BAB IV

IMPLEMENTASI DAN PENGUJIAN

4.1 Implementasi

Proses Implementasi merupakan tahapan dimana aplikasi yang sudah dibuat dan dirancang, diuji kelayakannya. Tahapan ini dilakukan setelah tahap perancangan selesai dilakukan. Adapun hasil dari perancangan tersebut kemudian diimplementasikan pada bahasa pemrograman. Setelah implementasi, maka dilakukan pengujian terhadap aplikasi. Pengujian tersebut terdiri dari pengujian alpha dan beta.

4.1.1 Implementasi Perangkat Keras

Perangkat keras yang digunakan untuk melakukan implementasi dan pengujian yaitu sebagai berikut :

1. Processor Intel Dual Core T4200 2GHz
2. Memori 2 GB
3. VGA card NVIDIA GEFORCE G102M 512MB
4. Monitor 1366 x 768 pixel
5. Mouse dan keyboard
6. Speaker

4.1.2 Implementasi Perangkat Lunak

Perangkat lunak yang digunakan untuk melakukan implementasi dan pengujian aplikasi yaitu sebagai berikut :

1. Sistem Operasi Windows 7
2. Greenfoot
3. JDK (*Java Runtime Environment)* minimal versi 7

4.1.3 Implementasi Antar Muka

Implementasi antarmuka dilakukan dengan menampilkan setiap tampilan sistem yang dibangun dengan pengkodeannya dalam file program. Pada bagian ini akan ditampilkan implementasi antarmuka menu utama, menu petunjuk, menu pengaturan dan menu mulai.

4.1.3.1 Implementasi Antar Muka Menu Utama

Berikut adalah Implementasi Antarmuka Menu Utama yang ditunjukan oleh gambar di bawah ini.



Gambar 4.1. Antarmuka Menu Utama

4.1.3.2 Implementasi Antar Muka Menu Petunjuk

Berikut adalah Implementasi Antarmuka Menu Petunjuk yang ditunjukan oleh gambar 4.2 di bawah ini.



Gambar 4.2. Antarmuka Menu Petunjuk

4.1.3.3 Implementasi Antar Muka Menu Pengaturan

Berikut adalah Implementasi Antarmuka Menu Pengaturan yang ditunjukan oleh gambar 4.3 di bawah ini.



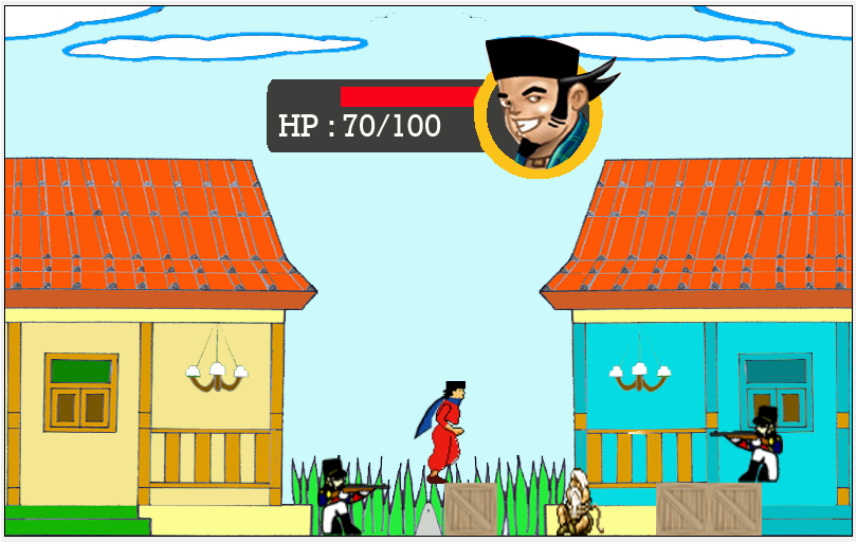
Gambar 4.3. Antarmuka Menu Pengaturan

4.1.3.4 Implementasi Antar Muka Menu Mulai

Berikut adalah Implementasi Antarmuka Menu Mulai yang ditunjukan oleh gambar di bawah ini dimana terdapat antarmuka level 1, antarmuka level 2 dan antarmuka level 3.

1. Antarmuka Level 1

Antarmuka level 1 merupakan tampilan pada saat memainkan level 1. Tampilan antarmuka level 1 dapat dilihat pada gambar 4.4 berikut.



Gambar 4.4. Antarmuka Level 1

2. Antarmuka Level 2

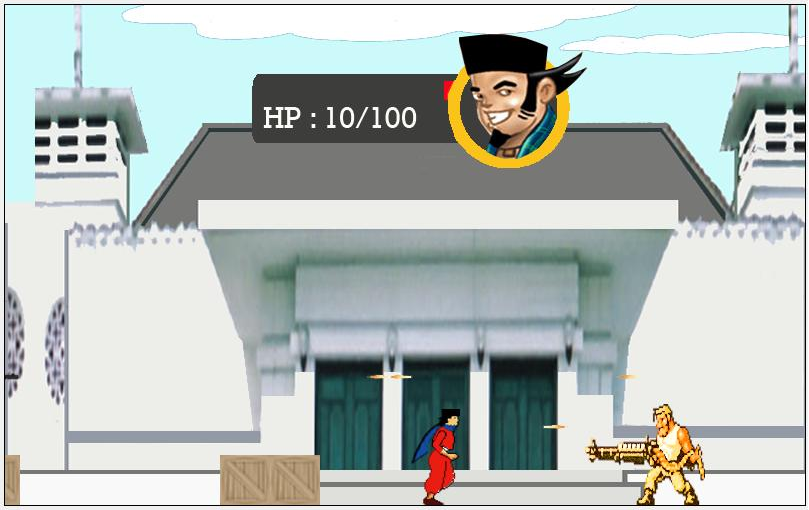
Antarmuka level 2 merupakan tampilan pada saat memainkan level 2. Tampilan antarmuka level 2 dapat dilihat pada gambar 4.5 berikut.



Gambar 4.5. Antarmuka Level 2

3. Antarmuka Level 3

Antarmuka level 3 merupakan tampilan pada saat memainkan level 3. Tampilan antarmuka level 3 dapat dilihat pada gambar 4.6 berikut.



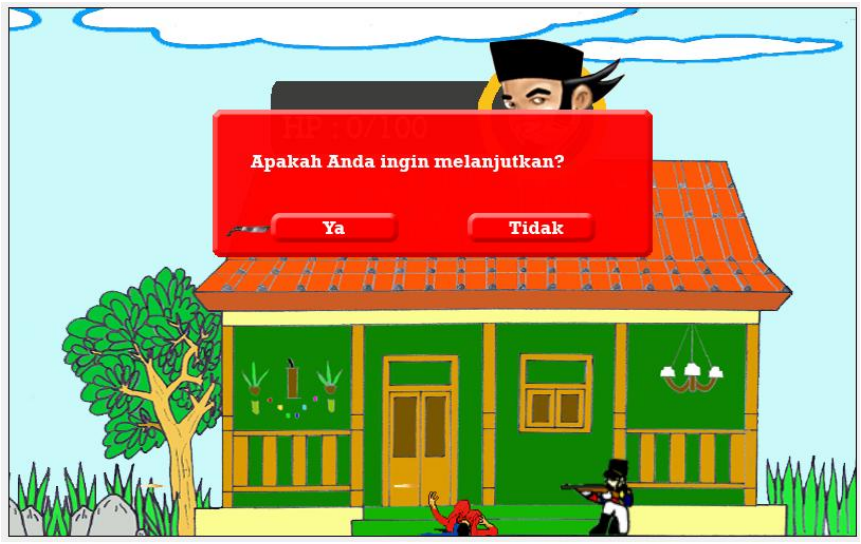
Gambar 4.6. Antarmuka Level 3

4.1.4 Implementasi Pesan

Pada bagian implementasi pesan ini akan ditampilkan implementasi pesan lanjut permianan.

4.1.4.1 Implementasi Pesan Lanjut Permainan

Berikut adalah Implementasi Pesan Lanjut Permainan yang ditunjukkan oleh gambar 4.7 berikut.



Gambar 4.7. Pesan Lanjut Permainan

4.1.4.2 Implementasi Pesan Exit Permainan



Gambar 4.8. Pesan Exit Permainan

4.2 Pengujian Perangkat Lunak

Dalam melakukan pengujian terhadap game Petualangan Si Pitung dilakukan dua macam teknik, yaitu pengujian black-box dan pengujian white-box. Pengujian black-box berfokus pada menemukan kesalahan program, sedangkan pengujian white-box berfokus pada struktur internal (source code) program untuk mengetahui apakah masih terjadi error.

4.2.1 Pengujian *Blackbox*

Pada tahap ini, pengujian dilakukan untuk memastikan aplikasi *game* Petualangan Si Pitung telah berjalan dengan benar dan sesuai dengan kebutuhan yang diinginkan. Pengujian alpha menitikberatkan kepada persyaratan fungsional perangkat lunak.

4.2.1.1 Skenario Pengujian

Skenario pengujian menjelaskan pengujian terhadap sistem yang ada pada game Petualangan Si Pitung. Skenario pengujian dari aplikasi permainan ini dapat dilihat pada tabel 4.1 berikut.

Tabel 4.1. Skenario Pengujian

No	Komponen yang diuji	Skenario pengujian	Hasil Pengujian
1	Menu Utama	Memilih tombol Mulai	*Black Box*
		Memilih tombol Pengaturan	*Black Box*
		Memilih tombol Petunjuk	*Black Box*
2	Keluar Dari Game	Memilih tombol Keluar	*Black box*
3	Memainkan *Level 1*	Karakter bergerak ke depan	*Black box*
		Karakter bergerak ke belakang	*Black box*
		Karakter bergerak ke atas	*Black box*
		Karakter memukul	*Black box*
		Healthpoint karakter berkurang apabila terkena	*Black box*

		senjata musuh	
		Healtpoint bertambah apabila karakter membebaskan tawanan	*Black box*
4		Karakter bergerak ke depan	*Black box*
		Karakter bergerak ke belakang	*Black box*
		Karakter bergerak ke atas	*Black box*
	Memainkan *Level 2*	Karakter memukul	*Black box*
		Healthpoint karakter berkurang apabila terkena senjata musuh	*Black box*
		Healtpoint bertambah apabila karakter membebaskan tawanan	*Black box*
5		Karakter bergerak ke depan	*Black box*
		Karakter bergerak ke belakang	*Black box*
		Karakter bergerak ke atas	*Black box*
	Memainkan *Level* 3	Karakter memukul	*Black box*
		Healthpoint karakter berkurang apabila terkena	*Black box*

		senjata musuh	
		Healtpoint bertambah apabila karakter membebaskan tawanan	*Black box*
		Kembali ke menu utama	*Black box*
6	Lanjut Permainan	Memilih tombol ya atau tidak	*Black box*
7	Exit Permainan	Memilih tombol ya atau tidak	*Black box*
8	Algoritma *A* (A Star)*	Pergerakan pada musuh	*White box*

4.2.1.2 Kasus dan Hasil Pengujian

Berikut ini adalah hasil dari pengujian perangkat lunak yang sudah dibangun menggunakan metode blackbox berdasarkan pengamatan yang sudah dilakukan dan menghasilkan kesimpulan dari pengujian tersebut.

Tabel 4.2 Skenario dan Hasil Pengujian

No	Kasus yang akan diuji	Skenario dan Hasil Pengujian		
		Skenario uji	**Hasil yang diharapkan**	**Kesimpulan**
1.	Menu Utama	Memilih tombol Mulai	Sistem menampilkan level permainan	[√] Berhasil [ ] Tidak berhasil
		Memilih tombol	Sistem menampilk	[√] Berhasil

		Pengaturan	an halaman pengaturan suara	[ ] Tidak berhasil
		Memilih tombol Petunjuk	Sistem menampilkan halaman petunjuk game	[√] Berhasil [ ] Tidak berhasil
2.	Keluar Dari Game	Memilih tombol Keluar	Sistem akan menutup aplikasi	[√] Berhasil [ ] Tidak berhasil
3.	Memainkan *Level 1*	Karakter bergerak ke depan	Sistem menggerakan karakter gerakan depan	[√] Berhasil [ ] Tidak berhasil
		Karakter bergerak ke belakang	Sistem menggerakan karakter gerakan belakang	[√] Berhasil [ ] Tidak berhasil
		Karakter bergerak ke atas	Sistem menggerakan karakter gerakan atas	[√] Berhasil [ ] Tidak berhasil
		Karakter Memukul	Sistem menampilkan point bertambah	[√] Berhasil [ ] Tidak berhasil
		Healthpoint karakter berkurang	Sistem menampilkan	[√] Berhasil

		apabila terkena senjata musuh	healtpoint berkurang	[ ] Tidak berhasil
		Healtpoint bertambah apabila karakter membebaskan tawanan	Sistem menampilkan healtpoint bertambah	[√] Berhasil [ ] Tidak berhasil
4.	Memainkan *Level 2*	Karakter bergerak ke depan	Sistem menggerakan karakter gerakan depan	[√] Berhasil [ ] Tidak berhasil
		Karakter bergerak ke belakang	Sistem menggerakan karakter gerakan belakang	[√] Berhasil [ ] Tidak berhasil
		Karakter bergerak ke atas	Sistem menggerakan karakter gerakan atas	[√] Berhasil [ ] Tidak berhasil
		Karakter Memukul	Sistem menampilkan point bertambah	[√] Berhasil [ ] Tidak berhasil
		Healthpoint karakter berkurang apabila terkena senjata musuh	Sistem menampilkan healtpoint berkurang	[√] Berhasil [ ] Tidak berhasil

		Healtpoint bertambah apabila karakter membebaskan tawanan	Sistem menampilkan healtpoint bertambah	[√] Berhasil [ ] Tidak berhasil
5.	Memainkan Level 3	Karakter bergerak ke depan	Sistem menggerakan karakter gerakan depan	[√] Berhasil [ ] Tidak berhasil
		Karakter bergerak ke belakang	Sistem menggerakan karakter gerakan belakang	[√] Berhasil [ ] Tidak berhasil
		Karakter bergerak ke atas	Sistem menggerakan karakter gerakan atas	[√] Berhasil [ ] Tidak berhasil
		Karakter Memukul	Sistem menampilkan point bertambah	[√] Berhasil [ ] Tidak berhasil
		Healthpoint karakter berkurang apabila terkena senjata musuh	Sistem menampilkan healtpoint berkurang	[√] Berhasil [ ] Tidak berhasil
		Healtpoint bertambah apabila karakter membebaskan	Sistem menampilkan healtpoint bertambah	[√] Berhasil [ ] Tidak berhasil

		tawanan		
		Kembali ke menu utama	Sistem menampilkan menu utama	[√] Berhasil [ ] Tidak berhasil
6.	Lanjut Permainan	Memilih tombol ya atau tidak	Sistem menampilkan menu utama atau lanjut permainan	[√] Berhasil [ ] Tidak berhasil
7	Exit Permainan	Memilih tombol ya atau tidak	Sistem menampilkan menu utama lanjut permainan	

4.2.1.3 Kesimpulan Pengujian *Blackbox*

Berdasarkan hasil pengujian terhadap butir uji yang telah dilakukan memberikan kesimpulan bahwa proses sudah benar. Pesan kesalahan yang ada sudah tampil cukup maksimal. Secara fungsionalitas game sudah dapat menghasilkan output yang diharapkan.

4.2.1.4 Pengujian *Whitebox*

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. Serta untuk menguji alur logika, perulangan, dan validasi parameter masukannya. Di bawah ini merupakan kasus Algoritma A*(A Star) untuk menguji perangkat lunak yang sudah dibangun dengan menggunakan

metode white-box, berdasarkan skenario pengujian aplikasi yang sudah dijabarkan.

Tabel *Source Code* A* (A-Star)

	Pseudocode
	public void astar(){
1.	int k = 0;
2.	if(path.size()==0){
3.	while(path.size()<100){
4.	cekNode();
	}
5.	}else{
6.	if(path.get(0).nx()>hero.get(0).getX()){
7.	moveLeft();
8.	}else if(path.get(0).nx()<hero.get(0).getX()){
9.	moveRight();
	}
	}
	}
	public void cekNode(){
10.	intgoalx = hero.get(0).getX(), goaly = hero.get(0).getY();
11.	if(open.size()==0){
12.	intnx = this.getX(), ny = this.getY();
	intgn = 0;
13.	for(int i=0;i<9;i++){
14.	if(i==0){ //tengah
	gn += 1;

```
            open.add(new node(nx,ny,gn,goalx,goaly));
        }
        else if(i==1){ //kanan
            open.add(new node(nx+1,ny,gn,goalx,goaly));
        }
        else if(i==2){ //kananatas
        open.add(new node(nx+1,ny-1,gn,goalx,goaly));
        }
        else if(i==3){ //atas
        open.add(new node(nx,ny-1,gn,goalx,goaly));
        }
        else if(i==4){ //kiriatas
            open.add(new node(nx-1,ny-1,gn,goalx,goaly));
        }
        else if(i==5){ //kiri
        open.add(new node(nx-1,ny,gn,goalx,goaly));
        }
        else if(i==6){ //kiribawah
        open.add(new node(nx-1,ny+1,gn,goalx,goaly));
        }
        else if(i==7){ //bawah
        open.add(new node(nx,ny+1,gn,goalx,goaly));
        }
        else if(i==8){ //kananbawah
        open.add(new node(nx+1,ny+1,gn,goalx,goaly));
        }
        }
```

```
        }
15.     else if(open.size()!=0){
16.     int f=0;
        intnx=0, ny=0, gn=0;
//pengecekan cost terkecil
//============================================================
=========
17.
        for(int k=0;k<open.size();k++){
18.
            if(f==0){
                f = open.get(k).f();
            nx = open.get(k).nx();
        ny = open.get(k).ny();
            gn = open.get(k).gn();
        }else{
            if(f >open.get(k).f()){
            f = open.get(k).f();
                        nx = open.get(k).nx();
        ny = open.get(k).ny();
        gn = open.get(k).gn();
            }
            }
            }
//============================================================
========
19.
//best node disimpandalam array path
            path.add(new node(nx,ny,gn,goalx,goaly));
20.         open.clear();
21.         for(int i=0;i<9;i++){
```

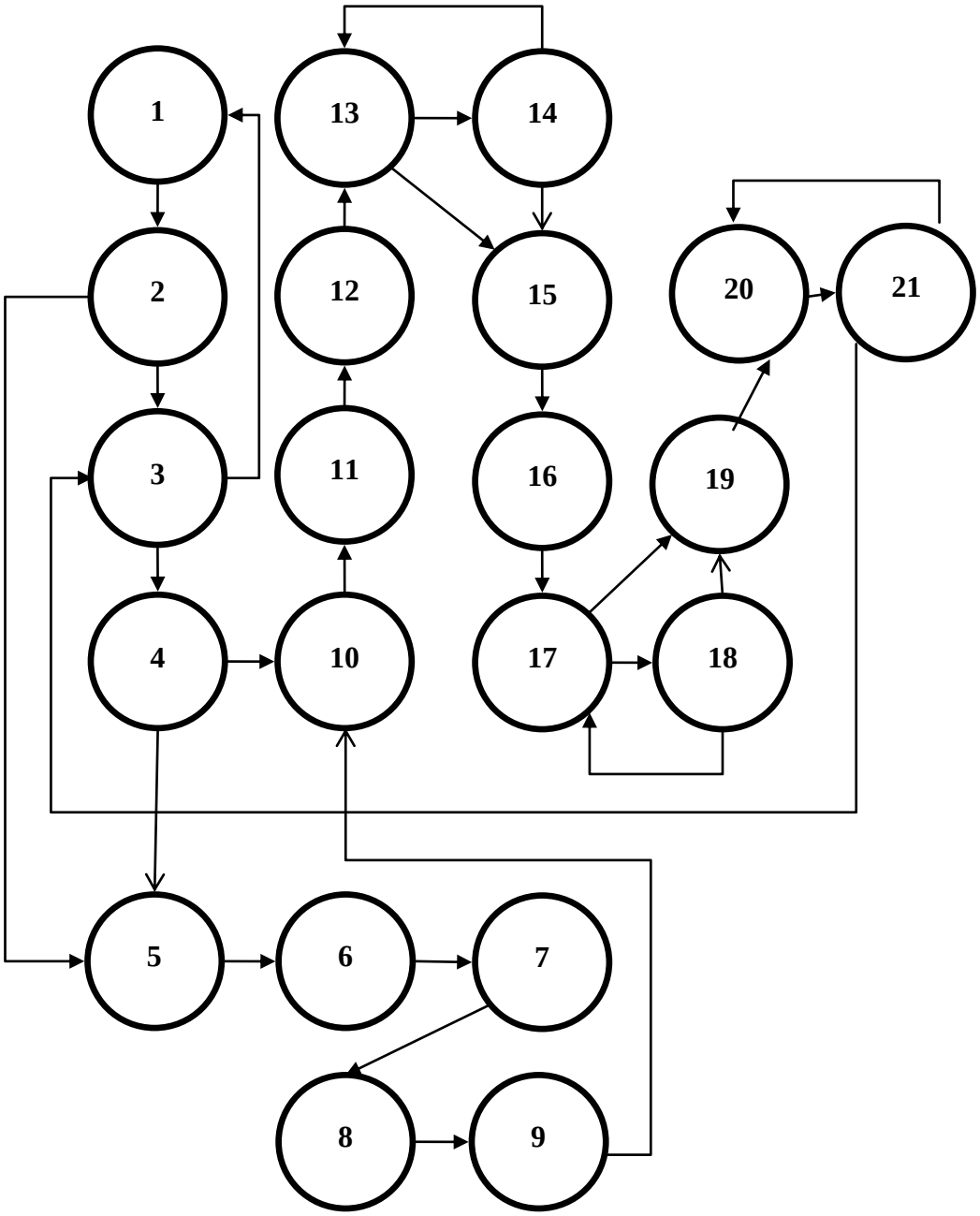
```
            if(i==0){ //tengah
            gn += 1;
            open.add(new node(nx,ny,gn,goalx,goaly));
        }
        else if(i==1){ //kanan
        open.add(new node(nx+1,ny,gn,goalx,goaly));
        }
        else if(i==2){ //kananatas
        open.add(new node(nx+1,ny-1,gn,goalx,goaly));
        }
        else if(i==3){ //atas
        open.add(new node(nx,ny-1,gn,goalx,goaly));
        }
        else if(i==4){ //kiriatas
        open.add(new node(nx-1,ny-1,gn,goalx,goaly));
        }
        else if(i==5){ //kiri
        open.add(new node(nx-1,ny,gn,goalx,goaly));
        }
        else if(i==6){ //kiribawah
        open.add(new node(nx-1,ny+1,gn,goalx,goaly));
        }
        else if(i==7){ //bawah
        open.add(new node(nx,ny+1,gn,goalx,goaly));
        }
        else if(i==8){ //kananbawah
        open.add(new node(nx+1,ny+1,gn,goalx,goaly));
```

```
            }
                    }
            }
}
```

Tabel 4.3. *Source Code* Algoritma A* (A-Star)

1. Flow Graph

Berikut merupakan Flow Graphdari algoritma A* (A-Star) yang dapat dilihat pada gambar 4.8 berikut.



Gambar 4.8. Flow Graph A* (A-Star)

Keterangan :

= Menggambarkan kondisi

= Menggambarkan Aksi

1. Cyclomatic Complexity
   E = 25, N = 21
   V(G) = E – N + 2
   V(G) = 25 – 21 + 2
   V(G) = 6
   Keterangan :
   E : Jumlah busur pada flowgraph
   N : Jumlah simpul pada flowgraph
   V(G) : Cyclomatic Complexity

2. Independent Path
   Path 1 : 1-2-3-4-5-6-7-8-9-10-11-12-13-14-13-15-16-17-18-17-19-20-21
   Path 2 : 1-2-3-4-5-6-7-8-9-10-11-12-13-14-15-16-17-18-19-20-21
   Path 3 : 1-2-3-4-5-6-7-8-9-10-11-12-13-15-16-17-18-19-20-21
   Path 4 : 1-2-3-4-5-6-7-8-9-10-11-12-13-14-15-16-17-19-20-21
   Path 5 : 1-2-5-6-7-8-9-10-11-12-13-15-16-17-19-20-21
   Path 6 : 1-2-3-4-10-11-12-13-15-16-17-19-20-21

3. Graph Matriks
   Berikut ini adalah *graph* matriks dari algoritma A*(A-Star) yang dapat dilihat pada Table 4.4.

Tabel 4.4 Graph Matriks

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	N = (E+1)
1		1																				
2			1		1																	1
3				1																		
4					1					1												1
5						1																
6							1															
7								1														
8									1													
9										1												
10											1											
11												1										

12												1									
13													1	1							1
14														1							
15															1						
16																1					
17																	1	1			1
18																1		1			1
19																			1		
20																				1	
Jumlah																					N = 5+1 = 6

Keterangan :

1. Baris dan kolom merepresentasikan simpul
2. Nilai 1 merepresentasikan adanya keterhubungan antar simpul

2. Kesimpulan

Dari hasil pengujian algoritma A* (A Star) dari entitas NPC yang dilakukan, didapat kesimpulan bahwa seluruh node yang ada pada setiap independent path telah dikerjakan, serta tidak ada pengulangan tak berhingga.

4.2.2 Pengujian Beta

Pengujian beta merupakan pengujian yang dilakukan secara objektif, dimana dilakukan pengujian secara langsung terhadap pengguna dengan menggunakan kuesioner mengenai aplikasi yang telah dibangun. Adapun metode penilaian pengujian yang digunakan adalah metode kuantitatif berdasarkan data dari pengguna.

4.2.2.1 Kuesioner

Kuesioner disebarkan menggunakan teknik Simple Random Sampling dimana anggota sample dari populasi dilakukan secara acak tanpa memperhatikan strata yang ada dalam populasi. Kuesioner disebarkan kepada 50 orang dari umur 6 tahun ke atas, dari hasil kuesioner tersebut akan dilakukan perhitungan agar dapat diambil kesimpulan terhadap penilaian penerapan sistem yang telah dibangun. Kuesioner terdiri dari 5 pernyataan dengan menggunakan skala 1 sampai 4. Untuk mencari nilai persentase dari masing-masing jawaban digunakan rumus sebagai berikut :

$$P = \frac{S}{skor\ ideal} \times 100$$

Keterangan :

P = Nilai persentase yang dicari

S = Jumlah frekuensi dikalikan dengan nilai yang ditetapkan jawaban

Skor ideal = Nilai tertinggi dikalikan dengan jumlah sampel

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

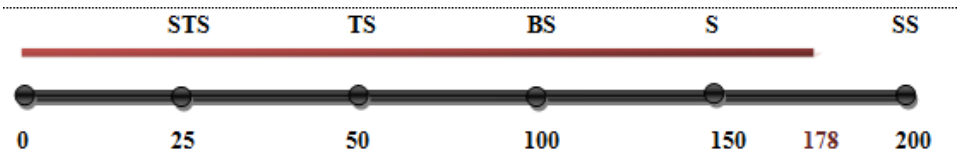
1. Apakah dengan *game* Petualngan Si Pitung ini sudah membantu memperkenalkan karakter Si Pitung ?

Tabel 4.5. Perhitungan pesentase pertanyaan ke-1

Pertanyaan	Keterangan	Skor	Frekuensi	S
1	Sangat Setuju	4	32	128
	Setuju	3	16	48
	Biasa Saja	2	0	0
	Tidak Setuju	1	2	2
	Sangat Tidak Setuju	0	0	0
Jumlah			**50**	**178**

P = 178 / 200 x 100% = 89%

Dapat disimpulkan bahwa penilaian terhadap pengenalan tokoh Si Pitung yang ada pada *game* Petualangan Si Pitung secara umum adalah 89% dari yang diharapkan 100%, atau dapat dikategorikan sebagai sangat setuju, Berikut kriteria interpretasi skor yang dapat dilihat pada gambar 4.9 :



Gambar 4.9. Interpretasi Skor Pertanyaan ke-1

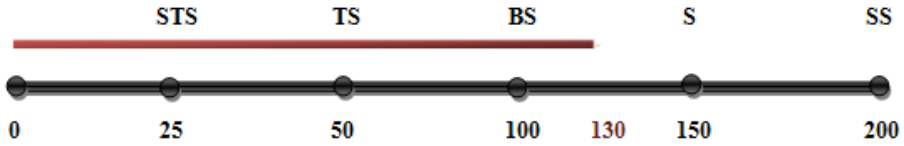
2. Apakah anda setuju bahwa penyampaian cerita Si Pitung dalam *game* Petualngan Si Pitung ini dapat anda pahami dengan baik?

Tabel 4.6. Perhitungan persentase pertanyaan ke-2

Pertanyaan	Keterangan	Skor	Frekuensi	S
2	Sangat Setuju	4	2	8
	Setuju	3	32	96
	Biasa Saja	2	10	20
	Tidak Setuju	1	6	6
	Sangat Tidak Setuju	0	0	0
Jumlah			**50**	**130**

P = 130 / 200 x 100% = 65%

Berdasarkan hasil persentase diatas, maka dapat disimpulkan bahwa penilaian terhadap penyampaian cerita tokoh Si Pitung setelah memainkan permainan ini adalah 65% dari yang diharapkan 100%, atau dapat dikategorikan sebagai setuju. Berikut kriteria interpretasi skor yang dapat dilihat pada gambar 4.10 :



Gambar 4.10. Interpretasi Skor Pertanyaan ke-2

3. Apakah Aplikasi *game* ini menarik untuk dimainkan?

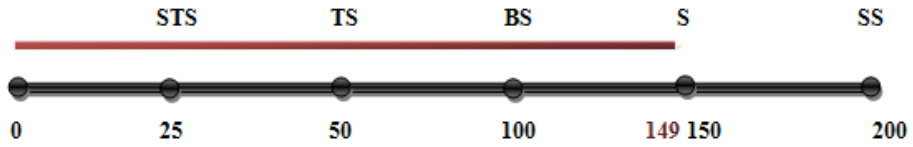
Tabel 4.7. Perhitungan persentase pertanyaan ke-3

Pertanyaan	Keterangan	Skor	Frekuensi	S
3	Sangat Setuju	4	4	16
	Setuju	3	43	129

	Biasa Saja	2	1	2
	Tidak Setuju	1	2	2
	Sangat Tidak Setuju	0	0	0
Jumlah			**50**	**149**

P = 149 / 200 x 100% = 74,5%

Berdasarkan hasil persentase diatas, maka dapat disimpulkan bahwa penilaian terhadap pengetahuan mengenal cerita Si Pitung dengan adanya permainan ini adalah 74,5% dari yang diharapkan 100%, atau dapat dikategorikan sebagai setuju. Berikut kriteria interpretasi skor yang dapat dilihat pada gambar 4.11 :



Gambar 4.11. Interpretasi Skor Pertanyaan ke-3

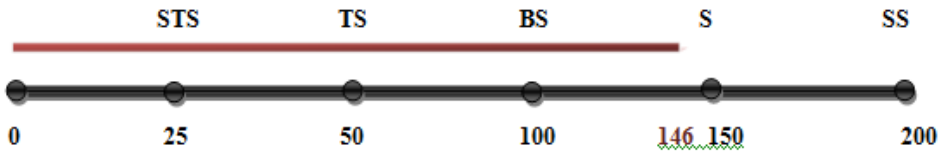
4. Apakah setelah memainkan *game* Petualngan Si Pitung anda mengetahui tentang cerita Si Pitung?

Tabel 4.8. Perhitungan persentase pertanyaan ke-4

Pertanyaan	**Keterangan**	**Skor**	**Frekuensi**	**S**
4	Sangat Setuju	4	6	24
	Setuju	3	36	108
	Biasa Saja	2	6	12
	Tidak Setuju	1	2	2
	Sangat Tidak Setuju	0	0	0
Jumlah			**50**	**146**

P = 146 / 200 x 100% = 73%

Berdasarkan hasil persentase diatas, maka dapat disimpulkan bahwa penilaian terhadap tertariknya memainkan permainan ini adalah 73% dari yang diharapkan 100%, atau dapat dikategorikan sebagai setuju. Berikut kriteria interpretasi skor yang dapat dilihat pada gambar 4.12 :



Gambar 4.12. Interpretasi Skor Pertanyaan ke-4

5. Apakah setelah memainkan *game* Petualangan Si Pitung anda menjadi lebih tertarik untuk mengenal tokoh *RobinHood* Indonesia yaitu Si Pitung?

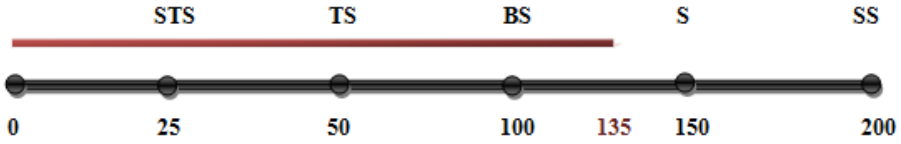
Tabel 4.9. Perhitungan persentase pertanyaan ke-5

Pertanyaan	**Keterangan**	**Skor**	**Frekuensi**	**S**
5	Sangat Setuju	4	1	4
	Setuju	3	35	105
	Kurang Setuju	2	12	24
	Tidak Setuju	1	2	2
	Sangat Tidak Setuju	0	0	0
Jumlah			**50**	**135**

P = 135 / 200 x 100% = 67,5%

Berdasarkan hasil persentase diatas, maka dapat disimpulkan bahwa penilaian terhadap tertariknya memainkan permainan ini adalah 67,5% dari yang

diharapkan 100%, atau dapat dikategorikan sebagai setuju. Berikut kriteria interpretasi skor yang dapat dilihat pada gambar 4.13 :



Gambar 4.13. Interpretasi Skor Pertanyaan ke-5

4.2.2.2 Kesimpulan Hasil Kuesioner

Hasil dari pengujian beta (kuesioner) dapat disimpulkan bahwa :

1. *Game* Petualangan Si Pitung sudah membantu memperkenalkan karater Si Pitung.
2. Penyampaikan cerita Si Pitung dalam *game* Petualangan Si Pitung sudah cukup dipahami dengan baik.
3. *Game* Petualangan Si Pitung sudah cukup menarik untuk dimainkan.
4. Pemain dapat lebih mengetahui cerita Si Pitung setelah memainkan game Petualangan Si Pitung.
5. Setelah memainkan game Petualangan Si Pitung pemain sudah cukup mengenal tokoh Si Pitung.