BAB 4

IMPLEMENTASI DAN PENGUJIAN

Pada bab ini akan dilakukan implementasi dan pengujian terhadap aplikasi permainan yang dibangun, yaitu *Pirates of Money*. Hasil perancangan pada tahap sebelumnya kemudian diimplementasikan ke dalam bahasa pemrograman.Setelah tahap implementasi selesai, maka dilakukan tahap pengujian terhadap aplikasi permainan *Pirates of Money*.

4.1 Implementasi

Tahap implementasi merupakan tahap penciptaan perangkat lunak dan juga tahap kelanjutan dari kegiatan perancangan aplikasi.Tahap ini merupakan tahap dimana aplikasi siap untuk dioperasikan, yang terdiri dari penjelasan mengenai lingkungan implementasi, baik itu lingkungan perangkat keras, maupun lingkungan perangkat lunak, serta implementasi program.

4.1.1 Implementasi Perangkat Keras

Perangkat keras yang digunakan untuk menjalankan aplikasi permainan *Pirates of Money* tidak harus yang berspesifikasi tinggi. Spesifikasi minimum yang dapat digunakan untuk menjalankan aplikasi permainan *Pirates of Money* adalah sebagai berikut :

1. *Processor* 2,0 GHz
2. RAM 1 GB

3. VGA 512 MB
4. *Harddisk* 80 GB
5. Monitor dengan resolusi 1024 x 768
6. *Mouse*
7. *Keyboard*

4.1.2 Implementasi Perangkat Lunak

Adapun spesifikasi perangkat lunak yang digunakan untuk menjalankan aplikasi permainan *Pirates of Money*adalah perangkat komputer yang telah terinstal JVM (*Java Virtual Machine*).

4.1.3 Implementasi Aplikasi

Pirates of Money merupakan aplikasi permainan ber-*genretower defense*.Pemain mendapatkan misi setiap *level*-nya yaitu untuk mencegah para tikus mencapai brankas dengan cara menempatkan *hero* yang ada pada *shop* untuk membunuh para tikus tersebut, dan setiap *hero* dapat dibeli sesuai dengan jumlah koin yang dimiliki pemain.

Permainan dikatakan *game over* apabila jumlah darah pemain berkurang menjadi 0, dan sebaliknya permainan dikatakan berhasil apabila pemain dapat menyelesaikan misi disetiap *level*-nya dengan baik.

Aplikasi permainan ini berbasis Java-Desktop yang dibangun dengan menggunakan bahasa pemrograman *Java*.Sehingga aplikasi permainan ini hanya

dapat dijalankan apabila pemain telah menginstal JVM (*Java Virtual Machine*) pada perangkat komputernya.

4.1.4 Implementasi Antarmuka

Adapun implementasi antarmuka pada aplikasi permainan *Pirates of Money* adalah sebagai berikut :

1. Antarmuka Menu Utama

   Antarmuka menu utama merupakan tampilan menu utama pada saat pertama kali pemain membuka aplikasi permainan. Tampilan dari antarmuka menu utama dapat dilihat pada gambar 4.1.



Gambar 4.1 Tampilan Antarmuka Menu Utama

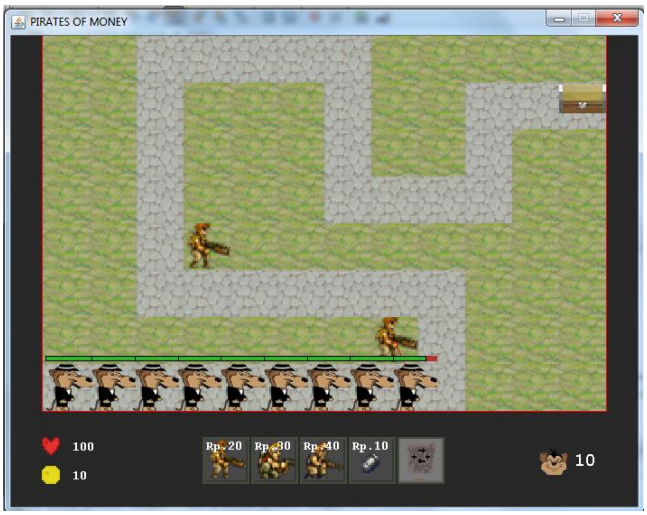
2. Antarmuka Mulai Permainan

   Antarmuka mulai permainanmerupakan tampilan ketika pemain mengklik tombol mulai permainan sehingga langsung masuk ke *level* 1. Tampilan dari antarmuka mulai permainan *level* 1 dapat dilihat pada gambar 4.2.



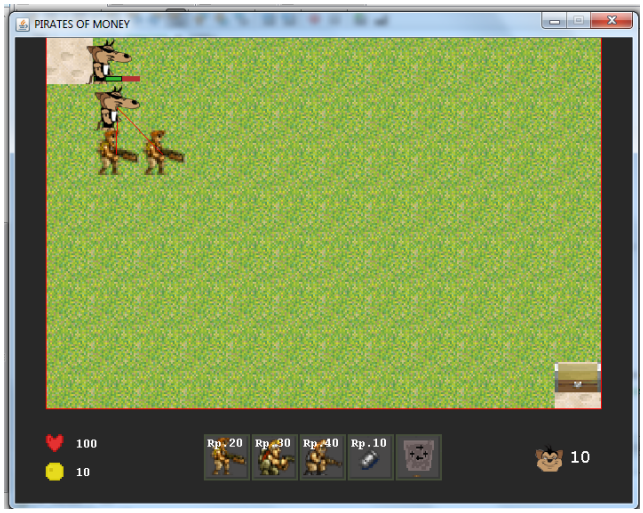
Gambar 4.2 Tampilan Antarmuka Mulai Permainan *Level 1*

Tampilan antarmuka *level* 2 dapat dilihat pada gambar 4.3.



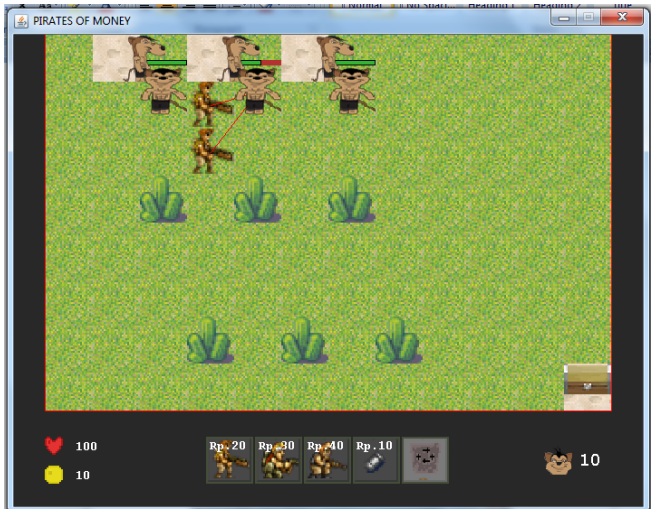
Gambar 4.3 Tampilan Antarmuka Mulai Permainan *Level*2

Tampilan antarmuka *level* 3 dapat dilihat pada gambar 4.4.



Gambar 4.4 Tampilan Antarmuka Mulai Permainan *Level* 3

Tampil antarmuka *level* 4 dapat dilihat pada gambar 4.5.



Gambar 4.5 Tampilan Antarmuka Mulai Permainan *Level* 4

4.2 Pengujian

Dalam melakukan pengujian terhadap aplikasi permainan *Pirates of Money* dilakukan dengan dua mcama teknik, yaitu pengujian *black-box* dan

pengujian *white-box*.Adapun pengujian *black box* berfokus pada menemukan kesalahan program, sedangkan pengujian *white box* berfokus kepada struktur internal (*source* code) program untuk mengetahui apakah masih terjadi *error*. Pengujian dilakukan dengan dua tahap, yaitu tahap pengujian *alpha* dan tahap pengujian *bet*a.

4.2.1 Pengujian *Alpha*

Pada tahap ini, pengujian dilakukan untuk memastikan aplikasi permainan *Pirates of Money* telah berjalan dengan benar dan sesuai dengan kebutuhan yang diinginkan.Pengujian *alpha* menitikberatkan kepada persyaratan fungsional perangkat lunak.

4.2.1.1 Skenario Pengujian Aplikasi

Skenario pengujian menjelaskan pengujian terhadap sistem yang ada pada aplikasi permainan *Pirates of Money.*Skenario pengujian dari aplikasi permainan *Pirates of Money* dapat dilihat pada tabel 4.1.

Tabel 4.1 Skenario Pengujian Aplikasi Permainan *Pirates of Money*

No	Komponen yang di uji	Skenario Pengujian	Jenis Pengujian
1	Menu Utama	Klik Menu Mulai Permainan	*Black Box*
		Klik Menu Petunjuk	*Black Box*
		Klik Menu Kredit	*Black Box*
		Klik Menu Keluar	*Black Box*
2	Menu Mulai Permainan	Klik Menu Mulai Permainan	*Black Box*
		Permainan *Level* 1	*Black Box*
3	Menu Petunjuk	Klik Menu Petunjuk	*Black Box*
		Klik Tombol Kembali	*Black Box*
4	Menu Kredit	Klik Menu Kredit	*Black Box*

		Klik Tombol Kembali	*Black Box*
5	Menu Keluar	Klik Menu Keluar	*Black Box*
6	*Collision Detection*	Deteksi Tabrakan Entitias Peluru Dengan Entitas Musuh	*White Box*
		Deteksi Tabrakan Entitas Musuh Dengan Entitas Brankas	*White Box*
7	Algoritma A* (A star)	Algoritma A* Pada *Level* 3	*White Box*
		Algoritma A* Pada *Level* 4	*White Box*

4.2.1.2 Kasus dan Hasil Pengujian (*Black Box*)

Kasus dan hasil uji perangkat lunak yang sudah dibangun dengan menggunakan metode *black box* berdasarkan pengamatan yang dilakukan dan menghasilkan kesimpulan dari pengujian tersebut, diantaranya sebagai berikut :

A. **Pengujian Mulai Permainan**

Pengujian mulai permainan dengan kasus dan hasil uji, kemudian diamati dan menghasilkan kesimpulan dapat dilihat pada tabel 4.2.

Tabel 4.2 Pengujian Mulai Permainan

Kasus dan Hasil Uji(Data benar)			
Masukan	Keluaran yang diharapkan	Pengamatan	Kesimpulan
Klik mulai permainan	Menampilkan map *level* 1	Dapat menampilkan map *level* 1	[√] Diterima [ ] Ditolak
Menyelesaikan misi *level* 1	Menampilkan map *level* 2	Dapat menampilkan map *level* 2	[√] Diterima [ ] Ditolak
Menyelesaikan misi *level* 2	Menampilkan map *level* 3	Dapat menampilkan map *level* 3	[√] Diterima [ ] Ditolak

Menyelesaikan misi *level* 3	Menampilkan map *level* 4	Dapat menampilkan map *level* 4	[√] Diterima [ ] Ditolak
Menyelesaikan misi *level* 4	Menampilkan keterangan menang	Dapat menampilkan keterangan menang	[√] Diterima [ ] Ditolak

B. **Pengujian Tampil Kredit**

Pengujian mulai permainan dengan kasus dan hasil uji, kemudian diamati dan menghasilkan kesimpulan dapat dilihat pada tabel 4.3.

Tabel 4.3 Pengujian Tampil Kredit

Kasus dan Hasil Uji(Data benar)			
Masukan	Keluaran yang diharapkan	Pengamatan	Kesimpulan
Menekan tombol kredit	Menampilkan informasi pembuat aplikasi permainan	Dapat menampilkan informasi pembuat aplikasi permainan	[√] Diterima [ ] Ditolak
Menekan tombol kembali	Menampilkan menu utama	Dapat menampilkan menu utama	[√] Diterima [ ] Ditolak

C. **Pengujian Tampil Petunjuk**

Pengujian mulai permainan dengan kasus dan hasil uji, kemudian diamati dan menghasilkan kesimpulan dapat dilihat pada tabel 4.4.

Tabel 4.4 Pengujian Tampil Petunjuk

Kasus dan Hasil Uji(Data benar)			
Masukan	Keluaran yang diharapkan	Pengamatan	Kesimpulan
Menekan tombol petunjuk	Menampilkan informasi petunjuk permainan	Dapat menampilkan informasi petunjuk permainan	[√] Diterima [ ] Ditolak

Menekan tombol kembali	Menampilkan menu utama	Dapat menampilkan menu utama	[√] Diterima [ ] Ditolak

4.2.1.3Kasus dan Pengujian (*White Box*)

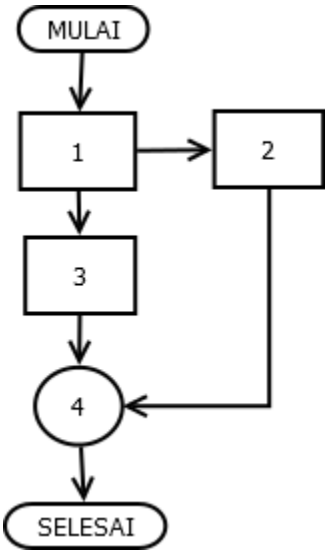
Di bawah ini merupakan kasus *collision detection* dan algoritma A* (A *star*) untuk menguji perangkat lunak yang sudah dibangun dengan menggunakan metode *white box*, berdasarkan skenario pengujian aplikasi yang sudah dijabarkan.

1. Pengujian *Collision Detection*

   Pengujian *collision detection* merupakan proses dimana akan diuji deteksi tubrukanantara peluru dengan musuh dan juga antara musuh dengan brankas pada aplikasi permainan *Pirates of Money*.

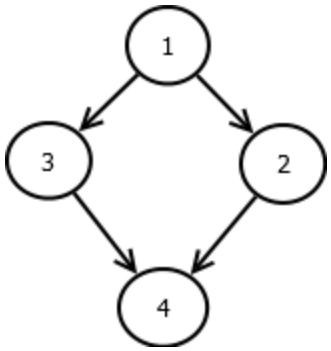
   a. *Collision Detection* Peluru dengan Musuh

      Pengujian dilakukan berdasarkan *pseudo code* pada tabel 3.4, adapun *Flowchart* pengujian pada *collision detection* peluru dengan musuh dapat dilihat pada gambar 4.6.



Gambar 4.6 *Flowchart* Pengujian *Collision Detection* Peluru dengan Musuh

Flowgraph yang merupakan hasil dari perubahan *flowchart* dapat dilihat pada gambar 4.7.



Gambar 4.7 *Flow Graph Collision Detection* Peluru dengan Musuh

Keterangan :  = Menggambarkan kondisi

 = Menggambarkan aksi

b. Tahap Pengujian *Collision Detection* Peluru dengan Musuh

1. *Cyclomatic Complexity* V(G)

Edge = 4, Node = 4

V(G) = Edge – Node + 2

V(G) = 4 – 4 + 2

V(G) = 0 + 2

V(G) = 2

2. *Independent Path*

Path 1 = 1, 2, 4

Path 2 = 1, 3, 4

3. *Predicate Node*

V(G) = Jumlah node yang memiliki lebih dari 1 jalur + 1

V(G) = 1 + 1

V(G) = 2

4. *Graph Matriks*

   Graph Matriks pengujian pada *collision detection* peluru dengan musuh dapat dilihat pada tabel 4.5.

Tabel 4.5 *Graph Matriks Collision Detection* Peluru dengan Musuh

	1	2	3	4	
1		1	1		1
2				1	0
3				1	0
4					0
hasil					1

   V(G) = Jumlah *Graph Matriks* + 1

   V(G) = 1 + 1

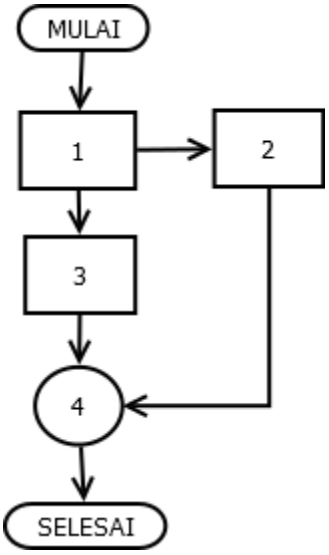
   V(G) = 2

5. Kesimpulan

   Berdasarkan hasil pengujian dihasilkan nilai *Cyclomatic Complexity* yang sama yaitu 2, maka dapat disimpulkan bahwa pada proses *collision detection* peluru dengan musuh berjalan dengan baik, karena setiap pengujian menghasilkan nilai yang sama.

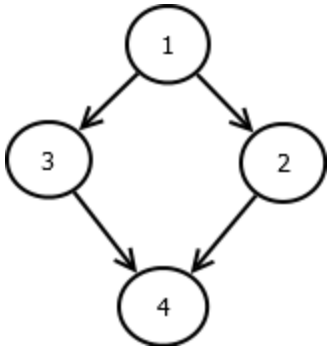
c. *Collision Detection* Musuh dengan Brankas

   Pengujian dilakukan berdasarkan *pseudo code* pada tabel 3.6, adapun *Flowchart* pengujian pada *collision detection* musuh dengan brankas dapat dilihat pada gambar 4.8.



Gambar 4.8 *Flowchart* Pengujian *Collision Detection* Musuh dengan Brankas

Flowgraph yang merupakan hasil dari perubahan *flowchart* dapat dilihat pada gambar 4.9.



Gambar 4.9 *Flowgraph Collision Detection* Musuh dengan Brankas

Keterangan : = Menggambarkan kondisi

= Menggambarkan aksi

d. Tahap Pengujian *Collision Detection* Peluru dengan Musuh

1. *Cyclomatic Complexity* V(G)

Edge = 4, Node = 4

V(G) = Edge – Node + 2

V(G) = 4 – 4 + 2

V(G) = 0 + 2

V(G) = 2

2. *Independent Path*

Path 1 = 1, 2, 4

Path 2 = 1, 3, 4

3. *Predicate Node*

V(G) = Jumlah node yang memiliki lebih dari 1 jalur + 1

V(G) = 1 + 1

V(G) = 2

4. *Graph Matriks*

Graph Matriks pengujian pada *collision detection* peluru dengan musuh dapat dilihat pada tabel 4.6.

Tabel 4.6 *Graph Matriks Collision Detection* Musuh dengan Brankas

	1	2	3	4	
1		1	1		1
2				1	0
3				1	0
4					0
hasil					1

V(G) = Jumlah *Graph Matriks* + 1

V(G) = 1 + 1

V(G) = 2

5. Kesimpulan

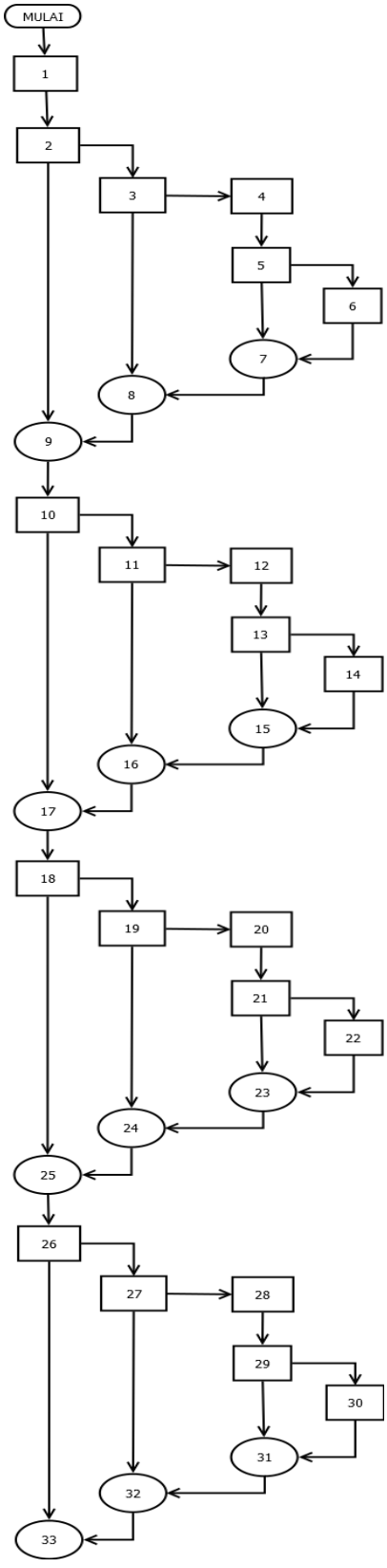
   Berdasarkan hasil pengujian dihasilkan nilai *Cyclomatic Complexity* yang sama yaitu 2, maka dapat disimpulkan bahwa pada proses *collision detection* peluru dengan musuh berjalan dengan baik, karena setiap pengujian menghasilkan nilai yang sama.

2. Pengujian Algoritma A* (A *star*)

   Pengujian Algoritma A* (A *star*) merupakan proses dimana akan diuji terhadap musuh dalam melakukan pencarian rute terpendek yang terdapat di *level* 3 dan 4 pada aplikasi permainan *Pirates of Money*.

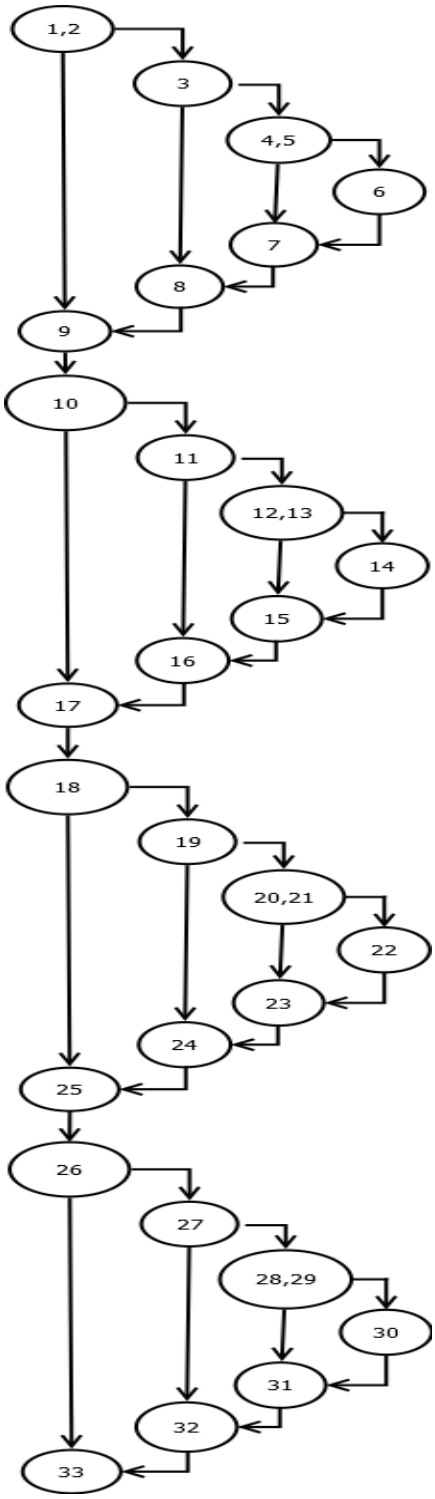
   a. Algoritma A* (A *star*) *Level* 3

      Pengujian dilakukan berdasarkan *pseudo code* pada tabel 3.7, adapun *Flowchart* pengujian pada algoritma A* (A *star*) *Level* 3 dapat dilihat pada gambar 4.10.



Gambar 4.10 *Flowchart* Pengujian Algoritma A* *Level* 3

Flowgraph yang merupakan hasil dari perubahan *flowchart* dapat dilihat pada gambar 4.11.



Gambar 4.11 *Flowgraph* Algoritma A* *Level* 3

Keterangan :  = Menggambarkan kondisi

 = Menggambarkan aksi

e. Tahap Pengujian Algoritma A* (A *star*) *Level* 3

1. *Cyclomatic Complexity* V(G)

Edge = 39, Node = 28

V(G) = Edge – Node + 2

V(G) = 39 – 28 + 2

V(G) = 11 + 2

V(G) = 13

2. *Independent Path*

Path 1 = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 17, 18, 25, 26, 33

Path 2 = 1, 2, 3, 4, 5, 7, 8, 9, 10, 17, 18, 25, 26, 33

Path 3 = 1, 2, 3, 8, 9, 10, 17, 18, 25, 26, 33

Path 4 = 1, 2, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 25, 26, 33

Path 5 = 1, 2, 9, 10, 11, 12, 13, 14, 16, 17, 18, 25, 26, 33

Path 6 = 1, 2, 9, 10, 11, 16, 17, 18, 25, 26, 33

Path 7 = 1, 2, 9, 10, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 33

Path 8 = 1, 2, 9, 10, 17, 18, 19, 20, 21, 23, 24, 25, 26, 33

Path 9 = 1, 2, 9, 10, 17, 18, 19, 24, 25, 26, 33

Path 10 = 1, 2, 9, 10, 17, 18, 19, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33

Path 11 = 1, 2, 9, 10, 17, 18, 19, 24, 25, 26, 27, 28, 29, 31, 32, 33

Path 12 = 1, 2, 9, 10, 17, 18, 19, 24, 25, 26, 27, 32, 33

Path 13 = 1, 2, 9, 10, 17, 18, 19, 24, 25, 26, 33

3. *Predicate Node*

   V(G) = Jumlah node yang memiliki lebih dari 1 jalur + 1

   V(G) = 12 + 1

   V(G) = 13

4. *Graph Matriks*

   Graph Matriks pengujian pada *collision detection* peluru dengan musuh dapat dilihat pada tabel 4.7.

Tabel 4.7 *Graph Matriks* Algoritma A* (A *star*) *Level* 3

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	
1		1																																
2			1						1																									1
3				1				1																										1
4					1																													
5						1	1																											1
6							1																											
7								1																										
8									1																									
9									1																									
10										1						1																		1
11											1				1																			1
12												1																						
13													1	1																				
14														1																				1
15															1																			
16																1																		
17																	1																	
18																		1							1									1
19																			1					1										1
20																				1														
21																					1	1												1
22																						1												
23																							1											
25																									1									
26																											1						1	1
27																												1				1		1
28																													1					
29																														1	1			1
30																															1			
31																																1		
32																																	1	
33																																		
Hasil																																		12

   V(G) = Jumlah *Graph Matriks* + 1

   V(G) = 12 + 1

   V(G) = 13

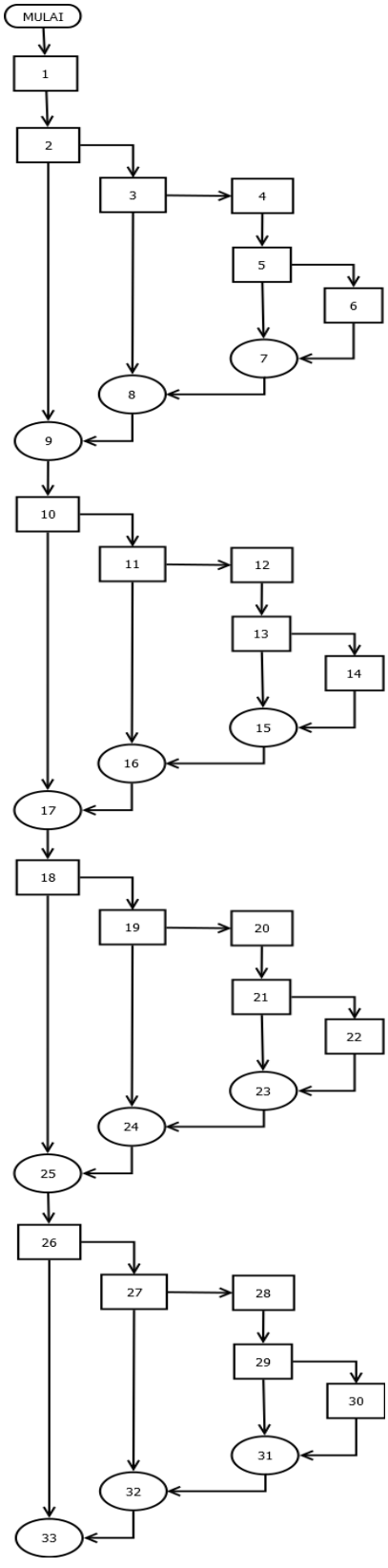
5. Kesimpulan

   Berdasarkan hasil pengujian dihasilkan nilai *Cyclomatic Complexity* yang sama yaitu 13, maka dapat disimpulkan bahwa

pada proses pencarian dengan algoritma A* (A *star*) berjalan dengan baik, karena setiap pengujian menghasilkan nilai yang sama.

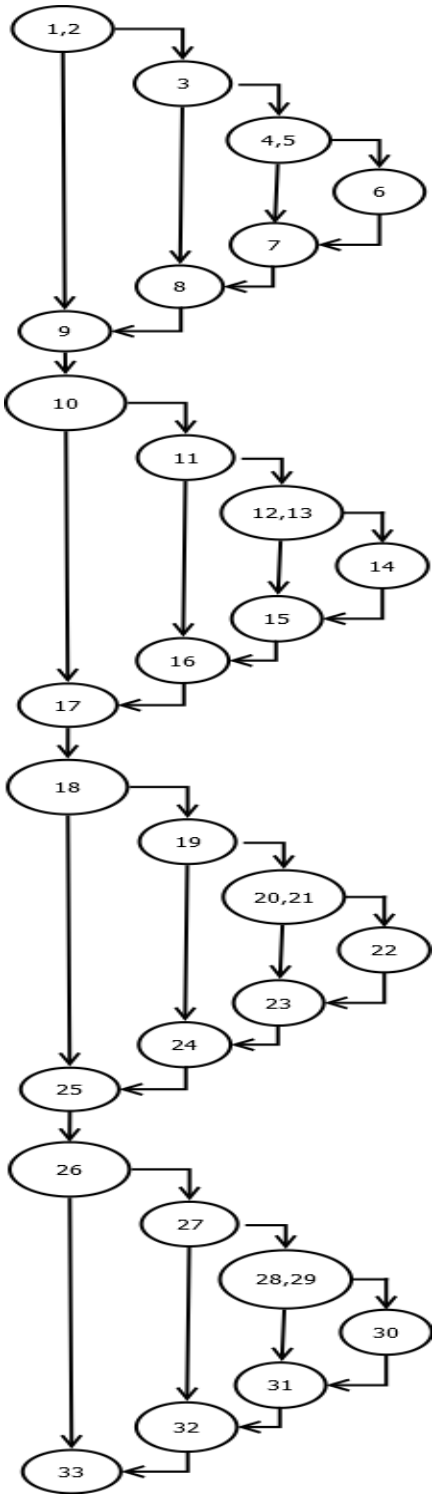
b. Algoritma A* (A *star*) *Level* 4

Pengujian dilakukan berdasarkan *pseudo code* pada tabel 3.7, adapun *Flowchart* pengujian pada algoritma A* (A *star*) *Level* 4 dapat dilihat pada gambar 4.12.



Gambar 4.12 *Flowchart* Pengujian Algoritma A* *Level* 4

Flowgraph yang merupakan hasil dari perubahan *flowchart* dapat dilihat pada gambar 4.13.



Gambar 4.13 *Flowgraph* Algoritma A* *Level* 4

Keterangan :  = Menggambarkan kondisi

 = Menggambarkan aksi

f. Tahap Pengujian Algoritma A* (A *star*) *Level* 4

6. *Cyclomatic Complexity* V(G)

Edge = 39, Node = 28

V(G) = Edge – Node + 2

V(G) = 39 – 28 + 2

V(G) = 11 + 2

V(G) = 13

7. *Independent Path*

Path 1 = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 17, 18, 25, 26, 33

Path 2 = 1, 2, 3, 4, 5, 7, 8, 9, 10, 17, 18, 25, 26, 33

Path 3 = 1, 2, 3, 8, 9, 10, 17, 18, 25, 26, 33

Path 4 = 1, 2, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 25, 26, 33

Path 5 = 1, 2, 9, 10, 11, 12, 13, 14, 16, 17, 18, 25, 26, 33

Path 6 = 1, 2, 9, 10, 11, 16, 17, 18, 25, 26, 33

Path 7 = 1, 2, 9, 10, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 33

Path 8 = 1, 2, 9, 10, 17, 18, 19, 20, 21, 23, 24, 25, 26, 33

Path 9 = 1, 2, 9, 10, 17, 18, 19, 24, 25, 26, 33

Path 10 = 1, 2, 9, 10, 17, 18, 19, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33

Path 11 = 1, 2, 9, 10, 17, 18, 19, 24, 25, 26, 27, 28, 29, 31, 32, 33

Path 12 = 1, 2, 9, 10, 17, 18, 19, 24, 25, 26, 27, 32, 33

Path 13 = 1, 2, 9, 10, 17, 18, 19, 24, 25, 26, 33

8. *Predicate Node*

   V(G) = Jumlah node yang memiliki lebih dari 1 jalur + 1

   V(G) = 12 + 1

   V(G) = 13

9. *Graph Matriks*

   Graph Matriks pengujian algoritma A* (A *star*) *level* 4 dapat dilihat pada tabel 4.8.

Tabel 4.8 *Graph Matriks* Algoritma A* (A *star*) *Level* 4

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	
1		1																																
2			1						1																									1
3				1				1																										1
4					1																													
5						1	1																											1
6							1																											
7								1																										
8									1																									
9									1																									
10											1						1																	1
11												1				1																		1
12													1																					
13														1	1																			
14															1																			1
15																1																		
16																	1																	
17																		1																
18																			1						1									1
19																				1				1										1
20																					1													
21																						1	1											1
22																							1											
23																								1										
25																									1									
26																										1							1	1
27																											1					1		1
28																												1						
29																														1	1			1
30																															1			
31																																1		
32																																	1	
33																																		
Hasil																																		12

   V(G) = Jumlah *Graph Matriks* + 1

   V(G) = 12 + 1

   V(G) = 13

10. Kesimpulan

    Berdasarkan hasil pengujian dihasilkan nilai *Cyclomatic Complexity* yang sama yaitu 13, maka dapat disimpulkan bahwa

pada proses pencarian dengan algoritma A* (A *star*) berjalan dengan baik, karena setiap pengujian menghasilkan nilai yang sama.

4.2.1.4 Kesimpulan Pengujian *Alpha*

Dari hasil pengujian yang dilakukan, dapat disimpulkan bahwa sistem sudah berjalan seperti yang diharapkan dan secara fungsional sudah dapat menghasilkan keluaran yang diharapkan.

4.2.2 Pengujian *Beta*

Pengujian *beta* merupakan pengujian yang dilakukan secara objektif, dimana pengujian langsung melibatkan para pengguna aplikasi permainan secara langsung dengan menggunakan kuisioner mengenai kepuasan para pengguna terhadap aplikasi permainan yang dibangun. Pengujian dilakukan terhadap 30 responden dengan usia minimal 7 tahun dan setiap responden diberikan 5 pernyataan untuk kemudian dilakukan penilaian dengan memilih 5 pilihan yang telah disediakan untuk setiap pernyataannya.

4.2.2.1 Kuesioner

Data hasil kuisioner yang telah diberikan dapat dicari presentasinya dari masing-masing jawaban dengan menggunakan rumus[21] :

$$P = \frac{\sum s}{skorideal} \times 100\%$$

dimana,

P = Nilai presentase yang dicari

$\sum s$ = Jumlah skor yang didapatkan dari setiap pilihan jawaban dikalikan dengan nilai yang telah ditetapkan sebelumnya

skorideal = Nilai tertinggi pada point pilihan dikalikan dengan jumlah responden

Data yang didapat dari hasil kuisioner terhadap 30 orang responden adalah sebagai berikut :

1. Tampilan aplikasi permainan *Pirates of Money* secara umum sudah baik.

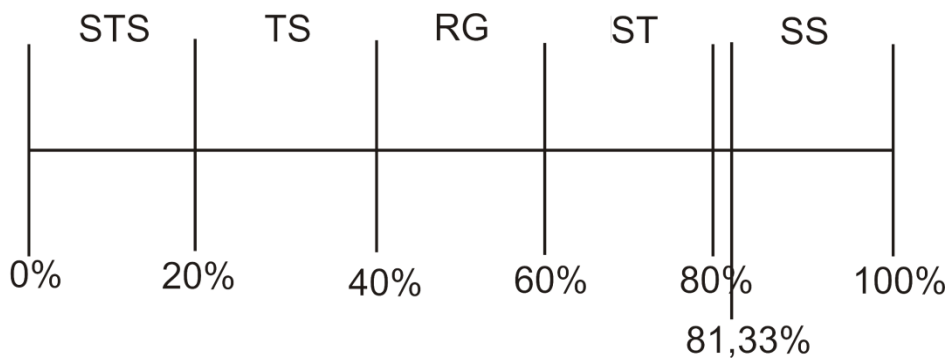
   Hasil kuisioner untuk pernyataan ke-1 dapat dilihat pada tabel 4.9.

Tabel 4.9 Hasil Kuisioner Tampilan Aplikasi Permainan

Pernyataan	Jawaban	Skor	Responden	S
1	Sangat Setuju	5	10	50
	Setuju	4	18	72
	Ragu-Ragu	3	2	6
	Tidak Setuju	2	0	0
	Sangat Tidak Setuju	1	0	0
JUMLAH			**30**	**122**

$$P = \frac{122}{150} \times 100\% = 81.33\%$$

Dapat disimpulkan bahwa penilaian terhadap tampilan secara umur baik tidaknya pada aplikasi permainan *Pirates of Money* adalah 81,67% dari yang diharapkan 100%, atau dapat dikategorikan sebagai sangat setuju, untuk lebih jelasnya dapat dilihat pada gambar 4.14.



Gambar 4.14 Interpretasi Skor Pertanyaan ke-1

2. Musuh yang terdapat pada aplikasi permainan *Pirates of Money* sudah cerdas.

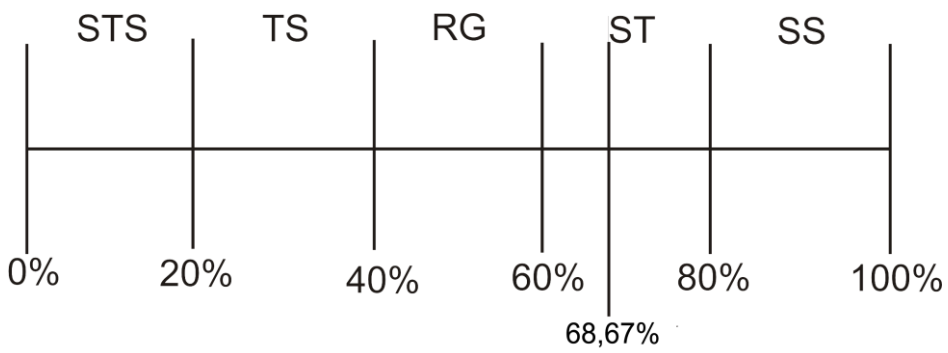
   Hasil kuisioner untuk pernyataan ke-2 dapat dilihat pada tabel 4.10.

Tabel 4.10 Hasil kuisioner kecerdasan musuh pada aplikasi permainan

Pernyataan	Jawaban	Skor	Responden	S
2	Sangat Setuju	5	0	0
	Setuju	4	16	64
	Ragu-Ragu	3	11	33
	Tidak Setuju	2	3	6
	Sangat Tidak Setuju	1	0	0
JUMLAH			**30**	**103**

$$P = \frac{103}{150} \times 100\% = 68.67\%$$

Dapat disimpulkan bahwa penilaian terhadap kecerdasan musuh pada aplikasi permainan *Pirates of Money* adalah 68,67% dari yang diharapkan 100%, atau dapat dikategorikan sebagai setuju, untuk lebih jelasnya dapat dilihat pada gambar 4.15.



Gambar 4.15 Interpretasi Skor Pertanyaan ke-2

3. Deteksi tubrukan antara objek satu dengan objek lainnya pada aplikasi *Pirates of Money* sudah bagus.

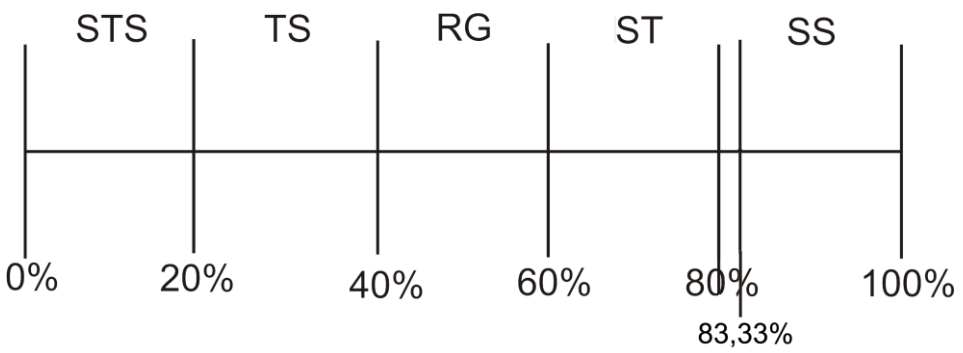
Hasil kuisioner untuk pernyataan ke-3 dapat dilihat pada tabel 4.11.

Tabel 4.11 Hasil kuisioner deteksi tubrukan pada aplikasi permainan

Pernyataan	Jawaban	Skor	Responden	S
3	Sangat Setuju	5	9	45
	Setuju	4	17	68
	Ragu-Ragu	3	4	12
	Tidak Setuju	2	0	0
	Sangat Tidak Setuju	1	0	0
JUMLAH			**30**	**125**

$$P = \frac{125}{150} \times 100\% = 83.33\%$$

Dapat disimpulkan bahwa penilaian terhadap deteksi tubrukan pada aplikasi permainan *Pirates of Money* adalah 83,33% dari yang diharapkan 100%, atau dapat dikategorikan sebagai setuju, untuk lebih jelasnya dapat dilihat pada gambar 4.16.



Gambar 4.16 Interpretasi Pertanyaan ke-3

4. Audio pada aplikasi permainan *Pirates of Money* enak didengar.

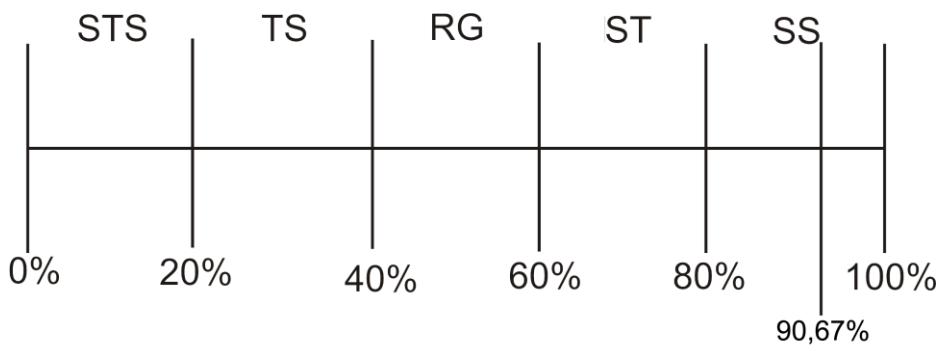
   Hasil kuisioner untuk pernyataan ke-4 dapat dilihat pada tabel 4.12.

Tabel 4.12 Hasil kuisioner audio pada aplikasi permainan

Pernyataan	Jawaban	Skor	Responden	S
4	Sangat Setuju	5	19	95
	Setuju	4	9	36
	Ragu-Ragu	3	1	3
	Tidak Setuju	2	1	2
	Sangat Tidak Setuju	1	0	0
JUMLAH			**30**	**136**

$$P = \frac{136}{150} \times 100\% = 90.67\%$$

Dapat disimpulkan bahwa penilaian terhadap audio pada aplikasi permainan *Pirates of Money* adalah 90,67% dari yang diharapkan 100%, atau dapat dikategorikan sebagai sangat setuju, untuk lebih jelasnya dapat dilihat pada gambar 4.17.



Gambar 4.17 Interpretasi Pertanyaan ke-4

5. Aplikasi permainan *Pirates of Money* menarik untuk dimainkan.

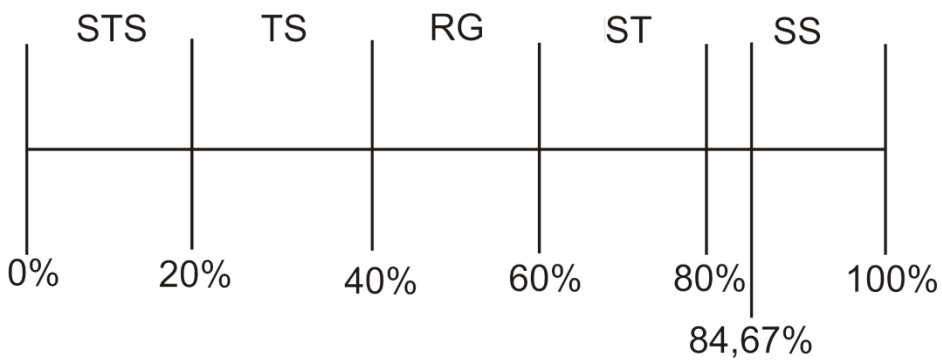
   Hasil kuisioner untuk pernyataan ke-5 dapat dilihat pada tabel 4.13.

Tabel 4.13 Hasil kuisioner ketertarikan pada aplikasi permainan

Pernyataan	Jawaban	Skor	Responden	S
5	Sangat Setuju	5	10	50
	Setuju	4	17	68
	Ragu-Ragu	3	3	9
	Tidak Setuju	2	0	0
	Sangat Tidak Setuju	1	0	0
JUMLAH			**30**	**127**

$$P = \frac{127}{150} \times 100\% = 84.67\%$$

Dapat disimpulkan bahwa penilaian terhadap ketertarikan pada aplikasi permainan *Pirates of Money* adalah 84,67% dari yang diharapkan 100%, atau dapat dikategorikan sebagai sangat setuju, untuk lebih jelasnya dapat dilihat pada gambar 4.18.



Gambar 4.18 Interpretasi Pertanyaan ke-5