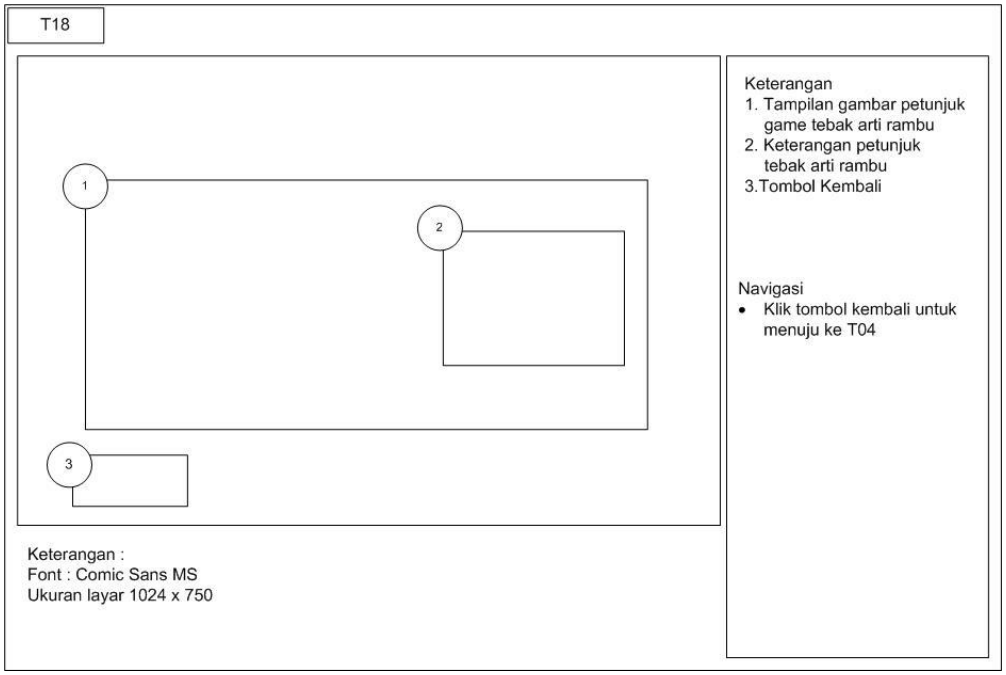
Perancangan antarmuka petunjuk *game* periksa kendaraan lihat gambar 3.48.



Gambar 3. 48 Antarmuka Petunjuk *Game* Periksa Kendaraan

3.2.3.17. Perancangan Antarmuka Petunjuk *Game* Tebak Arti Rambu

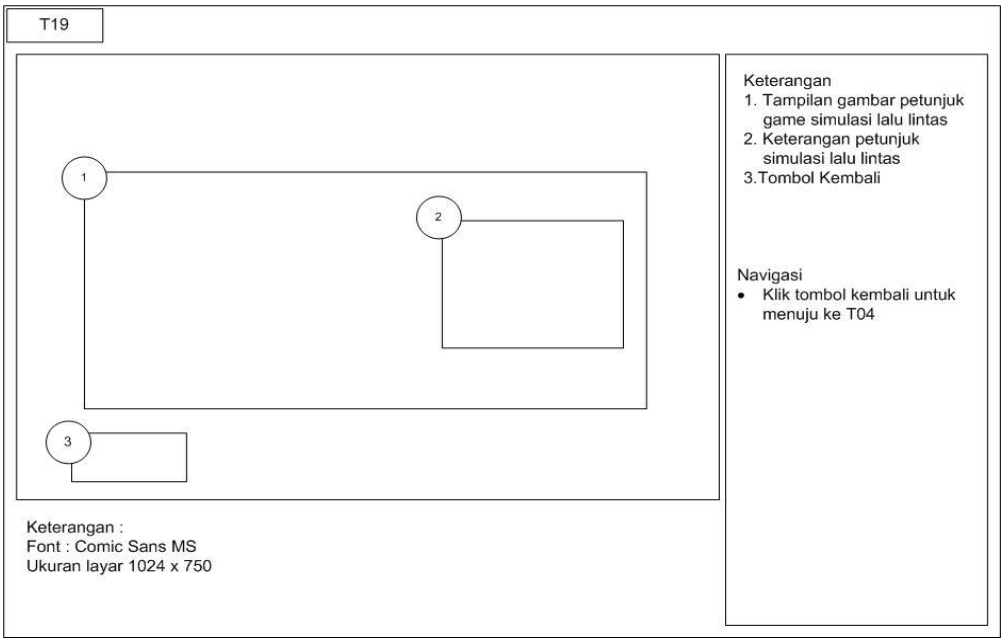
Perancangan antarmuka petunjuk *game* tebak arti rambu lihat gambar 3.49.



Gambar 3. 49 Antarmuka Petunjuk *Game* Tebak Arti Rambu

3.2.3.18. Perancangan Antarmuka Petunjuk *Game* Simulasi Lalu Lintas

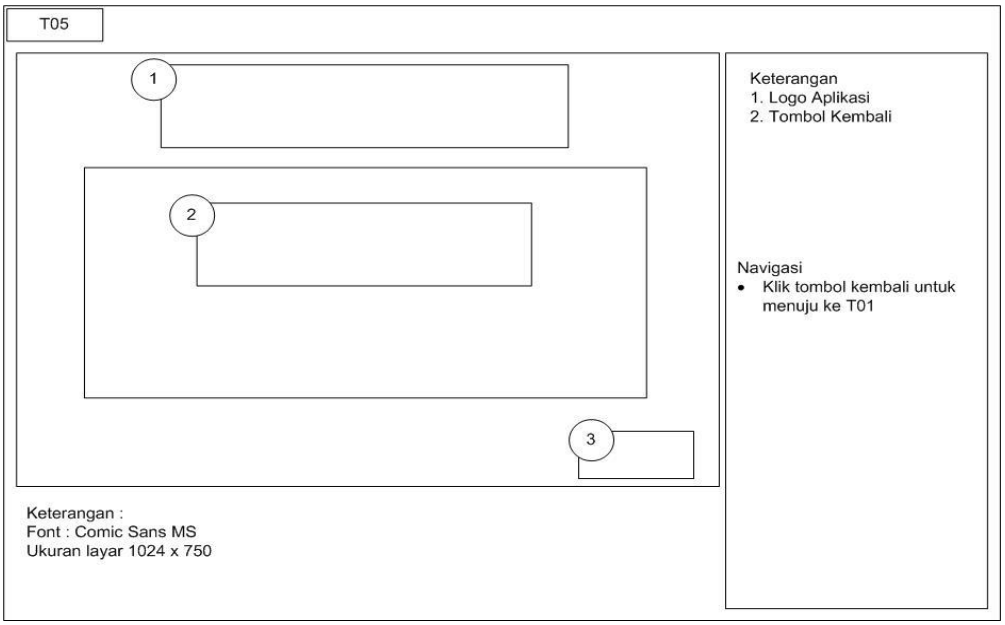
Perancangan antarmuka petunjuk *game* simulasi lalu lintas lihat gambar 3.50.



Gambar 3. 50 Antarmuka Petunjuk *Game* Simulasi Lalu Lintas

3.2.3.19. Perancangan Antarmuka Pembuat *Game*

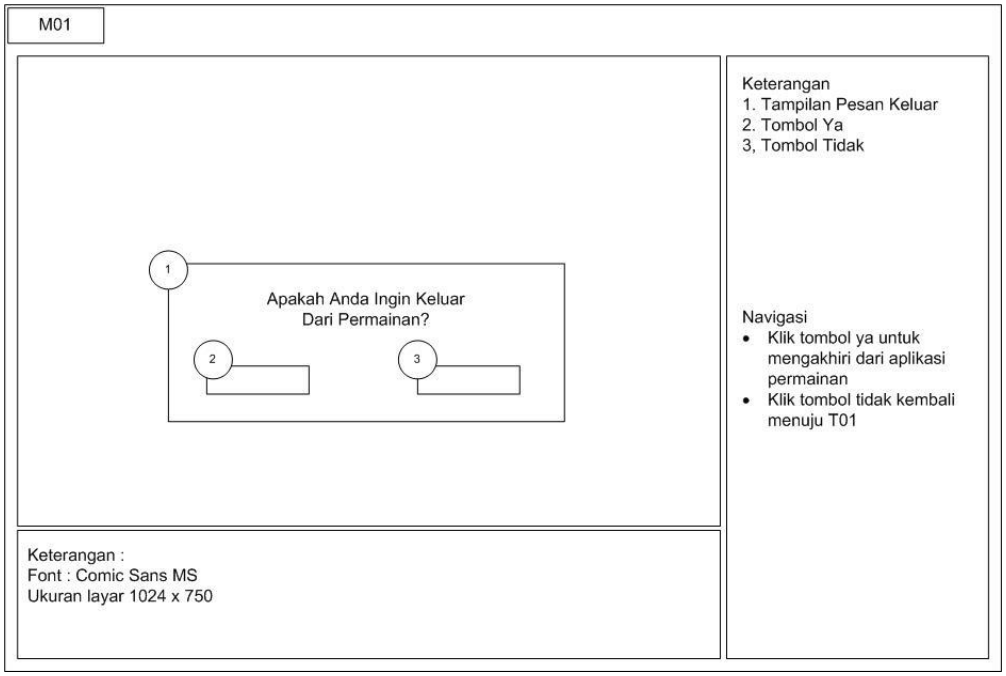
Perancangan antarmuka pembuat *game* lihat gambar 3.51.



Gambar 3. 51 Antarmuka Pembuat *Game*

3.2.3.20. Perancangan Antarmuka Pesan Keluar

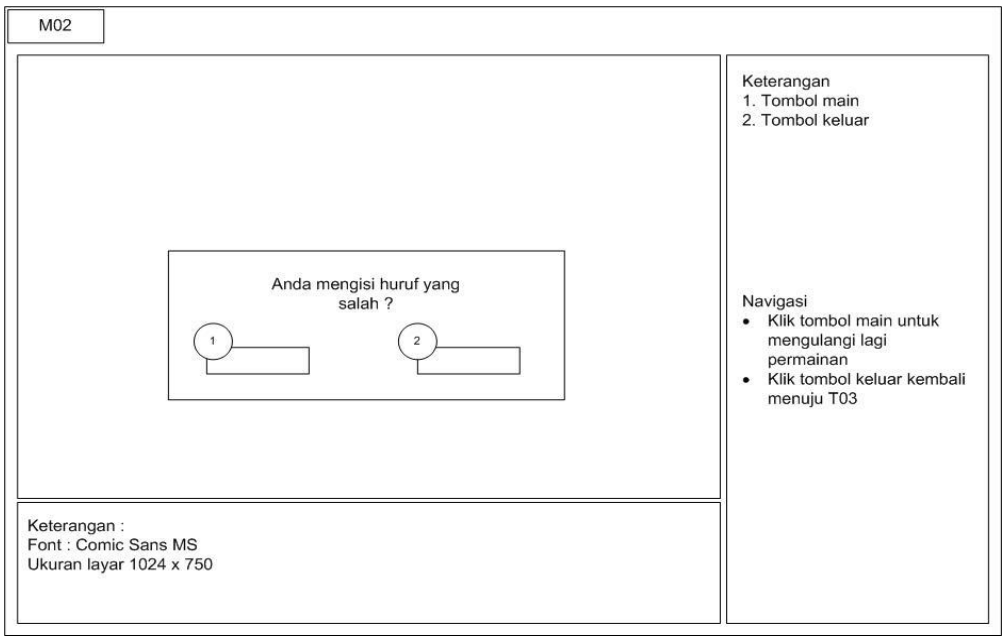
Perancangan antarmuka pesan keluar *game* lihat gambar 3.52.



Gambar 3. 52 Antarmuka Pesan Keluar

3.2.3.21. Perancangan Antarmuka Pesan Konfirmasi Kesalahan

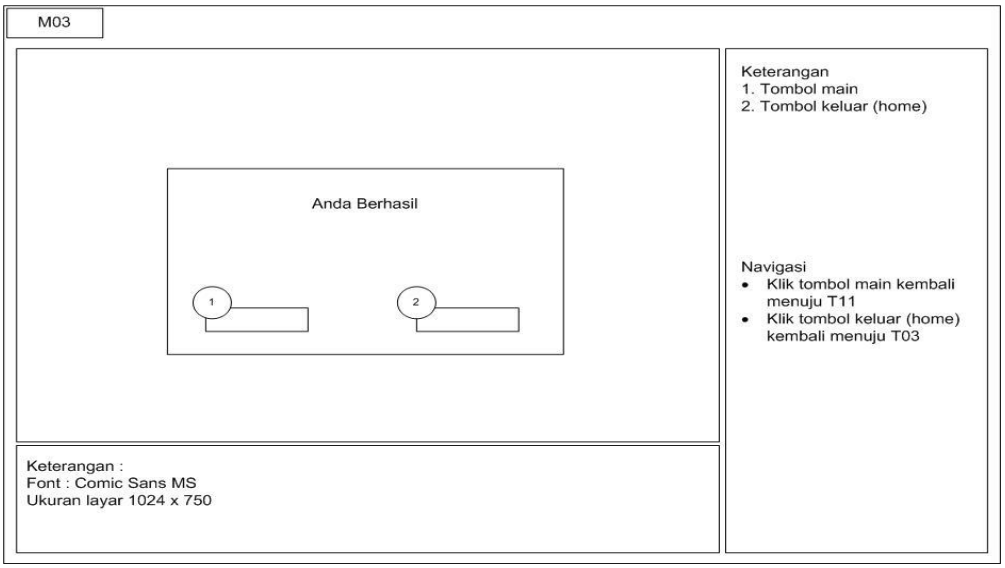
Perancangan antarmuka pesan konfirmasi kesalahan *game* lihat gambar 3.53.



Gambar 3. 53 Antarmuka Pesan konfirmasi Kesalahan

3.2.3.22. Perancangan Antarmuka Pesan Konfirmasi Berhasil

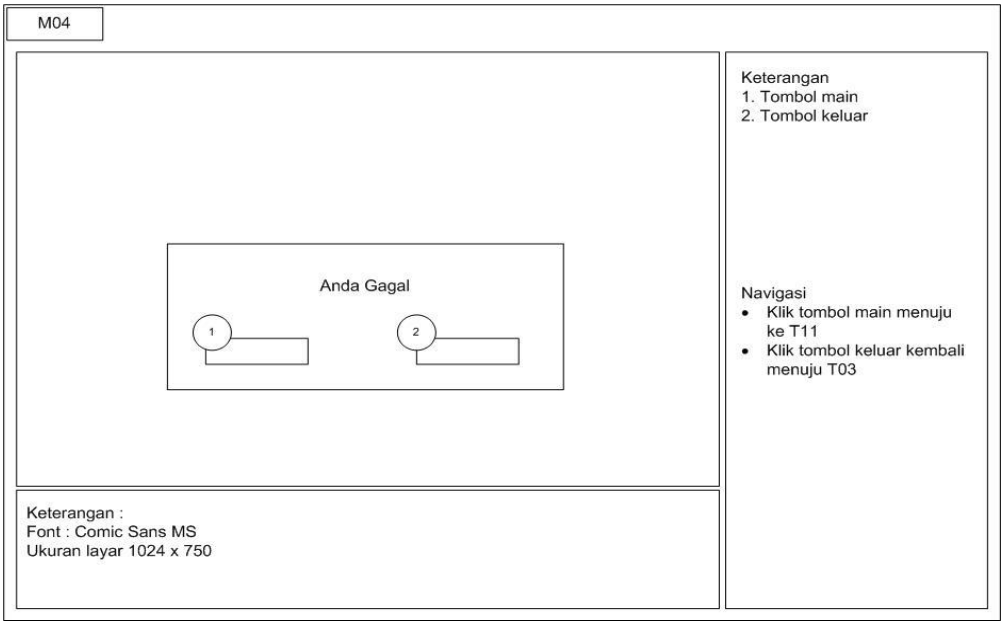
Perancangan antarmuka pesan konfirmasi berhasil *game* periksa kendaraan lihat gambar 3.54.



Gambar 3. 54 Antarmuka Pesan Konfirmasi Berhasil

3.2.3.23. Perancangan Antarmuka Pesan Konfirmasi Gagal

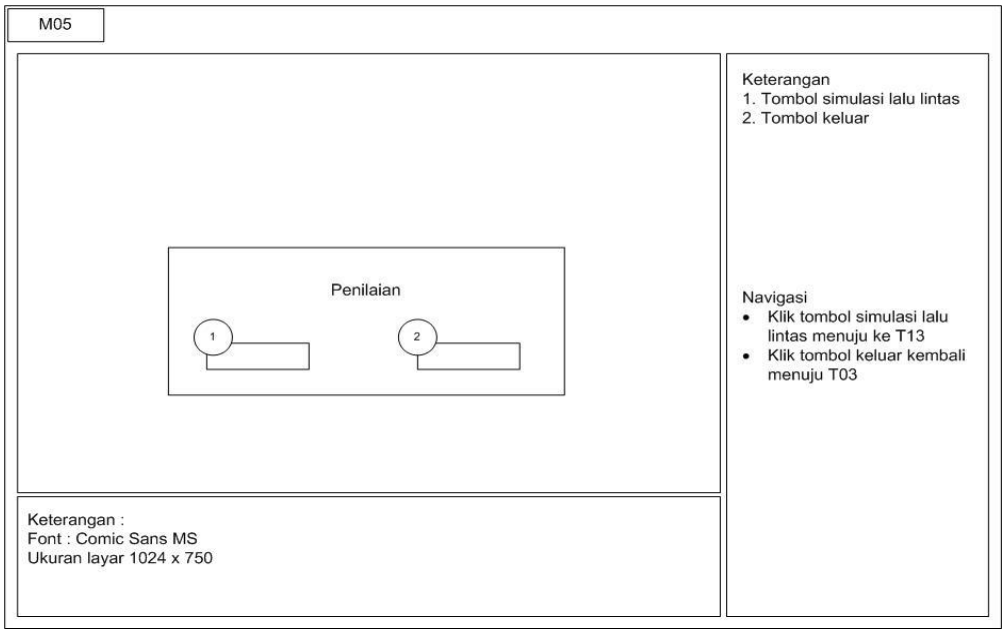
Perancangan antarmuka pesan konfirmasi berhasil *game* periksa kendaraan lihat gambar 3.55.



Gambar 3. 55 Antarmuka Pesan Gagal

3.2.3.24. Perancangan Antarmuka Pesan Konfirmasi Penilaian

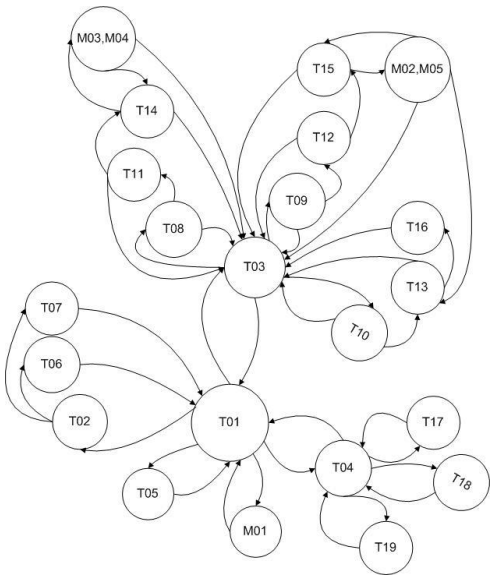
Perancangan antarmuka pesan konfirmasi berhasil *game* tebak arti lalu lintas lihat gambar 3.56.



Gambar 3. 56 Antarmuka Pesan Konfirmasi Penilaian

3.2.4. Jaringan Semantik

Jaringan semantik menggambarkan hubungan dari masing-masing tampilan yang bisa diakses pengguna. Jaringan semantik dapat dilihat pada gambar 3.57.



Gambar 3. 57 Jaringan Simantik

Berikut ini adalah yang berisi keterangan untuk jaringan semantik. Lihat tabel 3.10.

Tabel 3. 12 Keterangan Jaringan Simantik

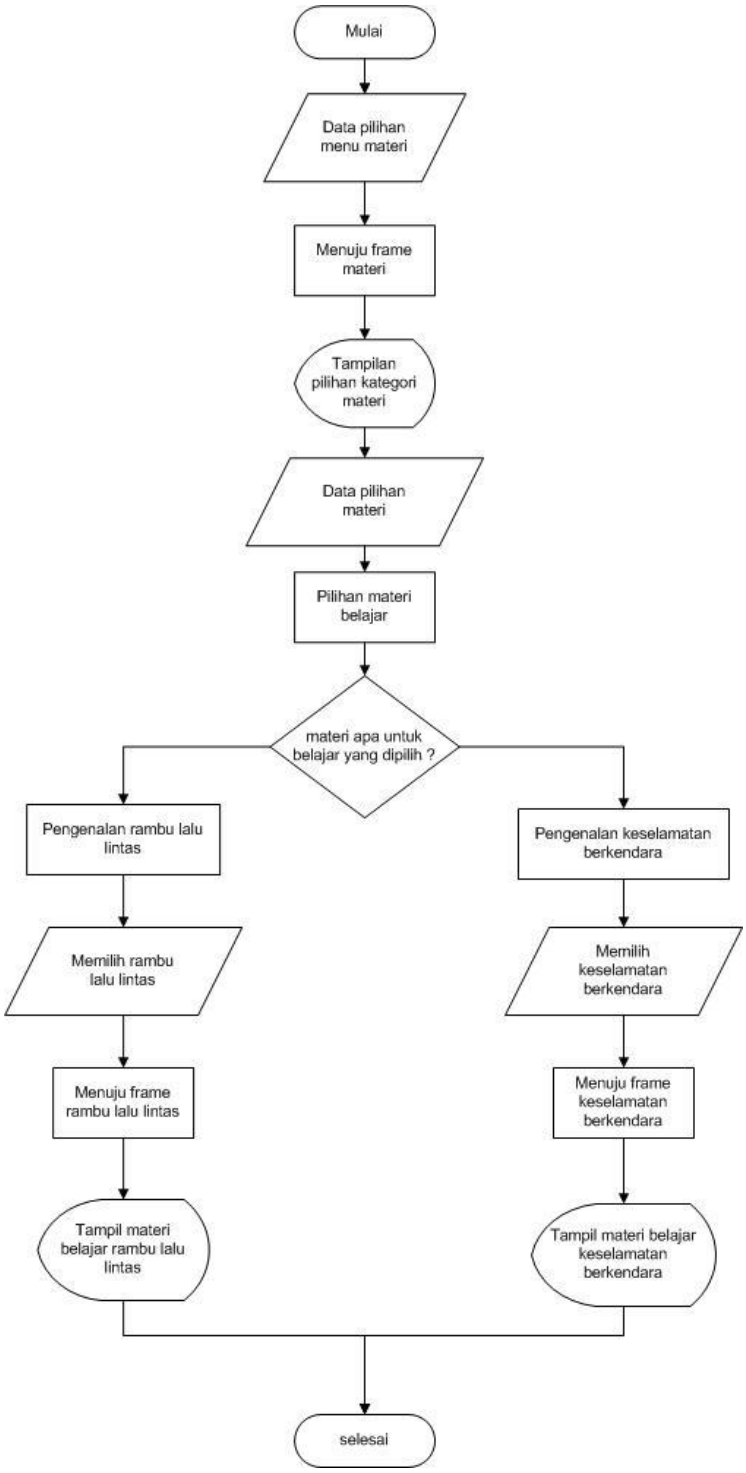
No	Kode	Keterangan
1	T01	Antarmuka Menu Utama
2	T02	Antarmuka Menu Kategori Materi
3	T03	Antarmuka Menu Bermain
4	T04	Antarmuka Menu Petunjuk *Game*
5	T05	Antarmuka Menu Pembuat
6	T06	Antarmuka Menu Menu Materi Rambu
7	T07	Antarmuka Menu Materi Keselamatan Berkendara
8	T08	Antarmuka Misi *Game* Periksa Kendaraan
9	T09	Antarmuka Misi *Game* Tebak Arti Rambu
10	T10	Antarmuka Misi Game Simulasi Lalu Lintas
11	T11	Antarmuka *Storyline Game* Periksa Kendaraan
12	T12	Antarmuka *Storyline Game* Tebak Arti Rambu
13	T13	Antarmuka *Storyline Game* Simulasi Lalu Lintas
14	T14	Antarmuka *Game* Periksa Kendaraan
15	T15	Antarmuka *Game* Tebak Arti Rambu
16	T16	Antarmuka *Game* Simulasi Lalu Lintas
17	T17	Antarmuka Petunjuk *Game* Periksa Kendaraan
18	T18	Antarmuka Petunjuk *Game* Tebak Arti Rambu
19	T19	Antarmuka Petunjuk *Game* Simulasi Lalu Lintas
20	M01	Perancangan Pesan Konfirmasi Keluar
21	M02	Perancangan Pesan Konfirmasi Kesalahan
22	M03	Perancangan Pesan Konfirmasi Berhasil
23	M04	Perancangan Pesan Konfirmasi Gagal
24	M05	Perancangan Pesan Konfirmasi Penilaian

3.2.5. Perancangan Prosedural

Perancangan prosedural berisi perancangan perilaku dan kinerja prosedur atau fungsi di dalam sistem. Perancangan prosedural untuk *game* ini sebagai berikut.

3.2.5.1. Prosedural Materi Belajar

Perancangan prosedural memilih materi belajar pada *game safety riding* dapat dilihat pada gambar 3.58.



Gambar 3. 58 FlowChart Materi Belajar

3.2.5.2. Prosedural Bermain

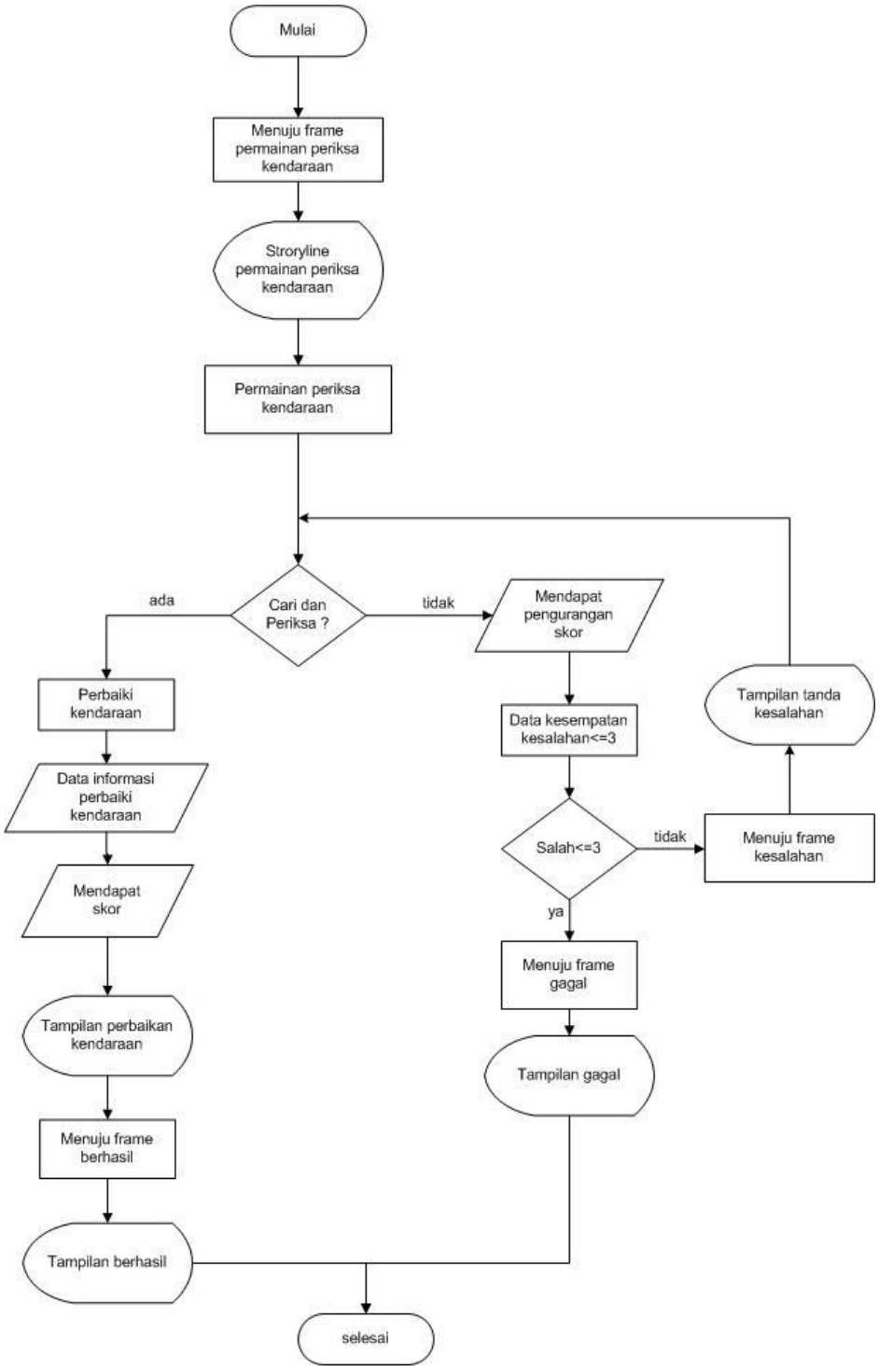
Perancangan prosedural memilih bermain pada *game safety riding* dapat dilihat pada gambar 3.59.



Gambar 3. 59 FlowChart Bermain

3.2.5.3. Prosedural *Game* Periksa Kendaraan

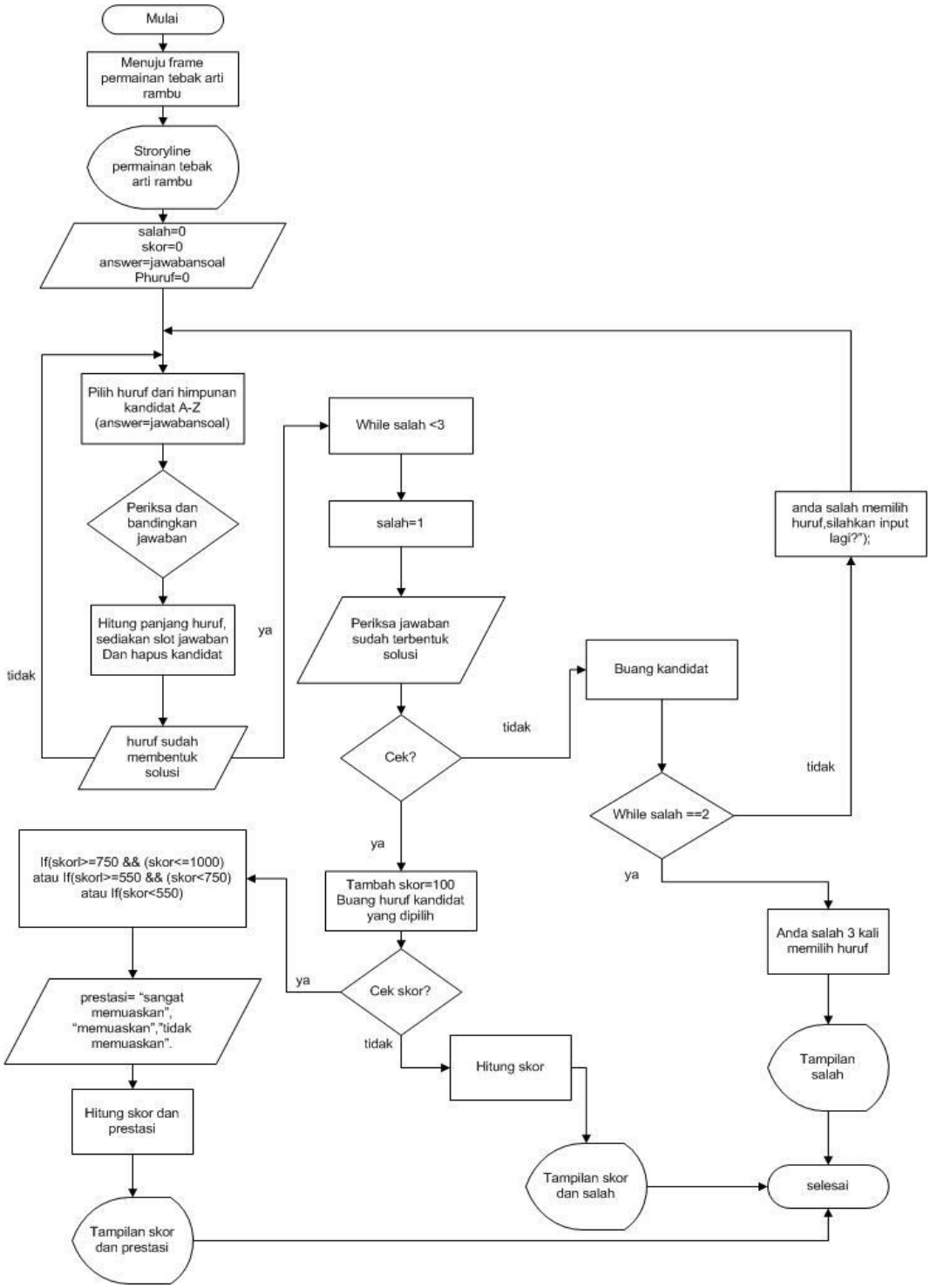
Perancangan prosedural memilih permainan periksa kendaraan pada *game safety riding* dapat dilihat pada gambar 3.60.



Gambar 3. 60 FlowChart *Game* Periksa Kendaraan

3.2.5.4. Prosedural *Game* Tebak Arti Rambu

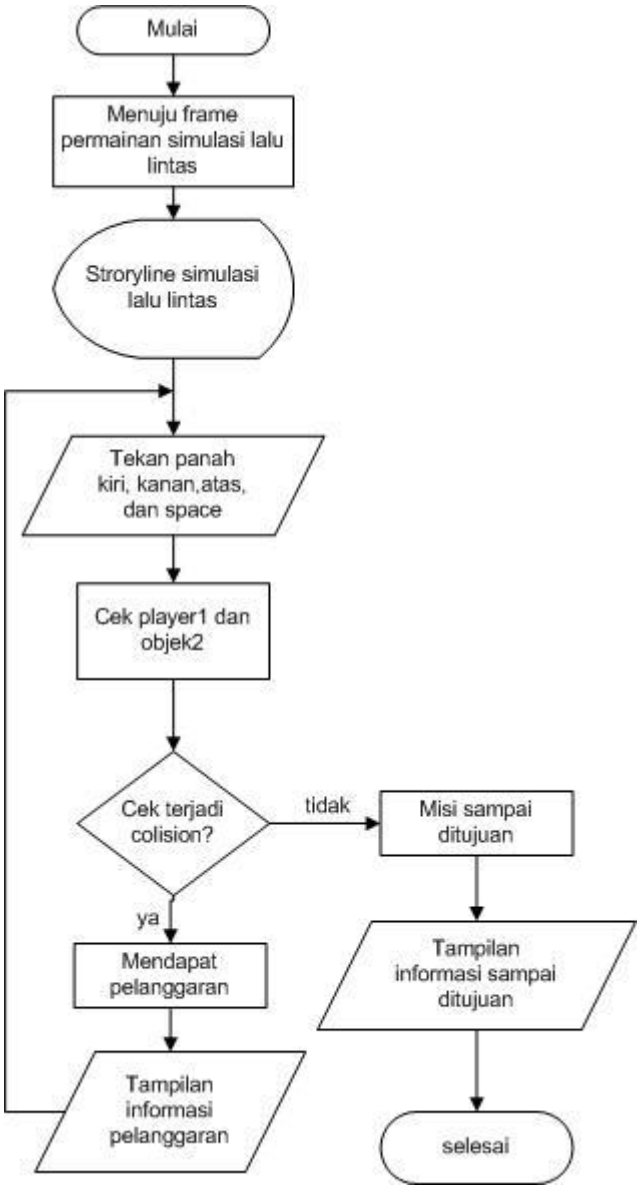
Perancangan prosedural memilih permainan tebak arti rambu pada *game safety riding* dapat dilihat pada gambar 3.61.



Gambar 3. 61 FlowChart Tebak arti Rambu

3.2.5.5. Prosedural *Game* Simulasi Lalu Lintas

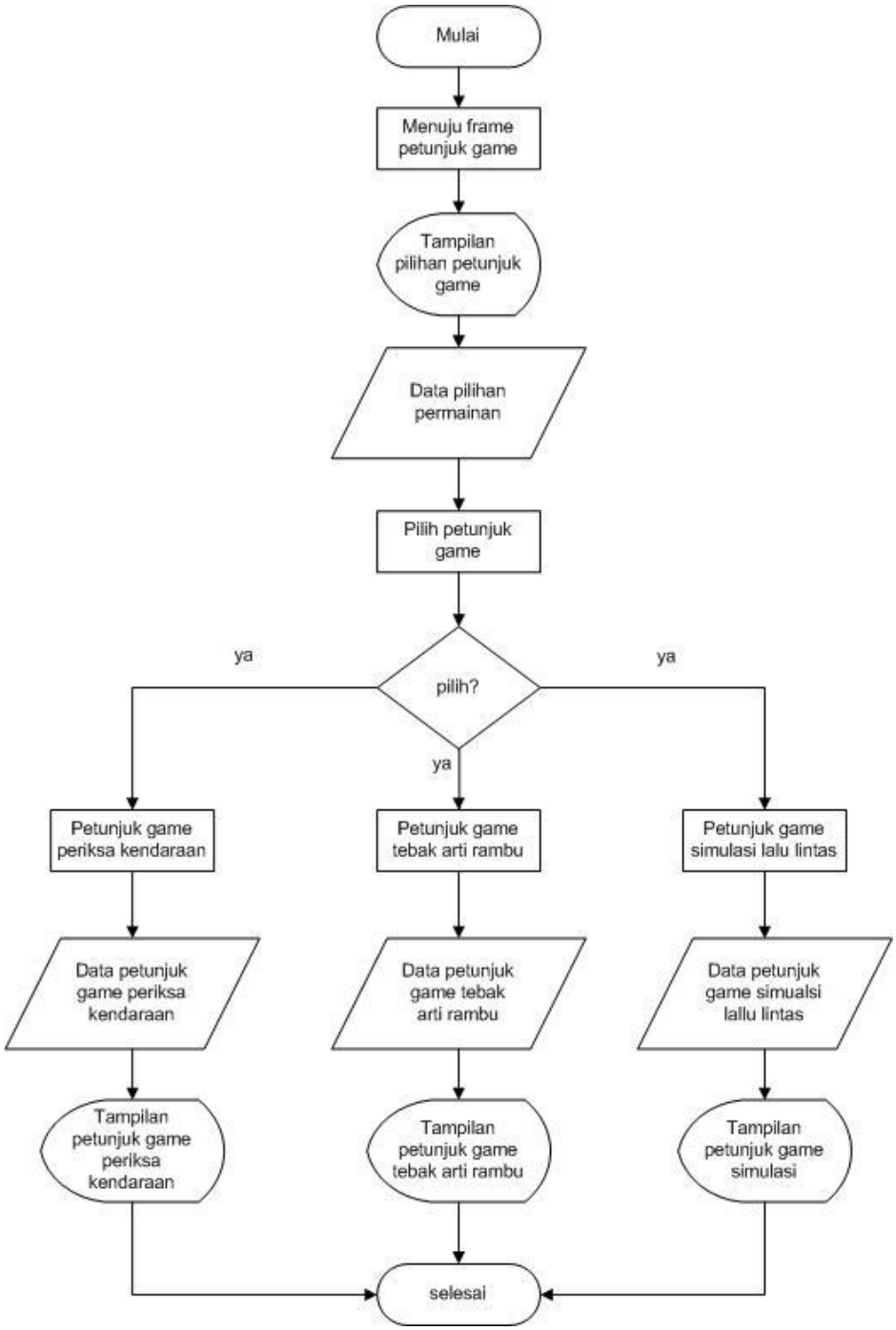
Perancangan prosedural memilih permainan simulasi lalu lintas pada *game safety riding* dapat dilihat pada gambar 3.62.



Gambar 3. 62 FlowChart Simulasi Lalu Lintas

3.2.5.6. Prosedural Petunjuk *Game*

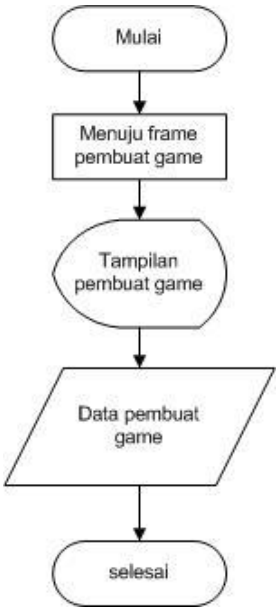
Perancangan prosedural memilih petunjuk *game* pada *game safety riding* dapat dilihat pada gambar 3.63.



Gambar 3. 63 FlowChart Petunjuk *Game*

3.2.5.7. Prosedural Pembuat *Game*

Perancangan prosedural memilih pembuat pada *game safety riding* dapat dilihat pada gambar 3.64.



Gambar 3. 64 FlowChart Pembuat *Game*