BAB 3

ANALISIS DAN PERANCANGAN

3.1 Analisis Sistem

Pada bagian analisis sistem ini, akan dibangun berbagai macam analisis yang berhubungan dengan *game* yang akan dibangun. Tahap analisis sistem yang akan dibahas yaitu analisis masalah, analisis kebutuhan non fungsional dan analisis kebutuhan fungsional.

3.1.1 Analisis Masalah

Indonesia merupakan salah satu negara yg memiliki curah hujan cukup banyak. hujan terus-menerus pada musim penghujan dapat menyebabkan banjir yang melanda daerah-daerah padat penduduk. Banjir pada awal tahun seperti di Jakarta, sepanjang Sungai Bengawan Solo, dan beberapa wilayah di Indonesia membuktikan bahwa daya serap dan sungai yang ada tidak mampu menampung curah hujan yang sangat tinggi. Sikap masyarakat yang kurang sadar tehadap lingkungan mejadi penyebab utama terjadinya banjir seperti melakukan penebangan pohon secara liar di daerah pegunungan, merubahan fungsi lahan hijau sebagai daerah resapan air menjadi bangunan beton dan membuang sampah tidak pada tempatnya. Upaya yang dapat dilakukan dalam meningkatkan kesadaran masyarakat adalah melalui game. Selain menyenangkan, adanya gambaran dan keaktifan anak saat bermain *game* dapat memberikan pemahaman yang lebih jelas dalam upaya mencegah banjir.

3.1.2 Analisis *Game* Sejenis

Pada analisis *game* sejenis, akan dilakukan observasi terhadap *game* yang memiliki kesamaan *gameplay* dan *genre* dengan *game* yang akan dibangun.

3.1.2.1 *Game* Titok: Pahlawan Kebersihan



Gambar 3.1 Titok: Pahlawan Kebersihan

Game Titok: Pahlawan Kebersihan adalah *game* dengan *genre casual game* dimana game ini mengusung *gameplay* yang sederhana dan bisa dimainkan oleh semua kalangan. *Game* yang terdapat di sebuah mini website flash *game* portal ini menggunakan *platform flash* (swf). Dalam *game* ini, karakter utamanya adalah seorang anak yang bernama Titok. *Gameplay* dalam *game* Titok: Pahlawan Kebersihan adalah sebagai berikut:

1. Karakter merupakan anak yang bernama Titok.
2. Pemain harus mengumpulkan sampah-sampah yang ada di sungai kemudian memasukkan ke dalam bak sampah.
3. Karakter dapat bergerak sesuai dengan arah mouse.
4. Pemain harus mengumpulkan sampah sebelum parameter air penuh.
5. Jika pemain gagal mengumpulkan sampah maka game akan berakhir.

6. Pemain dapat melihat nilai yang diperoleh setelah permainan usai.
7. Pemain akan mendapatkan reward apabila dapat mencapai target yang telah ditentukan.

3.1.3 Analisis Game Yang Akan Dibangun

Pada bagian ini akan dijelaskan mengenai analisis-analisis *game* yang akan dibuat dari pengenalan *game* sampai dengan metode dan algoritma yang akan digunakan.

3.1.3.1 Deskripsi

Nama pada aplikasi *game* ini adalah Cegah Banjir. *Game* ini merupakan *game* ber-*genre* petualangan (*Adventure Game*). Pemberian nama *game* ini dilihat dari tujuan pembuatan permainan ini yaitu mengaplikasikan teknologi sebagai jembatan dalam memberikan pemahaman pada pemain bagaimana mencegah terjadinya banjir secara *interaktif* dan *edukatif*. *Game* ini merupakan *game* 2D dan dimainkan dengan menggunakan komputer. Berikut ini adalah fitur-fitur yang ada pada *game* ini:

1. Sistem *single player*
2. Mengangkat tema menjaga lingkungan dalam mencegah banjir
3. Grafik *game* 2D yang lebih berwarna
4. Jenis *game* bertipe *Adventure* yang disajikan secara interaktif
5. Aplikasi *game* bersifat bermain dan belajar karena dalam *game* disajikan materi-materi mengenai cara mencegah terjadinya banjir.

3.1.3.2 Storyline

Bencana Banjir sering terjadi di kota tinggalnya Ara. Ara merupakan seorang gadis pecinta alam yang selalu peduli dengan keberadaan lingkungannya. Ara yang penasaran akan seringnya banjir di kotanya kemudian pergi ke gunung untuk mencari jawaban dari seringnya banjir tersebut. Sesampainya di atas gunung yang dituju, Ara kaget karena mendapati pohon yang berada di gunung tersebut telah gundul karena penebangan pohon secara liar. Ara yang tidak terima

akan penebang pohon liar dan menyebabkan banjir di kotanya kemudian mencari pelaku penebangan pohon liar tersebut.

Air hujan yang mengalir dari pegunungan akan masuk ke bendungan. Tetapi penjaga bendungan ada keperluan mendadak, kemudian dia menelepon Ara untuk membantunya. Ara pun langsung bergegas menuju bendungan dan melihat keadaan di bendungan tersebut. Sesampainya disana, penjaga bendungan kemudian memberi tahu cara membersihkan sampah yang masuk ke bendungan. Disana terdapat robot yang dapat dikendalikan melalui ruang kontrol untuk memudahkan pembersihan sampah tersebut.

Setelah membantu penjaga bendungan, kemudian Ara kembali ke kotanya untuk beristirahat. Tetapi ditengah jalan Ara melihat orang yang membuang sampah di sungai. Pembuangan sampah tersebut akan mengakibatkan meluapnya air di sungai dan menyebabkan banjir.

3.1.3.3 *Gameplay*

Tugas pemain adalah melakukan petualangan dalam upaya mencegah terjadinya banjir. Pada *Game* ini terdapat tiga level berupa area yang akan dikunjungi, yaitu:

1. Level Satu

   Pada tahap awal pemain hanya dapat memilih area pegunungan. Pada area ini pemain akan diberikan Health Point (HP) sebagai nyawa. Pemain harus menangkap penebang pohon dengan menggunakan tali dan menyelesaikan misi menanam pohon pada lubang yang kosong dalam batas waktu yang telah ditentukan. Ketika pemain bertubrukan dengan penebang pohon akan mengurangi HP pemain. Apabila HP dan waktu pemain habis maka pemain harus mengulang misi tersebut. Pemain akan diberikan nilai 10 point untuk 1 tindakan yang dilakukan. Pemain dapat menggunakan *keyboard* untuk melakukan aksi seperti tombol panah untuk berjalan, tombol A untuk menyerang dan tombol S untuk menanam pohon.

2. Level Dua

   Setelah menyelesaikan area pegunungan, pemain dapat memilih area bendungan. Pada area ini pemain akan menggunakan robot untuk membersihkan sampah yang masuk ke kawasan bendungan dalam waktu yang telah ditentukan. Apabila ada 1 sampah yang masuk ke area pengaliran air maka HP pemain akan berkurang. Apabila pemain membiarkan 10 sampah masuk ke area pengaliran maka akan terjadi penyumbatan, akibatnya volume air meningkat sehingga bendungan jebol dan pemain harus mengulang misi tersebut. Pemain akan diberikan nilai 1 point untuk setiap pengambilan sampah yang memasuki bendungan. Robot yang digunakan pemain untuk membersihkan sampah dapat bergerak dengan menggunakan tombol panah pada *keyboard*.

3. Level Tiga

   Setelah menyelesaikan area bendungan, selanjutnya pemain dapat memilih area pemukiman. Pada area ini pemain harus menangkap orang yang membuang sampah sembarangan menggunakan tali dan menyelesaikan misi membersihkan sungai dengan menggunakan jaring yang harus diambil di area dalam waktu yang telah ditentukan. Ketika pemain bertubrukan dengan orang yang membuang sampah sembarangan akan mengurangi HP pemain. Apabila HP dan waktu pemain habis maka pemain harus mengulang misi tersebut. Pemain akan diberikan nilai 10 point untuk 1 tindakan yang dilakukan. Pemain dapat menggunakan *keyboard* untuk melakukan aksi seperti tombol panah untuk menggerakan karakter utama, tombol A untuk menyerang pembuang sampah dan tombol S untuk mengambil sampah.

3.1.4 Analisis Pengguna

Sasaran pengguna dari *game* Cegah Banjir adalah dari kelompok usia minimal 10 atau setara dengan SD kelas hingga SMA. Berdasarkan psikologi perkembangan anak pada usia 10 tahun memiliki ciri-ciri sebagai berikut:

1. Pendengar yang baik.

2. Ekspresif, suka menjelaskan dan aktif berbicara.
3. Bekerjasama dan bersaing.
4. Daya ingat cukup produktif.
5. Menyukai aturan dan hal-hal yang masuk akal.
6. Mampu konsentrasi dengan baik dan bisa membaca dalam waktu yg relatif lama.
7. Menjadi orang yang mampu menyelesaikan masalah dengan baik.

Berdasarkan informasi pada situs berita jatengonline.com, pada hari jumat tanggal 20 September 2013, Hotel jaringan Accor, Novotel dan Ibis Solo kembali menggelar program *corporate social responsibility* (CSR) alias tanggungjawab sosial perusahaan kepada masyarakat sekitar. Kali ini, program itu diimplementasikan dalam bentuk sosialisasi penanggulangan banjir kepada siswa Sekolah Dasar (SD) Joyotakan, Serengan. Dalam kesempatan itu, Green Team Novotel Ibis Solo memberikan penyuluhan tentang banjir mulai dari faktor penyebab maupun cara penanggulangan banjir. Kegiatan diikuti 60 siswa dari kelas V dan VI SD Joyotakan beserta perwakilan wali murid dan pemerintah setempat. Sosialisasi penyuluhan penanggulangan banjir dapat dilihat pada gambar 3.2.



Gambar 3.2 Sosialisasi Penyuluhan Penanggulangan Banjir di SD Joyotakan

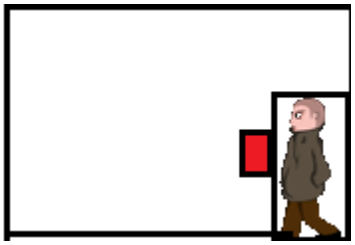
Adanya pemberitaan sosialisasi banjir dan ciri-ciri perkembangan psikologi anak dapat disimpulkan bahwa anak usia 10 tahun merupakan usia minimal yang paling efektif untuk dibekali pengetahuan tentang banjir. Selain itu, adanya mata pelajaran TIK di Sekolah Dasar menjadi penunjang bagi anak-anak untuk bisa menggunakan komputer.

3.1.5 Analisis Metode

Pada penelitian ini terdapat dua metode yang diterapkan yaitu metode *Collision Detection* dan algoritma Greedy. Dimana metode *Collision Detection* untuk mendeteksi tabrakan yang terjadi, sedangkan algoritma Greedy digunakan untuk pergerakan penebang pohon dan pembuang sampah sembarangan saat melawan karakter pemain.

3.1.5.1 *Collision Detection*

Metode *Collision Detection* adalah pendeteksian proses tabrakan antara dua objek atau lebih. *Collision Detection* sendiri terjadi ketika pemain menyerang penebang pohon dengan menggunakan senjata begitupun sebaliknya, sehingga terjadi proses tabrakan antara dua objek. Ketika tabrakan terjadi akan terpicu sebuah event yaitu berkurangnya HP pada objek yang terkena serangan. Contoh metode collision senjata dengan penebang pohon ditunjukan pada gambar 3.3.



Gambar 3.3 *Collision Detection*

Adapun pseudo code dari tahapan penerapan collision detection tersebut adalah sebagai berikut:

```
procedure CollisionRectangle(input : x1,y1,x2,y2)
{I.S : Menentukan nilai collision rectangle}
```

```
{F.S : Mendapatkan nilai condition berdasarkan hasil
collision}

Kamus :
x1,y1,x2,y2 : integer

Algoritma :
read x1,y1,x2,y2
if collision(x1,y1,x2,y2)=true then
condition = true
else condition = false
endif
write condition
```

3.1.5.2 Algoritma Greedy

Penggunaan algoritma greedy memberikan keputusan dengan cepat serta keputusan yang terbaik menjadikan algoritma greedy cocok digunakan pada permainan ini. Karena, dalam permainan ini dibutuhkan kecepatan berfikir untuk memutuskan pengambilan tindakan karakter yang harus diserang. Penggunaan algoritma greedy tersebut diterapkan pada semua karakter lawan.

Penerapan algoritma greedy yang digunakan pada karakter lawan tersebut, digunakan pada saat karakter lawan memutuskan untuk menyerang karakter utama, pohon atau sungai berdasarkan perkiraan jarak paling dekat dengan perkiraan health point (HP).

1. Himpunan Kandidat

   Untuk penerapan pada level pertama pada game ini, terdapat 4 kandidat untuk di serang, yaitu 1 karakter utama dan 3 pohon. Kandidat tersebut kemudian di inisialisasikan dengan C atau himpunan kandidat.

2. Himpunan Solusi

   Himpunan solusi merupakan kandidat-kandidat yang menjadi solusi untuk diserang, dengan perkiraan jarak paling dekat dan hp kandidat tersebut lebih kecil dari karakter lawan. Himpunan solusi tersebut diinisialisasikan dengan S.

3. Fungsi Seleksi

   Kandidat yang ada kemudian diseleksi berdasarkan jarak paling dekat. Untuk kemudian diuji kelayakannya.

4. Fungsi Kelayakan

   Kemudian kandidat tersebut diperiksa kelayakannya dengan cara memeriksa apakah nilai HP dari kandidat tersebut lebih kecil dari HP lawan atau tidak. Jika HP dari kandidat tersebut lebih kecil dari HP lawan kandidat maka dinyatakan layak dan dimasukan ke dalam solusi.

5. Fungsi Obyektif

   Kandidat yang menjadi solusi tersebut kemudian menjadi solusi optimum untuk diputuskan bahwa kandidat tersebut harus diserang.

Adapun pseudo code dari tahapan penerapan algoritma greedy tersebut adalah sebagai berikut:

```
procedure greedy(kandidat:karakter utama,pohon/sungai)
{I.S  : Menentukan kandidat untuk untuk diserang}
{F.S : Mendapatkan kandidat yang menjadi solusi optimum
untuk diserang}

Kamus :
c[a]← kandidat
s ← {}
a ← {}

Algoritma :
read C[kandidat]
if HP >= 100 then
A ← jarak kandidat paling dekat //fungsi seleksi
   if HP >= A then //fungsi kelayakan
   S ← A
   C[A]-=1
        if S=A then //fungsi objektif
        attack = true
```

```
            else attack = false
            endif
        else attack = false
        endif
    else attack = false
    endif
    end
```

3.1.6 Analisis Kebutuhan Non Fungsional

Analisis non fungsional merupakan analisis yang dibutuhkan untuk menentukan spesifikasi kebutuhan sistem. Spesifikasi ini juga meliputi elemen atau komponen-komponen apa saja yang dibutuhkan untuk sistem yang akan dibangun sampai dengan sistem tersebut diimplementasikan. Analisis kebutuhan ini juga menentukan spesifikasi masukan yang diperlukan sistem, keluaran yang akan dihasilkan sisten dan proses yang dibutuhkan untuk mengolah masukan sehingga menghasilkan suatu keluaran yang diinginkan.

Pada analisis kebutuhan sistem non fungsional ini dijelaskan analisis kebutuhan perangkat lunak, analisis kebutuhan perangkat keras dan analisis pengguna.

3.1.6.1 Analisis dan Kebutuhan Perangkat Lunak *Developer*

Kebutuhan perangkat lunak adalah untuk memberikan perintah-perintah perangkat keras agar saling berinteraksi merupakan bagian dari sistem. Kebutuhan perangkat lunak yang dibutuhkan dalam membangun aplikasi *game* Cegah Banjir adalah sebagai berikut:

1. Sistem operasi Window 7,
2. Game Maker 8 Pro,
3. GIMP 2.0

3.1.6.2 Analisis dan Kebutuhan Perangkat Lunak *User*

Adapun untuk kebutuhan perangkat lunak dari sisi *user* untuk memainkan *game* Cegah Banjir adalah menggunakan sistem operasi window 7.

3.1.6.3 Analisis Kebutuhan Perangkat Keras *Developer*

Perangkat keras saling berhubungan dengan perangkat lunak dan menjalankan perintah-perintah sistem. Perangkat keras yang digunakan pada *game* Cegah Banjir adalah sebagai berikut:

1. RAM 2GB,
2. Harddisk 500 GB,
3. VGA 2 GB,
4. Monitor
5. Keyboard dan mouse

3.1.6.4 Analisis Kebutuhan Perangkat Keras *User*

Adapun untuk kebutuhan perangkat keras yang dibutuhkan *user* untuk menjalankan *game* Cegah Banjir adalah sebagai berikut:

1. RAM 1GB,
2. Harddisk 50 GB,
3. VGA 512 MB,
4. Monitor
5. Keyboard dan mouse

3.1.7 Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional merupakan gambaran, perencanaan dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah ke dalam satu kesatuan yang utuh dan berfungsi. Alat bantu untuk menggambarkan sistem menggunakan *Unified Modeling Language* (UML), dimana tahapan-tahapan yang terdapat pada *game* Cegah Banjir adalah *Use Case Diagram, Activitiy Diagram, Class Diagram* dan *Sequence Diagram*.

3.1.7.1 Deskripsi Global Perangkat Lunak

Game yang akan dibangun merupakan *game* dengan jenis *adventure*, dimana pemain sebagai karakter utama harus menyelesaikan misi yang berbeda pada setiap level. Pemain dapat menjalankan game ini dengan bantuan *keyboard*

dan *mouse*. Pemain memiliki senjata untuk menyerang lawan. Setiap tindakan yang dilakukan pemain ketika menyelesaikan misi akan diberikan *point*.

3.1.7.2 Fungsi Produk

Game ini mempunyai fungsi-fungsi utama, fungsi utama tersebut diuraikan sebagai berikut:

1. Permainan

   Game ini dapat berjalan disebuah komputer dan menampilkan sebuah permainan.
2. Pengaturan

   Pada *game* ini pengguna dapat melakukan pengaturan suara latar, pengaturan suara efek dan pengaturan layar untuk permainan.
3. Nilai Tertinggi

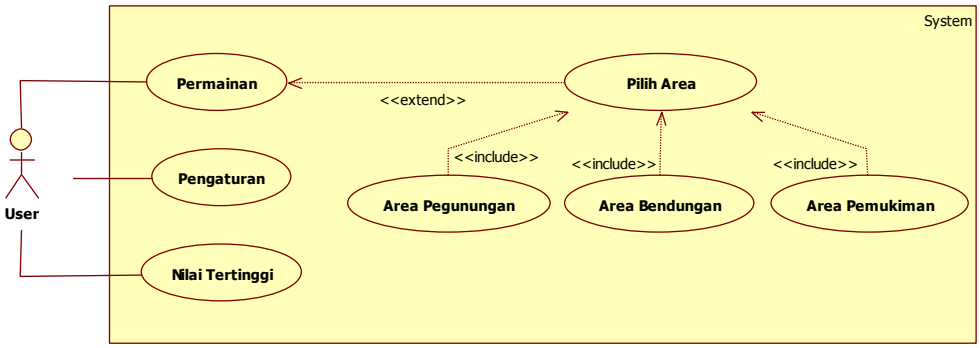
   Pada *game* ini dapat melihat informasi nilai tertinggi yang didapat oleh pemain seusai setiap permainan.
4. Keluar

   Pada *game* ini dapat menutup aplikasi game setelah dimainkan.

3.1.7.3 Use Case Diagram

Use Case atau diagram use case merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. Use Case mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat [8].

Berikut adalah perancangan use case diagram pada game cegah banjir dapat dilihat pada gambar 3.4.



Gambar 3.4 Use Case Diagram *Game* Cegah Banjir

Use Case terdiri dari tiga definisi bagian yaitu definisi *actor*, definisi *use case* dan skenario *use case*.

3.1.7.4 Definisi Actor

Definisi *actor* menjelaskan *actor* yang terlibat pada *use case diagram*. Berikut ini adalah tabel yang menerangkan definisi *actor*.

Tabel 3.1 Definisi *Actor*

No.	*Actor*	Deskripsi
1.	User	Merupakan orang yang memainkan *game*.

3.1.7.5 Definisi *Use Case*

Definisi *use case* menjelaskan fungsi *use case* yang terdapat pada *use case diagram*. Berikut ini adalah tabel yang menerangkan definisi *use case*.

Tabel 3.2 Definisi *Use Case*

No.	Use Case	Deskripsi
1.	Permainan	Proses untuk memilih memainkan *game*.
2.	Pengaturan	Proses untuk memilih pengaturan pada *game*.
3.	Nilai Tertinggi	Proses untuk melihat informasi nilai tertinggi yang didapatkan oleh pemain seusai permainan.
4.	Pilih Area	Proses untuk memilih area yang akan digunakan pada

		game.
5.	Area Pegunungan	Proses untuk memulai permainan di area Pegunungan.
6.	Area Bendungan	Proses untuk memulai permainan di area Bendungan.
7.	Area Pemukiman	Proses untuk memulai permainan di area Pemukiman.

3.1.7.6 Skenario *Use Case*

Skenario *use case* menggambarkan alur penggunaan sistem dimana setiap skenario digambarkan dari sudut pandang aktor, seseorang atau piranti yang berinteraksi dengan perangkat lunak dalam berbagai cara.

Tabel 3.3 Skenario *Use Case* Permainan

Identifikasi	
Nomor	1.
Nama	Mulai Permainan.
Tujuan	Memulai Permainan.
Deskripsi	Proses untuk memulai permainan.
Aktor	*User*
Skenario Utama	
Kondisi Awal	*User* berada di halaman menu utama
Aksi Aktor	**Reaksi Sistem**
1. Memilih menu permainan	
	2. Menampilkan halaman pilih area
3. Memilih tombol *back*.	
	4. Menampilkan halaman menu utama
Kondisi Akhir	Sistem berada di halaman menu utama

Tabel 3.4 Skenario *Use Case* Pengaturan

Identifikasi	
Nomor	2.
Nama	Pengaturan.
Tujuan	Mengatur pengaturan pada *game*.
Deskripsi	Proses untuk mengatur pengaturan *game*.
Aktor	*User*
Skenario Utama	
Kondisi Awal	*User* berada di halaman pengaturan
Aksi Aktor	**Reaksi Sistem**
1. Memilih musik aktif atau tidak.	
	2. Jika memilih mengaktifkan musik, musik diaktifkan dan menampilkan halaman pengaturan.
	3. Jika memilih mematikan musik, musik dimatikan dan menampilkan halaman pengaturan.
4. Memilih mengatur pengaturan layar.	
	5. Jika memilih mengaktifkan fungsi layar penuh, fungsi layar penuh diaktifkan dan menampilkan halaman pengaturan.
	6. Jika memilih mematikan suara fungsi layar penuh, fungsi layar penuh dimatikan dan menampilkan halaman pengaturan.
7. Memilih tombol *back*.	
	8. Menampilkan halaman menu utama

Kondisi Akhir	Sistem berada di halaman menu utama

Tabel 3.5 Skenario *Use Case* Nilai Tertinggi

Identifikasi	
Nomor	3.
Nama	Nilai Tertinggi.
Tujuan	Melihat informasi.
Deskripsi	Proses untuk melihat informasi nilai yang didapatkan dari *game*.
Aktor	*User*
Skenario Utama	
Kondisi Awal	*User* berada di halaman menu utama
Aksi Aktor	**Reaksi Sistem**
1. Memilih nilai tertinggi	
	2. Menampilkan nilai tertinggi
3. Memilih tombol *back*.	
	4. Menampilkan halaman menu utama
Kondisi Akhir	Sistem berada di halaman menu utama

Tabel 3.6 Skenario *Use Case* Pilih Area

Identifikasi	
Nomor	4.
Nama	Pilih Area.
Tujuan	Memilih area yang akan dipakai dalam permainan.
Deskripsi	Proses untuk memilih area pada *game*.
Aktor	*User*
Skenario Utama	
Kondisi Awal	*User* berada di menu halaman pilih area
Aksi Aktor	**Reaksi Sistem**
1. Memilih area.	

	2. Cek ketersediaan area. Jika ada ditampilkan.
	3. Jika tidak ada tidak ditampilkan
Kondisi Akhir	Sistem menampilkan menu halaman pilih area

Tabel 3.7 Skenario *Use Case* Area Pegunungan

Identifikasi	
Nomor	5.
Nama	Area Pegunungan
Tujuan	Memulai Permainan.
Deskripsi	Proses untuk memulai permainan.
Aktor	*User*
Skenario Utama	
Kondisi Awal	*User* berada di halaman pilih area
Aksi Aktor	**Reaksi Sistem**
1. Memilih area pegunungan	
	2. Menampilkan materi stage
3. Memilih close materi stage	
	4. Menampilkan cerita stage
5. Memilih lewati cerita stage.	
	6. Menampilkan misi stage
7. Memilih ok misi stage	
	8. Menampilkan cara bermain
9. Memilih ok cara bermain	
	10. Memulai permainan dan menampilkan stage game.
	11. Cek HP karakter dan waktu sama dengan 0 atau tidak. Jika ya menampilkan pesan kalah.
	12. Jika tidak menampilkan stage game.

	13. Cek HP lawan sama dengan 0 dan semua lubang telah ditanami atau tidak. Jika ya menampilkan pesan menang.
	14. Jika tidak, menampilkan stage game.
15. Menggerakan karakter dengan menekan tombol *keyboard.*	
	16. Jika menekan tombol *keyboard* "→", karakter bergerak ke kanan.
	17. Jika menekan tombol *keyboard* "←", karakter bergerak ke kiri.
	18. Jika menekan tombol *keyboard* "↑", karakter bergerak ke atas.
	19. Jika menekan tombol *keyboard* "↓", karakter bergerak ke bawah.
	20. Jika menekan tombol *keyboard* "A", karakter melemparkan tali.
	21. Jika menekan tombol *keyboard* "S", karakter menanam pohon.
22. Jika memilih mengulang game	
	23. Menampilkan pesan yakin akan mengulang permainan atau tidak. Jika "Ya" memulai permainan dan menampilkan misi stage.
	24. Jika memilih "Tidak", menampilkan stage game.

25. Memilih keluar stage game	
	26. Menampilkan pesan yakin akan keluar permainan atau tidak. Jika memilih "Ya" keluar dari stage game.
	27. Menampilkan halaman menu utama
	28. Jika memilih "Tidak" kembali ke stage game
Kondisi Akhir	Sistem berada di halaman pilih area

Tabel 3.8 Skenario *Use Case* Area Bendungan

Identifikasi	
Nomor	6.
Nama	Area Bendungan
Tujuan	Memulai Permainan.
Deskripsi	Proses untuk memulai permainan.
Aktor	*User*
Skenario Utama	
Kondisi Awal	*User* berada di halaman pilih area
Aksi Aktor	**Reaksi Sistem**
1. Memilih area bendungan	
	2. Menampilkan materi stage
3. Memilih close materi stage	
	4. Menampilkan cerita stage
5. Memilih lewati cerita stage.	
	6. Menampilkan misi stage
7. Memilih ok misi stage	
	8. Menampilkan cara bermain
9. Memilih ok cara bermain	

	10. Memulai permainan dan menampilkan stage game.
	11. Cek HP karakter sama dengan 0 atau tidak. Jika ya menampilkan pesan kalah.
	12. Jika tidak menampilkan stage game.
	13. Cek waktu main sama dengan 0. Jika ya menampilkan pesan menang.
	14. Jika tidak menampilkan stage game
15. Menggerakan karakter dengan menekan tombol *keyboard.*	
	16. Jika menekan tombol *keyboard* "→", karakter bergerak ke kanan.
	17. Jika menekan tombol *keyboard* "←", karakter bergerak ke kiri.
	18. Jika menekan tombol *keyboard* "↑", karakter bergerak ke atas.
	19. Jika menekan tombol *keyboard* "↓", karakter bergerak ke bawah.
20. Jika memilih mengulang game	
	21. Menampilkan pesan yakin akan mengulang permainan atau tidak. Jika "Ya" memulai permainan dan menampilkan misi stage.
	22. Jika memilih "Tidak",

	menampilkan stage game.
23. Memilih keluar stage game	
	24. Menampilkan pesan yakin akan keluar permainan atau tidak. Jika memilih "Ya" keluar dari stage game
	25. Menampilkan halaman menu utama
	26. Jika memilih "Tidak" kembali ke stage game
Kondisi Akhir	Sistem berada di halaman pilih area

Tabel 3.9 Skenario *Use Case* Area Pemukiman

Identifikasi	
Nomor	7.
Nama	Area Pemukiman
Tujuan	Memulai Permainan.
Deskripsi	Proses untuk memulai permainan.
Aktor	*User*
Skenario Utama	
Kondisi Awal	*User* berada di halaman pilih area
Aksi Aktor	**Reaksi Sistem**
1. Memilih area pemukiman	
	2. Menampilkan materi stage
3. Memilih close materi stage	
	4. Menampilkan cerita stage
5. Memilih lewati cerita stage.	
	6. Menampilkan misi stage
7. Memilih ok misi stage	
	8. Menampilkan cara bermain

9. Memilih ok cara bermain	
	10. Memulai permainan dan menampilkan stage game.
	11. Cek HP karakter dan waktu sama dengan 0 atau tidak. Jika ya menampilkan pesan kalah.
	12. Jika tidak menampilkan stage game.
	13. Cek HP lawan sama dengan 0 dan semua sungai bersih atau tidak. Jika ya menampilkan pesan menang.
	14. Jika tidak, menampilkan stage game.
15. Menggerakan karakter dengan menekan tombol *keyboard.*	
	16. Jika menekan tombol *keyboard* "→", karakter bergerak ke kanan.
	17. Jika menekan tombol *keyboard* "←", karakter bergerak ke kiri.
	18. Jika menekan tombol *keyboard* "↑", karakter bergerak ke atas.
	19. Jika menekan tombol *keyboard* "↓", karakter bergerak ke bawah.
	20. Jika menekan tombol *keyboard* "A", karakter melemparkan tali.
	21. Jika menekan tombol *keyboard* "S", karakter membersihkan sungai.

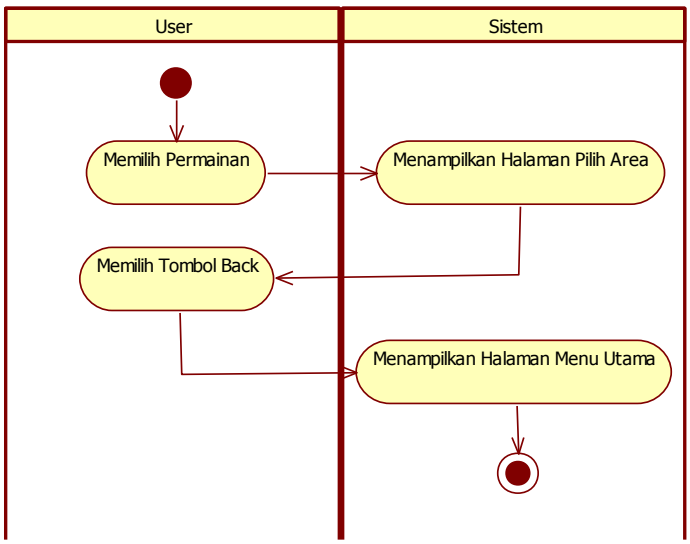
22. Jika memilih mengulang game	
	23. Menampilkan pesan yakin akan mengulang permainan atau tidak. Jika "Ya" memulai permainan dan menampilkan misi stage.
	24. Jika memilih "Tidak", menampilkan stage game.
25. Memilih keluar stage game	
	26. Menampilkan pesan yakin akan keluar permainan atau tidak. Jika memilih "Ya" keluar dari stage game.
	27. Menampilkan halaman menu utama
	28. Jika memilih "Tidak" kembali ke stage game
Kondisi Akhir	Sistem berada di halaman pilih area

3.1.7.7 Activity Diagram

Diagram aktifitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktifitas dari sebuah sistem atau proses bisnis [8]. Dalam diagram aktifitas yangg terlibat adalah antara *user* dan sistem. Terdapat tujuh aktifitas yang tergambar sesuai prosesnya masing-masing yaitu, *activity diagram* permainan, *activity diagram* pengaturan, *activity diagram* nilai tertinggi, *activity diagram* mulai permainan dan *activity diagram* pilih map.

3.1.7.7.1 Activity Diagram Permainan

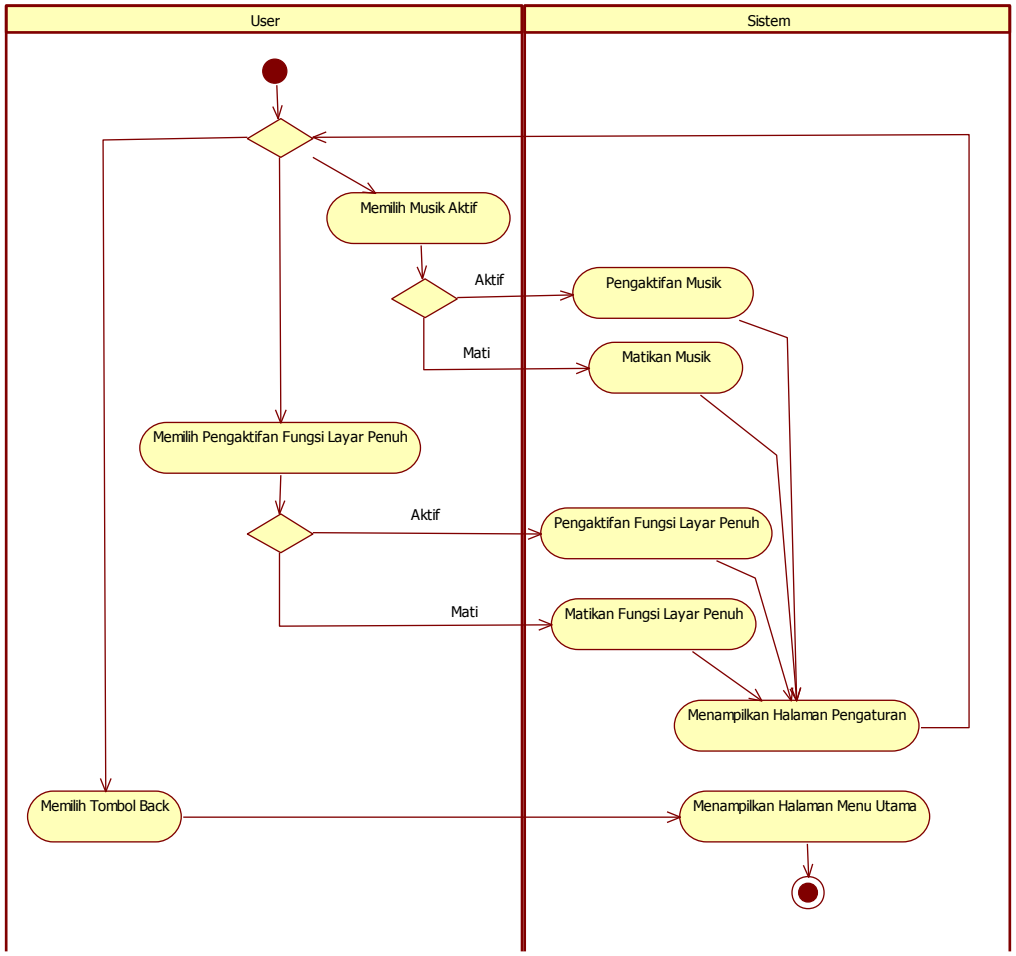
Activity diagram permainan merupakan alur aktifitas dari proses permainan pada game cegah banjir.



Gambar 3.5 Activity Diagram Permainan

3.1.7.7.2 Activity Diagram Pengaturan

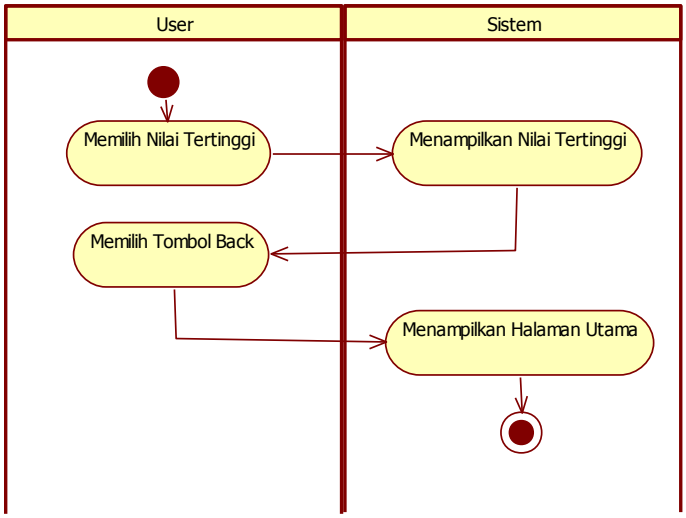
Activity diagram pengaturan merupakan alur aktifitas dari proses pengaturan pada game cegah banjir.



Gambar 3.6 Activity Diagram Pengaturan

3.1.7.7.3 Activity Diagram Nilai Tertinggi

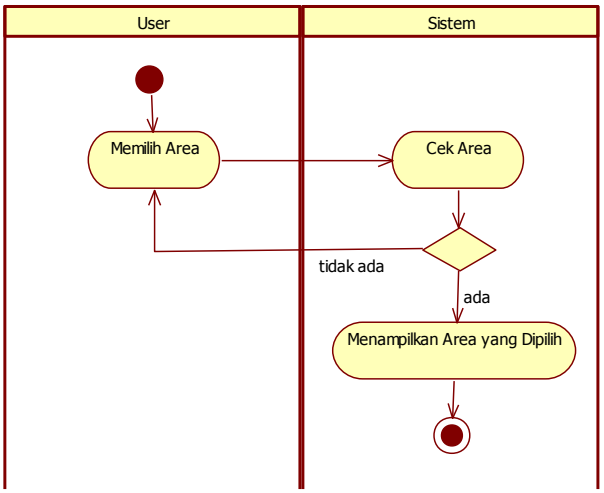
Activity diagram prestasi merupakan alur aktifitas dari proses nilai tertinggi pada game cegah banjir.



Gambar 3.7 Activity Diagram Nilai Tertinggi

3.1.7.7.4 Activity Diagram Pilih Area

Activity diagram pilih area merupakan alur aktifitas dari proses pilih area pada game cegah banjir.



Gambar 3.8 Activity Diagram Pilih Area

3.1.7.7.5 Activity Diagram Area Pegunungan

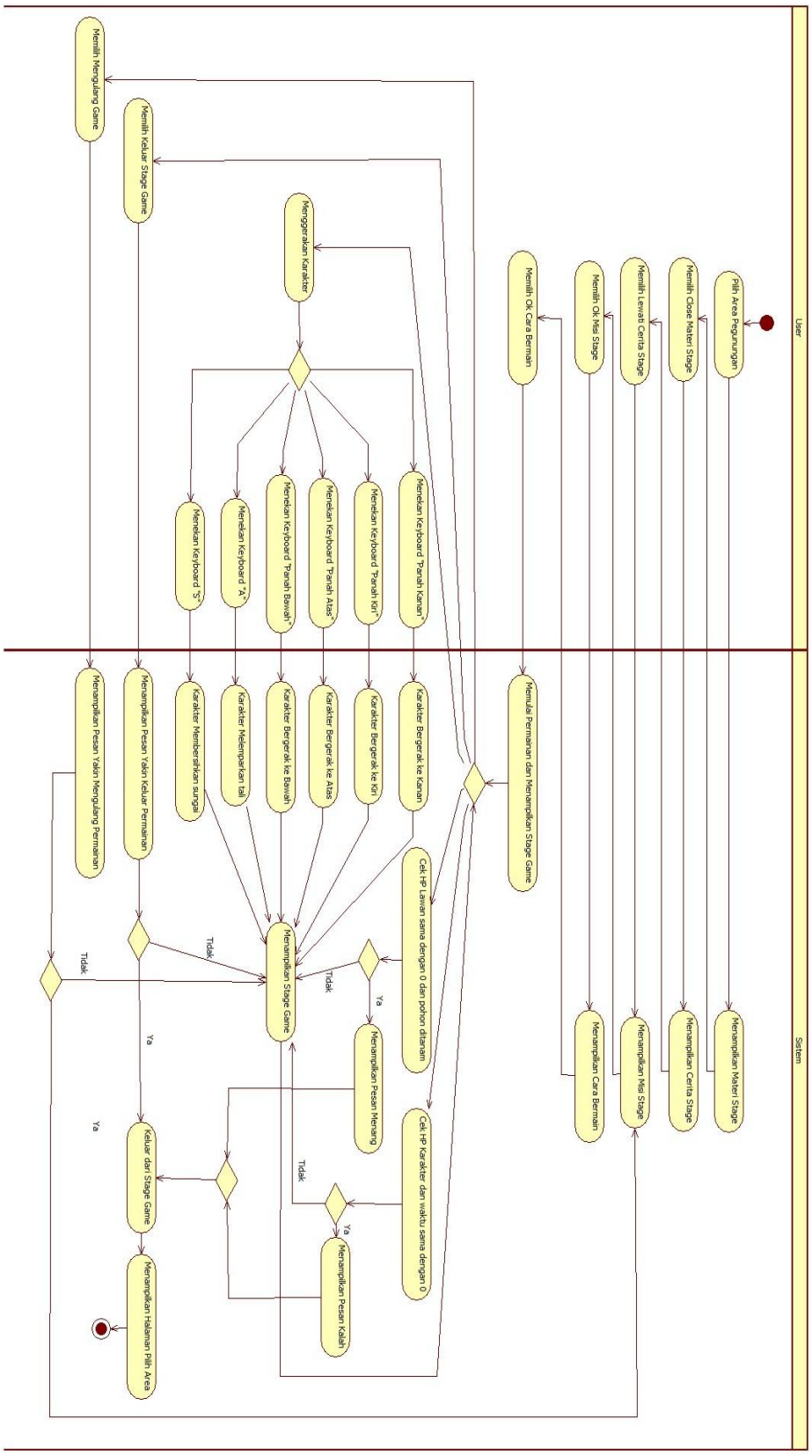
Activity diagram area pegunungan merupakan alur aktifitas dari proses mulai permainan pada area pegunungan dapat dilihat pada gambar 3.9.

3.1.7.7.6 Activity Diagram Area Bendungan

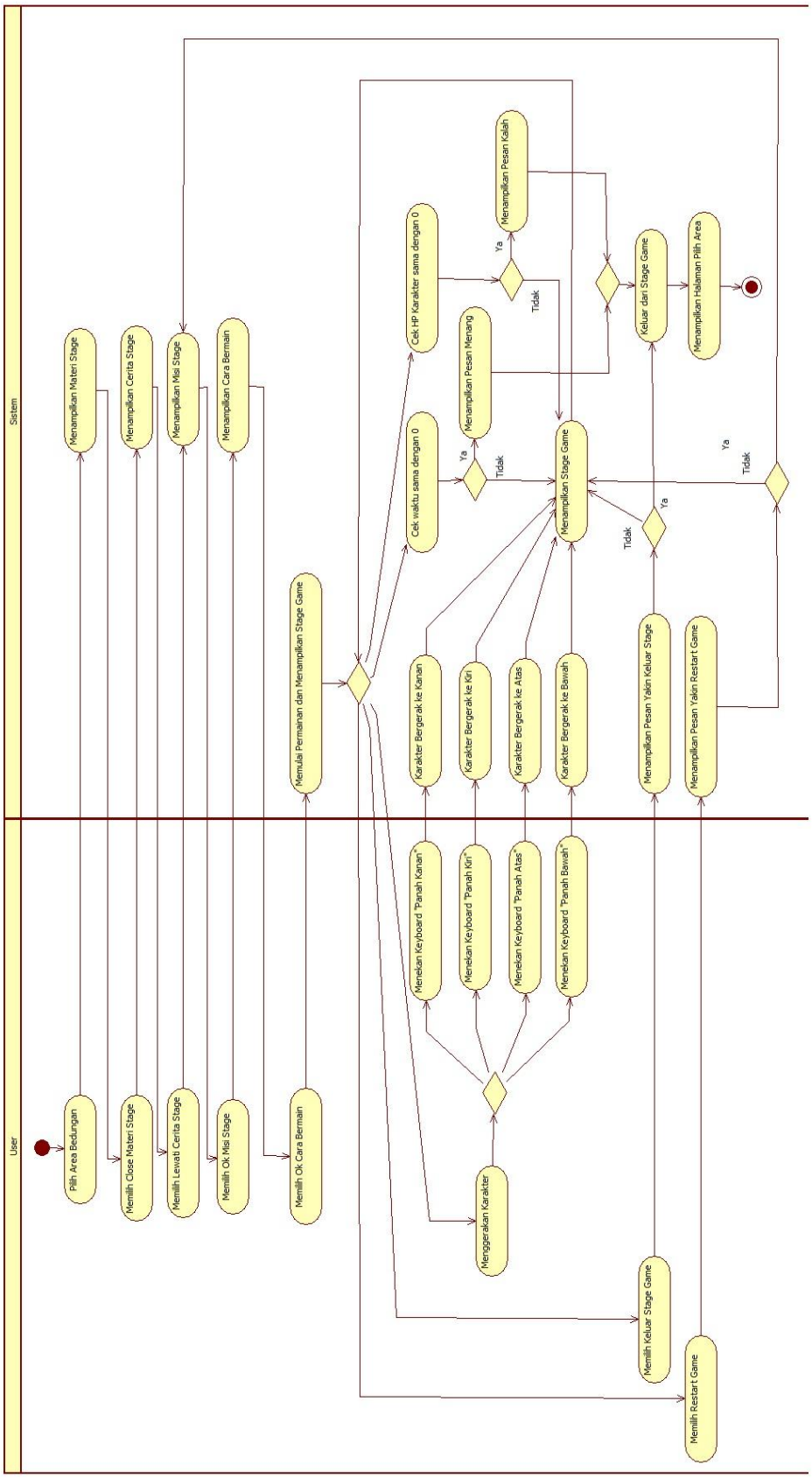
Activity diagram area bendungan merupakan alur aktifitas dari proses mulai permainan pada area bendungan dapat dilihat pada gambar 3.10.

3.1.7.7.7 Activity Diagram Area Pemukiman

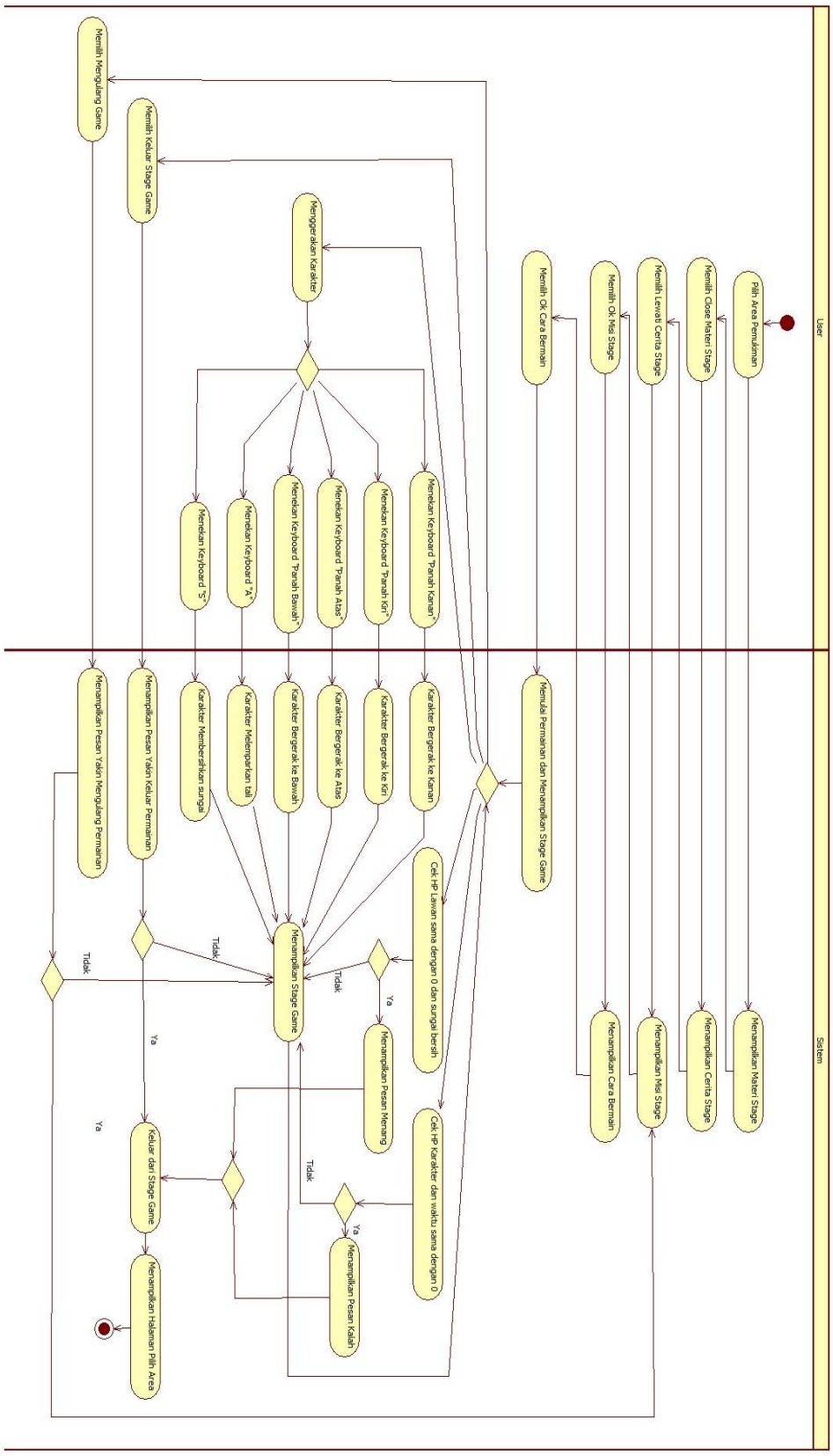
Activity diagram area pemukiman merupakan alur aktifitas dari proses mulai permainan pada area pemukiman dapat dilihat pada gambar 3.11.



Gambar 3.9 Activity Diagram Area Pegunungan



Gambar 3.10 Activity Diagram Area Bendungan



Gambar 3.11 Activity Diagram Area Pemukiman

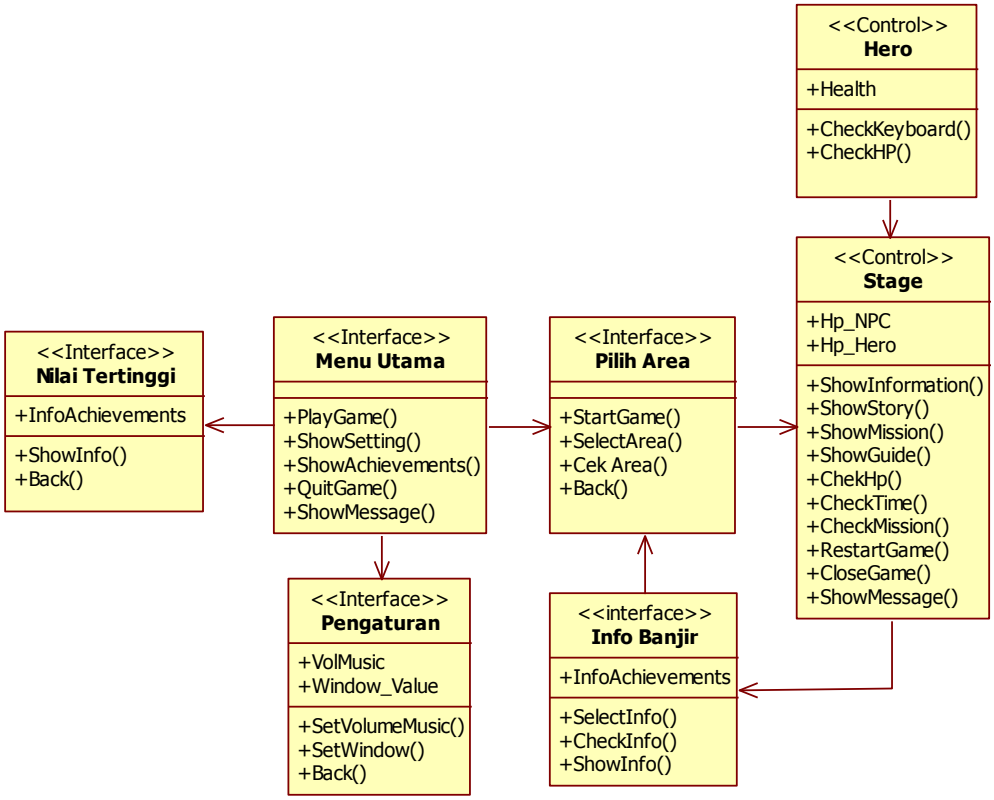
3.1.7.8 Class Diagram

Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem [8]. Kelas-kelas yang terdapat pada *game* Cegah Banjir adalah kelas menu utama, kelas pengaturan, kelas nilai tertinggi, kelas area, kelas mulai permainan dan kelas stage. Defini kelas dapat dilihat pada tabel 3.10.

Tabel 3.10 Definisi Kelas

Jenis Kelas	Nama Kelas
Boundary	Menu Utama
	Pengaturan
	Nilai Tertinggi
	Pilih Area
	Pilih Info
Control	Hero
	Stage

Kelas diagram pada *game* cegah banjir dapat dilihat pada gambar 3.12.

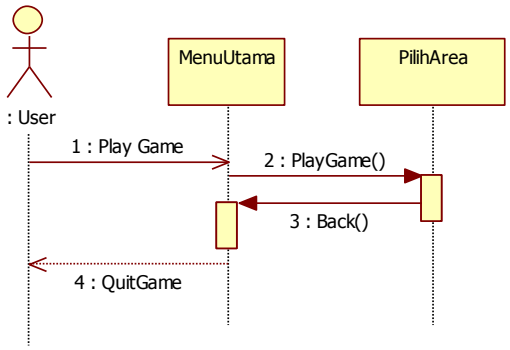


Gambar 3.12 Class Diagram Game Cegah Banjir

3.1.7.9 Sequence Diagram

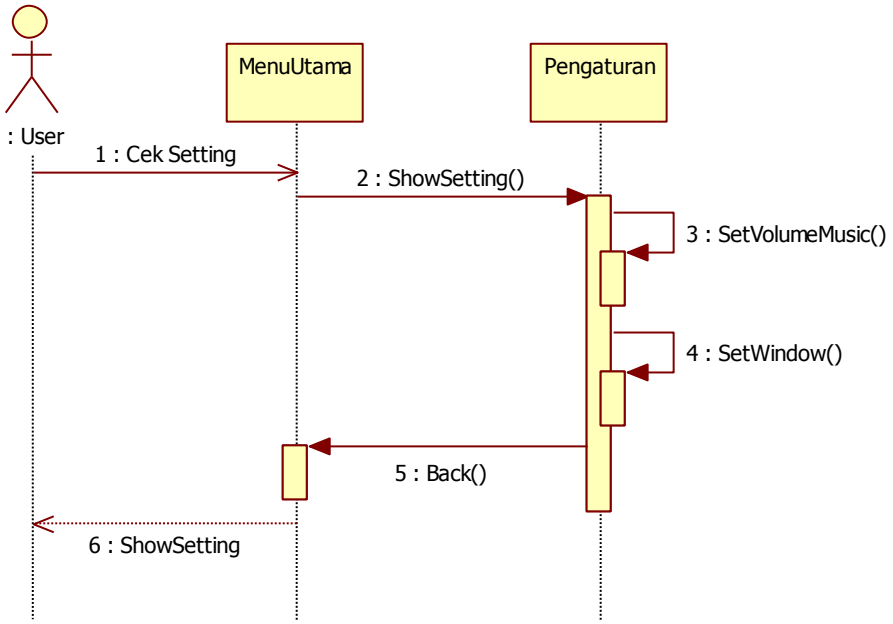
Diagram sekuen atau *sequence diagram* menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek [8].

3.1.7.9.1 Sequence Diagram Permainan



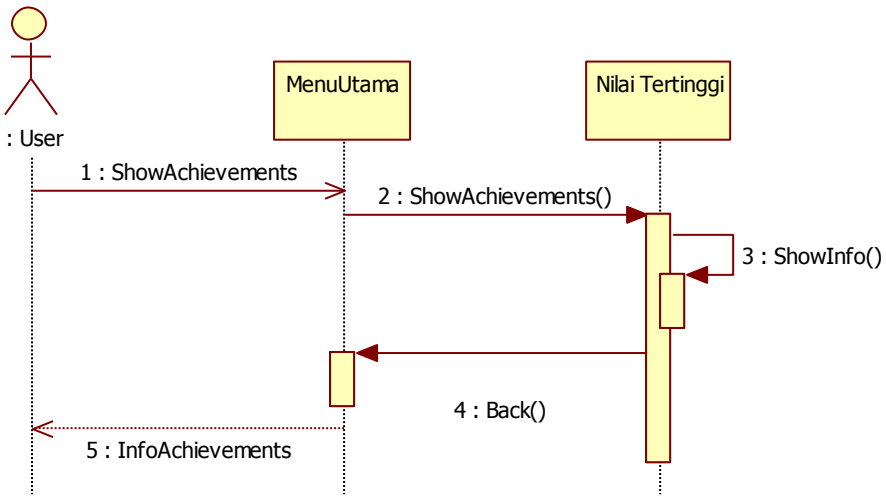
Gambar 3.13 Sequence Diagram Permainan

3.1.7.9.2 Sequence Diagram Pengaturan



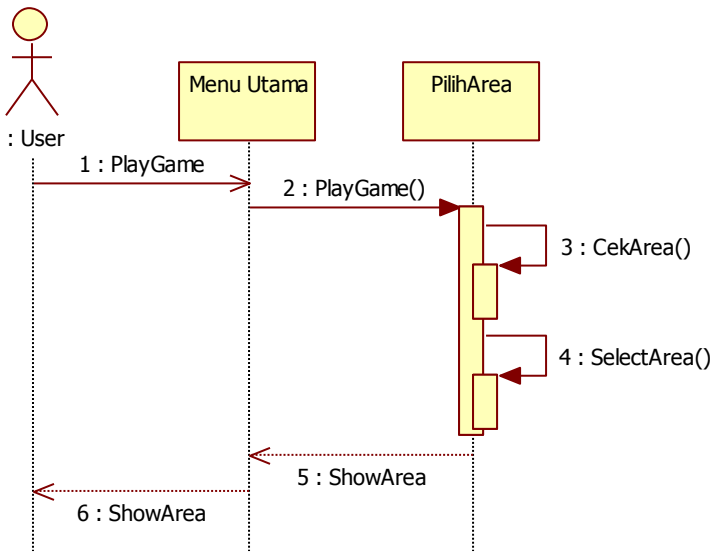
Gambar 3.14 Sequence Diagram Pengaturan

3.1.7.9.3 Sequence Diagram Nilai Tertinggi



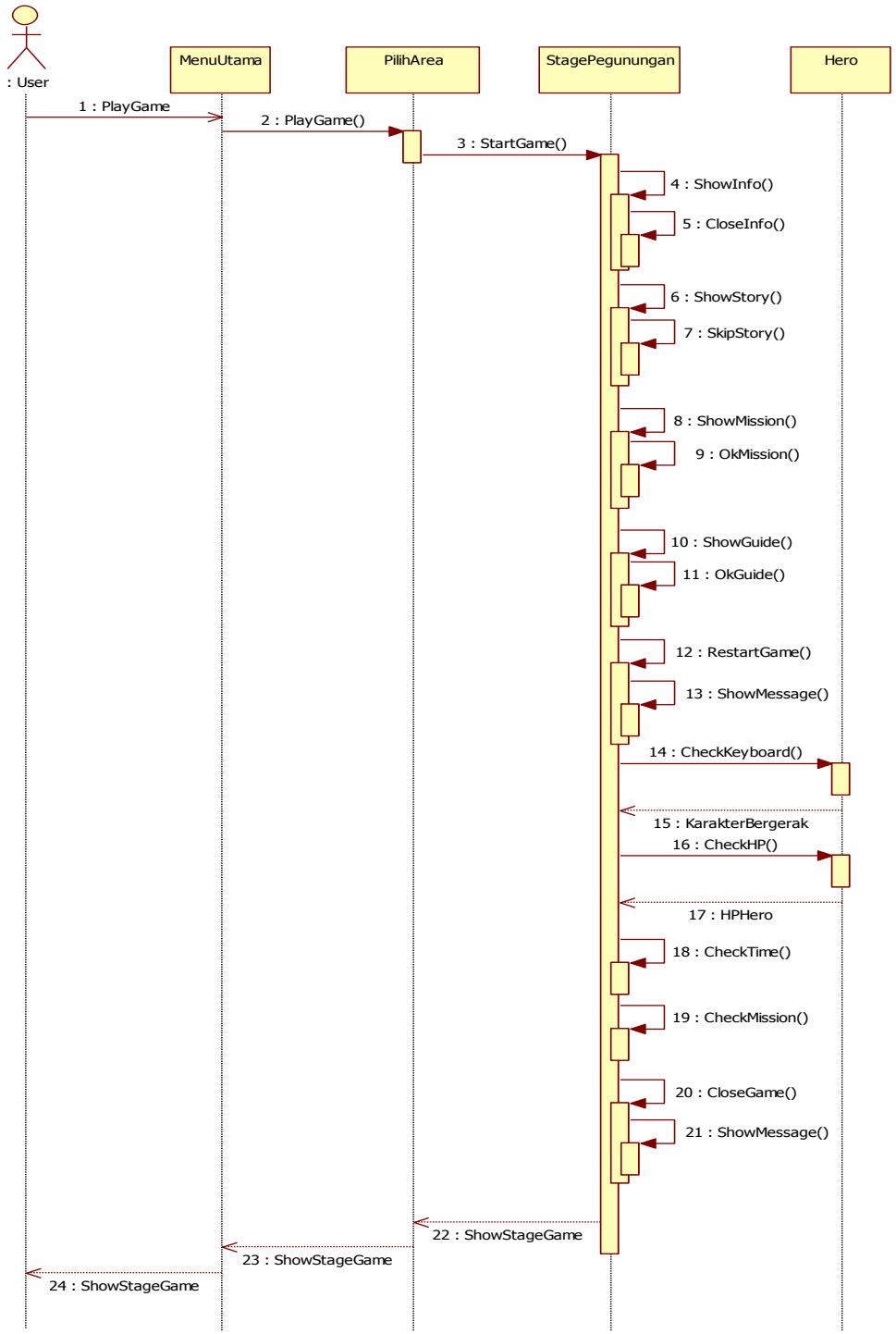
Gambar 3.15 Sequence Diagram Nilai Tertinggi

3.1.7.9.4 Sequence Diagram Pilih Area



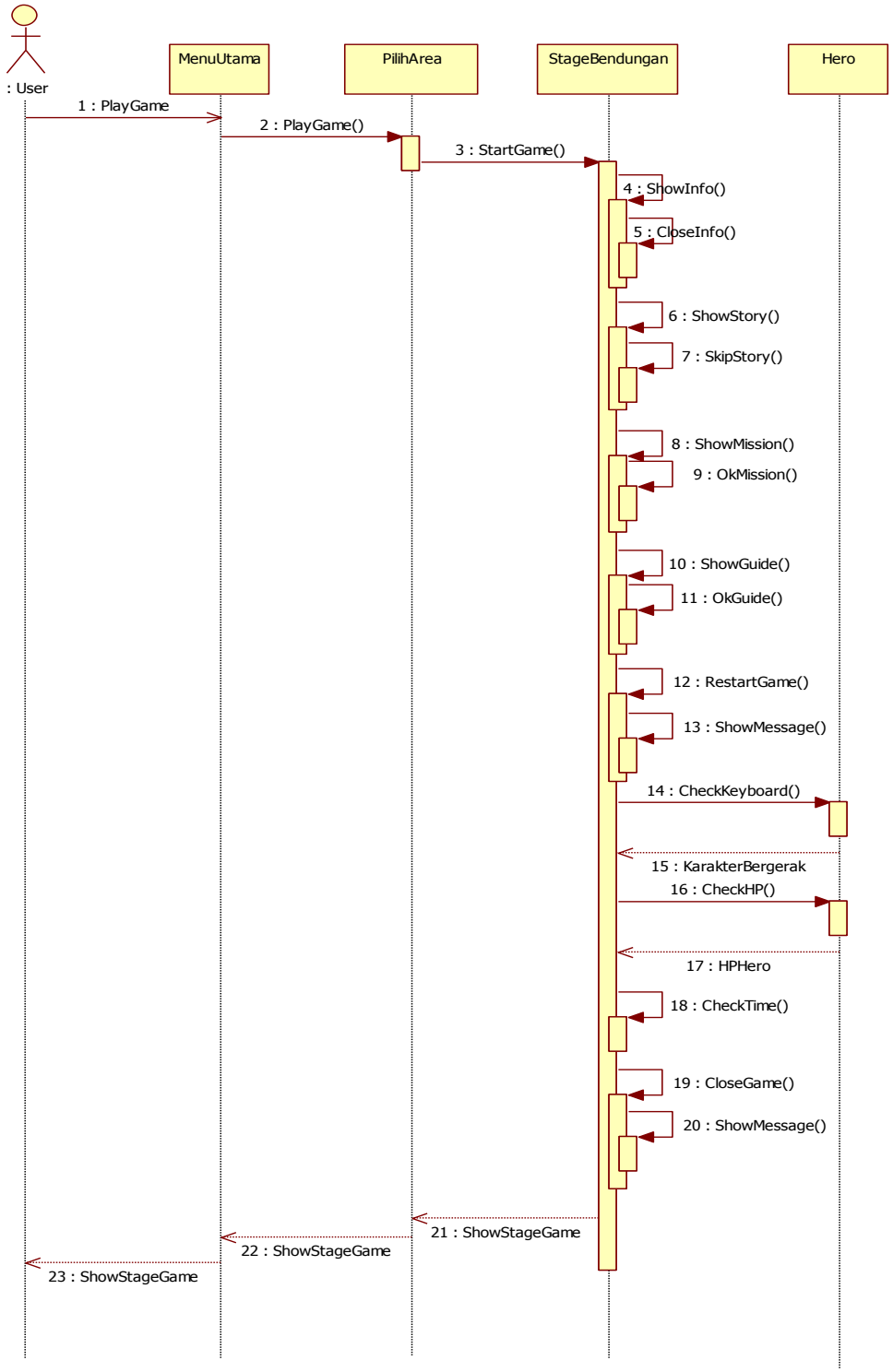
Gambar 3.16 Sequence Diagram Pilih Area

3.1.7.9.5 Sequence Diagram Area Pegunungan



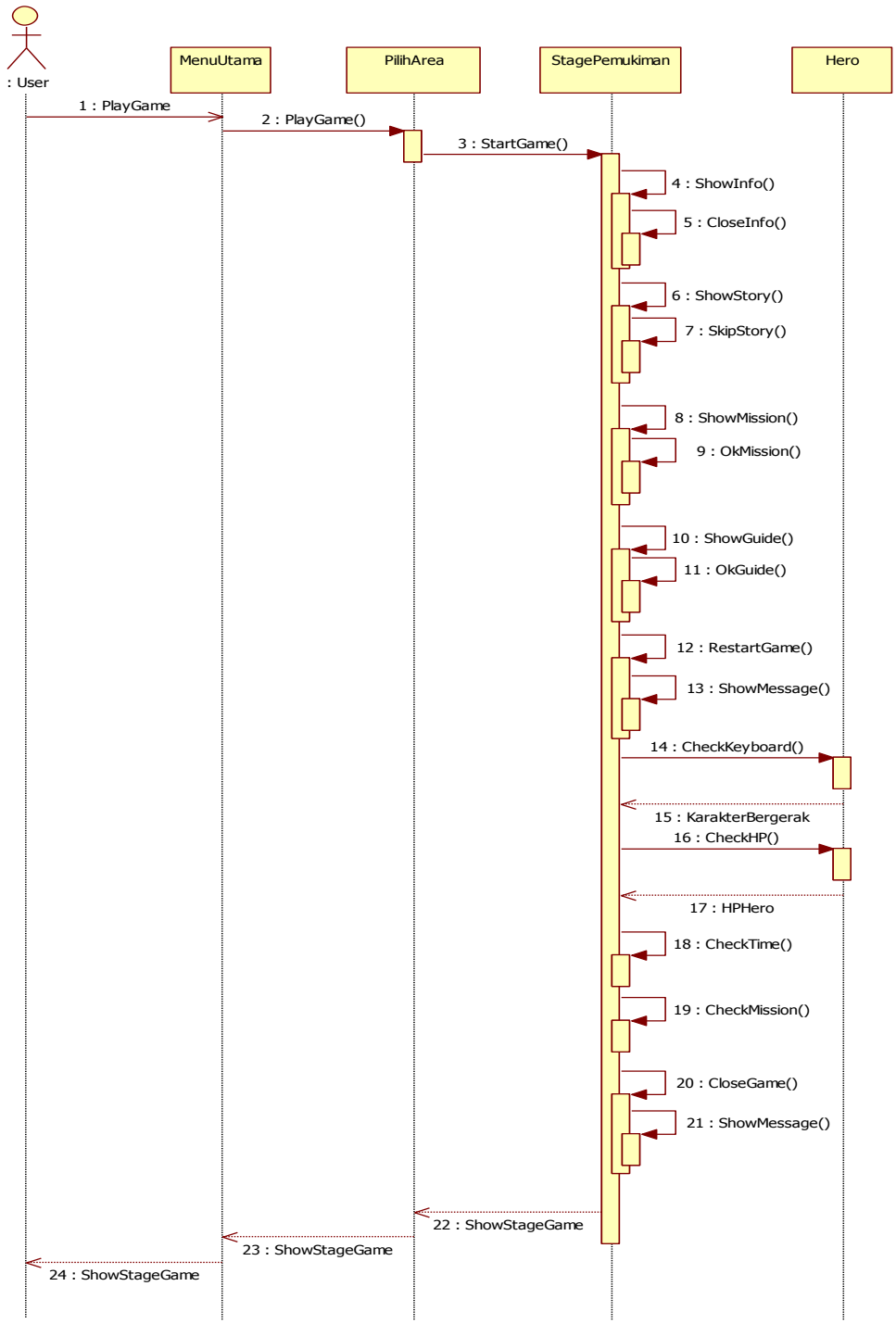
Gambar 3.17 Sequence Diagram Area Pegunungan

3.1.7.9.6 Sequence Diagram Area Bendungan



Gambar 3.18 Sequence Diagram Area Bendungan

3.1.7.9.7 Sequence Diagram Area Pemukiman



Gambar 3.19 Sequence Diagram Area Pemukiman

3.2 Perancangan

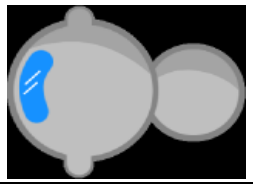
Pada perancangan akan mencakup pengenalan karakter, storyboard, perancangan antar muka, perancangan pesan, jaringan semantik dan perancangan method.

3.2.1 Pengenalan Karakter

Pengenalan karakter merupakan pembahasan mengenai karakter-karakter yang terlibat pada game Cegah Banjir. Karakter-karakter tersebut dapat terlihat pada tabel 3.14.

Tabel 3.14 Pengenalan Karakter

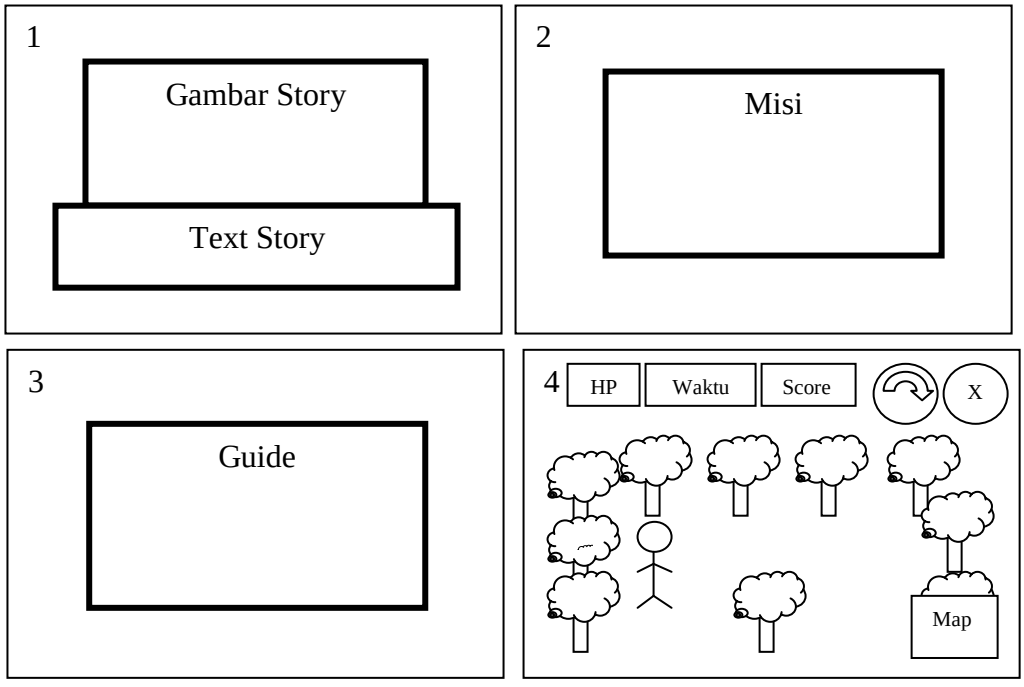
No.	Nama	Gambar	Keterangan
1.	Ara		Karakter yang dimainkan oleh *user*. Memiliki kemampuan untuk menangkap pelaku penebang pohon dan orang yang membuang sampah sembarangan.
2.	Pelaku yang menebang Pohon		Merupakan karakter hero lawan. Memiliki kemampuan untuk menebang pohon dan menyerang hero utama.

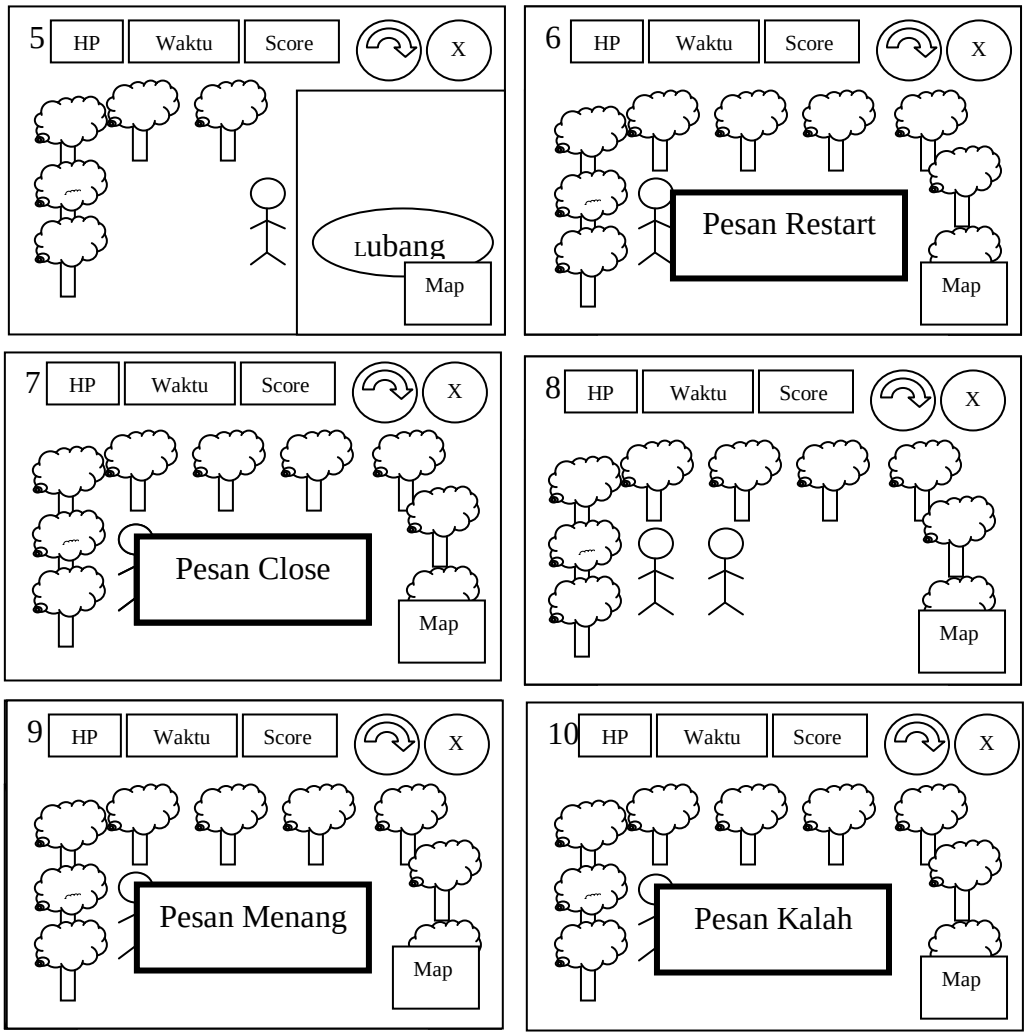
3.	Pelaku yang membuang sampah sembarangan		Merupakan karakter hero lawan. Memiliki kemampuan untuk membuang sampah sembarangan dan menyerang hero utama.
4.	Robot Pengambil Sampah		Merupakan karakter pembantu saat di level 2. Memiliki kemampuan untuk mengambil sampah di area bendungan.

3.2.2 Storyboard

Storyboard merupakan sketsa atau rancangan gambar dari sebuah permainan, dimana fungsinya menjadikan ide atau persepsi agar mudah dipahami oleh orang lain. Terdapat tiga *storyboard* pada game Cegah Banjir, yaitu storyboard area pegunungan, area bendungan dan area pemukiman.

3.2.2.1 Storyboard Area Pegunungan





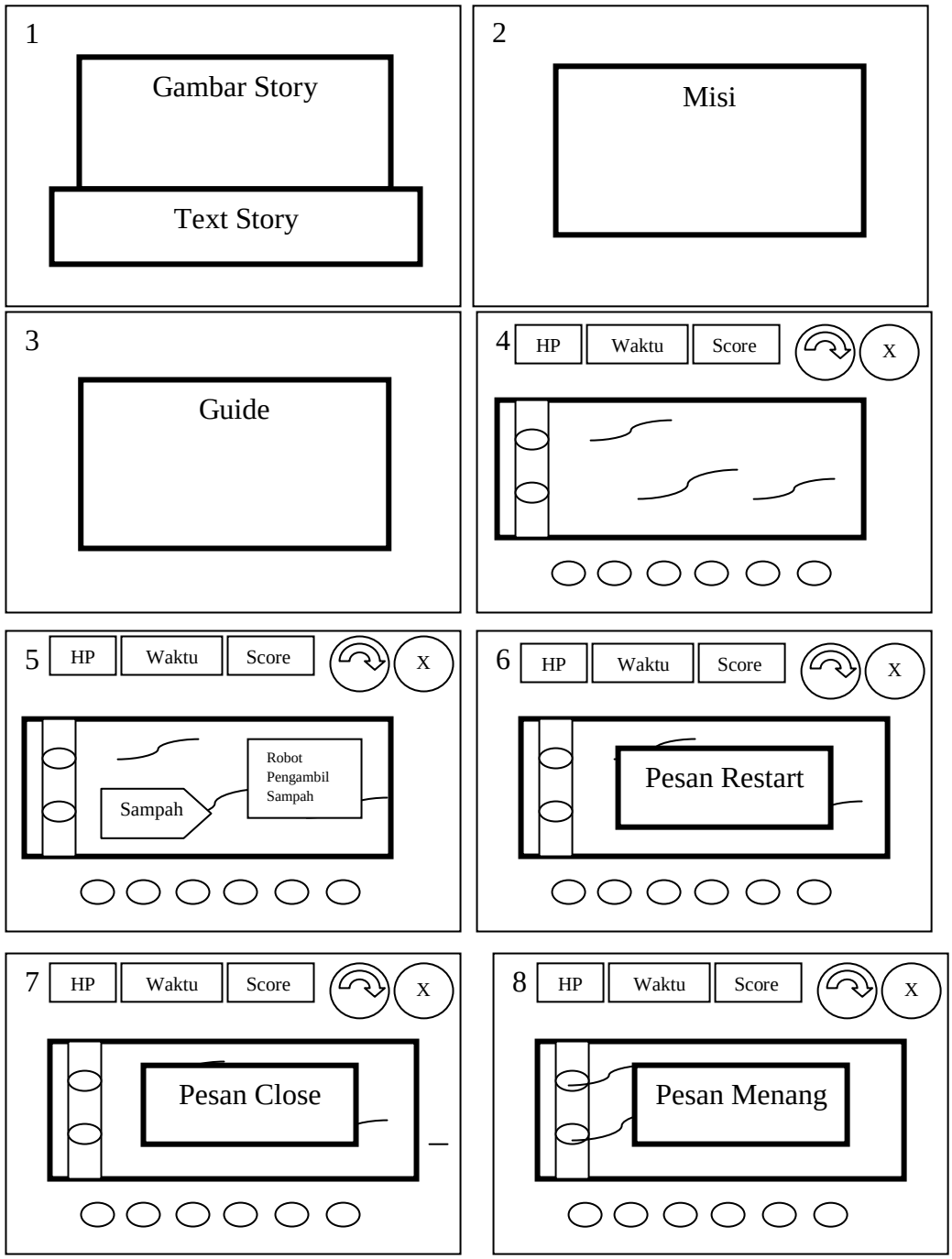
Gambar 3.20 Storyboard Area Pegunungan

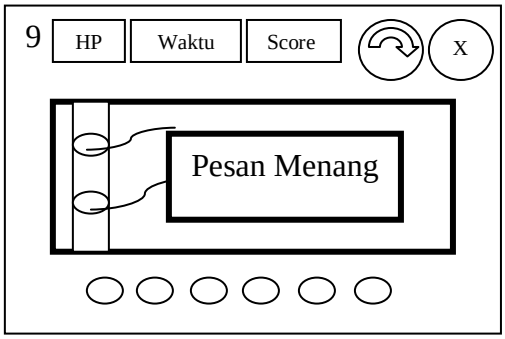
Penjelasaan gambar dari story board area pegunungan adalah sebagai berikut.

1. Gambar pertama adalah story untuk area pegunungan atau stage awal.
2. Setelah story selesai, maka akan menampilkan misi yang harus ditempuh.
3. Menampilkan guide atau panduan untuk cara memainkan game.
4. Gambar awal di area pegunungan.
5. Gambar dimana karakter hero menanam bibit pohon
6. Gambar dimana saat user memilih tombol mengulang
7. Gambar disaat user memilih tombol keluar
8. Gambar ketika hero terkena serangan.

9. Ketika hero lawan tertangkap dan telah menyelesaikan misi menanam pohon, muncul pesan menang.
10. Ketika HP hero telah habis, muncul pesan kalah.

3.2.2.2 Storyboard Area Bendungan



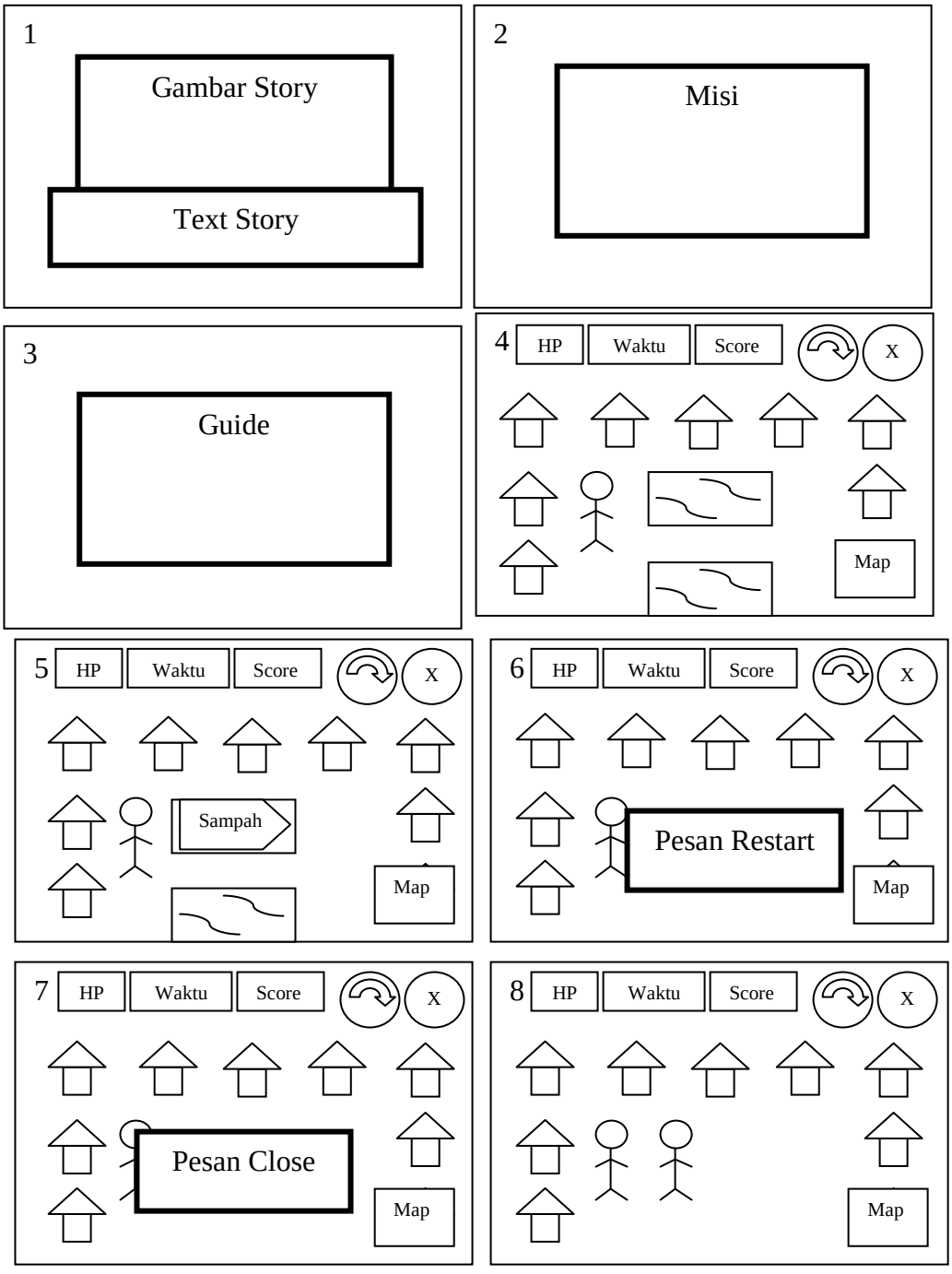


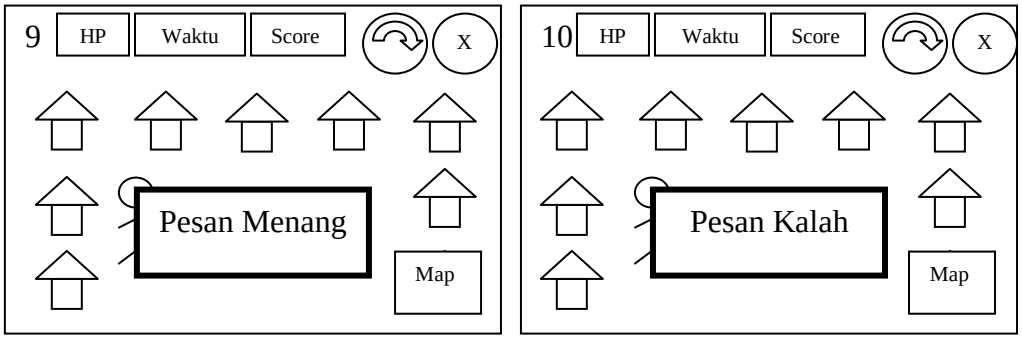
Gambar 3.21 Storyboard Area Bendungan

Penjelasaan gambar dari story board area bendungan adalah sebagai berikut.

1. Gambar pertama adalah story untuk area bendungan atau stage kedua.
2. Setelah story selesai, maka akan menampilkan misi yang harus ditempuh.
3. Menampilkan guide atau panduan untuk cara memainkan game.
4. Gambar awal di area bendungan.
5. Gambar dimana user menggunakan robot untuk mengambil sampah.
6. Gambar dimana saat user memilih tombol mengulang.
7. Gambar disaat user memilih tombol keluar.
8. Ketika waktu habis dan HP $\neq 0$, muncul pesan menang.
9. Ketika HP = 0, muncul pesan kalah.

3.2.2.3 Storyboard Area Pemukiman





Gambar 3.22 Storyboard Area Pemukiman

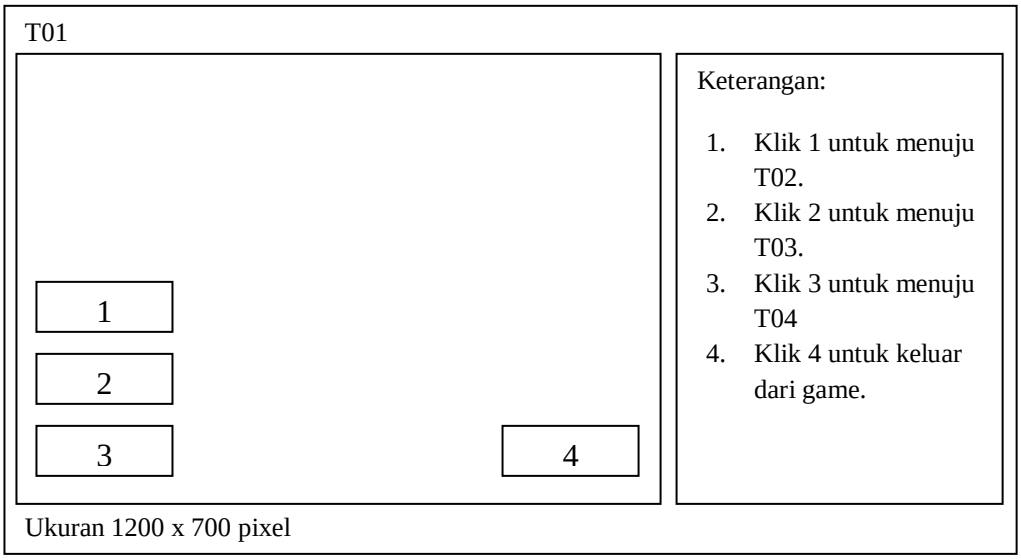
Penjelasaan gambar dari story board area bendungan adalah sebagai berikut.

1. Gambar pertama adalah story untuk area pemukiman atau stage ketiga.
2. Setelah story selesai, maka akan menampilkan misi yang harus ditempuh.
3. Menampilkan guide atau panduan untuk cara memainkan game.
4. Gambar awal di area pemukiman.
5. Gambar dimana user membersihkan sungai.
6. Gambar dimana saat user memilih tombol mengulang.
7. Gambar disaat user memilih tombol keluar.
8. Gambar ketika user melawan pelaku yang membuang sampah sembarangan.
9. Ketika hero lawan tertangkap dan misi membersihkan sungai selesai, muncul pesan menang.
10. Ketika HP hero telah habis, muncul pesan kalah.

3.2.3 Perancangan Antar Muka

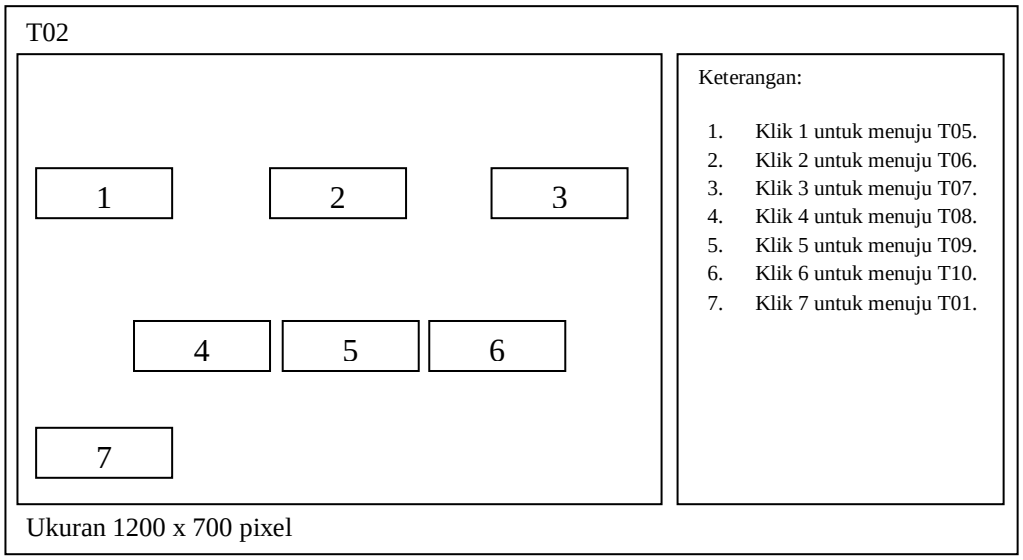
Perancangan antar muka merupakan gambaran untuk mewakili aplikasi yang akan dibangun, sehingga mempermudah dalam mengimplementasikan disaat pembuatannya. Berikut ini adalah antar muka dari game Cegah Banjir.

3.2.3.1 Perancangan Antar Muka Menu Utama Game Cegah Banjir



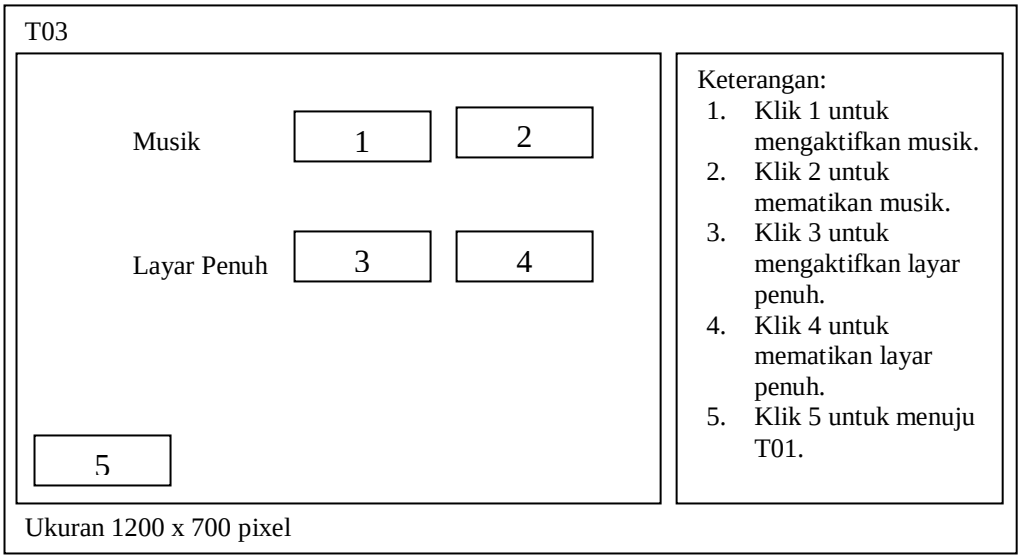
Gambar 3.23 Antar Muka Menu Utama

3.2.3.2 Perancangan Antar Muka Pilih Area



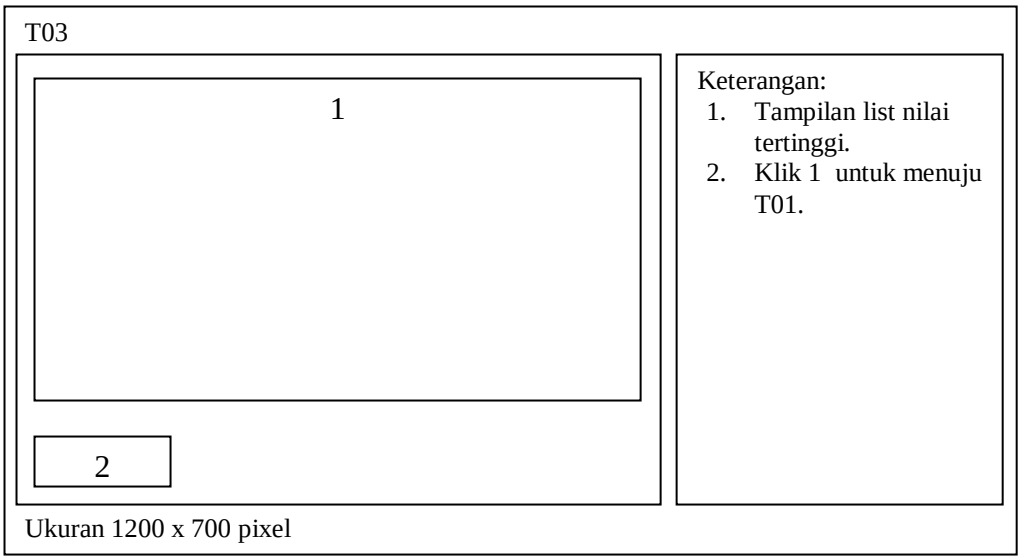
Gambar 3.24 Antar Muka Pilih Area

3.2.3.3 Perancangan Antar Muka Pengaturan



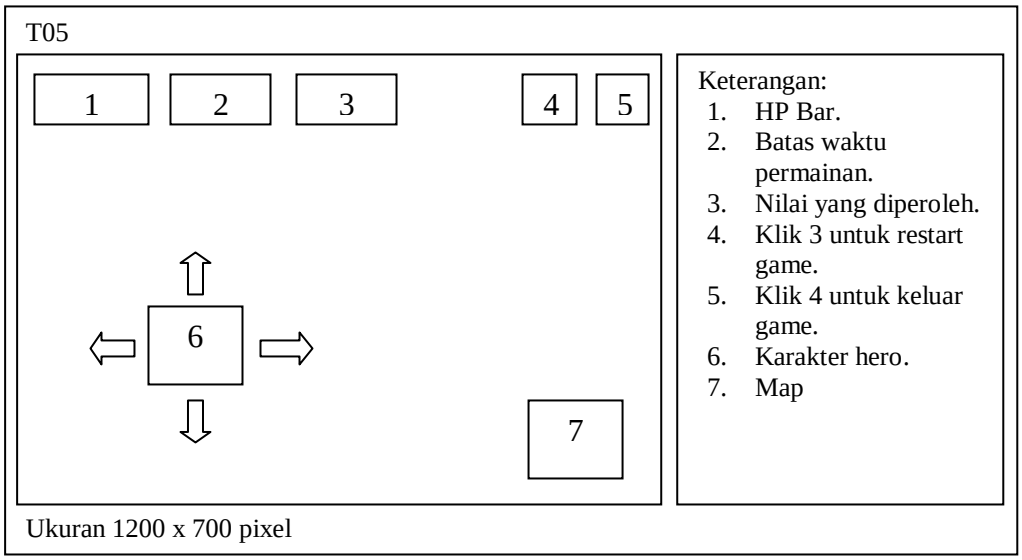
Gambar 3.25 Antar Muka Pengaturan

3.2.3.4 Perancangan Antar Muka Nilai Tertinggi



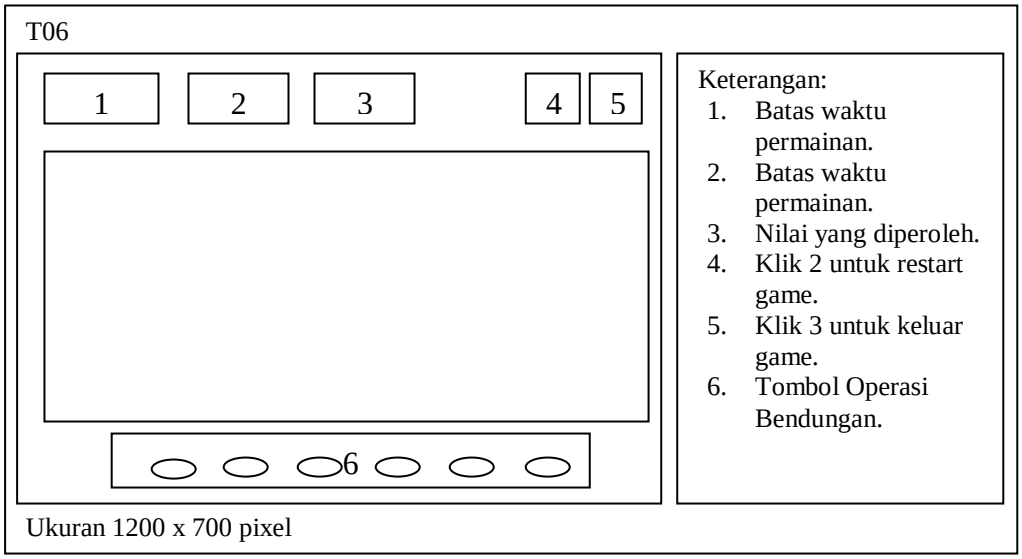
Gambar 3.26 Antar Muka Nilai Tertinggi

3.2.3.5 Perancangan Antar Muka Stage Game Area Pegunungan



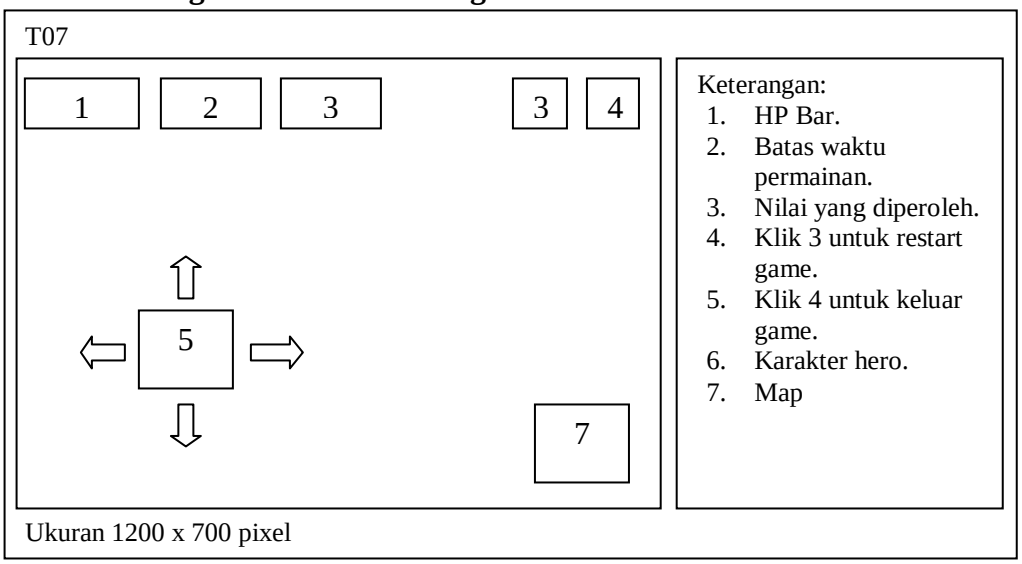
Gambar 3.327 Antar Muka Stage Area Pegunungan

3.2.3.6 Perancangan Antar Muka Stage Game Area Bendungan



Gambar 3.28 Antar Muka Stage Area Bendungan

3.2.3.7 Perancangan Antar Muka Stage Game Area Pemukiman

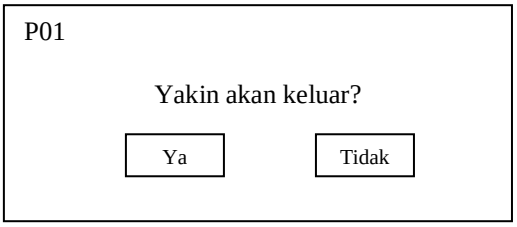


Gambar 3.29 Antar Muka Stage Area Pemukiman

3.2.4 Perancangan Pesan

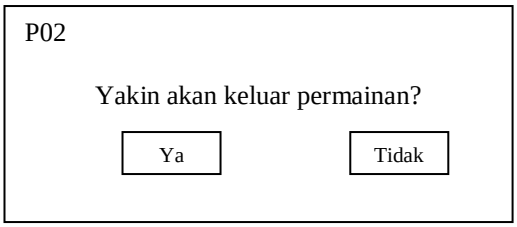
Perancangan pesan menampilkan gambaran tampilan dialog konfirmasi yang terdapat dalam *game* cegah banjir.

3.2.4.1 Pesan Keluar Game Cegah Banjir



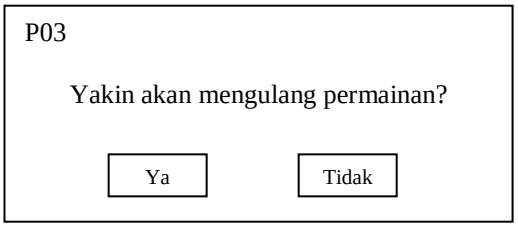
Gambar 3.30 Pesan Keluar Game Cegah Banjir

3.2.4.2 Pesan Keluar Stage Permainan Game Cegah Banjir



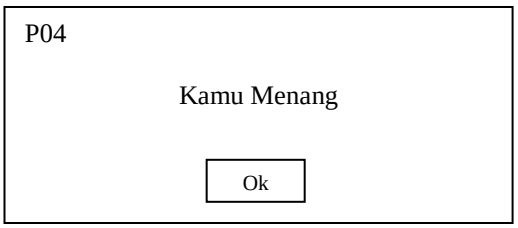
Gambar 3.31 Pesan Keluar Stage Permainan

3.2.4.3 Pesan Mengulang Stage Game



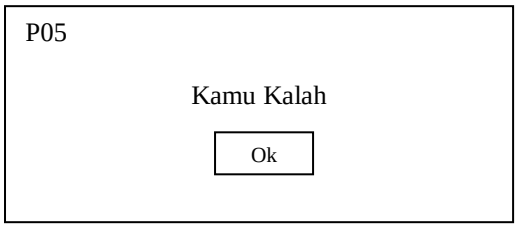
Gambar 3.32 Pesan Mengulang Stage Permainan

3.2.4.4 Pesan Menang Permainan



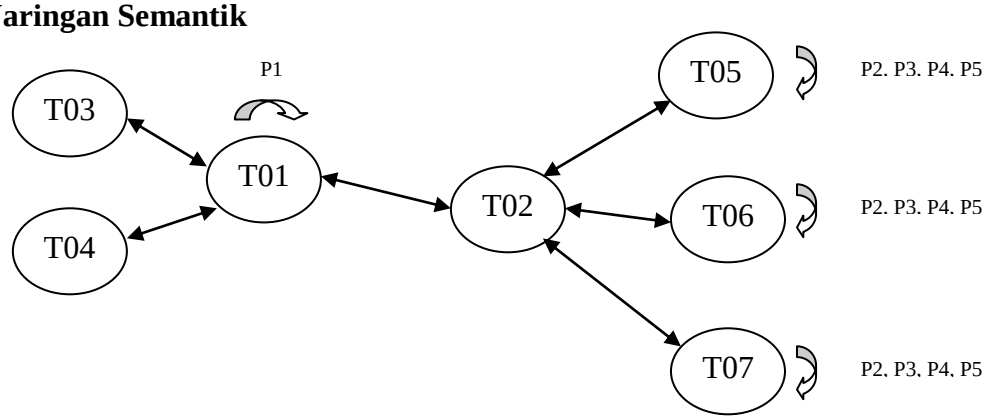
Gambar 3.33 Pesan Menang Permainan

3.2.4.5 Pesan Kalah Permainan



Gambar 3.34 Pesan Kalah Permainan

3.2.5 Jaringan Semantik



Gambar 3.35 Jaringan Semantik

3.2.6 Perancangan Method

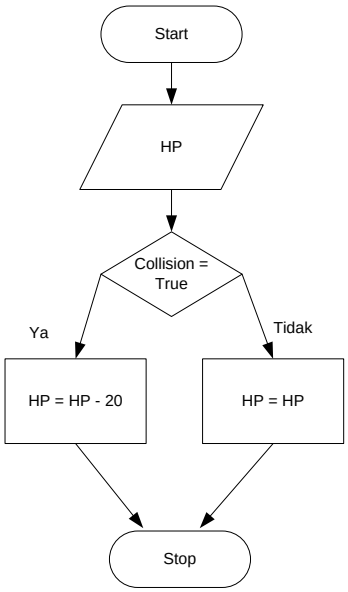
Perancangan method ini digunakan untuk menetapkan detail algoritma yang digunakan. Dalam perancangan method game Cegah Banjir ini menggunakan algoritma dan flowchart.

3.2.6.1 Method Pengurangan Health Point Oleh Hero Lawan

Rincian proses dari method pengurangan *hit point* oleh *hero* lawan adalah seperti berikut:

1. Start
2. Baca nilai health point (HP)
3. Jika *hero* collision dengan *hero* lawan

4. Health Point = Health Point – 20
5. Jika tidak Health Point = Health Point
6. Stop



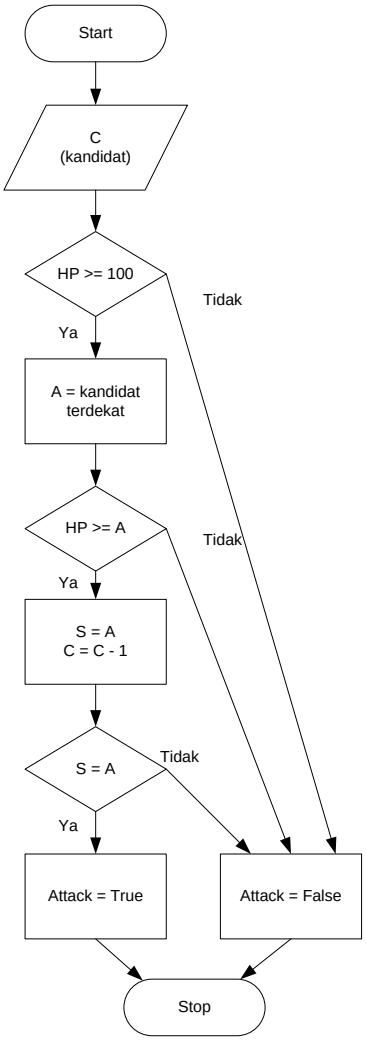
Gambar 3.36 Flowchart Method Pengurangan Health Point Oleh Hero Lawan

3.2.6.2 Method Hero Lawan Menggunakan Algoritma Greedy

Method ini digunakan saat hero lawan memutuskan menyerang Ara atau menlanjutkan perusakan alam. Perusakan alam diterapkan pada penebang pohon yang melakukan penebangan pohon dan pembuang sampah yang membuang sampah sembarangan. Rincian prosesnya sebagai berikut:

1. Start
2. Read semua kandidat
3. Jika HP lebih besar dari 100
4. A merupakan kandidat paling dekat
5. Jika tidak attack = false
6. Jika HP lebih besar dari kadidat A
7. A menjadi solusi (S) dan jumlah kandidat berkurang 1
8. Jika tidak attack = false
9. Jika S = A
10. Attack = true
11. Jika tidak attack = false

12. Stop



Gambar 3.37 Flowchart Method Algoritma Greedy