BAB 4

IMPLEMENTASI DAN PENGUJIAN

4.1 Implementasi

Pada bagian tahap implementasi ini merupakan terjemahan perancangan yang berdasarkan hasil analisis pada bab sebelumnya kedalam bahasa pemrograman yang dapat dimengerti oleh komputer. Bahasa pemrograman dalam sistem ini menggambarkan Game Maker Language, kemudian GIMP 2.8 sebagai tools untuk desain gambar.

4.2 Perangkat Pendukung yang Digunakan

Dalam proses pembuatan aplikasi ini, tentunya menggunakan perangkat keras (*hardware*) dan perangkat lunak (*software*). Berikut merupakan penjelasan dari perangkat keras dan perangkat lunak yang digunakan tersebut.

4.2.1 Perangkat Keras (Hardware)

Perangkat keras yang digunakan selama pembangunan aplikasi ini memiliki speseifikias sebagai berikut:

1. CPU : Intel Core (TM) i3-2350M CPU @2.30GHz
2. Memory : 2 GB
3. Keyboard : 1 buah
4. Mouse : 1 buah
5. Speaker : Ya

4.2.2 Perangkat Lunak (Software)

Perangkat lunak yang digunakan untuk menjalankan game cegah banjir ini adalah sistem operasi windows 7.

4.3 Implementasi Aplikasi

Untuk memastikan game cegah banjir dapat berjalan dengan baik, maka langkah-langkah yang harus dilakukan adalah sebagai berikut:

1. Klik file execute (.exe) game cegah banjir.
2. Game sudah dapat dimainkan.

4.4 Implementasi Antar Muka

4.4.1 Sisi User

1. Implementasi Antar Muka Menu Utama Game Cegah Banjir



Gambar 4.1 Tampilan Menu Utama

Tampilan ini merupakan tampilan awal game cegah banjir dimana terdapat empat pilihan menu yang dapat digunakan yaitu permainan, pengaturan, nilai tertinggi dan keluar.

2. Implementasi Antar Muka Menu Pilih Area



Gambar 4.2 Tampilan Menu Pilih Area

Gambar 4.2 merupakan tampilan dari menu pilih area game cegah banjir, dimana pada menu ini pemain dapat memilih map yang ingin digunakan..

3. Implementasi Antar Muka Pengaturan



Gambar 4.3 Tampilan Menu Pengaturan

Tampilan ini merupakan tampilan menu pengaturan game cegah banjir, dimana pada menu ini adalah untuk mengatur musik dan ukuran layar.

4. Implementasi Antar Muka Nilai Tertinggi



Gambar 4.4 Tampilan Menu Nilai Tertinggi

Gambar 4.4 merupakan tampilan menu nilai tertinggi game cegah banjir, dimana pada menu ini terdapat nilai yang telah diperoleh disaat permainan berlangsung.

5. Implementasi Antar Muka Stage Game

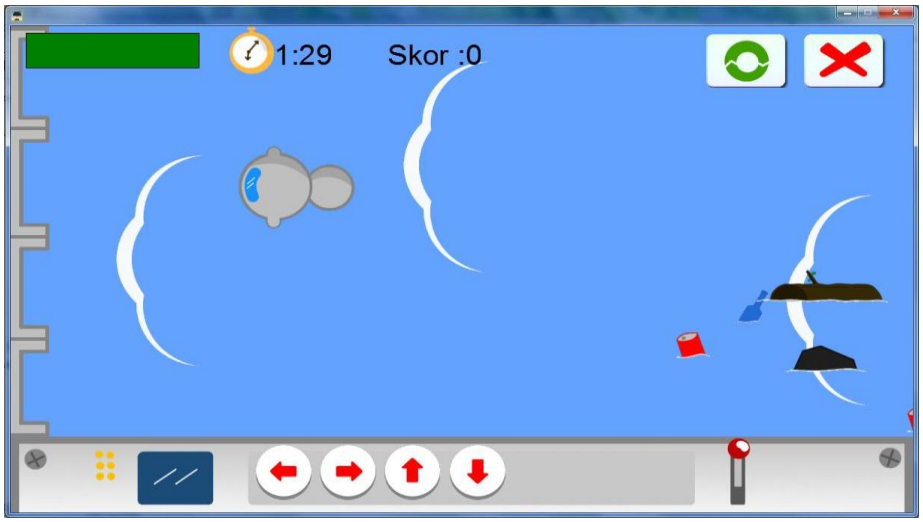
a. Tampilan Area Pegunungan



Gambar 4.5 Tampilan Area Pegunungan

Tampilan ini merupakan tampilan pada area pertama dari game cegah banjir. Pada map ini berlatarkan hutan sebagai backgroundnya.

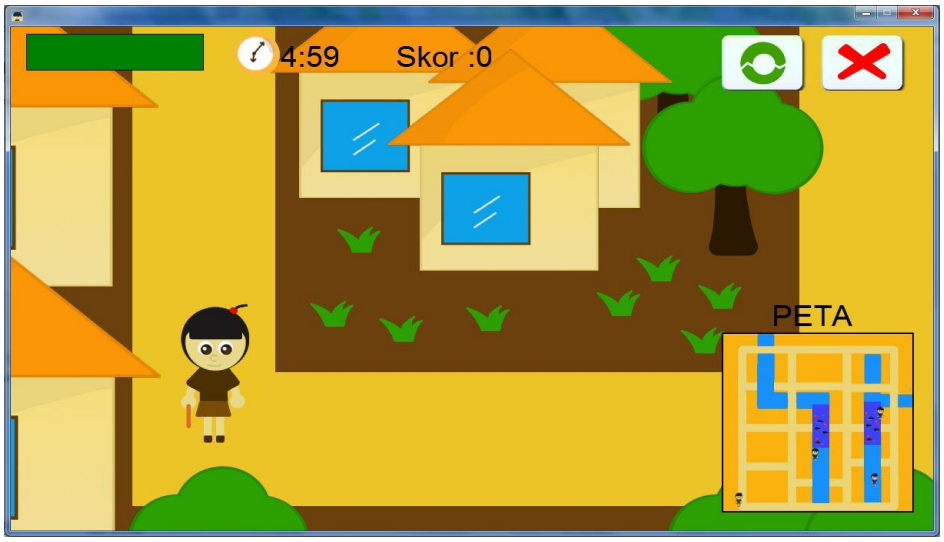
b. Tampilan Area Bendungan



Gambar 4.6 Tampilan Area Bendungan

Gambar 4.6 merupakan tampilan pada area kedua dari game cegah banjir. Pada map ini berlatarkan sungai yang memasuki bendungan sebagai backgroundnya.

c. Tampilan Area Pemukiman



Gambar 4.7 Tampilan Area Pemukiman

Tampilan ini merupakan tampilan pada area ketiga dari game cegah banjir. Pada map ini berlatarkan pemukiman penduduk dan sungai kecil sebagai backgroundnya.

6. **Implementasi Pesan Keluar Game**



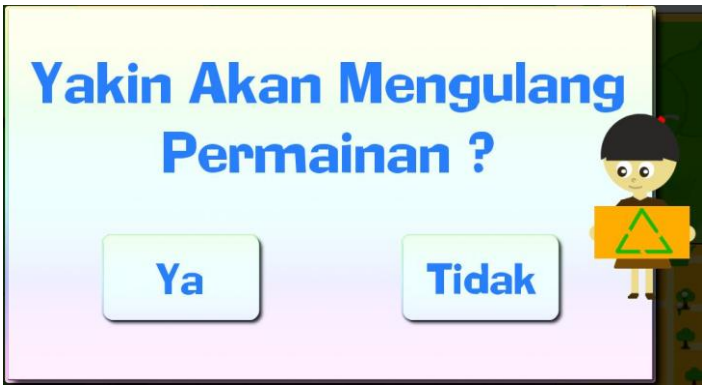
Gambar 4.8 Tampilan Pesan Keluar Game

7. **Implementasi Pesan Keluar Stage Game**



Gambar 4.9 Tampilan Pesan Keluar Stage Game

8. **Implementasi Pesan Mengulang Stage Game**



Gambar 4.10 Tampilan Pesan Mengulang Stage Game

9. Implementasi Pesan Menang Permainan



Gambar 4.11 Tampilan Pesan Menang Permainan

10. Implementasi Pesan Kalah Permainan



Gambar 4.12 Tampilan Pesan Kalah Permainan

4.4.2 Sisi Developer



Gambar 4.13 Tampilan Pengembangan Game

Gambar 4.13 merupakan tampilan pengembangan game dari game Cegah Banjir.

4.5 Pengujian Sistem

Pengujian sistem merupakan hal terpenting yang dilakukan untuk menemukan kekurangan atau kesalahan pada perangkat lunak yang diuji. Pengujian bermaksud untuk mengetahui perangkat lunak yang dibuat sudah memenuhi kriteria yang sesuai dengan tujuan perancangan perangkat lunak tersebut.

4.5.1 Pengujian Alpha (Fungsional)

Pengujian *fungsional* yang digunakan untuk menguji sistem yang baru adalah metode *alpha*. Pengujian *alpha* berfokus pada persyaratan fungsional perangkat lunak.

4.5.1.1 Rencana Pengujian

Rencana pengujian yang akan dilakukan pada aplikasi ini yaitu pengujian black box dan pengujian white box.

a. Pengujian Black Box

Rencana pengujian yang akan dilakukan pada pengujian *black box* ini selengkapnya terlihat pada tabel berikut ini.

Tabel 4.1 Rencana Pengujian Black Box

No	Kelas Uji	Detail Pengujian	Jenis Uji
1.	Menu Utama	Memilih tombol permainan	Black Box
		Memilih tombol pengaturan	Black Box
		Memilih tombol nilai tertinggi	Black Box
		Memilih tombol keluar	Black Box
2.	Menu Pengaturan	Memilih tombol aktif untuk musik	Black Box

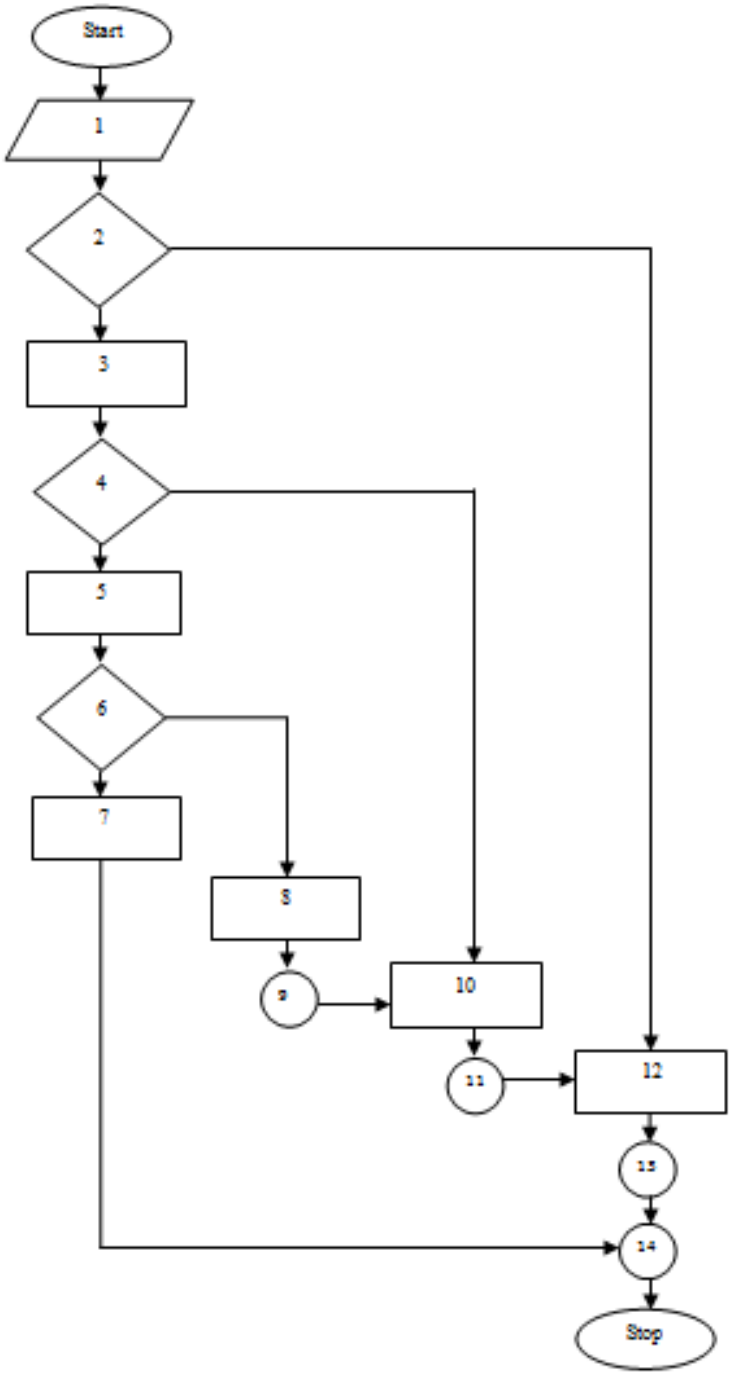
		Memilih tombol mati untuk musik	Black Box
		Memilih tombol aktif untuk layar penuh	Black Box
		Memilih tombol mati untuk layar penuh	Black Box
		Memilih tombol kembali ke menu utama	Black Box
3.	Menu Nilai Tertinggi	Memilih tombol kembali ke menu utama	Black Box
4.	Menu Pilih Area	Memilih area pegunungan	Black Box
		Memilih area bendungan	Black Box
		Memilih area pemukiman	Black Box
		Memilih tombol kembali ke menu utama	Black Box
5.	Karakter Bergerak	Menekan keyboard “→”	Black Box
		Menekan keyboard “←”	Black Box
		Menekan keyboard “↑”	Black Box
		Menekan keyboard “↓”	Black Box
		Menekan keyboard “A”	Black Box
		Menekan keyboard “S”	Black Box
6.	Stage	Memilih tombol lewati cerita	Black Box
		Memilih tombol ok untuk petunjuk misi	Black Box
		Memilih tombol ok untuk petunjuk tombol	Black Box
		Memilih tombol mengulang permainan	Black Box
		Memilih tombol keluar permainan	Black Box
7.	Pergerakan Musuh	Algoritma Greedy	White Box

b. Pengujian White Box

Pengujian white box merupakan pengujian yang didasarkan pada pengecekan terhadap detail perancangan, menggunakan struktur *control* dari desain aplikasi secara prosedural untuk membagi pengujian ke dalam beberapa kasus pengujian untuk mengetahui apakah setiap node pada *independent paths* dikerjakan. Serta untuk menguji alur logika, perulangan dan validasi parameter masukannya.

1. Mengubah bentuk *pseudocode* menjadi bentuk *flowchart.*
   a. Penerapan Algoritma Greedy

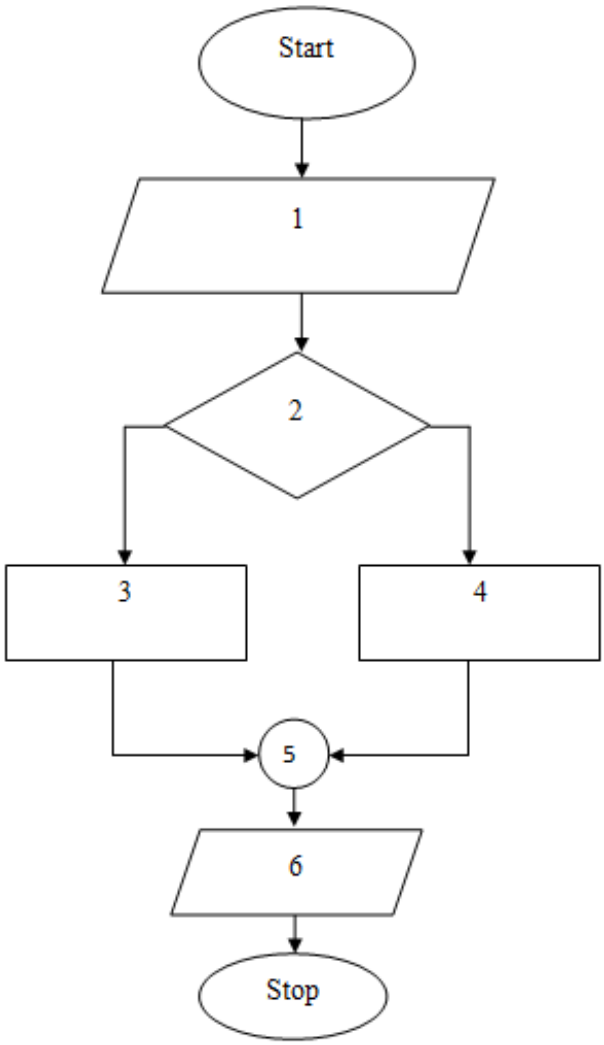
```
1   read C[kandidat]
2   if HP >= 100 then
3   A ← kandidat paling dekat
4         if HP >= A then
5         S ← A
          C [kandidat] -= 1
6                 if S=A then
7                 attack =  true
8                   else attack = false
9                  endif
10               else attack = false
11          endif
12      else attack = false
13  endif
14  end
```



Gambar 4.14 Flowchart Penerapan Algoritma Greedy

b. Penerapan Collision Detection

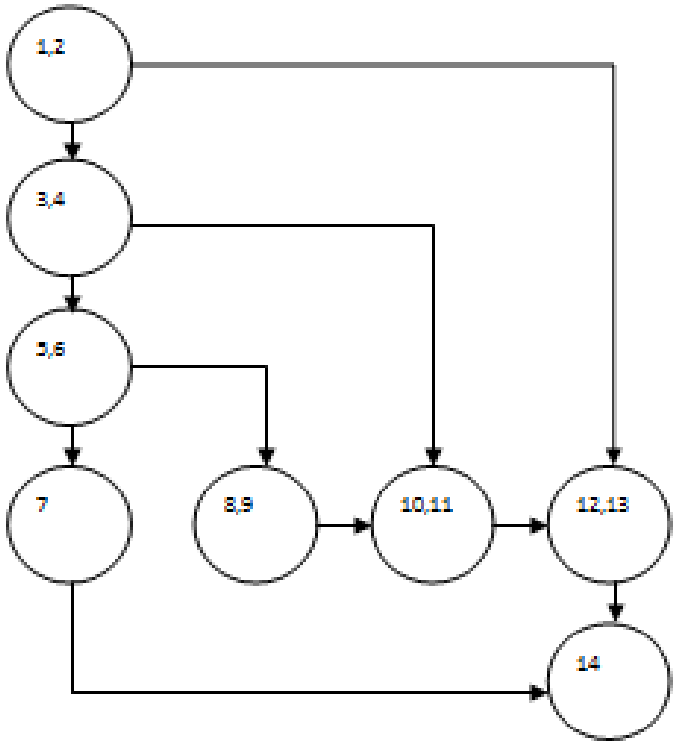
```
1  read x1,y1,x2,y2
2  if collision(x1,y1,x2,y2)=true then
3  condition = true
4  else condition = false
5  endif
6  write condition
```



Gambar 4.15 Flowchart Penerapan Collision Detection

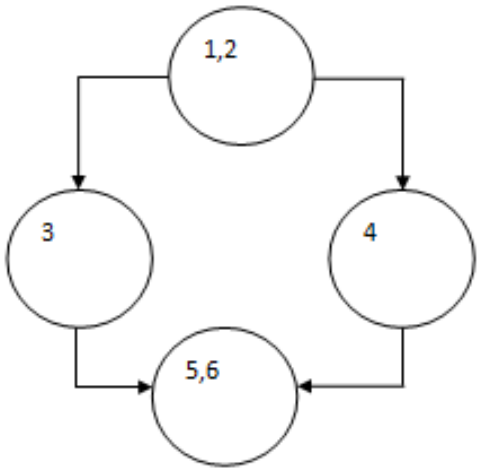
2. *Flowgraph*.

a. Penerapan Algoritma Greedy



Gambar 4.16 Flowgraph Penerapan Algoritma Greedy

b. Penerapan Collision Detection



Gambar 4.17 Flowgraph Penerapan Collision Detection

3. *Cyclomatic Complexity*
   a. Penerapan Algoritma Greedy

      E = 10, N = 8

      V(G) = E-N+2

      V(G) = 10 – 8 + 2

      V(G) = 4

      Keterangan :

      E : Jumlah busur pada *flow graph*

      N : Jumlah simpul (*node*) pada *flow graph*

      V(G) : *Cyclomatic Complexity*

   b. Penerapan Collision Detection

      E = 4, N = 4

      V(G) = E-N+2

      V(G) = 4 – 4 + 2

      V(G) = 2

      Keterangan :

      E : Jumlah busur pada *flow graph*

      N : Jumlah simpul (*node*) pada *flow graph*

      V(G) : *Cyclomatic Complexity*

4. *Independent Path*
   a. Penerapan Algoritma Greedy

      Path 1 : 1-2-3-4-5-6-7-14

      Path 2 : 1-2-12-13-14

      Path 3 : 1-2-3-4-10-11-12-13-14

      Path 4 : 1-2-3-4-5-6-8-9-10-11-12-13-14

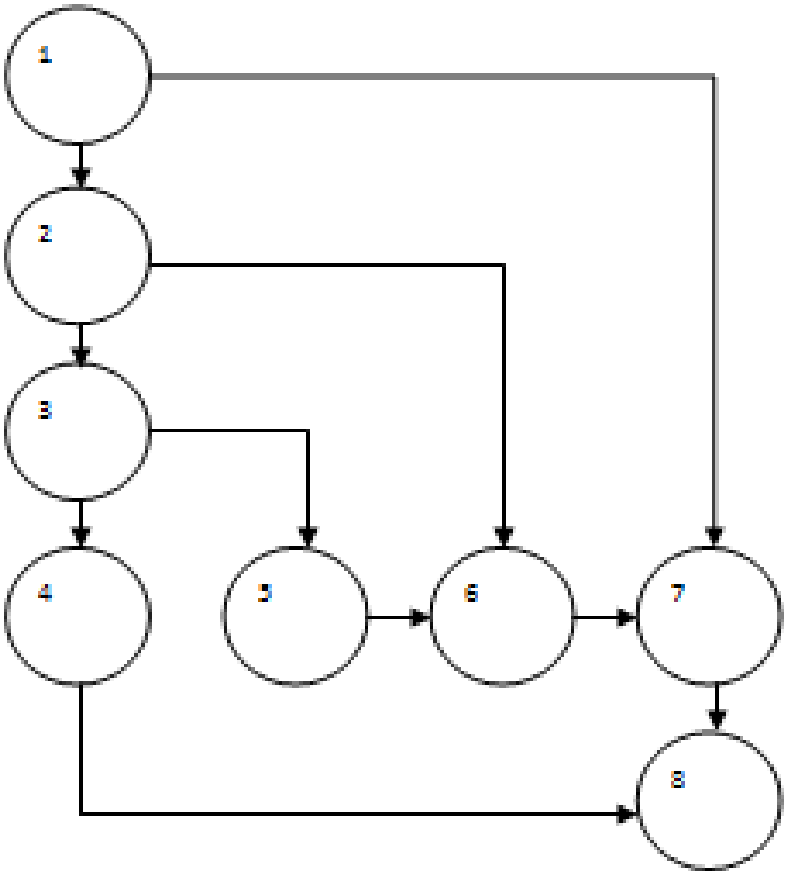
b. Penerapan Collision Detection

Path 1 : 1-2-3-5-6

Path 2 : 1-2-4-5-6

5. *Graph Matrics Testing*

a. Penerapan Algoritma Greedy



Gambar 4.18 Flowgraph Penerapan Algoritma Greedy Setelah Penomoran Ulang

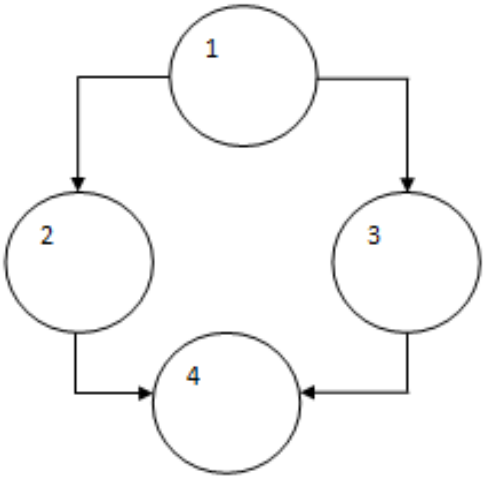
Tabel 4.2 Graph Matrics Testing Penerapan Algoritma Greedy

N	1	2	3	4	5	6	7	8	N(E)-1
1		1					1		1
2			1			1			1
3				1	1				1
4								1	0
5						1			0
6							1		0
7								1	0
8									0
Jumlah									3

V(G) = Jumlah *graph matrices* + 1

V(G) = 4

b. Penerapan Collision Detection



Gambar 4.19 Flowgraph Penerapan Collision Detection Setelah Penomoran Ulang

Tabel 4.3 Graph Matrics Testing Penerapan Collision Detection

N	1	2	3	4	N(E)-1
1		1	1		1
2				1	0
3				1	0
4					0
Jumlah					1

V(G) = Jumlah *graph matrices* + 1

V(G) = 2

4.5.1.2 Kasus dan Hasil Pengujian

Berdasarkan kasus pengujian yang telah disusun, maka dapat dilakukan pengujian sebagai berikut.

a. Pengujian Black Box

Tabel 4.4 Kasus dan Hasil Uji Black Box

No	Kelas Uji	Detail Pengujian	Hasil yang Diharapkan	Hasil Pengujian
1.	Menu Utama	Memilih tombol permainan	Menampilkan menu pilih Area	[√] Berhasil [ ] Tidak Berhasil
		Memilih tombol pengaturan	Menampilkan menu pengaturan	[√] Berhasil [ ] Tidak Berhasil
		Memilih tombol nilai tertinggi	Menampilkan menu nilai tertinggi	[√] Berhasil [ ] Tidak Berhasil
		Memilih tombol	Keluar dari game	[√] Berhasil

		keluar		[ ] Tidak Berhasil
2.	Menu Pengaturan	Memilih tombol aktif untuk musik	Musik game menyala	[√] Berhasil [ ] Tidak Berhasil
		Memilih tombol mati untuk musik	Musik game mati	[√] Berhasil [ ] Tidak Berhasil
		Memilih tombol aktif untuk layar penuh	Layar game fullscreen	[√] Berhasil [ ] Tidak Berhasil
		Memilih tombol mati untuk layar penuh	Layar game window	[√] Berhasil [ ] Tidak Berhasil
		Memilih tombol kembali pada menu pengaturan	Menampilkan menu utama game	[√] Berhasil [ ] Tidak Berhasil
3.	Menu Nilai Tertinggi	Memilih tombol kembali pada menu nilai tertinggi	Menampilkan menu utama game	[√] Berhasil [ ] Tidak Berhasil
4.	Menu Pilih Area	Memilih area pegunungan	Menampilkan stage game area pegunungan	[√] Berhasil [ ] Tidak Berhasil
		Memilih area bendungan	Menampilkan stage game area bendungan	[√] Berhasil [ ] Tidak Berhasil
		Memilih area pemukiman	Menampilkan stage game area pemukiman	[√] Berhasil [ ] Tidak Berhasil

		Memilih tombol kembali pada menu pilih area	Menampilkan menu utama game	[√] Berhasil [ ] Tidak Berhasil
5.	Karakter Bergerak	Menekan keyboard "→"	Karakter bergerak ke kanan	[√] Berhasil [ ] Tidak Berhasil
		Menekan keyboard "←"	Karakter bergerak ke kiri	[√] Berhasil [ ] Tidak Berhasil
		Menekan keyboard "↑"	Karakter bergerak ke atas	[√] Berhasil [ ] Tidak Berhasil
		Menekan keyboard "↓"	Karakter bergerak ke bawah	[√] Berhasil [ ] Tidak Berhasil
		Menekan keyboard "A"	Karakter menyerang musuh	[√] Berhasil [ ] Tidak Berhasil
		Menekan keyboard "S"	Karakter menanam pohon atau membersihkan sungai	[√] Berhasil [ ] Tidak Berhasil
6.	Stage	Memilih tombol lewati cerita	Menampilkan petunjuk misi	[√] Berhasil [ ] Tidak Berhasil
		Memilih tombol ok untuk petunjuk misi	Menampilkan petunjuk tombol	[√] Berhasil [ ] Tidak Berhasil
		Memilih tombol ok untuk petunjuk	Menampilkan stage game	[√] Berhasil [ ] Tidak

		tombol		Berhasil
		Memilih tombol mengulang permainan	Menampilkan pesan mengulang permainan	[√] Berhasil [ ] Tidak Berhasil
		Memilih tombol keluar permainan	Menampilkan pesan keluar stage	[√] Berhasil [ ] Tidak Berhasil

b. Pengujian White Box

a. Penerapan Algoritma Greedy

Tabel 4.5 Hasil Uji White Box Penerapan Algoritma Greedy

Kasus dan Hasil Uji				
No. Path	Butir Uji Algoritma Greedy	Node	Pengamatan	Kesimpulan
1	HP >= 100 dan HP >= A dan A = S, Hero memilih menyerang	1,2,4,5,6,7,14	Musuh menyerang	Path dikerjakan
2	HP< 100, Hero memilih tidak menyerang	1,2,12,13,14	Musuh tidak menyerang	Path dikerjakan
3	HP<A, Hero memilih tidak menyerang	1,2,3,4,10,11,12,13,14	Musuh tidak menyerang	Path dikerjakan
4	HP != A, Hero memilih tidak menyerang	1,2,3,4,5,6,8,9,10,11,12,13,14	Musuh tidak menyerang	Path dikerjakan

b. Penerapan Collision Detection

Tabel 4.6 Hasil Uji White Box Penerapan Collision Detection

Kasus dan Hasil Uji				
No. Path	Butir Uji Bounding Box Test	Node	Pengamatan	Kesimpulan
1	Collision true, Condition = true	1,2,4	Condition true	Path dikerjakan
2	Collision false, Condition = false	1,3,4	Condition false	Path dikerjakan

4.5.2 Pengujian Beta

Untuk mengetahui tanggapan dan penilaian pengguna terhadap game ini. Telah disebarkan kuesioner kepada 30 responden dengan usia berdasarkan pada sasaran user. Perhitungan untuk kuisioner menggunakan *skala likert*. Dimana terdapat lima penyataan yaitu, sangat setuju, setuju, sedang, tidak setuju dan sangat tidak setuju. Setiap pernyataan tersebut mempunyai bobot nilai seperti berikut :

1. Sangat Setuju : 5 poin
2. Setuju : 4 poin
3. Sedang : 3 poin
4. Tidak Setuju : 2 poin
5. Sangat Tidak Setuju : 1 poin

Data hasil kuisioner yang telah diberikan dapat dicari presentasinya dari masing-masing jawaban dengan menggunakan rumus :

$$P = \frac{S}{skorideal} x100\%$$

Keterangan :

P = Nilai persentase yang dicari

S = Jumlah frekuensi jawaban dikalikan dengan nilai pernyataan

Skorideal = Nilai pernyataan tertinggi dikalikan dengan jumlah sampel

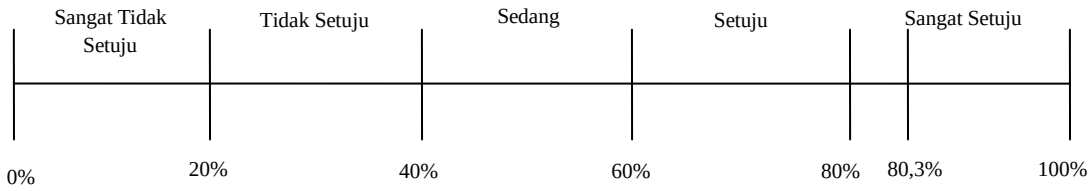
Berikut ini adalah presentase masing-masing jawaban yang sudah dihitung nilainya dengan menggunakan rumus di atas.

1. Apakah tampilan game Cegah Banjir ini menarik bagi anda?

Tabel 4.7 Perhitungan Presentase Pertanyaan Kesatu

Pertanyaan	Pernyataan	Skor	Frekuensi	S
1.	Sangat Setuju	5	15	75
	Setuju	4	10	40
	Sedang	3	4	12
	Tidak Setuju	2	0	0
	Sangat Tidak Setuju	1	1	1
Jumlah			30	128

$$p = \frac{128}{150} \times 100 = 85{,}3\%$$



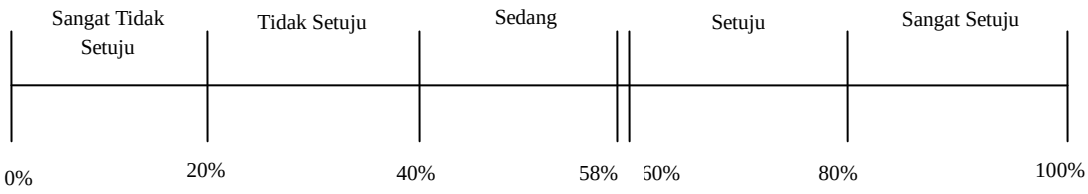
Gambar 4.20 Hasil Perhitungan Kuisioner Pertanyaan Kesatu

2. Apakah anda kesulitan menyelesaikan misi pada game?

Tabel 4.8 Perhitungan Presentase Pertanyaan Kedua

Pertanyaan	Pernyataan	Skor	Frekuensi	S
2.	Sangat Setuju	5	1	5
	Setuju	4	5	20
	Sedang	3	15	45
	Tidak Setuju	2	8	16
	Sangat Tidak Setuju	1	1	1
Jumlah			30	87

$$p = \frac{87}{150} \times 100 = 58\%$$



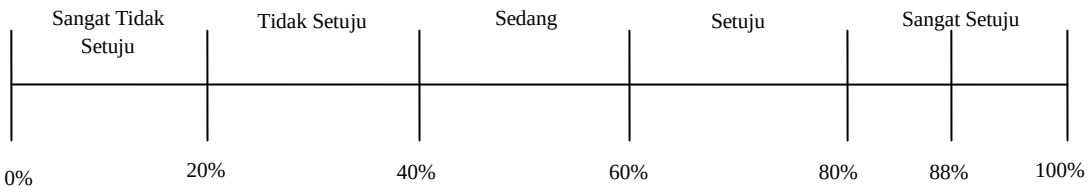
Gambar 4.21 Hasil Perhitungan Kuisioner Pertanyaan Kedua

3. Menurut anda, apakah game ini mudah digunakan?

Tabel 4.9 Perhitungan Presentase Pertanyaan Ketiga

Pertanyaan	Pernyataan	Skor	Frekuensi	S
3.	Sangat Setuju	5	18	90
	Setuju	4	8	32
	Sedang	3	2	6
	Tidak Setuju	2	2	4
	Sangat Tidak Setuju	1	0	0
Jumlah			30	132

$$p = \frac{132}{150} \times 100 = 88\%$$



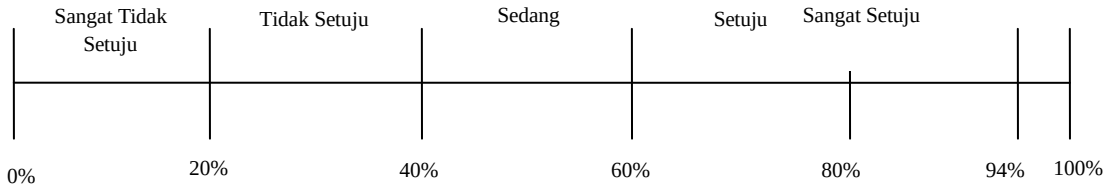
Gambar 4.22 Hasil Perhitungan Kuisioner Pertanyaan Ketiga

4. Apakah dengan adanya game ini anda lebih tertarik untuk lebih mengetahui penyebab terjadinya banjir?

Tabel 4.10 Perhitungan Presentase Pertanyaan Keempat

Pertanyaan	Pernyataan	Skor	Frekuensi	S
4.	Sangat Setuju	5	24	120
	Setuju	4	4	16
	Sedang	3	1	3
	Tidak Setuju	2	1	2
	Sangat Tidak Setuju	1	0	0
Jumlah			30	141

$$p = \frac{141}{150} \times 100 = 94\%$$



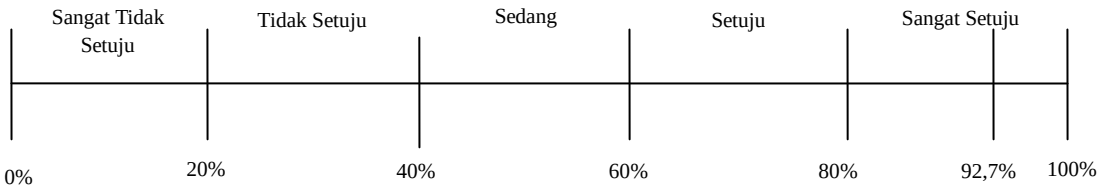
Gambar 4.23 Hasil Perhitungan Kuisioner Pertanyaan Keempat

5. Apakah dengan adanya game ini pengetahuan anda tentang cara mencegah banjir menjadi bertambah?

Tabel 4.11 Perhitungan Presentase Pertanyaan Kelima

Pertanyaan	Pernyataan	Skor	Frekuensi	S
5.	Sangat Setuju	5	21	105
	Setuju	4	7	28
	Sedang	3	2	6
	Tidak Setuju	2	0	0
	Sangat Tidak Setuju	1	0	0
Jumlah			30	139

$$p = \frac{139}{150} \times 100 = 92{,}7\%$$



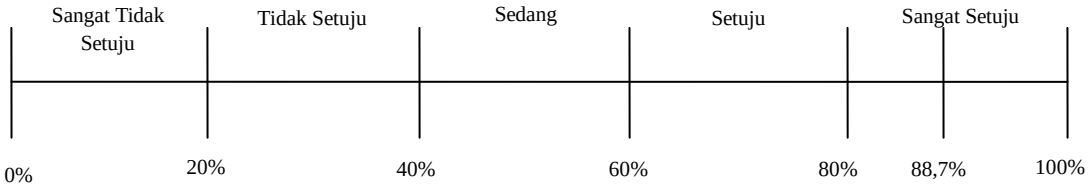
Gambar 4.24 Hasil Perhitungan Kuisioner Pertanyaan Kelima

6. Apakah dengan adanya game ini anda akan menjaga lingkungan agar terhindar dari banjir?

Tabel 4.12 Perhitungan Presentase Pertanyaan Keenam

Pertanyaan	Pernyataan	Skor	Frekuensi	S
6.	Sangat Setuju	5	16	80
	Setuju	4	13	52
	Sedang	3	0	0
	Tidak Setuju	2	0	0
	Sangat Tidak Setuju	1	1	1
Jumlah			30	133

$$p = \frac{133}{150} \times 100 = 88{,}7\%$$



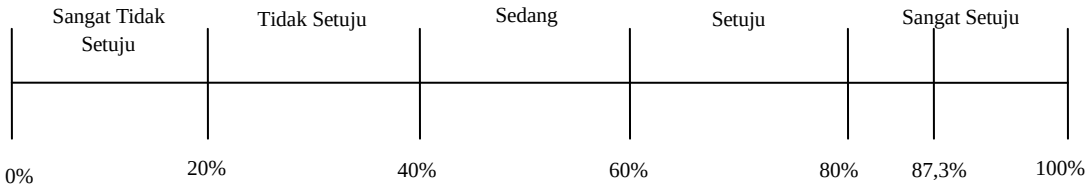
Gambar 4.25 Hasil Perhitungan Kuisioner Pertanyaan Keenam

7. Apakah dengan adanya game ini anda akan menegur orang-orang yang melakukan perusakan lingkungan seperti orang-orang yang membuang sampah sembarangan?

Tabel 4.13 Perhitungan Presentase Pertanyaan Ketujuh

Pertanyaan	Pernyataan	Skor	Frekuensi	S
7.	Sangat Setuju	5	13	65
	Setuju	4	16	64
	Sedang	3	0	0
	Tidak Setuju	2	1	2
	Sangat Tidak Setuju	1	0	0
Jumlah			30	131

$$p = \frac{131}{150} \times 100 = 87{,}3\%$$



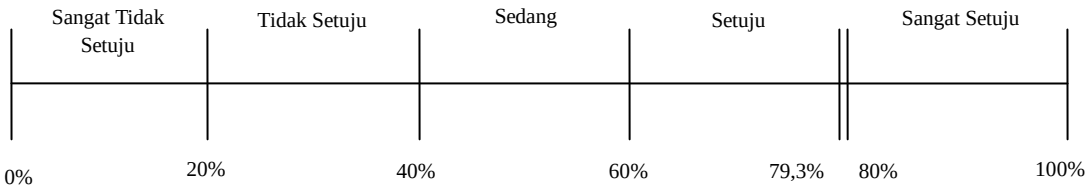
Gambar 4.26 Hasil Perhitungan Kuisioner Pertanyaan Ketujuh

8. Apakah anda tertarik untuk memainkan game ini kembali?

Tabel 4.14 Perhitungan Presentase Pertanyaan Kedelapanr

Pertanyaan	Pernyataan	Skor	Frekuensi	S
8.	Sangat Setuju	5	7	35
	Setuju	4	15	60
	Sedang	3	8	24
	Tidak Setuju	2	0	0
	Sangat Tidak Setuju	1	0	0
Jumlah			30	119

$$p = \frac{119}{150} \times 100 = 79{,}3\%$$



Gambar 4.30 Hasil Perhitungan Kuisioner Pertanyaan Kedelapan

Berdasarkan hasil yang didapat, maka kesimpulannya adalah sebagai berikut:

1. Responden sangat setuju bahwa tampilan game sangat menarik.
2. Responden menyatakan bahwa tingkat kesulitan untuk menyelesaikan misi game tergolong sedang.
3. Responden sangat setuju bahwa game ini mudah untuk mereka mainkan.
4. Responden sangat setuju bahwa mereka tertarik untuk lebih mengetahui penyebab terjadinya banjir setelah memainkan game ini.
5. Responden sangat setuju bahwa pengetahuan mereka tentang cara mencegah banjir manjadi bertambah setelah memainkan game ini.
6. Responden sangat setuju bahwa mereka akan menjaga lingkungan agar terhindar dari banjir setelah memainkan game ini.
7. Responden sangat setuju bahwa mereka akan menegur orang-orang yang melakukan perusakan lingkungan seperti orang-orang yang membuang sampah sembarangan setelah memainkan game ini.
8. Responden setuju bahwa mereka tertarik untuk memainkan game ini kembali.