BAB 3

ANALISIS DAN PERANCANGAN

Pada tahap analisis dan perancangan diperlukan tahapan sistematis agar supaya aplikasi yang di buat sesuai dengan kegunaan dan tujuannya. Tahap pertama dari analisis adalah melakukan analisis sistem, mulai dari analisis pengguna, analisis fungsional dan analisis non-fungsional. Sedangkan untuk tahap perancangan di mulai dengan melakukan perancangan sistem dan perancangan antarmuka yang akan digunakan untuk diterapkan pada aplikasi.

3.1 Analisis Sistem

Analisis merupakan suatu proses penguraian konsep yang utuh ke dalam bagian–bagian yang lebih sederhana dengan maksud untuk mengetahui, mengidentifikasi dan mengevaluasi masalah-masalah yang ada, hambatan-hambatan, kemungkinan-kemungkinan dari setiap solusi dan kebutuhan dari sistem atau aplikasi yang akan di buat. Pada sub bab ini akan membahas mengenai sistem atau prosedur yang sedang berjalan pada aplikasi *Game* edukasi IPA dan Matematika untuk siswa SMP.

3.1.1 Analisis Masalah

Berdasarkan analisis sistem yang dilakukan dengan cara mengamati dan melakukan evaluasi dalam kegiatan pembelajaran yang dilakukan selama ini adalah sebagai berikut :

1. Berbedanya tingkat pemahaman masing-masing siswa terhadap materi yang disampaikan oleh pengajar.
2. Kurangnya peralatan dan objek yang bisa digunakan untuk melakukan penelitian.
3. IPA akan mudah dipelajari jika dibarengi dengan praktek ketika menjelaskan teori sehingga diperlukan aplikasi *game* agar pembelajaran terlihat menarik.

3.1.2 Pengenalan *Game* Petualangan Mitfas

Game Petualangan Mitfas adalah *game* yang menerapkan unsur edukasi, unsur edukasi yang disampaikan berupa pembelajaran mengenai matematika dan fisika.

1. *Gameplay Game* Petualangan Mitfas

Berikut *Gameplay* untuk *game* petualangan mitfas :

a. Permainan ini tentang seorang anak SMP yang harus mengasah kemampuan akademisnya, khususnya di bidang Matematika dan Fisika.
b. Anak yang akan diperankan oleh pemain *gamer* (dalam *game* ini pemain berperan sebagai miftas) akan pergi menjelajahi berbagai tempat untuk menjalani berbagai macam tantangan yang akan menguji kepintarannya dalam bidang matematika dan fisika.
c. Apabila dia berhasil menyelesaikan semua tantangan tersebut, maka dia dapat menunjukkan pada ayahnya kalau dia bukan anak bodoh walau sempat mendapatkan nilai rapor yang kurang memuaskan.

d. *Game* ini tersusun atas 5 level dan 1 nilai level, masing-masing level mempunyai keunikan *game* tersendiri.

e. Level-level tersebut terdiri 2 level bertema matematika dan 3 level bertema fisika selain itu disediakan 1 level rapor untuk melihat nilai dan ending dari *game* ini.

f. Uniknya lagi *game* ini memiliki *multiple ending*. Jika rata-rata nilai yang dihasilkan kurang dari 70 maka pemain akan mengalami *sad ending.*

g. Sedangkan jika rata-rata nilai yang dihasilkan lebih dari atau sama dengan 70 maka pemain akan mengalami *happy ending*.

2. Tampilan *Game Petualangan Mitfas*



Gambar 3. 1 Tampilan Menu Utama *Game* Petualangan Mitfas [4]



Gambar 3. 2 Tampilan Permainan *Game* Petualangan Mitfas [4]

Berdasarkan pengamatan ada catatan dari *game* petualangan miftas ini :

1. Pada *game* petualangan miftas sudah memiliki unsur edukasi yang cukup bagus.
2. Namun ada bererapa kekurangan dalam *game* petualangan miftas. Seperti dalam setiap *game* hanya ada sat tingkat kesulitan dan tidak adanya penjelasan materi tentang soal yang ditampilakan.

3.1.3 Pengenalan *Game* edukasi IPA dan Matematika untuk siswa SMP

Game yang akan di buat adalah *Game* edukasi IPA dan Matematika untuk siswa SMP yang mempunyai genre *game* casual. *Game* ini dibangun dengan menggunakan kemajuan teknologi untuk membantu siswa untuk memahami mata pelajaran yang di maksud. *Game* ini berbasis 2D (2 dimensi). *Game* ini juga menerapkan sistem *single player*.

Berikut ini adalah batasan yang terdapat pada *game* ini :

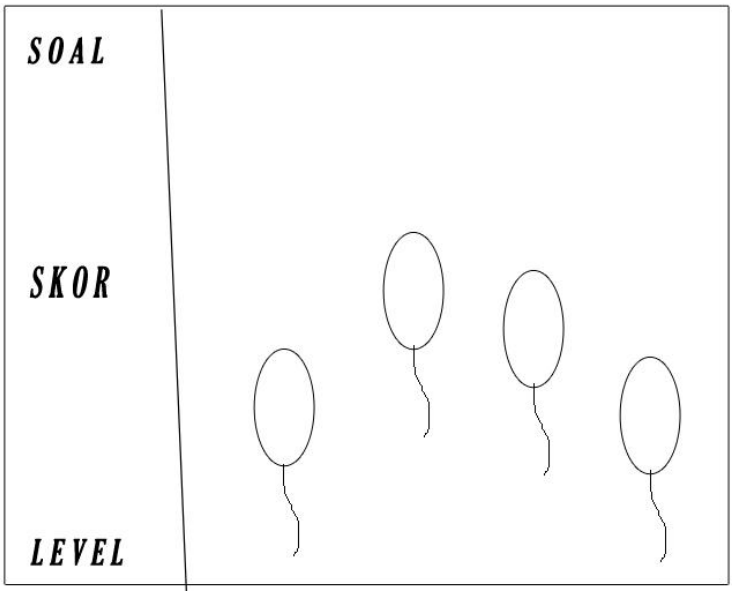
1. Sistem *Single Player*.
2. Grafik 2 dimensi.
3. Mengangkat Materi Pelajaran IPA dan Matematika sebagai bahan *game*.
4. Aplikasi *Game* bersifat bermain sambil belajar.

Dalam pembangunan aplikasi *game*, ada beberapa hal yang perlu diperhatikan untuk membuat sebuah *game*, diantaranya yaitu pembuatan *storyboard* dan *gameplay*. Berikut masing-masing penjelasan untuk *storyboard* dan *gameplay* dari *Game* edukasi IPA dan Matematika untuk siswa SMP :

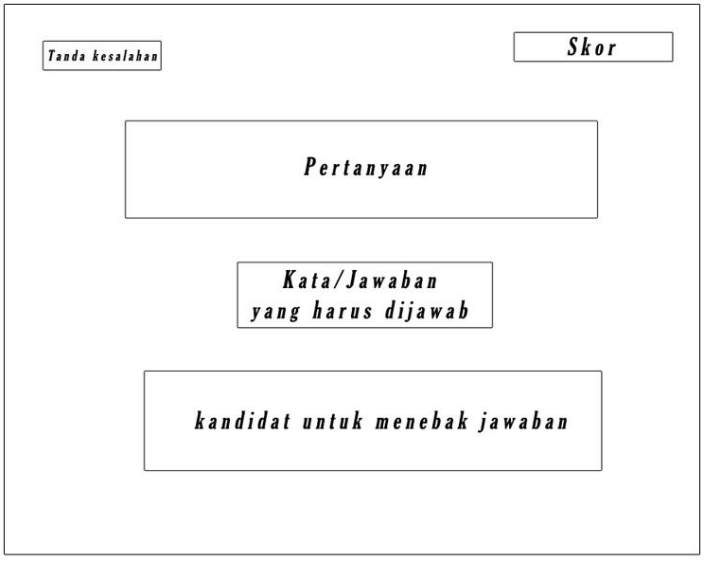
A. Storyboard

Storyboard adalah sketsa gambar yang disusun berurutan sesuai dengan naskah, *storyboard* dapat menyampaikan ide cerita kepada orang lain dengan lebih mudah, karena dapat menggiring khayalan seseorang mengikuti gambar-gambar yang tersaji, sehingga menghasilkan persepsi yang sama pada ide cerita yang telah disusun.

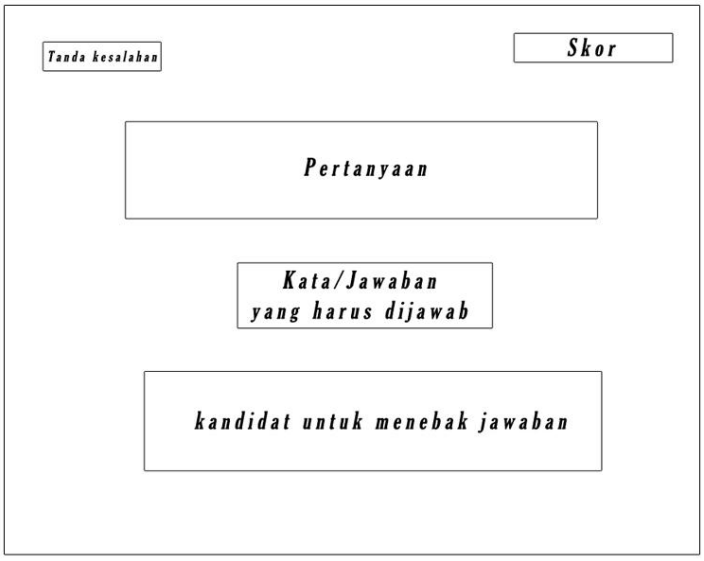
Berikut merupakan *storyboard* dari Aplikasi *Game* edukasi IPA dan Matematika untuk siswa SMP :



Gambar 3. 3 *Storyboard Game* Matematika



Gambar 3. 4 *Storyboard Game* Biologi



Gambar 3. 5 *Storyboard Game* Fisika

B. Gameplay

Untuk masing-masing *game* memiliki *gameplay* yang berbeda. Berikut penjelasan tentang *gameplay* dari masing-masing *Game* edukasi IPA dan Matematika untuk siswa SMP :

1. *Game* Matematika

a) Pemain akan menjawab pertanyaan yang ditampilkan didalam permainan.

b) Jawaban dari pertanyaan tadi dimasukkan dalam sebuah balon dan disertai pula balon yang yang lain yang mempunyai jawaban yang salah.

c) Setiap pertanyaan yang benar akan mendapat poin seratus (100) dan jawaban salah akan dikurangi lima puluh (50). Sedangkan untuk pemain yang masih mempunyai poin nol (0) ketika salah dalam menjawab poin tidak dikurangi dan tetap nol(0).

d) Setiap babak akan memuat 10 pertanyaan.

e) Soal akan diacak, sehingga kemungkinan kecil untuk mendapat urutan soal yang sama.

f) Untuk *game* matematika ini terdapat dua pilihan, yaitu *game* bilangan dan *game* bangun datar dan ruang.

g) Untuk *game* bilangan materinya diambil dari bab bilangan bulat, aljabar dan aritmetika, persamaan dan pertidaksamaan linear dengan satu variabel serta perbandingan.

h) Sedangkan pilihan yang kedua adalah *game* bangun datar dan ruang isi materinya terdiri dari bab bangun datar serta bangun ruang.

i) Setiap pilihan mempunyai 3 level yang mempunyai tingkat kesulitan berbeda dari paling mudah sampai paling sulit. Kesulitannya berupa kecepatan gerak balon jawaban dan soal yang ditampilkan akan semakin besar nilainya.

2. *Game* Biologi

a) Pemain akan menjawab pertanyaan yang ditampilkan didalam permainan.

b) Untuk menjawab pertanyaan, pemain harus menebak huruf apa saja yang menjadi bagian dari jawaban dari pertanyaan.

c) Setiap pertanyaan yang benar akan mendapat poin seratus (100) dan jawaban salah akan dikurangi lima puluh (50). Sedangkan untuk pemain yang masih mempunyai poin nol (0) ketika salah dalam menjawab poin tidak dikurangi dan tetap nol(0).

d) Setiap babak akan memuat 5 pertanyaan.

e) Soal akan diacak, sehingga kemungkinan kecil untuk mendapat urutan soal yang sama.

f) *Game* biologi terbagi dua pilihan, yaitu *game* sistem kehidupan dan *game* ekositem.

g) Untuk *game* sistem kehidupan materinya diambil dari bab sistem dalam kehidupan manusia dan sistem dalam kehidupan tumbuhan.

h) Sedangkan untuk *game* ekosistem materinya diambil dari dari bab ekosistem dan keanekaragaman makhluk hidup.

i) Untuk setiap *game* memiliki 2 level.

j) Untuk *level* pertama, kesempatan salah adalah 3 kali kali dengan waktu 30 detik.

k) Untuk *level* kedua, kesempatan salah adalah 2 dengan waktu 20 detik.

3. *Game* Fisika

a) Pemain akan menjawab pernyataan yang ditampilkan didalam permainan.

b) Pemain akan memilih, apakah pernyataan yang benar atau salah.

c) Setiap pernyataan yang cocok jawabannya akan mendapat poin seratus (100) dan jawaban salah akan dikurangi lima puluh (50). Sedangkan untuk pemain yang masih mempunyai poin nol (0) ketika salah dalam menjawab poin tidak dikurangi dan tetap nol(0).

d) *Game* ini memuat 10 pernyataan.

e) Soal akan diacak, sehingga kemungkinan kecil untuk mendapat urutan soal yang sama.

f) Untuk *game* fisika terbagi dua pilihan yaitu *game* gerak lurus dan *game* gaya.

g) Untuk *game* gerak lurus materinya diambil dari bahasan bab gerak lurus.

h) Sedangkan untuk *game* gaya materinya diambil dari bab gaya, hukum Newton serta Energi dan Usaha.

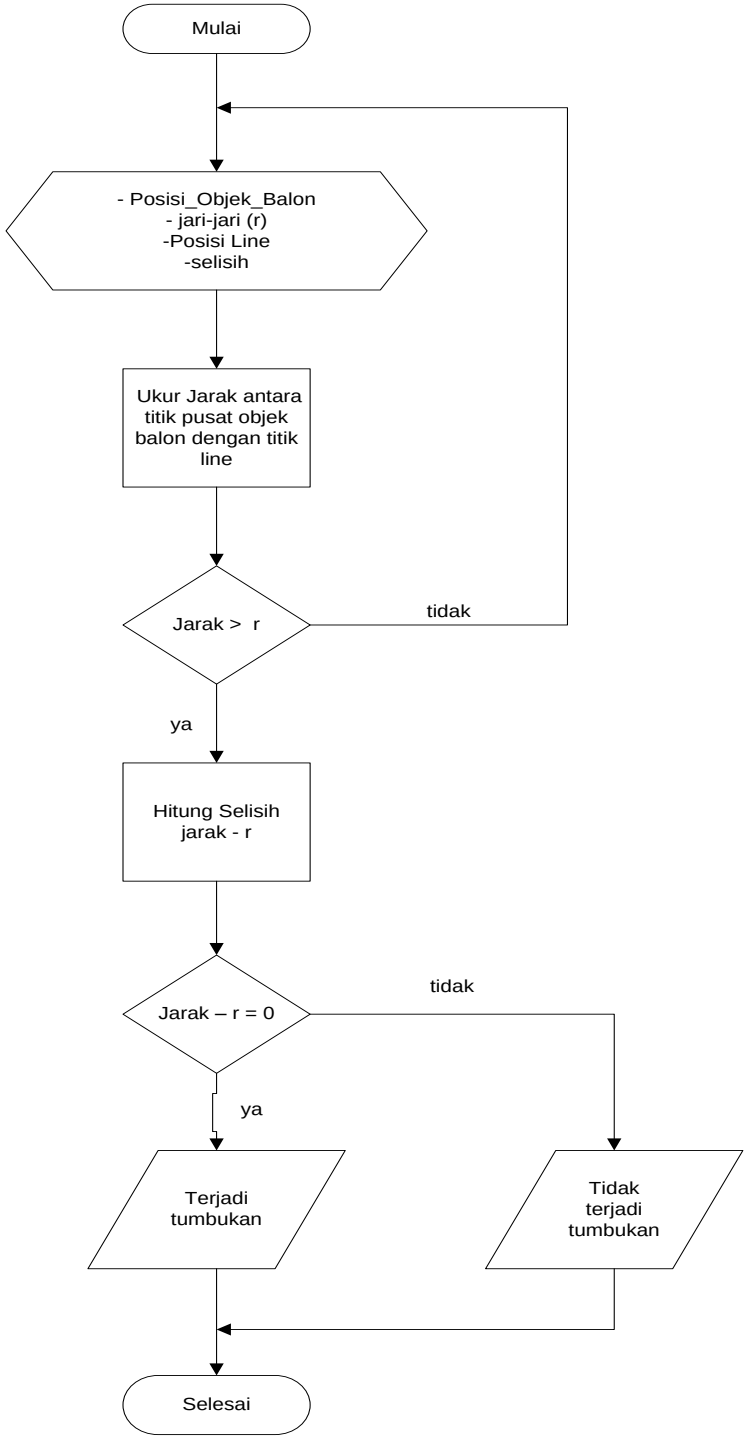
i) Untuk setiap *game* memiliki 2 level.

j) Untuk *level* pertama, kesempatan salah adalah 3 kali kali dengan waktu 30 detik.

k) Untuk *level* kedua, kesempatan salah adalah 2 dengan waktu 20 detik.

3.1.4 Analisis *Collision Detection*

Penggunaan teknik *Collision Detection* pada *Game* edukasi IPA dan Matematika untuk siswa SMP diterapkan pada *game* matematika, yang digunakan pada objek balon dan garis. Dengan metode ini objek balon akan meletus ketika bersentuhan dengan garis. Algoritma dari teknik *collision detection* dapat dijelaskan dalam flowchart berikut ini:



Gambar 3. 6 Flowchart algoritma *collision detection game* matematika

Diketahui :

r = 50

titik pusat awal objek balon = (0,650)

titik garis = (0,0)

naik = 100

nilai jarak awal = balon._y – garis._y

= 650 – 0 = 650

Posisi tumbukan = garis._y + r

= 0 + 50 = 50

Langkah – langkah teknik *collision detection* sebagai berikut :

1. Cek apakah jarak lebih besar dari tumbukan, jika tidak maka sudah pasti tidak akan terjadi tumbukan.
2. Jika ya, berikan nilai baru untuk jarak yang diambil dari balon._y dikurangi variabel naik
3. Cek apakah jarak sudah sama dengan tumbukan, jika ya maka terjadi tumbukan.
4. Jika tidak kembali ke langkah ke dua, ulangi sampai jarak lebih kecil sama dengan tumbukan.

Berdasarkan algoritma diatas, diketahui bahwa tumbukan akan terjadi ketika selisih jarak dengan jari-jari(r) adalah 0. Jika hasilnya bukan 0 maka dipastikan tumbukan tidak akan terjadi, dengan kemungkinan jika jarak masih lebih besar dari

jari-jari(r) maka algoritma akan diulang kembali dan kemungkinan tumbukan masih bias terjadi sedangkan jika jarak lebih kecil dari jari-jari(r) maka tumbukan tidak akan pernah terjadi.

3.1.5 Analisis Algoritma *Greedy*

Untuk *game* biologi Algoritma yang digunakan untuk menyelesaikan *game* adalah algoritma *greedy*. Berikut pemaparan jelas tentang algoritma *greedy* yang digunakan untuk menyelesaikan *game* biologi :

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,X,Y,Z}

2. Himpunan Solusi = S

   Himpunan Solusi (S) adalah jawaban dari pertanyaan yang ditampilkan.

3. Fungsi Seleksi

   Mengambil sebuah kandidat dari Himpunan Kandidat (C).

4. Fungsi Kelayakan

   Memeriksa apakah ada berapa huruf berdasarkan kandidat yang dipilih ada dalam barisan huruf dari jawaban. Dilakukan dengan cara mencocokkan pertanyaan yang ditampilkan dengan pertanyaan dari daftar soal, jika sudah ketemu bandingkan kandidat dengan jawaban tersebut. Jika ada tampilkan semua kandidat yang ada dalam jawaban, jika tidak buang kandidat tersebut dan jangan dipilih lagi.

5. Fungsi Objektif

   Batas maksimal kesalahan dalam memilih kandidat adalah tiga(3) kali.

Algoritma *greedy* dapat kita tuliskan sebagai berikut :

1. Inisialisasi S dengan kosong.
2. Pilih sebuah kandidat dari C dengan fungsi seleksi.
3. Kurangi C dengan kandidat yang telah terpilih di atas.
4. Periksa apakah kandidat yang dipilih tersebut bersama, 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.
5. 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 megulangi dari langkah dua (2).

Misal pertanyaan : Disebut apakah hewan pemakan daging ?

Berikut langkah untuk menyelesaikan pertanyaan di atas dengan menggunakan algoritma *greedy* :

1. Menentukan 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,X,Y,Z}

2. Menentukan himpunan solusi (S), untuk mendapatkan solusi kita harus 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.

S =

*	*	*	*	*	*	*	*	*

3. Berikan nilai nol (0) untuk vaiabel kesalahan (fungsi objektif).

   kesalahan = 0

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

   kandidat_yang_diambil = A

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

   C ={ B,C,D,E,F,G,H,I,J,K,L,M,N,O,P,Q,R,S,T,U,V,X,Y,Z}

6. Lakukan langkah no empat(4) dan lima(5), periksa apakah kandidat membentuk solusi jika ya tampilkan atau masukkan dalam S (himpunan solusi) jika tidak tambahkan satu (1) untuk variabel kesalahan.

   S =

*	A	*	*	*	*	*	*	A

   kesalahan = 0

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

   kandidat_yang_diambil = R

8. Lakukan langkah no tiga (3), kurangi C dengan kandidat yang diambil.

   C ={ B,C,D,E,F,G,H,I,J,K,L,M,N,O,P,Q, S,T,U,V,X,Y,Z}

9. Lakukan langkah no empat(4) dan lima(5), periksa apakah kandidat membentuk solusi jika ya tampilkan atau masukkan dalam S (himpunan solusi) jika tidak tambahkan satu (1) untuk variabel kesalahan.

   S =

*	A	R	*	*	*	*	R	A

   kesalahan = 0

10. Ulangi langkah no dua (2), ambil sebuah kandidat dari C.

    kandidat_yang_diambil = B

11. Lakukan langkah no tiga (3), kurangi C dengan kandidat yang diambil.

    C ={ C,D,E,F,G,H,I,J,K,L,M,N,O,P,Q,S,T,U,V,X,Y,Z}

12. Lakukan langkah no empat(4) dan lima(5), periksa apakah kandidat membentuk solusi jika ya tampilkan atau masukkan dalam S (himpunan solusi) jika tidak tambahkan satu (1) untuk variabel kesalahan.

    S =

*	A	R	*	*	*	*	R	A

    kesalahan = 1

13. Ulangi langkah no dua (2), ambil sebuah kandidat dari C.

    kandidat_yang_diambil = N

14. Lakukan langkah no tiga (3), kurangi C dengan kandidat yang diambil.

    C ={ C,D,E,F,G,H,I,J,K,L,M, O,P,Q, S,T,U,V,X,Y,Z}

15. Lakukan langkah no empat(4) dan lima(5), periksa apakah kandidat membentuk solusi jika ya tampilkan atau masukkan dalam S (himpunan solusi) jika tidak tambahkan satu (1) untuk variabel kesalahan.

    S =

*	A	R	N	*	*	*	R	A

    kesalahan = 1

16. Ulangi langkah no dua (2), ambil sebuah kandidat dari C.

    kandidat_yang_diambil = I

17. Lakukan langkah no tiga (3), kurangi C dengan kandidat yang diambil.

    C ={ C,D,E,F,G,H, J,K,L,M, O,P,Q, S,T,U,V,X,Y,Z}

18. Lakukan langkah no empat(4) dan lima(5), periksa apakah kandidat membentuk solusi jika ya tampilkan atau masukkan dalam S (himpunan solusi) jika tidak tambahkan satu (1) untuk variabel kesalahan.

    S =

*	A	R	N	I	*	*	R	A

    kesalahan = 1

19. Ulangi langkah no dua (2), ambil sebuah kandidat dari C.

    kandidat_yang_diambil = V

20. Lakukan langkah no tiga (3), kurangi C dengan kandidat yang diambil.

    C ={ C,D,E,F,G,H,I,J,K,L,M, O,P,Q, S,T,U,X,Y,Z}

21. Lakukan langkah no empat(4) dan lima(5), periksa apakah kandidat membentuk solusi jika ya tampilkan atau masukkan dalam S (himpunan solusi) jika tidak tambahkan satu (1) untuk variabel kesalahan.

    S =

*	A	R	N	I	V	*	R	A

    kesalahan = 1

22. Ulangi langkah no dua (2), ambil sebuah kandidat dari C.

    kandidat_yang_diambil = O

23. Lakukan langkah no tiga (3), kurangi C dengan kandidat yang diambil.

    C ={ C,D,E,F,G,H,I,J,K,L,M, P,Q, S,T,U,X,Y,Z}

24. Lakukan langkah no empat(4) dan lima(5), periksa apakah kandidat membentuk solusi jika ya tampilkan atau masukkan dalam S (himpunan solusi) jika tidak tambahkan satu (1) untuk variabel kesalahan.

    S =

*	A	R	N	I	V	O	R	A

    kesalahan = 1

25. Ulangi langkah no dua (2), ambil sebuah kandidat dari C.

    kandidat_yang_diambil = D

26. Lakukan langkah no tiga (3), kurangi C dengan kandidat yang diambil.

    C ={ C, E,F,G,H,I,J,K,L,M, O,P,Q, S,T,U, X,Y,Z}

27. Lakukan langkah no empat(4) dan lima(5), periksa apakah kandidat membentuk solusi jika ya tampilkan atau masukkan dalam S (himpunan solusi) jika tidak tambahkan satu (1) untuk variabel kesalahan.

    S =

*	A	R	N	I	V	O	R	A

    kesalahan = 2

28. Ulangi langkah no dua (2), ambil sebuah kandidat dari C.

    kandidat_yang_diambil = K

29. Lakukan langkah no tiga (3), kurangi C dengan kandidat yang diambil.

    C ={ C,D,E,F,G,H,I,J, L,M, O,P,Q, S,T,U, X,Y,Z}

30. Lakukan langkah no empat(4) dan lima(5), periksa apakah kandidat membentuk solusi jika ya tampilkan atau masukkan dalam S (himpunan solusi) jika tidak tambahkan satu (1) untuk variabel kesalahan.

    S =

K	A	R	N	I	V	O	R	A

    kesalahan = 2

3.1.6 Analisis Soal

Analisis soal diperlukan untuk mennetukan apakah soal layak untuk user yang memang menjadi target dalam pembuatan aplikasi ini. Berikut analisis soal dari masing-masing *game* :

1. *Game* Matematika

a. Setiap level terdiri dari 10 soal yang ditampilkan.

b. Soal akan diacak dari *array* yang disimpan dalam aplikasi.

c. Tingkat kesulitan soal akan bertambah ketika level naik, selain pergerakan objek yang bertambah cepat, nilai dari variabel pada soal akan meningkat pula.

2. *Game* Biologi

a. Ada lima pertanyaan yang harus dijawab ketika bermain *game* biologi.

b. Setiap satu pertanyaan diambil dari sebuah data array soal yang diambil secara acak atau random.

3. *Game* Fisika

a. Dalam *game* fisika ada 10 pernyataan yang harus dijawab oleh *user*.

b. Ada 10 data array dalam aplikasi, dimana dalam setiap satu array hanya akan ditampilkan satu pernyataan.

3.1.7 Analisis Kebutuhan Sistem

Dari analisis kebutuhan sistem yang dilakukan maka kebutuhan sistem dapat didefinisikan sebagai berikut :

1. Aplikasi ini dapat dijadikan sebagai alternatif media pembelajaran terhadap mata pelajaran IPA di SMP.
2. Aplikasi ini dapat digunakan untuk meningkatkan atau menumbuhkan minat siswa terhadap mata pelajaran IPA.

3.1.8 Analisis Kebutuhan Non-Fungsional

Analisis kebutuhan non-fungsional menggambarkan kebutuhan sistem yang menitikberatkan pada properti prilaku yang dimiliki oleh sistem, diantaranya kebutuhan perangkat keras, perangkat lunak, serta *user* sebagai bahan analisis kekurangan dan kebutuhan yang harus dipenuhi dalam sistem yang akan diterapkan.

3.1.8.1 Analisis Kebutuhan Perangkat Lunak

Berikut ini adalah spesifikasi perangkat lunak yang digunakan dalam membangun aplikasi ini adalah :

1. Sistem Operasi Windows 7.
2. Adobe Flash CS5.
3. Adobe Photoshop CS5.
4. Adobe Audition 1.5.

Adapun perangkat lunak yang dibutuhkan dalam penerapan aplikasi *game* edukasi ini adalah sebagai berikut :

1. Sistem Operasi *Windows* (*Windows XP, Vista, 7*).
2. Aplikasi *Game* edukasi IPA dan Matematika untuk siswa SMP.

3.1.8.2 Analisis Kebutuhan Perangkat Keras

Berikut ini adalah spesifikasi perangkat keras *minimum* yang mendukung aplikasi *game* edukasi ini adalah :

1. *Processor* : Dengan Kecepatan *Minimum* 2.0 GHZ
2. RAM : 1GB.

3. VGA 512 MB.
4. *Harddisk* 40 GB.
5. *Speaker* atau *Headset.*
6. Monitor.
7. *Mouse* dan *Keyboard*.

3.1.8.3 Analisis Kebutuhan *User*

Selain dibutuhkannya perangkat lunak dan perangkat keras, *user* juga sangat dibutuhkan dalam penggunakaan aplikasi *game* ini. Adapun spesifikasi *user* yang dibutuhkan:

1. User diutamakan siswa SMP kelas VII dan VIII karena materi yang ada dalam *game* diambil dari materi untuk siswa SMP kelas VII dan VIII. Namun *game* juga dapat dimainkan selain dari siswa SMP kelas VII dan VIII.
2. Memahami bagaimana mengoperasikan aplikasi *Game* edukasi IPA dan Matematika untuk siswa SMP.
3. Dapat menggunakan *mouse* dan *keyboard.*
4. Bisa membaca dan mengetahui bahasa Indonesia dengan baik.
5. Minimal mengenal ilmu matematika dasar seperti penjumlahan, pengurangan, pembagian dan perkalian.

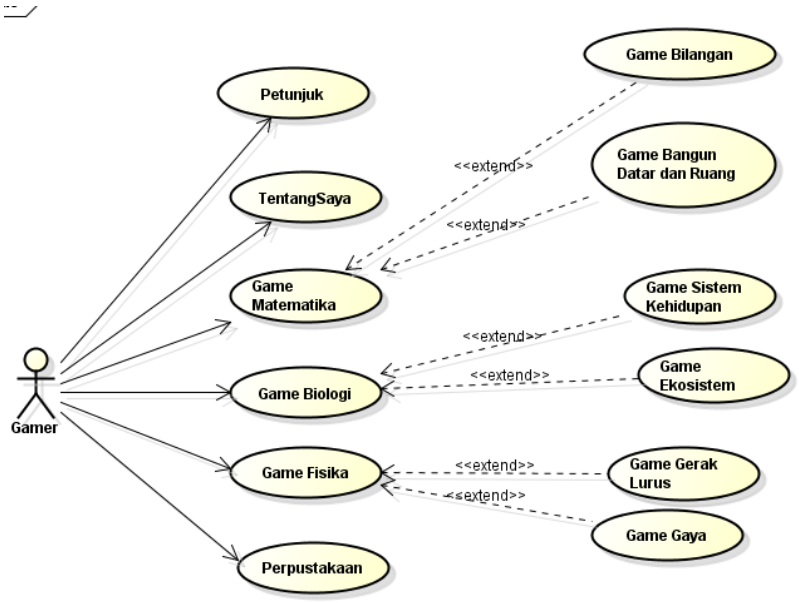
3.1.9 Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional menggambarkan proses kegiatan yang akan diterapkan dalam sebuah sistem dan menjelaskan kebutuhan yang diperlukan sistem agar sistem dapat berjalan dengan baik serta sesuai dengan kebutuhan.

Analisis yang dilakukan dimodelkan dengan menggunakan UML (*Unified Modeling Language*). Tahap-tahap pemodelan dalam analisis tersebut antara lain identifikasi aktor, *usecase diagram*, *scenario*, *sequence diagram*, dan *class diagram*.

3.1.9.1 Use Case Diagram

Use Case merupakan gambaran skenario dari interaksi antara *user* dengan sistem. Sebuah diagram *Use Case* menggambarkan hubungan antara aktor dan kegiatan yang dapat dilakukannya terhadap aplikasi.



Gambar 3. 7 Gambar *Use Case* Menu Utama

1. Definisi *Use Case*

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

Tabel 3. 1 Tabel Definisi *Use Case* menu utama

NO	Use Case	Definisi
1	*Petunjuk*	Proses untuk mengetahui cara-cara bermain *game*
2	*Tentang Saya*	Proses untuk mengetahui profile pembuat *game*
3	*Game* Matematika	Proses untuk memulai *game* matematika dan memilih materi *game* yang akan dimainkan.
4	*Game* Biologi	Proses untuk memulai *game* biologi dan memilih materi *game* yang akan dimainkan.
5	*Game* Fisika	Proses untuk memulai *game* fisika
6	Perpustakaan	Proses untuk belajar materi tentang soal-soal yang nada dalam *game*
7	*Game* Bilangan	Proses untuk memulai *game* matematika yang materinya berisi bilangan bulat, aljabar, persamamaan dan perbandingan.
8	*Game* Bangun Datar dan Ruang	Proses untuk memulai *game* matematika yang materinya berisi mengenai bangun datar dan

		bangun ruang.
9	*Game* Sistem Kehidupan	Proses untuk memulai *game* biologi yang materinya tentang Sistem kehidupan manusia dan sistem kehidupan hewan.
10	*Game* Ekosistem	Proses untuk memulai *game* biologi yang materinya tentang ekosistem.
11	*Game* Gerak Lurus	Proses untuk memulai *game* fisika yang materinya tentang gerak lurus.
12	*Game* Gaya	Proses untuk memulai *game* fisika yang materinya tentang gaya.

3.1.9.2 Skenario *Use Case*

Skenario setiap bagian pada *use case* menunjukkan proses apa yang terjadi pada setiap bagian di dalam *use case* tersebut, dimana *user* memberikan perintah pada setiap bagian dan respon apa yang diberikan oleh sistem kepada *user* setelah *user* memberikan perintah pada setiap bagian–bagian *use case*.

Tabel 3. 2 Tabel *Scenario Use Case* Pilihan *Petunjuk*

Identifikasi	
Nomor	1
Nama	*Petunjuk*
Tujuan	Mengetahui bagaiamana cara bermain

Deskripsi	Proses untuk mengetahui cara bermain
Aktor	*Gamer*
Skenario Utama	
Kondisi Awal	Pengguna berada di menu utama *Game* edukasi IPA dan Matematika untuk siswa SMP

Aksi Aktor	Reaksi Sistem
1. Memilih menu *Petunjuk*	2. Aplikasi akan merespon dengan masuk ke tampilan menu *Petunjuk*
3. Memilih *Petunjuk Game* Matematika	4. Aplikasi akan merespon dengan masuk ke tampilan *petunjuk game* Matematika
5. Memilih *Petunjuk Game* Biologi	6. Aplikasi akan merespon dengan masuk ke tampilan *petunjuk game* Biologi
7. Memilih *Petunjuk Game* Fisika	8. Aplikasi akan merespon dengan masuk ke tampilan *petunjuk game* Fisika

Kondisi Akhir	Aplikasi akan menampilkan intruksi atau *Petunjuk* tentang cara bermain dari masing-masing *game*.

Tabel 3. 3 Tabel *Scenario Use Case* Pilihan *Tentang Saya*

Identifikasi	
Nomor	2
Nama	*Tentang Saya*

Tujuan	Mengetahui tentang pembuat aplikasi	
Deskripsi	Proses untuk mengetahui pembuat aplikasi	
Aktor	*Gamer*	
Skenario Utama		
Kondisi Awal	Pengguna berada di menu utama *Game* edukasi IPA dan Matematika untuk siswa SMP	
Aksi Aktor		Reaksi Sistem
1. Memilih menu *Tentang Saya*		2. Aplikasi akan merespon dengan masuk ke tampilan *Tentang Saya*
Kondisi Akhir	Aplikasi akan menampilkan informasi mengenai *game* dan pembuat aplikasi	

Tabel 3. 4 Tabel *Scenario Use Case* Pilihan *Game* Matematika

Identifikasi	
Nomor	3
Nama	*Game* Matematika
Tujuan	Memilih *game* matematika
Deskripsi	Proses untuk memilih dan memainkan *game* matematika
Aktor	*Gamer*
Skenario Utama	
Kondisi Awal	Pengguna berada di menu utama *game* edukasi IPA dan

	Matematika untuk siswa SMP	
Aksi Aktor		Reaksi Sistem
1. Memilih *game* matematika		2. Aplikasi akan merespon dengan masuk ke menu pilihan *game* matematika
3. Memilih menu pilihan materi yang ditampilkan.		4. Aplikasi merespon dengan masuk ke *game* matematika yang dipilih oleh *user*.
Kondisi Akhir	Aplikasi akan masuk ke *game* matematika sesuai dengan materi yang dipilih.	

Tabel 3. 5 Tabel *Scenario Use Case Game* Biologi

Identifikasi		
Nomor	4	
Nama	*Game* Biologi	
Tujuan	Memilih *game* biologi	
Deskripsi	Proses untuk memilih dan memainkan *game* biologi	
Aktor	*Gamer*	
Skenario Utama		
Kondisi Awal	Pengguna berada di menu utama *game* edukasi IPA dan Matematika untuk siswa SMP	
Aksi Aktor		Reaksi Sistem

1. Memilih *game* biologi	2. Aplikasi akan merespon dengan masuk ke menu pilihan *game* biologi
3. Memilih menu pilihan materi yang ditampilkan.	4. Aplikasi merespon dengan masuk ke *game* biologi yang dipilih oleh *user*.
Kondisi Akhir	Aplikasi akan masuk ke *game* biologi sesuai dengan materi yang dipilih.

Tabel 3. 6 Tabel *Scenario Use Case* Pilihan *Game* Fisika

Identifikasi	
Nomor	5
Nama	*Game* fisika
Tujuan	Memilih *game* fisika
Deskripsi	Proses untuk memilih dan memainkan *game* fisika
Aktor	*Gamer*
Skenario Utama	
Kondisi Awal	Pengguna berada di menu utama *game* edukasi IPA dan Matematika untuk siswa SMP
Aksi Aktor	Reaksi Sistem
1. Memilih *game* fisika	2. Aplikasi akan merespon dengan masuk ke menu pilihan materi *game* fisika
3. Memilih menu pilihan materi yang ditampilkan.	4. Aplikasi merespon dengan masuk ke *game* biologi yang dipilih oleh *user*.

Kondisi Akhir	Aplikasi akan masuk ke *game* fisika sesuai dengan materi yang dipilih.

Tabel 3. 7 Tabel *Scenario Use Case* Perpustakaan

Identifikasi		
Nomor	6	
Nama	Perpustakaan	
Tujuan	Untuk mengetahui dan memahami materi yang ditampilkan dalam soal didalam *game*.	
Deskripsi	Proses untuk belajar materi pelajaran yang dibahas dalam *game*.	
Aktor	*Gamer*	
Skenario Utama		
Kondisi Awal	Pengguna berada di menu utama *Game* edukasi IPA dan Matematika untuk siswa SMP	
Aksi Aktor		Reaksi Sistem
1. Memilih menu Perpustakaan		2. Aplikasi akan merespon dengan masuk ke tampilan Perpustakaan
Kondisi Akhir	Aplikasi akan menampilkan menu perpustakaan, untuk memilih katalog pelajaran.	

Tabel 3. 8 Tabel *Scenario Use Case Game* Bilangan

Identifikasi		
Nomor	7	
Nama	*Game* Bilangan	
Tujuan	Memilih *game* matematika yang materinya bab bilangan	
Deskripsi	Proses untuk memilih dan memainkan *game* matematika yang materinya bab bilangan	
Aktor	*Gamer*	
Skenario Utama		
Kondisi Awal	Pengguna berada di menu pilihan *game* matematika	
Aksi Aktor		Reaksi Sistem
1. Memulai *game* matematika yang membahas materi bab bilangan.		2. Aplikasi akan merespon dengan masuk ke permainan.
3. Memilih jawaban yang benar dari pertanyaan yang ditampilkan.		4. Aplikasi merespon dengan memberikan nilai 100 jika tebakan jawaban benar dan nilai akan dikurangi 50 apabila tebakan jawaban salah.
5. Jika skor memenuhi target, user memilih akan melanjutkan *game* atau keluar dari *game*.		6. Aplikasi akan merespon dengan menampilkan *level* selanjutnya jika memilih melanjutkan dan keluar dari

	game matematika jika memilih keluar.
7. Jika skor tidak memenuhi target, *user* akan keluar dari *game* atau main lagi.	8. Aplikasi akan merespon dengan keluar dari *game* matematika atau kembali bermain dengan dimulai dari level paling bawah.
Kondisi Akhir	Aplikasi akan menampilkan hasil akhir dari permainan dan menentukan apakah *user* menang atau kalah dalam permainan.

Tabel 3. 9 Tabel *Scenario Use Case Game* Bangun Datar dan Ruang

Identifikasi		
Nomor	8	
Nama	*Game* Bangun Datar dan Ruang	
Tujuan	Memilih *game* matematika yang materinya bab bangun datar dan ruang.	
Deskripsi	Proses untuk memilih dan memainkan *game* matematika yang materinya bab bangun datar dan ruang.	
Aktor	*Gamer*	
Skenario Utama		
Kondisi Awal	Pengguna berada di menu pilihan *game* matematika.	
Aksi Aktor		Reaksi Sistem
1. Memulai *game* matematika yang		2. Aplikasi akan merespon dengan masuk

membahas materi bab bilangan.	ke permainan.
3. Memilih jawaban yang benar dari pertanyaan yang ditampilkan.	4. Aplikasi merespon dengan memberikan nilai 100 jika tebakan jawaban benar dan nilai akan dikurangi 50 apabila tebakan jawaban salah.
5. Jika skor memenuhi target, user memilih akan melanjutkan *game* atau keluar dari *game.*	6. Aplikasi akan merespon dengan menampilkan level selanjutnya jika memilih melanjutkan dan keluar dari *game* matematika jika memilih keluar.
7. Jika skor tidak memenuhi target, user akan keluar dari *game* atau main lagi.	8. Aplikasi akan merespon dengan keluar dari *game* matematika atau kembali bermain dengan dimulai dari level paling bawah.
Kondisi Akhir	Aplikasi akan menampilkan hasil akhir dari permainan dan menentukan apakah *user* menang atau kalah dalam permainan.

Tabel 3. 10 Tabel *Scenario Use Case Game* Sistem Kehidupan

Identifikasi	
Nomor	9
Nama	*Game* Sistem Kehidupan

Tujuan	Memilih *game* Sistem Kehidupan
Deskripsi	Proses untuk memilih dan memainkan *game* Sistem Kehidupan
Aktor	*Gamer*
Skenario Utama	
Kondisi Awal	Pengguna berada di menu pilihan *game* biologi.

Aksi Aktor	Reaksi Sistem
1. Memulai *game* biologi yang membahas materi bab Sistem Kehidupan.	2. Aplikasi akan merespon dengan masuk ke permainan.
3. Menjawab huruf apa saja yang menjadi bagian dari jawaban dari pertanyaan yang ditampilkan	4. Aplikasi merespon dengan menampilkan pesan benar dan nilai akan bertambah 200 jika tidak ada kesalahan, 150 jika sekali salah dan 100 jika dua kali salah. Sedangkan jika jawaban salah maka tidak akan mendapatkan nilai dan aplikasi akan menampilkan pesan salah atau gagal.
5. Jika jawaban soal sebelumnya benar, maka *gamer* akan menjawab	6. Aplikasi akan merespon dengan keluar dari *game* jika user

pertanyaan selanjutnya. Jika salah maka akan keluar dari *game* Sistem Kehidupan	memilih tombol salah dan akan menampilkan pertanyaan selanjutnya jika emilih tombol selanjutnya.
Kondisi Akhir	Aplikasi akan masuk ke *game* Sistem Kehidupan

Tabel 3. 11 Tabel *Scenario Use Case Game* Ekosistem

Identifikasi	
Nomor	10
Nama	*Game* Ekosistem
Tujuan	Memilih *game* Ekosistem
Deskripsi	Proses untuk memilih dan memainkan *game* ekosistem
Aktor	*Gamer*
Skenario Utama	
Kondisi Awal	Pengguna berada di menu pilihan *game* biologi.

Aksi Aktor	Reaksi Sistem
1. Memulai *game* biologi yang membahas materi bab Sistem Kehidupan.	2. Aplikasi akan merespon dengan masuk ke permainan.
3. Menjawab huruf apa saja yang menjadi bagian dari jawaban dari	4. Aplikasi merespon dengan menampilkan pesan benar dan nilai

pertanyaan yang ditampilkan	akan bertambah 200 jika tidak ada kesalahan, 150 jika sekali salah dan 100 jika dua kali salah. Sedangkan jika jawaban salah maka tidak akan mendapatkan nilai dan aplikasi akan menampilkan pesan salah atau gagal.
5. Jika jawaban soal sebelumnya benar, maka *gamer* akan menjawab pertanyaan selanjutnya. Jika salah maka akan keluar dari Ekosistem.	6. Aplikasi akan merespon dengan keluar dari *game* jika user memilih tombol salah dan akan menampilkan pertanyaan selanjutnya jika emilih tombol selanjutnya.
Kondisi Akhir	Aplikasi akan menampilkan hasil akhir dari permainan dan menentukan apakah *user* menang atau kalah dalam permainan.

Tabel 3. 12 Tabel *Scenario Use Case Game* Gerak Lurus

Identifikasi	
Nomor	11
Nama	*Game* Gerak Lurus
Tujuan	Memilih *game* Gerak Lurus
Deskripsi	Proses untuk memilih dan memainkan *game* Gerak Lurus

Aktor	*Gamer*	
Skenario Utama		
Kondisi Awal	Pengguna berada di menu pilihan *game* fisika	
Aksi Aktor		Reaksi Sistem
1. Memulai *game* fisika yang membahas materi bab Gerak Lurus.		2. Aplikasi akan merespon dengan masuk ke permainan.
3. Menjawab huruf apa saja yang menjadi bagian dari jawaban dari pertanyaan yang ditampilkan		4. Aplikasi merespon dengan menampilkan pesan benar dan nilai akan bertambah 200 jika tidak ada kesalahan, 150 jika sekali salah dan 100 jika dua kali salah. Sedangkan jika jawaban salah maka tidak akan mendapatkan nilai dan aplikasi akan menampilkan pesan salah atau gagal.
5. Jika jawaban soal sebelumnya benar, maka *gamer* akan menjawab pertanyaan selanjutnya. Jika salah maka akan keluar dari Ekosistem.		6. Aplikasi akan merespon dengan keluar dari *game* jika user memilih tombol salah dan akan menampilkan pertanyaan selanjutnya jika emilih tombol selanjutnya.
Kondisi Akhir	Aplikasi akan menampilkan hasil akhir dari permainan dan menentukan apakah *user* menang atau kalah dalam permainan.	

Tabel 3. 13 Tabel *Scenario Use Case Game* Gerak Lurus

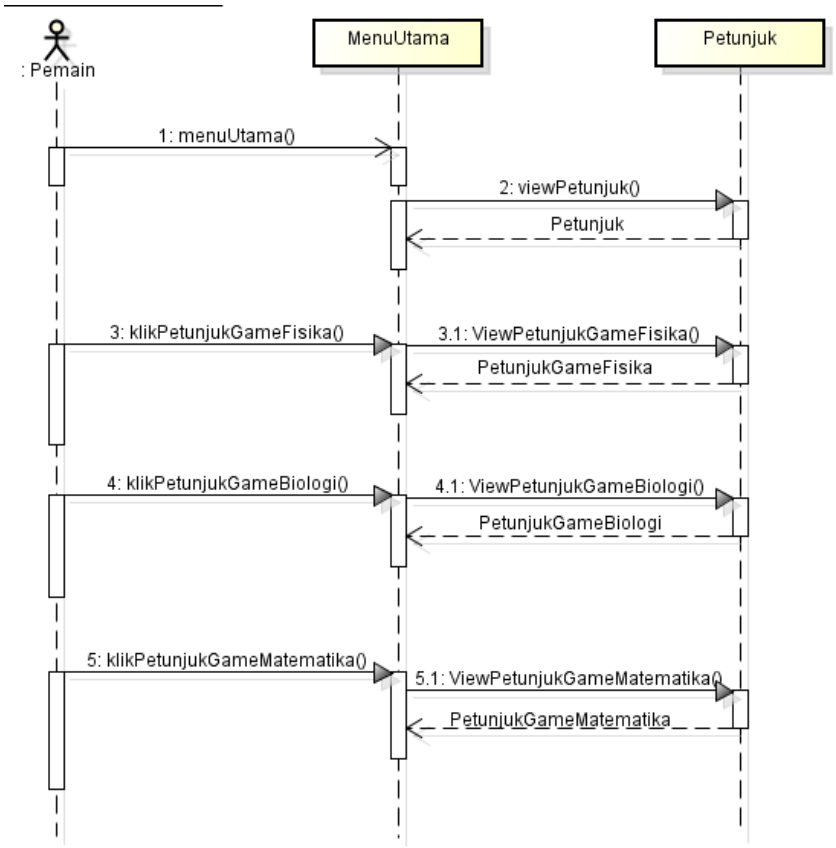
Identifikasi		
Nomor	12	
Nama	*Game* Gaya	
Tujuan	Memilih *game* Gaya	
Deskripsi	Proses untuk memilih dan memainkan *game* Gaya	
Aktor	*Gamer*	
Skenario Utama		
Kondisi Awal	Pengguna berada di menu pilihan *game* fisika	
Aksi Aktor		Reaksi Sistem
1. Memulai *game* fisika yang membahas materi bab Gaya		2. Aplikasi akan merespon dengan masuk ke permainan.
3. Menjawab huruf apa saja yang menjadi bagian dari jawaban dari pertanyaan yang ditampilkan		4. Aplikasi merespon dengan menampilkan pesan benar dan nilai akan bertambah 200 jika tidak ada kesalahan, 150 jika sekali salah dan 100 jika dua kali salah. Sedangkan jika jawaban salah maka tidak akan mendapatkan nilai dan aplikasi akan menampilkan pesan salah atau gagal.
5. Jika jawaban soal sebelumnya benar,		6. Aplikasi akan merespon dengan

maka *gamer* akan menjawab pertanyaan selanjutnya. Jika salah maka akan keluar dari Ekosistem.	keluar dari *game* jika user memilih tombol salah dan akan menampilkan pertanyaan selanjutnya jika emilih tombol selanjutnya.
Kondisi Akhir	Aplikasi akan menampilkan hasil akhir dari permainan dan menentukan apakah *user* menang atau kalah dalam permainan.

3.1.9.3 Sequence Diagram

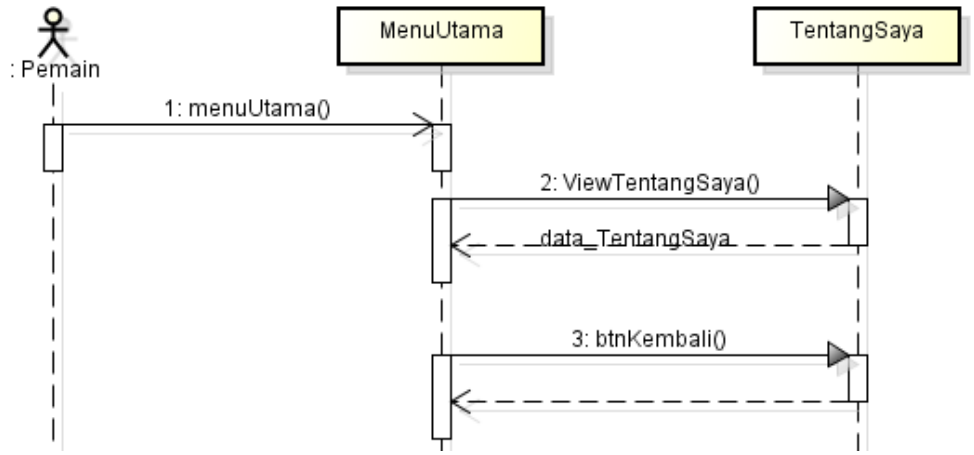
Diagram *sequence* atau *sequence diagram* menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek.

1. *Sequence Diagram Petunjuk*



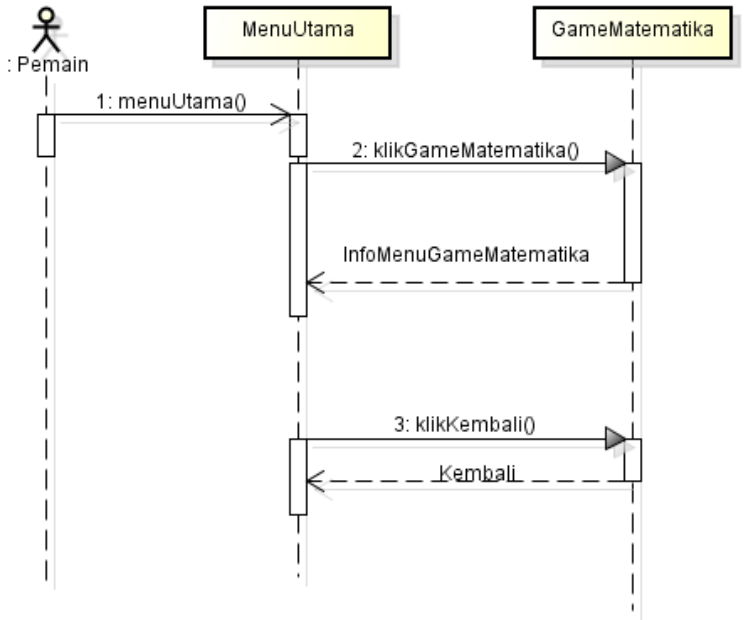
Gambar 3. 8 *Sequence Diagram Petunjuk*

2. *Sequence Diagram Tentang Saya*



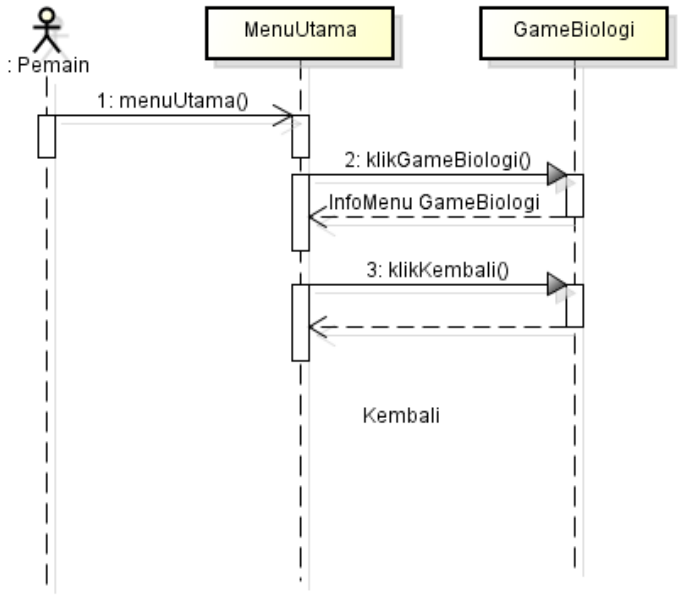
Gambar 3. 9 *Sequence Diagram Tentang Saya*

3. *Sequence Diagram Game* Matematika



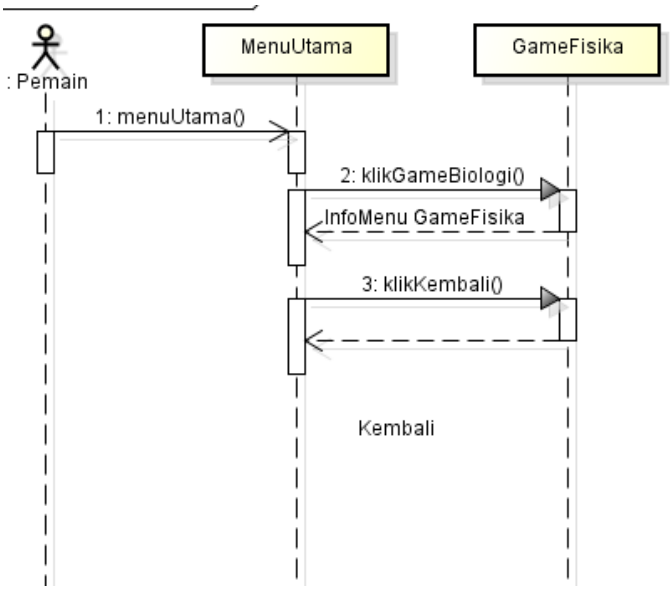
Gambar 3. 10 *Sequence Diagram Game* Matematika

4. *Sequence Diagram Game* Biologi



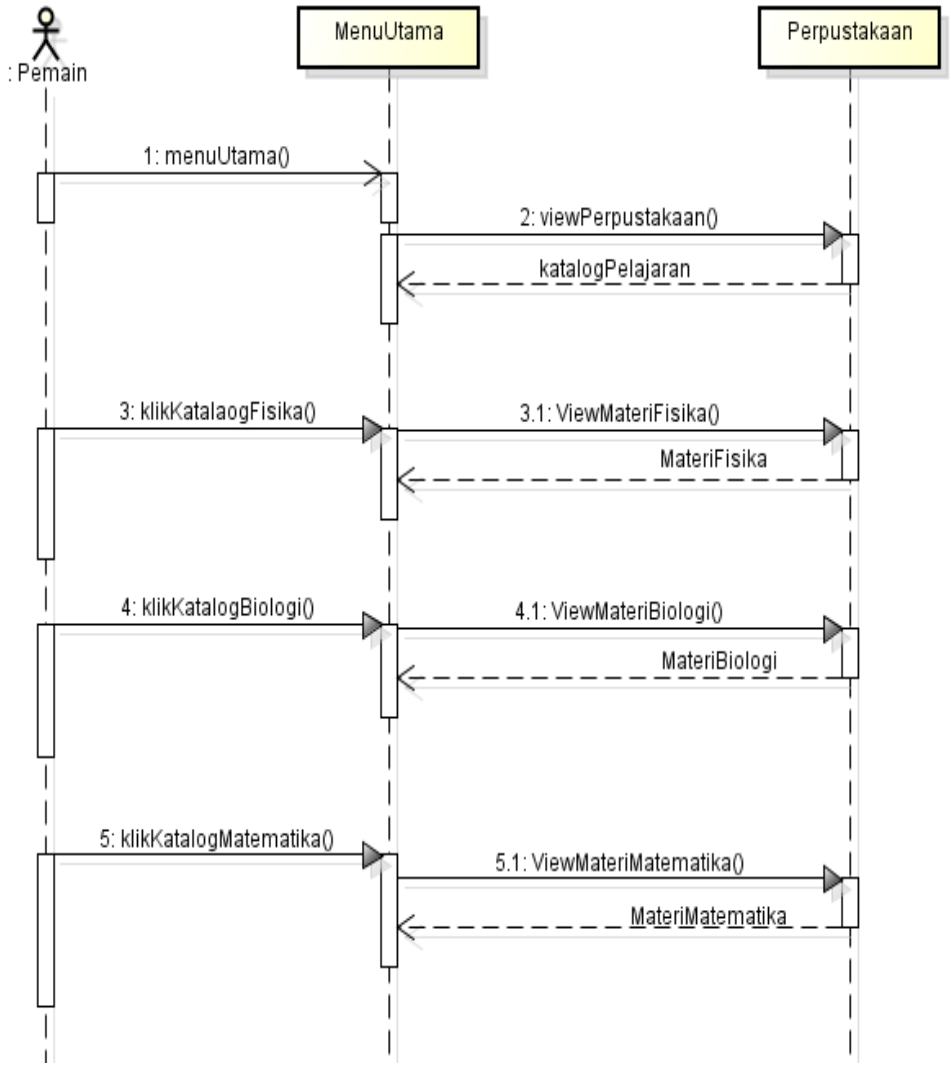
Gambar 3. 11 *Sequence Diagram Game* Biologi

5. *Sequence Diagram Game* Fisika



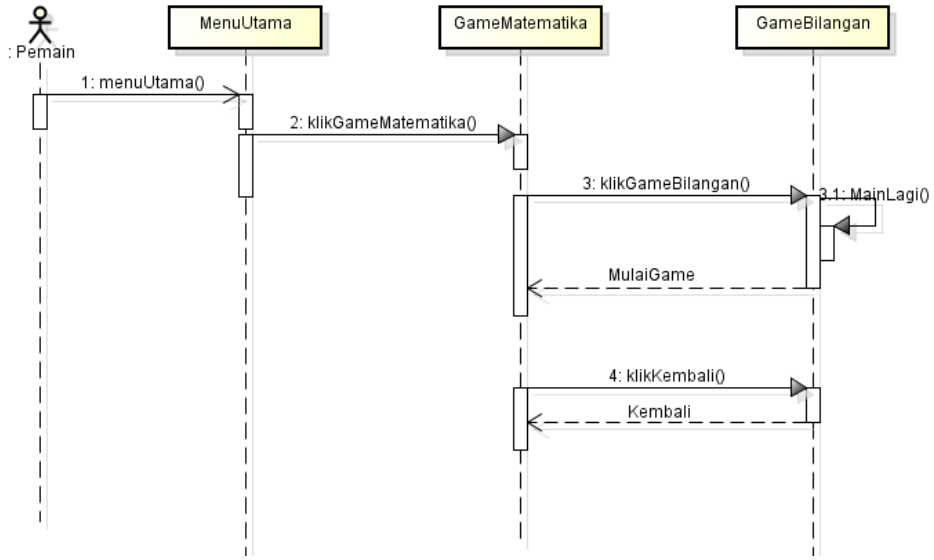
Gambar 3. 12 *Sequence Diagram Game* Fisika

6. *Sequence Diagram* Perpustakaan



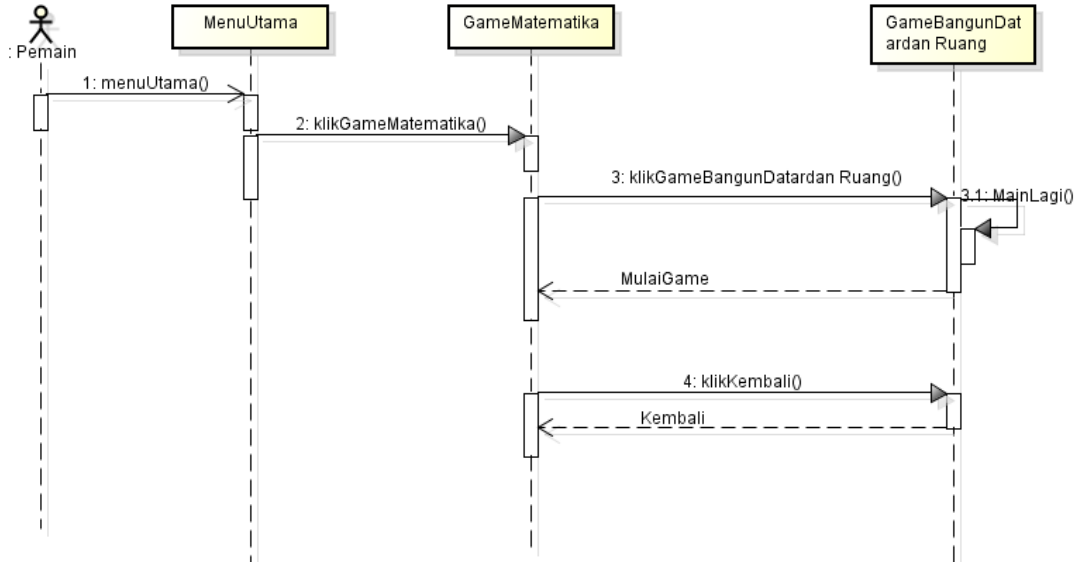
Gambar 3. 13 *Sequence Diagram* Perpustakaan

7. *Sequence Diagram Game* Bilangan



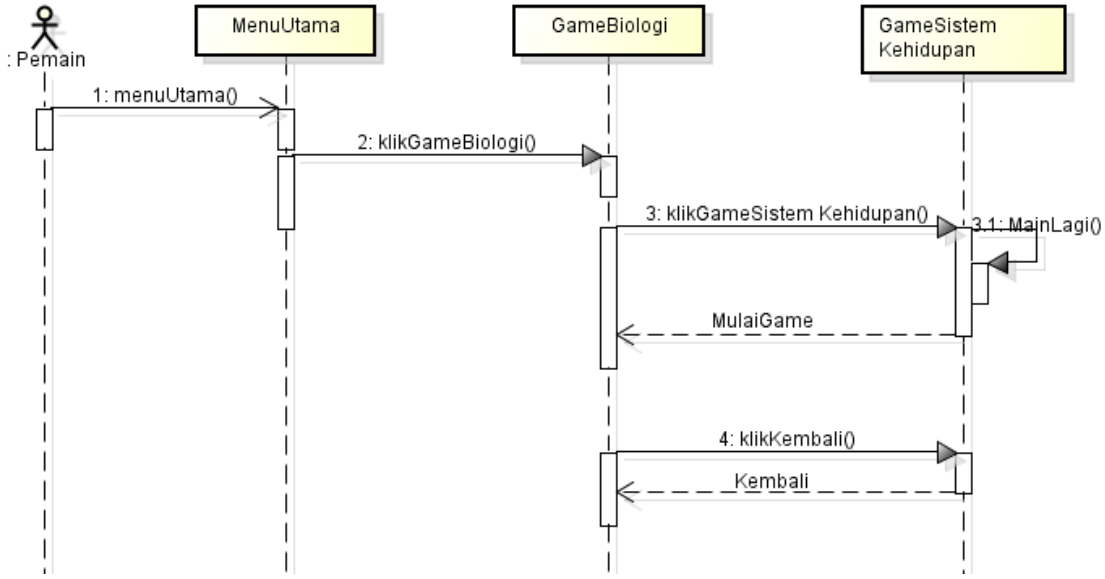
Gambar 3. 14 *Sequence Diagram Game* Bilangan

8. *Sequence Diagram Game* Bangun Datar dan Ruang



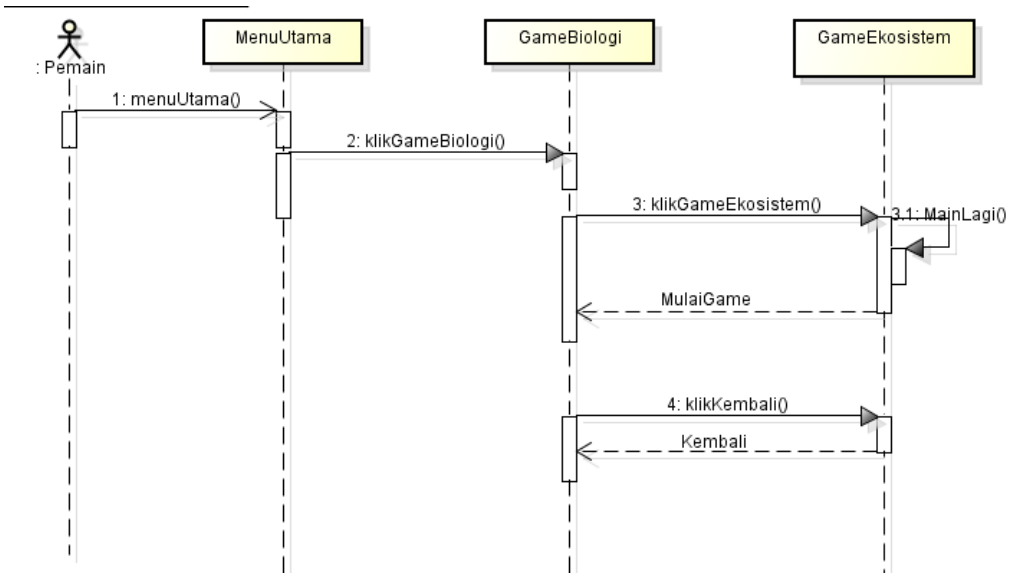
Gambar 3. 15 *Sequence Diagram Game* Bangun Datar dan Ruang

9. *Sequence Diagram Game* Sistem Kehidupan



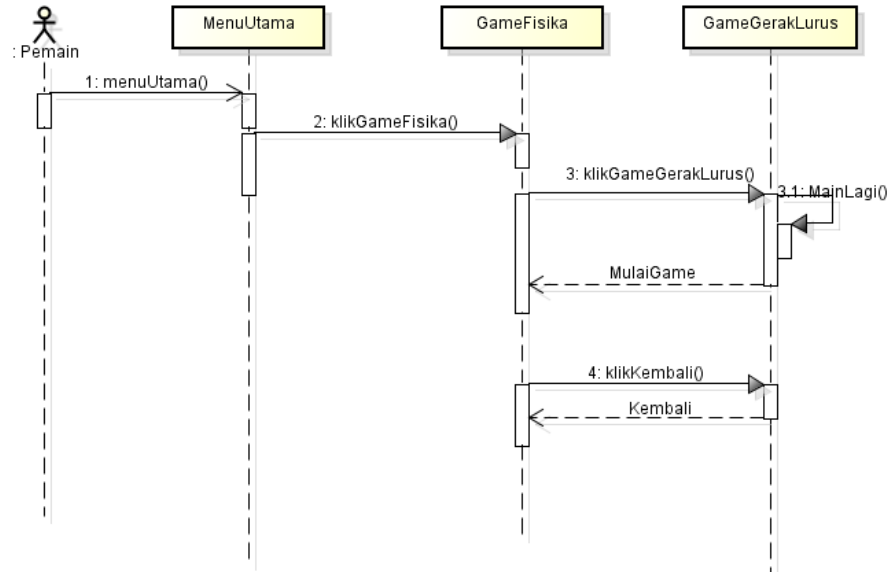
Gambar 3. 16 *Sequence Diagram Game* Sistem Kehidupan

10. *Sequence Diagram Game* Ekosistem



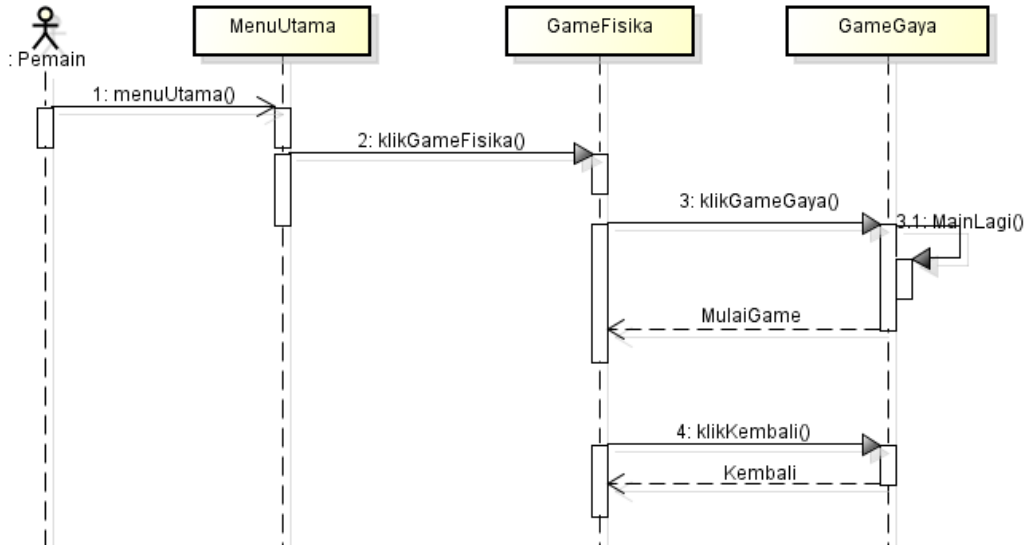
Gambar 3. 17 *Sequence Diagram Game* Ekosistem

11. *Sequence Diagram Game* Gerak Lurus



Gambar 3. 18 *Sequence Diagram Game* Gerak Lurus

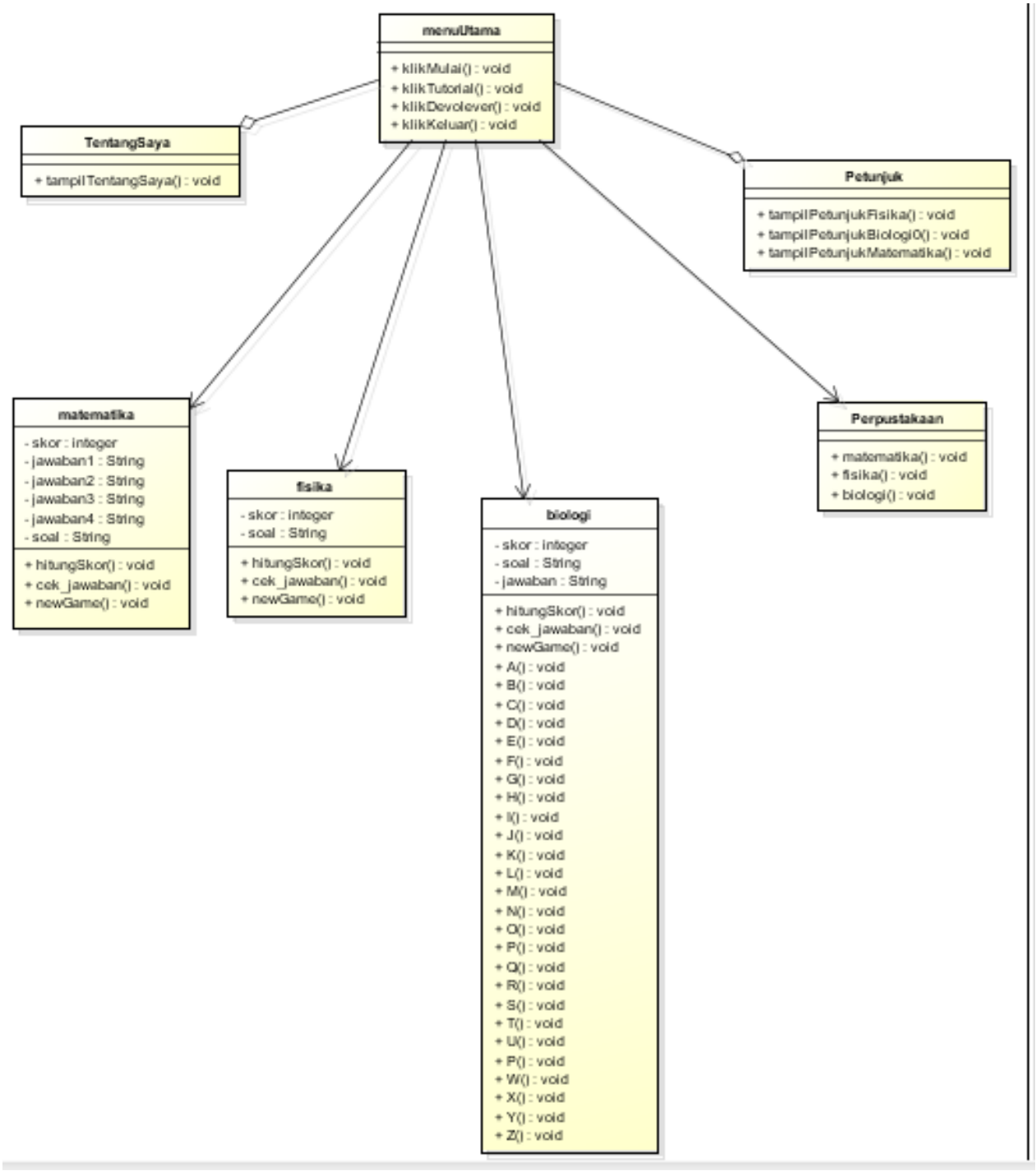
12. *Sequence Diagram Game* Gaya



Gambar 3. 19 *Sequence Diagram Game* Gaya

3.1.9.4 Class Diagram

Diagram kelas atau atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang dibuat untuk membangun sistem. Berikut ini adalah perancangan struktur sistem yang terdapat pada *Game* edukasi IPA dan Matematika untuk siswa SMP, yang digambarkan dengan *class diagram.*



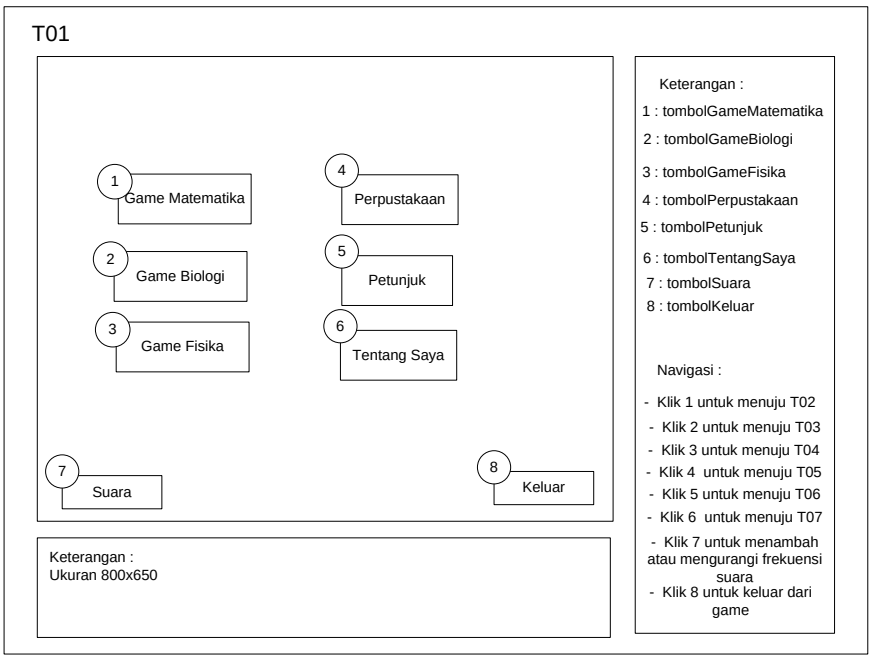
Gambar 3. 20 *Class Diagram*

3.2 Perancangan Sistem

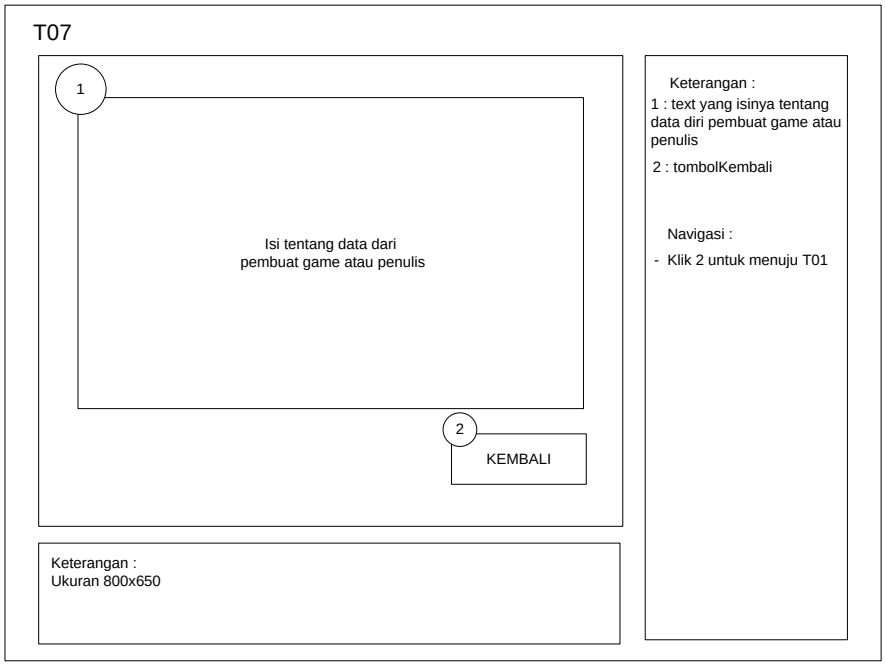
Perancangan adalah suatu bagian dari metodologi pengembangan pembangunan suatu perangkat lunak yang dilakukan setelah tahapan untuk memberikangambaran secara terperinci. Berdasarkan uraian diatas perancangan sistem merupakan tahapan dari siklus pengembangan sistem yang didefinisikan dari kebutuhan-kebutuhan fungsional dan persiapan untuk rancang bangun implementasi yang menggambarkan bagaimana suatu sistem dibentuk, yang dapat berupa penggambaran, perancangan, dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah kedalam satu kesatuan yang utuh dan berfungsi, juga menyangkut konfigurasi dari komponen-komponen perangkat keras dan perangkat lunak dari suatu sistem.

3.2.1 Perancangan Antar Muka

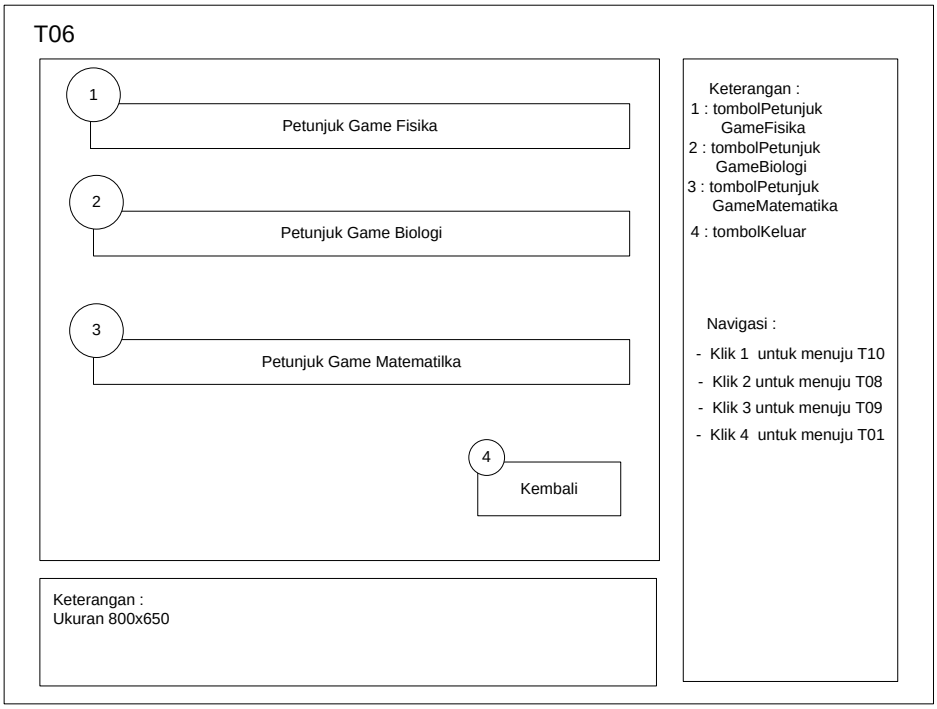
Perancangan antarmuka (*interface*) merupakan guna membentuk tampilan dari perangkat lunak yang berinteraksi dengan pemakai. Perancangan ini menggunakan aturan-aturan tertentu untuk menghasilkan antar muka yang baik dan benar.



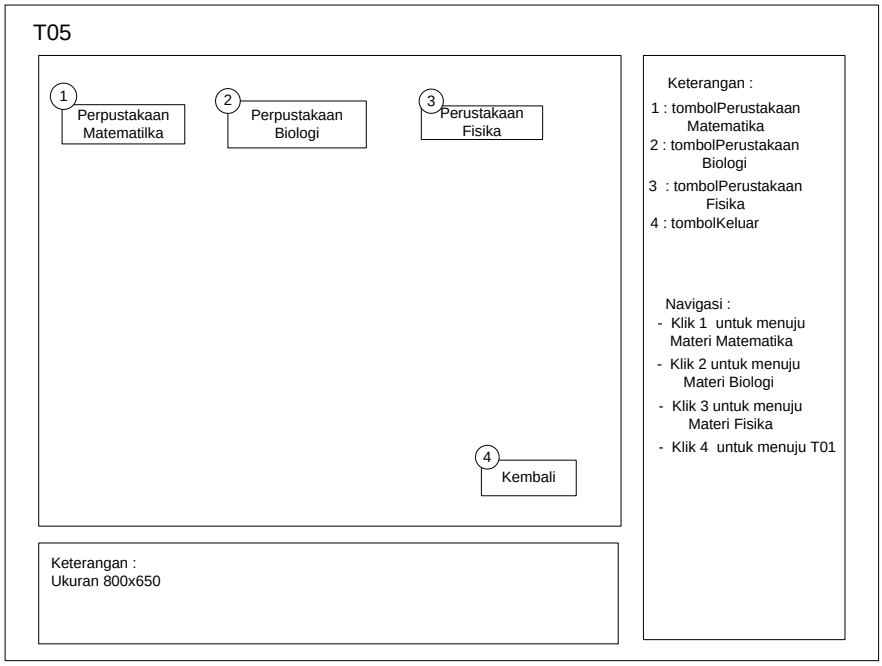
Gambar 3. 21 Perancangan Antar Muka Menu Utama



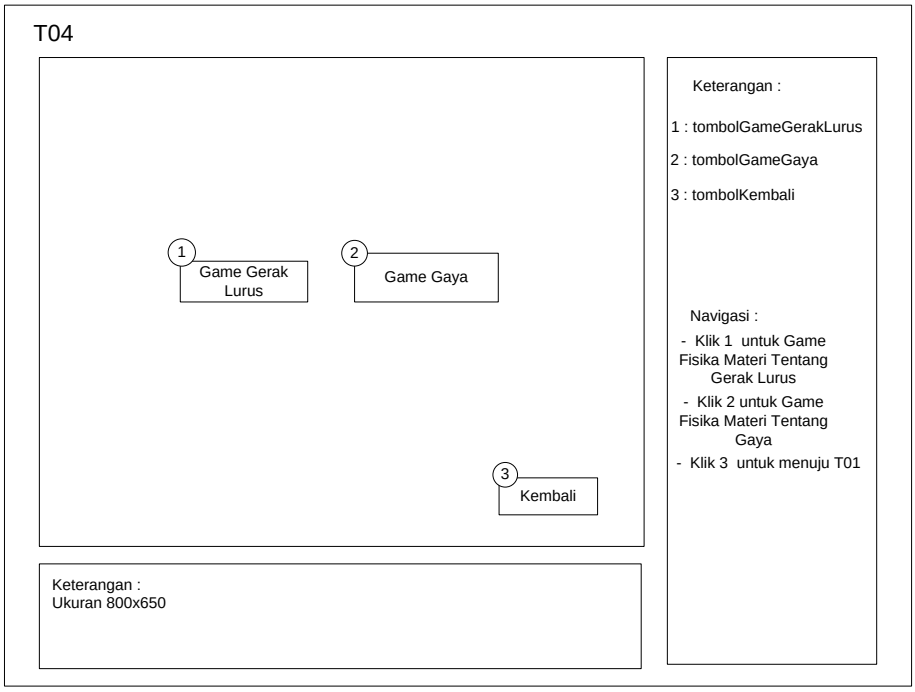
Gambar 3. 22 Perancangan Antar Muka Tentang Saya



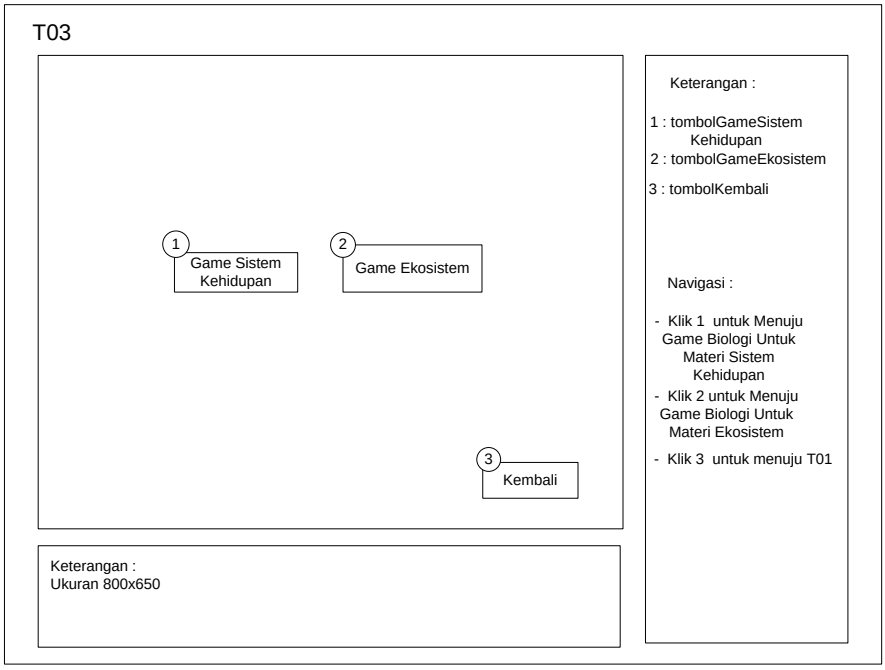
Gambar 3. 23 Perancangan Antar Muka Menu Petunjuk



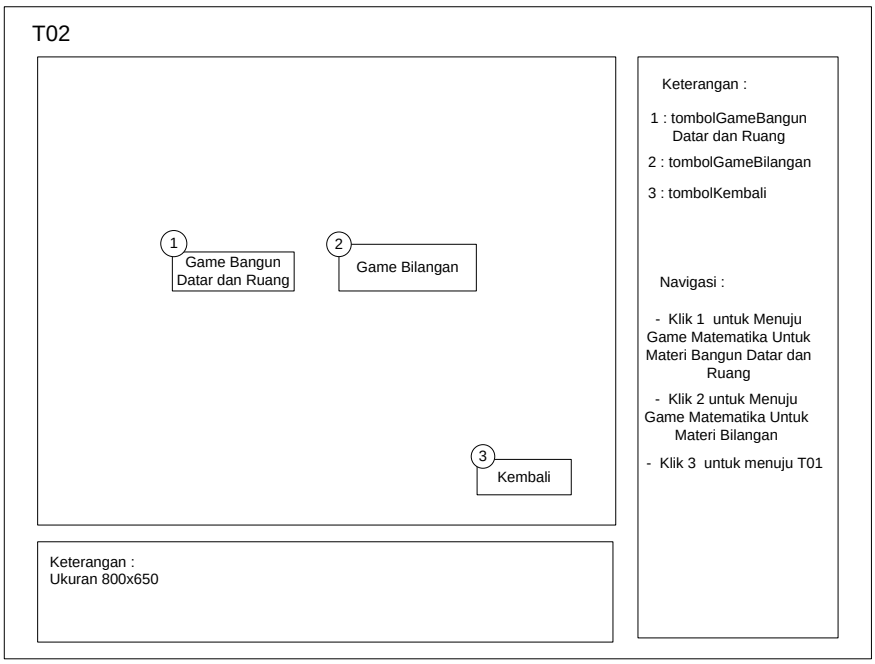
Gambar 3. 24 Perancangan Antar Muka Menu Perpustakaan



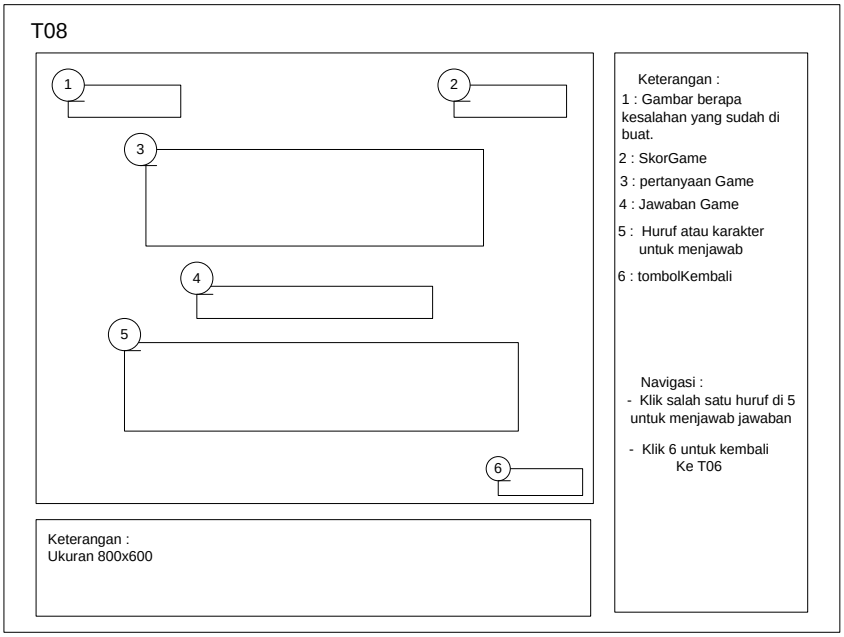
Gambar 3. 25 Perancangan Antar Muka Pilihan *Game* Fisika



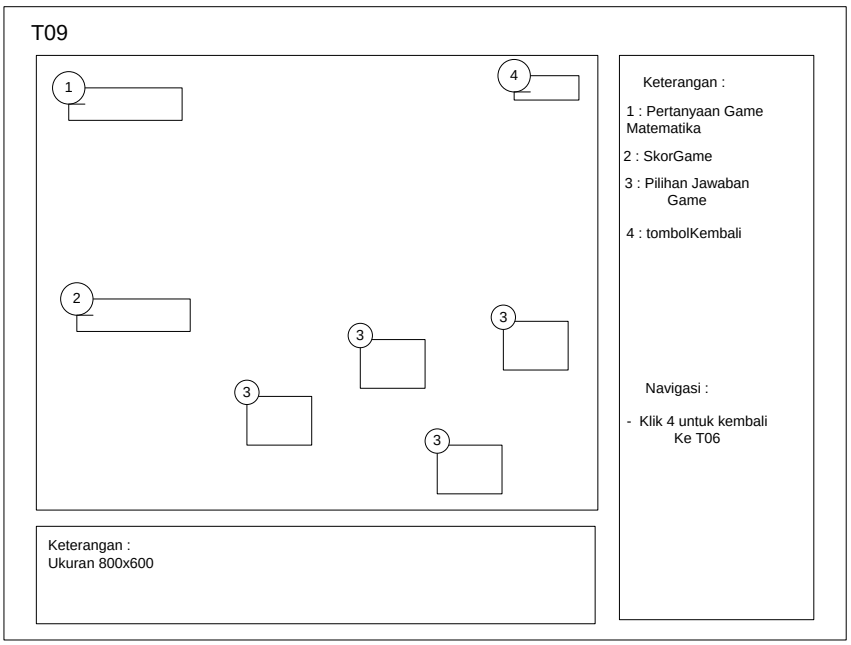
Gambar 3. 26 Perancangan Antar Muka Pilihan *Game* Biologi



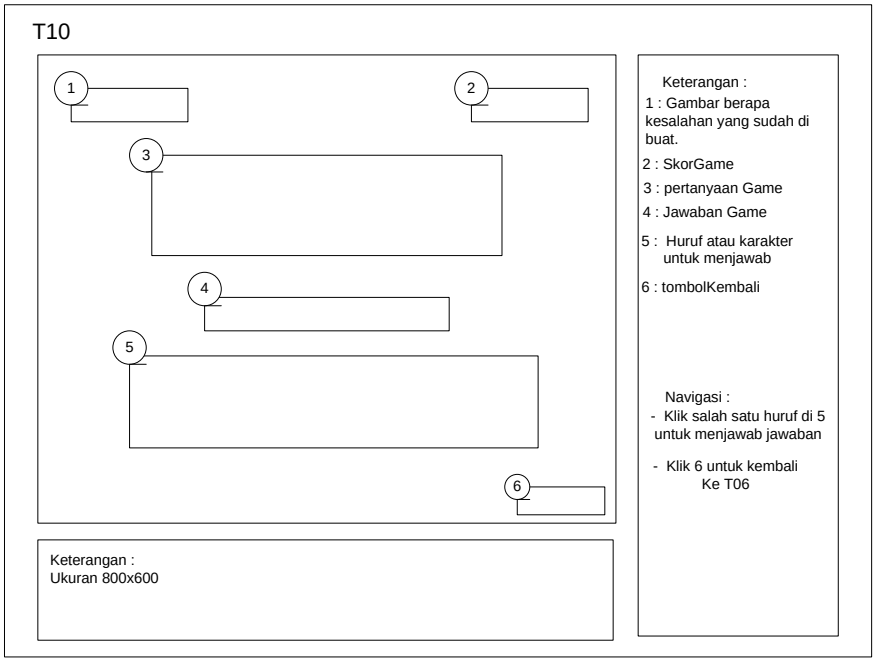
Gambar 3. 27 Perancangan Antar Muka Pilihan *Game* Matematika



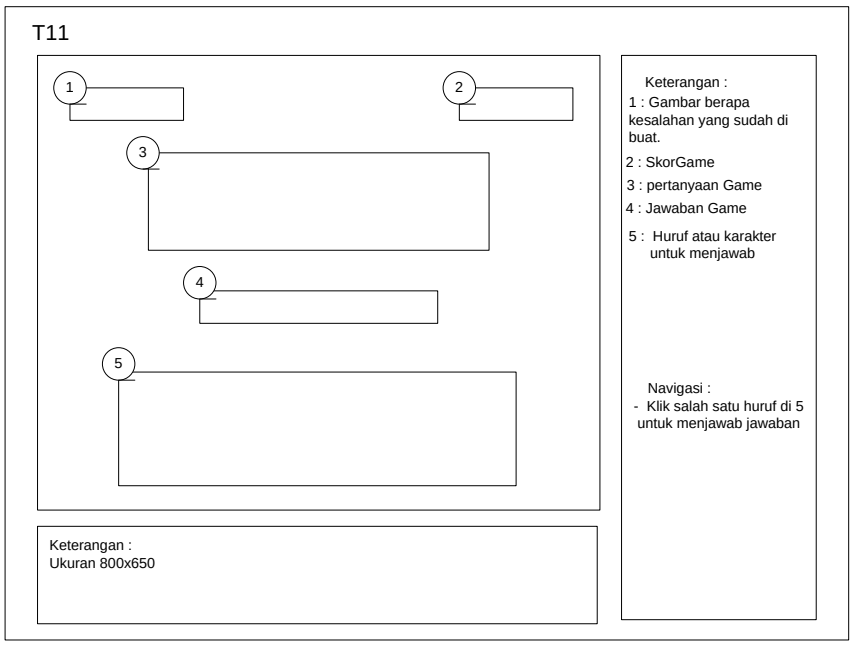
Gambar 3. 28 Perancangan Antar Muka Petunjuk *Game* Biologi



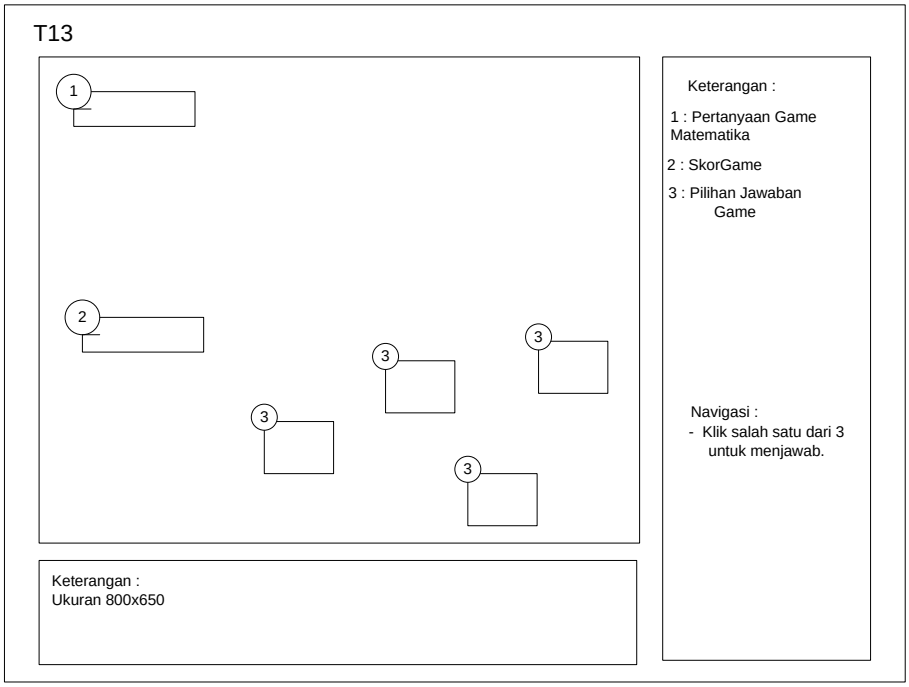
Gambar 3. 29 Perancangan Antar Muka Petunjuk *Game* Matematika



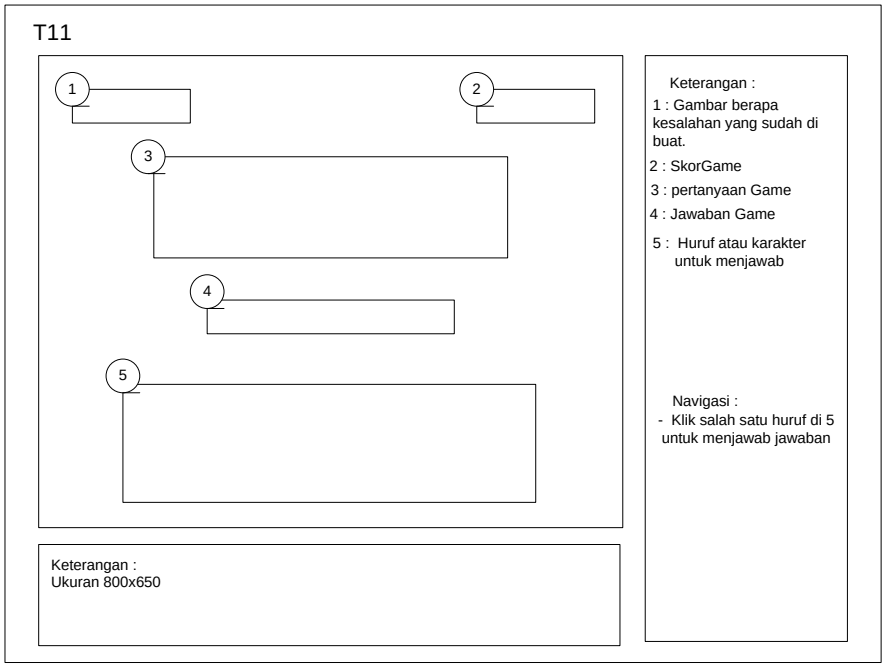
Gambar 3. 30 Perancangan Antar Muka Petunjuk *Game* Fisika



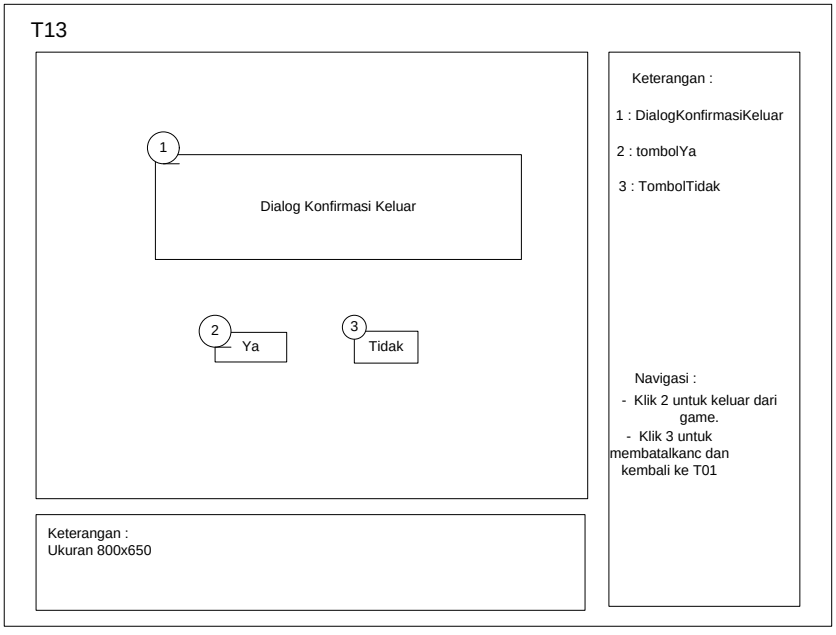
Gambar 3. 31 Perancangan Antar Muka *Game* Biologi



Gambar 3. 32 Perancangan Antar Muka *Game* Matematika



Gambar 3. 33 Perancangan Antar Muka *Game* Fisika



Gambar 3. 34 Perancangan Antar Muka Dialog Konfirmasi Keluar