BAB 4

IMPLEMENTASI DAN PENGUJIAN SISTEM

4.1. Implementasi Sistem

Pada bab ini merupakan tahap implementasi dari aplikasi permainan *The Legend Of Cursed Amulet*. Berdasarkan hasil analisis pada bab sebelumnya kedalam bahasa pemograman yang dapat dimengerti oleh komputer. Setelah tahap implementasi, maka akan dilakukan tahap pengujian terhadap aplikasi permainan *The Legend Of Cursed Amulet*.

4.2. Perangkat Pendukung yang Digunakan

Dalam proses pembuatan aplikasi ini, tentunya membutuhkan perangkat keras (*Hardware*) dan perangkat lunak (*Software*). Berikut merupakan penjelasan dari perangkat keras dan perangkat lunak yang digunakan.

4.2.1. Perangkat Keras (*Hardware*)

Perangkat keras yang digunakan selama pengujian aplikasi ini tidak harus yang berspesifikasi tinggi. Spesifikasi minimum yang dapat digunakan untuk menjalankan aplikasi permainan *The Legend Of Cursed Amulet*.

Spesifikasi perangkat keras:

1. Prosesor : 2.20 GHz
2. Memory : 2048MB RAM
3. VGA : 756MB
4. Hardisk : 320GB
5. Monitor
6. Mouse dan Keyboard.

4.2.2. Perangkat Lunak (*Software*)

Perangkat lunak yang digunakan selama pembangunan aplikasi ini memiliki spesifikasi sebagai berikut :

a. Sistem Operasi Windows 7
b. Greenfoot 2.2.1

4.2.3. Implementasi Kelas

Pada Tabel 4.1. merupakan keterangan tentang implementasi kelas khusus *boundary*, *control*, dan *entity* yang terdapat pada aplikasi permainan *The Legend Of Cursed Amulet*.

Tabel 4.1. Implementasi Kelas

Nama Kelas	Jenis	File Fisik
Player	*Entity*	Player.java
NPC	*Entity*	NPC.java
NPCNEW	*Entity*	NPCNEW.java
NPCFINAL	*Entity*	NPCFINAL.java
TileMap	*Entity*	TileMap.java
Sound	*Entity*	Sound.java
MusicPlayer	*Entity*	MusicPlayer.java
AStarPathFinder	*Entity*	AStarPathFinder.java
Path	*Entity*	Path.java
Intro	*Boundary*	Intro.java
Instructions_Menu	*Boundary*	Instructions_Menu.java
Menu	*Boundary*	Menu.java
GameWorldManager	*Boundary*	GameWorldManager.java
Animation	*Boundary*	Animation.java
Sprite	*Boundary*	Sprite.java
myWorld	*Control*	myWorld.java
Start_Button	*Control*	Start_Button.java
Instructions_Button	*Control*	Instructions_Button.java
Menu_Button	*Control*	Menu_Button.java

4.2.4. Implementasi Antarmuka

1. Tampilan Menu Utama

Layar ini berfungsi untuk tampilan menu utama yang akan tampil apabila pemain membuka aplikasi permainan *The Legend Of Cursed Amulet*. Terdapat dua buah tombol, tombol mulai untuk menampilkan intro dan tombol petunjuk untuk masuk menu petunjuk. Tampilan menu utama terdapat pada gambar 4.1.



Gambar 4.1. Tampilan Menu Selamat Datang

2. Tampilan Petunjuk

Layar ini berfungsi menampilkan petunjuk. Didalam menu petunjuk terdapat langkah-langkah untuk menentukan tombol yang digunakan dalam memainkan aplikasi permainan *The Legend Of Cursed Amulet*. Tampilan petunjuk terdapat pada gambar 4.2.



Gambar 4.2. Tampilan Petunjuk

3. Tampilan Intro

Layar ini berfungsi menampilkan intro pada permainan, apabila pemain memilih tombol mulai pada aplikasi permainan *The Legend Of Cursed Amulet*. Pada layar intro terdapat text yang akan menampilkan alur cerita pada aplikasi permainan *The Legend Of Cursed Amulet*. Tampilan intro terdapat pada gambar 4.3.



Gambar 4.3. Tampilan Intro

4. Tampilan *Level* 1

Layar ini berfungsi menampilkan arena *level* 1, terdapat NPC yang akan menyerang karakter hero dan mengurangi health poin apabila hero terkena serangan. Pada arena *level* 1 juga terdapat item-item yang akan membantu pemain dalam menghadapi rintangan pada aplikasi permainan *The Legend Of Cursed Amulet*. Tampilan menu utama terdapat pada gambar 4.4.



Gambar 4.4. Tampilan *Level* 1

5. Tampilan *Game OVER*

Layar ini berfungsi menampilkan tampilan permainan berakhir. Pada layar tercantum skor tertinggi yang akan di akumulasikan pada akhir permainan *The Legend Of Cursed Amulet*. Tampilan *game OVER* terdapat pada gambar 4.5.



Gambar 4.5. Tampilan *Game OVER*

4.3. Pengujian Sistem

Pengujian sistem merupakan hal terpenting yang dilakukan untuk menemukan kekurangan atau kesalahan pada perangkat lunak yang diuji. Dalam melakukan pengujian terhadap aplikasi permainan permainan *The Legend Of Cursed Amulet* dilakukan dengan dua cara, 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 beta.

4.3.1. Pengujian Alpha (Fungsional)

Pengujian dilakukan untuk memastikan aplikasi permainan *The Legend Of Cursed Amulet* telah berjalan dengan benar dan sesuai dengan kebutuhan yang diinginkan. Pengujian alpha berfokus kepada persyaratan fungsional perangkat lunak.

4.3.1.1. Rencana Pengujian

Rencana pengujian yang akan dilakukan pada aplikasi ini selengkapnya terlihat pada tabel 4.2.

Tabel 4.2. Rencana Pengujian Aplikasi

No.	Komponen yang diuji	Detail Pengujian	Jenis Uji
1.	Menu	Memilih Tombol Mulai	*Black box*
		Memilih Tombol Petunjuk	*Black box*
2.	Petunjuk	Menampilkan Halaman Petunjuk	*Black box*
		Memilih Tombol Kembali	*Black box*
3.	Intro	Menampilkan Halaman Intro	*Black box*
		Menampilkan Text	*Black box*
		Mengaktifkan Musik	*Black box*
		Menekan Spasi untuk skip Text	*Black box*
		Menekan Enter untuk memasuki arena *Level* 1	*Black box*

4.	Arena *Level* 1	Menampilkan arena *Level* 1	*Black box*
		Menampilkan Health	*Black box*
		Menampilkan Score	*Black box*
		Menampilkan cek *level*	*Black box*
		Menampilkan Karakter Hero	*Black box*
		Menampilkan Karakter NPC	*Black box*
		Menampilkan Item	*Black box*
		Menekan tombol spasi untuk melompat	*Black box*
		Menekan tombol kanan untuk bergerak ke kanan	*Black box*
		Menekan tombol kiri untuk bergerak ke kiri	*Black box*
		Menekan tombol "B" untuk menembak	*Black box*
5.	*Game OVER*	Menampilkan Halaman *Game OVER*	*Black box*
		Menampilkan Skor Tertinggi	*Black box*
6	*Collision detection*	Deteksi Tabrakan Entitias Peluru Dengan Entitas Musuh	*White Box*
		Deteksi Tabrakan Entitas Hero Dengan Entitas Musuh	*White Box*
7	Algoritma A* (*A star*)	Algoritma A* Pada *Level* 3	*White Box*
		Algoritma A* Pada *Level* 4	*White Box*

4.3.1.2. Kasus dan hasil pengujian

Berdasarkan hasil uji perangkat lunak yang telah disusun, maka dapat dilakukan hasil uji kemudian diamati dan menghasilkan kesimpulan dapat dilihat pada tabel 4.3.

Tabel 4.3. Kasus dan Hasil Uji

Kasus dan hasil uji (data normal)				
No.	**Kasus /diuji**	**Skenario Uji**	**Hasil yang diharapkan**	**Hasil Pengujian**
1.	Menu Utama	Memilih Tombol Mulai	Menampilkan halaman Intro	[√] diterima [ ] ditolak
		Memilih Tombol Petunjuk	Menampilkan halaman Menu Petunjuk	[√] diterima [ ] ditolak
2.	Intro	Menampilkan Halaman Intro	Intro Tampil	[√] diterima [ ] ditolak
		Mengaktifkan Musik	Suara Musik Aktif/Hidup	[√] diterima [ ] ditolak
		Menampilkan Text	Text Aktif/Hidup	[√] diterima [ ] ditolak
		Menekan Tombol Spasi	Skip Text	[√] diterima [ ] ditolak
		Menekan Tombol Enter	Menampilkan halaman *Level* 1	[√] diterima [ ] ditolak
3.	Petunjuk	Menampilkan Halaman Petunjuk	Petunjuk Tampil	[√] diterima [ ] ditolak

		Menekan Tombol Kembali	Kembali ke menu utama	[√] diterima [ ] ditolak
4.	*Level* 1	Menampilkan arena *Level* 1	arena *Level* 1Tampil	[√] diterima [ ] ditolak
		Menampilkan Health	Health Tampil	[√] diterima [ ] ditolak
		Menampilkan Score	Score Tampil	[√] diterima [ ] ditolak
		Menampilkan cek *level*	cek *level* Tampil	[√] diterima [ ] ditolak
		Menampilkan Karakter Hero	Karakter Hero Tampil	[√] diterima [ ] ditolak
		Menampilkan Karakter NPC	Karakter NPC Tampil	[√] diterima [ ] ditolak
		Menampilkan Item	Item Tampil	[√] diterima [ ] ditolak
		Menekan tombol spasi untuk melompat	Karakter Melompat	[√] diterima [ ] ditolak
		Menekan tombol kanan untuk bergerak ke kanan	Karakter bergerak ke kanan	[√] diterima [ ] ditolak
		Menekan tombol kiri untuk bergerak ke kiri	Karakter bergerak ke kiri	[√] diterima [ ] ditolak

		Menekan tombol “B” untuk menembak	Karakter menembak	[√] diterima [ ] ditolak
5.	*Game OVER*	Menampilkan Halaman *Game OVER*	*Game OVER* Tampil	[√] diterima [ ] ditolak
		Menampilkan Skor Tertinggi	Skor Tertinggi Tampil	[√] diterima [ ] ditolak

4.3.1.3. Kesimpulan Hasil 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.

4.3.1.4. Hasil 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 tubrukan antara sprite dengan sprite.

a. Pseudo Code *Collision* pada aplikasi permainan *The Legend Of Cursed Amulet*, dapat dilihat pada tabel 4.4.

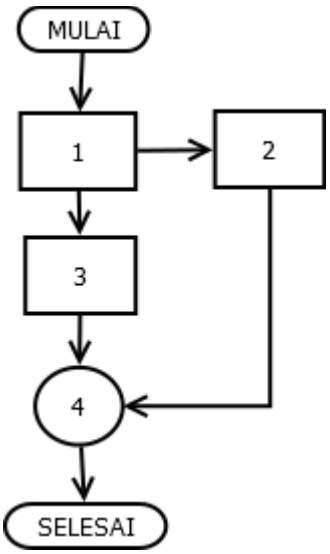
Tabel 4.4. Pseudo code *collision*

1 2	If sprite1_X < (sprite2_x + sprite2_lebar) AND Sprite2_X < (sprite1_X + sprite1_lebar) AND Sprite1_Y < (sprite2_Y + sprite2_lebar) AND Sprite2_Y < (sprite1_Y +sprite1_lebar)

3	*Collision* terjadi Else 4 *Collision* tidak terjadi Else if

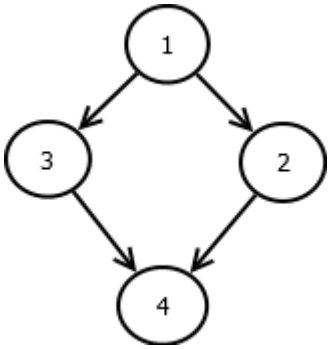
b. *Collision detection* Peluru dengan Musuh

Pengujian dilakukan berdasarkan pada tabel 4.3, 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

c. 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 Patch

   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

d. 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 penelusuran algoritma A* pada musuh

Pengujian proses algoritma A* merupakan proses pengujian dari pengimplementasian algoritma A* menggunakan pengujian whitebox, dan menguji apakah algoritma A* (*A star*) berjalan dengan seharusnya pada musuh.

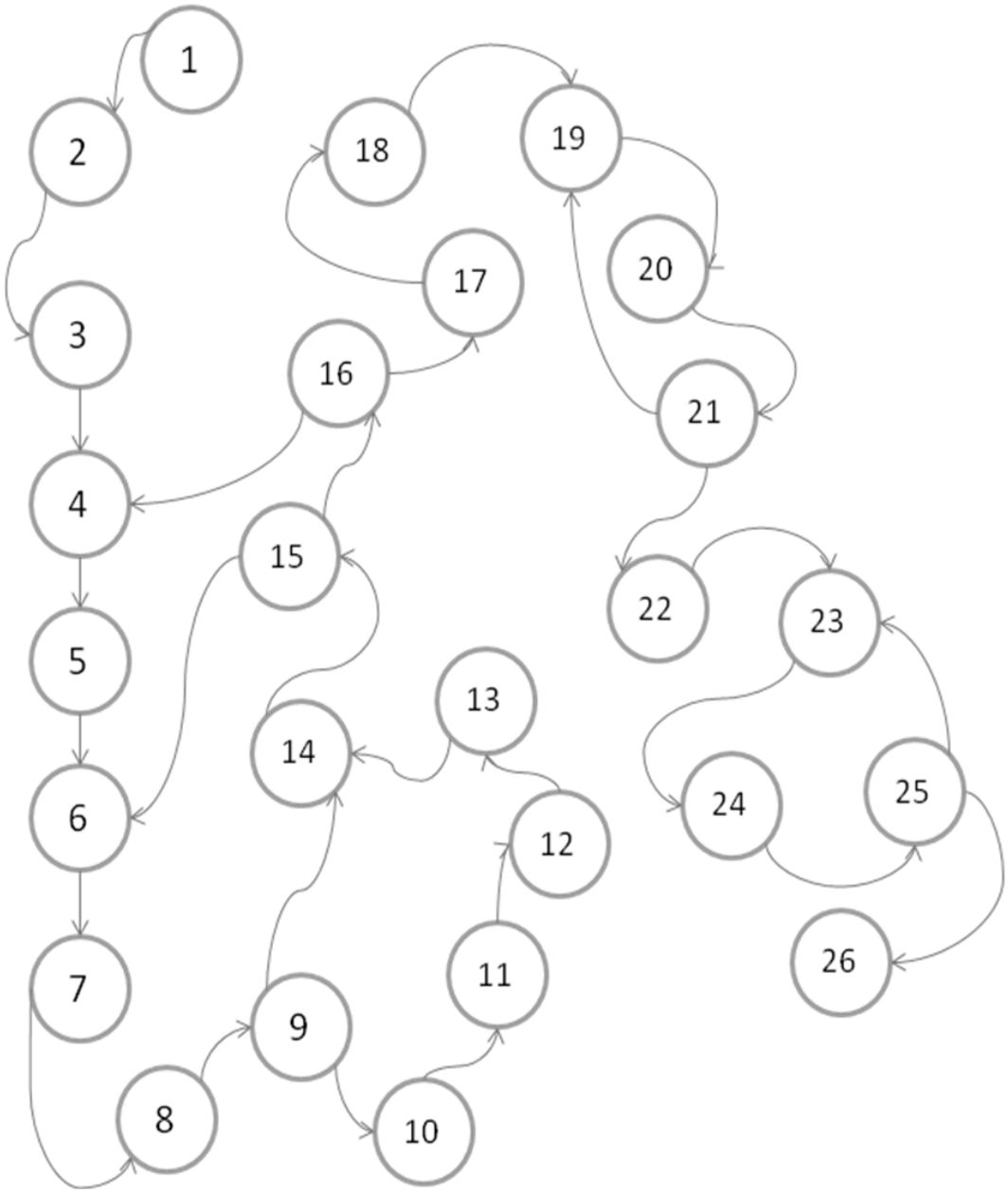
a. Pseudo Algoritma A*

Tabel 4.6. Pseudo Algoritma A*

```
1   create the OPEN LIST of nodes, initially containing
2   only our starting node
3      create the CLOSED LIST of nodes, initially empty
4      while (we have not reached our goal) {
5          consider the best node in the OPEN LIST (the
6   node with the lowest f value)
7          if (this node is the goal) {
8              then we're done
           }
9          else {
               move the current node to the CLOSED LIST
10  and consider all of its neighbors
               for (each neighbor) {
11
                   if (this neighbor is in the CLOSED
12  LIST and our current g value is lower) {
13
                       update the neighbor with the
14  new, lower, g value
15
                       change the neighbor's parent to
16  our current node
                   }
17
                   else if (this neighbor is in the
18  OPEN LIST and our current g value is lower) {
19
                       update the neighbor with the
20  new, lower, g value
                       change the neighbor's parent to
```

```
21  our current node
22                  }
                    else this neighbor is not in either
    the open or CLOSED LIST {
23
                        add the neighbor to the OPEN
24  LIST and set its g value
25                  }
26              }
            }
        }
```

b. *FLOW GRAPH*



Gambar 4.8. *FLOW GRAPH* A *

Keterangan :

= Menggambarkan kondisi

= Menggambarkan aksi

1. Cyclomatic Complexity (V(G)

   V(G) = E – N + 2

   V(G) = 30 – 26 + 2

   V(G) = 4 + 2

   V(G) = 6

   Di mana :

   E = jumlah edge pada grafik alir

   N = jumlah node pada grafik alir

2. Independent path

   Path 1 = 1-2-3-4-5-6-7-8-9-10-14-15-16-17-18-19-20-21-22-23-24-25-26

   Path 2 = 1-2-3-4-5-6-7-8-9-11-12-13-14-15-16-17-18-19-20-21-22-23-24-25-26

   Path 3 = 1-2-3-4-5-6-7-8-9-10-14-15-6-7-8-9-10-14-15-16-17-18-19-20-21-22-23-24-25-26

   Path 4 = 1-2-3-4-5-6-7-8-9-10-14-15-16-4-5-6-7-8-9-11-12-13-14-15-16-17-18-19-20-21-22-23-24-25-26

   Path 5 = 1-2-3-4-5-6-7-8-9-10-14-15-16-17-18-19-20-21-19-20-21-22-23-24-25-26

   Path 6 = 1-2-3-4-5-6-7-8-9-10-14-15-16-17-18-19-20-21-22-23-24-25-23-24-25-26

3. Graph Matrices

   Berikut ini adalah graph matrices dari algoritma A*yang dapat dilihat pada tabel 4.7.

Tabel 4.7. Graph Matrices

	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	Hasil
1		1																									0
2			1																								0
3				1																							0
4					1																						0
5						1																					0
6							1																				0
7								1																			0
8									1																		0
9										1	1																1
10														1													0
11												1															0
12													1														0
13														1													0
14															1												0
15						1										1											1
16				1													1										1
17																		1									0
18																			1								0
19																				1							0
20																					1						0
21																			1			1					1
22																							1				0
23																								1			0
24																									1		0
25																							1			1	1
26																											Sum(E)+1=6 5+1=6

Berikut ini adalah hasil dari pengujian whitebox penelusuran lgoritma A* pada objek musuh pada aplikasi permainan *The Legend Of Cursed Amulet*. Dapat dilihat pada tabel 4.8.

Tabel 4.8. Hasil Pengujian *White Box*

No path	Node (n)	Hasil yang diharapkan	Hasil tes	kesimpulan
1	1-2-3-4-5-6-7-8-9-10-14-15-16-17-18-19-20-21-22-23-24-25-26	Menampilkan rute ke player	Menampilkan rute ke player	Path dikerjakan
2	1-2-3-4-5-6-7-8-9-11-12-13-14-15-16-17-18-19-20-21-22-23-24-	Mengecek node dalam *OPEN LIST*	Node dalam *OPEN LIST* terdefinisi	Path dikerjakan

	25-26			
3	1-2-3-4-5-6-7-8-9-10-14-15-6-7-8-9-10-14-15-16-17-18-19-20-21-22-23-24-25-26	Mengecek node bukan berada dalam *OPEN LIST*	Node dalam *OPEN LIST* terdefinisi	Path dikerjakan
4	1-2-3-4-5-6-7-8-9-10-14-15-16-4-5-6-7-8-9-11-12-13-14-15-16-17-18-19-20-21-22-23-24-25-26	Membandingkan Nilai F antara nodeawal dengan node tetangga	Node tetangga menjadi current node	Path dikerjakan
5	1-2-3-4-5-6-7-8-9-10-14-15-16-17-18-19-20-21-19-20-21-22-23-24-25-26	Mencari nilai F setiap node terkecil dalam antrian	Nilai node yang mempunyai F terkecil terdefinisi	Path dikerjakan
6	1-2-3-4-5-6-7-8-9-10-14-15-16-17-18-19-20-21-22-23-24-25-23-24-25-26	Mengecek node terakhir sama dengan node awal	Node akhir = node awa	Path dikerjakan

Berdasarkan hasil pengamatan dan pengujian path maka dapat disimpulkan beberapa poin sebagai berikut :

1. Seluruh path dilalui dan dieksekusi
2. Setiap kondisi setidaknya diproses satu kali
3. Tidak ada perulangan tak hingga

4.4. 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.

1. Penafsiran bobot pengujian

   Pengujian menggunakan 3 kategori jawaban dengan bobot yang berbeda untuk setiap jawabannya seperti pada tabel 4.9 dan 4.10.

Tabel 4.9. Bobot Jawaban Pernyataan Positif

Jawaban	Sangat Setuju	Setuju	Ragu-Ragu	Tidak Setuju	Sangat Tidak Setuju
Bobot	5	4	3	2	1

Tabel 4.10.Bobot Jawaban Pernyataan Negatif

Jawaban	Sangat Setuju	Setuju	Ragu-Ragu	Tidak Setuju	Sangat Tidak Setuju
Bobot	1	2	3	4	5

2. Kualifikasi Skor

Persentase skor=(jumlah skor / jumlah skor nilai tertinggi) x100%

Berikut adalah kriteria penilaian intrepetasi skor, kualifikasi persentase = (bobot jawaban / bobot jawaban mak) x 100%

Data hasil kuisioner yang telah diberikan dapat dicari presentasinya dari masing-masing jawaban dengan menggunakan rumus[12]:

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

dimana,

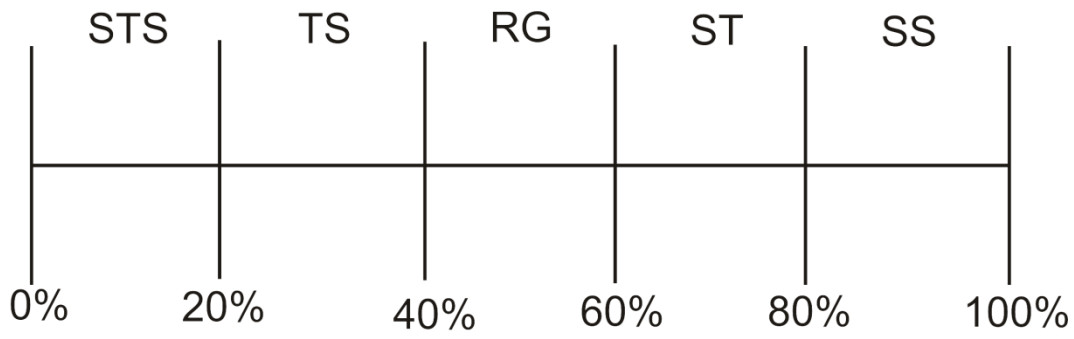
P = Nilai presentase yang dicari

$\sum s$ = 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 :

a. Persentase nilai Sangat Setuju = (5 / 5) x 100% = 100%
b. Persentase nilai Setuju = (4 / 5) x 100% = 80%
c. Persentase nilai Ragu-Ragu = (3 / 5) x 100% = 60%
d. Persentase nilai Tidak Setuju = (2 / 5) x 100% = 40%
e. Persentase nilai Sangat Tidak Setuju = (1 / 5) x 100% = 20%



Gambar 4.9. Kualifikasi Skala Pengukuran

Hasil pengujian Beta adalah sebagai berikut:

1. Permainan menjadi lebih menarik dengan prilaku NPC yang lebih aktif dan lebih agresif.

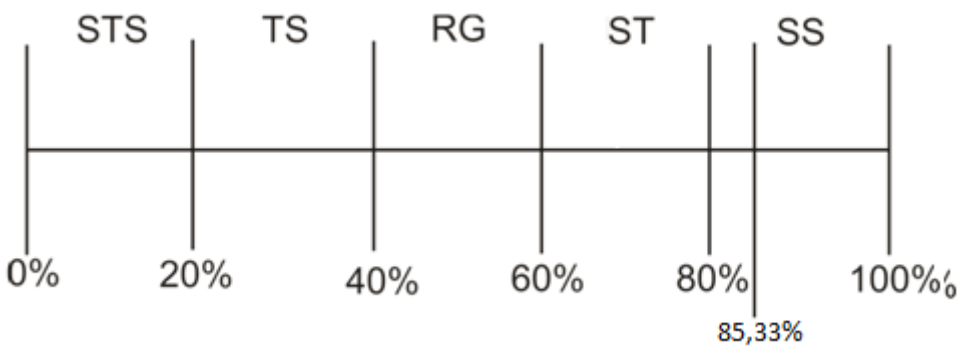
Hasil kuisioner untuk pernyataan ke-1 dapat dilihat pada tabel 4.11.

Tabel 4.11.Hasil Kuisioner Tampilan Aplikasi Permainan

Pernyataan	Jawaban	Skor	Responden	S
1	Sangat Setuju	5	13	64
	Setuju	4	13	52
	Ragu-Ragu	3	4	12
	Tidak Setuju	2	0	0
	Sangat Tidak Setuju	1	0	0
JUMLAH			**30**	**128**

$$\mathrm{P} = \frac{128}{150} \times 100\% = 85.33\%$$

Dapat disimpulkan bahwa penilaian terhadap perilaku musuh pada aplikasi permainan *The Legend Of Cursed Amulet* adalah 85,33% dari yang diharapkan 100%, atau dapat dikategorikan sebagai sangat setuju, untuk lebih jelasnya dapat dilihat pada gambar 4.10.



Gambar 4.10. Interpretasi Skor Pertanyaan ke-1

2. Tampilan lebih spesifik dan mudah dimengerti dalam penyampaian informasi penggunaan *Gameplay*-control.

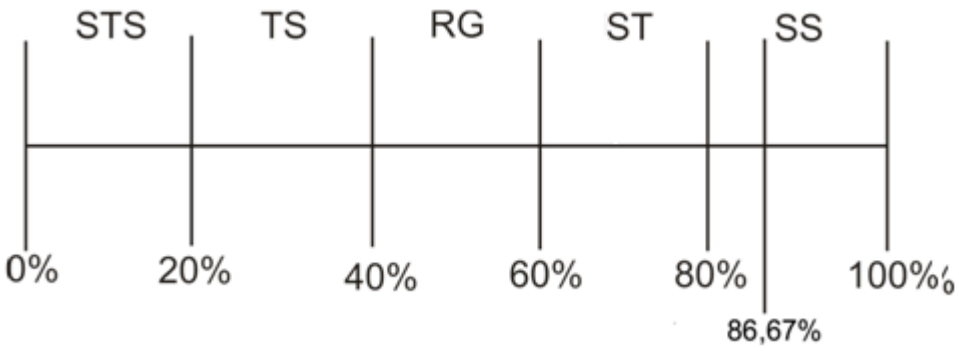
Hasil kuisioner untuk pernyataan ke-2 dapat dilihat pada tabel 4.12.

Tabel 4.12.Hasil kuisioner perilaku musuh pada aplikasi permainan

Pernyataan	Jawaban	Skor	Responden	S
2	Sangat Setuju	5	11	55
	Setuju	4	18	72
	Ragu-Ragu	3	1	3
	Tidak Setuju	2	0	0
	Sangat Tidak Setuju	1	0	0
JUMLAH			**30**	**130**

$$P = \frac{130}{150} \times 100\% = 86.67\%$$

Dapat disimpulkan bahwa penilaian terhadap tampilan dan petunjuk baik tidaknya pada aplikasi permainan *The Legend Of Cursed Amulet* adalah 86,67% dari yang diharapkan 100%, atau dapat dikategorikan sebagai setuju, untuk lebih jelasnya dapat dilihat pada gambar 4.11.



Gambar 4.11. Interpretasi Skor Pertanyaan ke-2

3. Musuh pada permainan *The Legend Of Cursed Amulet* mudah untuk dikalahkan.

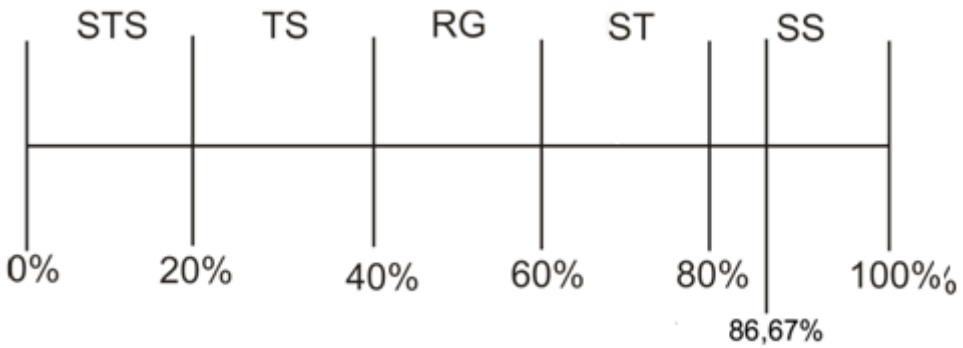
Hasil kuisioner untuk pernyataan ke-3 dapat dilihat pada tabel 4.13.

Tabel 4.13.Hasil kuisioner NPC pada aplikasi permainan

Pernyataan	Jawaban	Skor	Responden	S
3	Sangat Setuju	5	12	60
	Setuju	4	16	64
	Ragu-Ragu	3	2	6
	Tidak Setuju	2	0	0
	Sangat Tidak Setuju	1	0	0
JUMLAH			**30**	**130**

$$P = \frac{130}{150} \times 100\% = 86.67\%$$

Dapat disimpulkan bahwa penilaian terhadap tingkat kesulitan NPC pada aplikasi permainan *The Legend Of Cursed Amulet* adalah 86,67% dari yang diharapkan 100%, atau dapat dikategorikan sebagai setuju, untuk lebih jelasnya dapat dilihat pada gambar 4.12.



Gambar 4.12. Interpretasi Pertanyaan ke-3

4. Tombol pada aplikasi permainan *The Legend Of Cursed Amulet* mudah untuk digunakan.

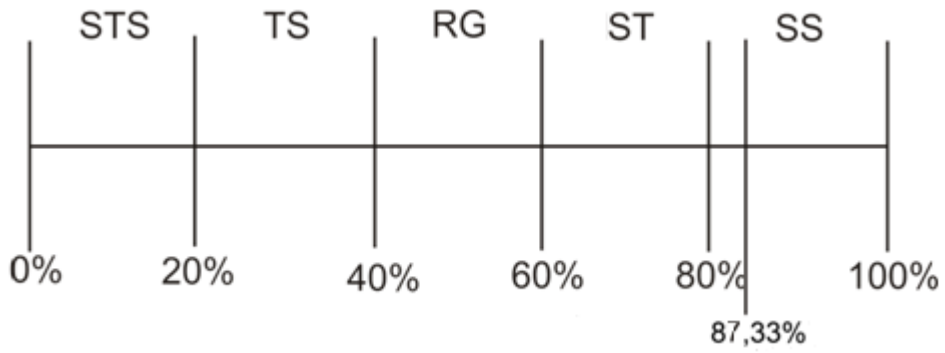
Hasil kuisioner untuk pernyataan ke-4 dapat dilihat pada tabel 4.14.

Tabel 4.14.Hasil kuisioner kontrol pada aplikasi permainan

Pernyataan	Jawaban	Skor	Responden	S
4	Sangat Setuju	5	14	70
	Setuju	4	13	52
	Ragu-Ragu	3	3	9
	Tidak Setuju	2	0	0
	Sangat Tidak Setuju	1	0	0
JUMLAH			**30**	**131**

$$\mathrm{P} = \frac{131}{150} \times 100\% = 87.33\%$$

Dapat disimpulkan bahwa penilaian terhadap kontrol pada aplikasi permainan *The Legend Of Cursed Amulet* adalah 87,33% dari ynang diharapkan 100%, atau dapat dikategorikan sebagai sangat setuju, untuk lebih jelasnya dapat dilihat pada gambar 4.13.



Gambar 4.13. Interpretasi Pertanyaan ke-4

5. Aplikasi permainan *The Legend Of Cursed Amulet* menarik untuk dimainkan.

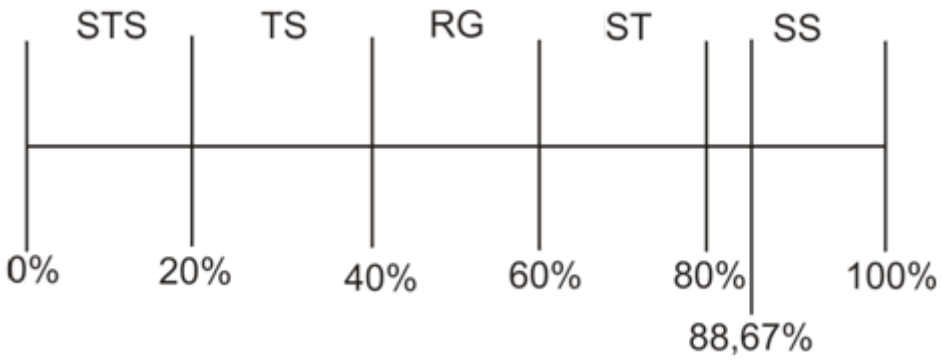
Hasil kuisioner untuk pernyataan ke-5 dapat dilihat pada tabel 4.15.

Tabel 4.15.Hasil kuisioner ketertarikan pada aplikasi permainan

Pernyataan	Jawaban	Skor	Responden	S
5	Sangat Setuju	5	13	65
	Setuju	4	17	68
	Ragu-Ragu	3	0	0
	Tidak Setuju	2	0	0
	Sangat Tidak Setuju	1	0	0
JUMLAH			**30**	**133**

$$\mathrm{P} = \frac{133}{150} \times 100\% = 88.67\%$$

Dapat disimpulkan bahwa penilaian terhadap ketertarikan pada aplikasi permainan *The Legend Of Cursed Amulet* adalah 88,67% dari yang diharapkan 100%, atau dapat dikategorikan sebagai sangat setuju, untuk lebih jelasnya dapat dilihat pada gambar 4.14.



Gambar 4.14. Interpretasi Pertanyaan ke-5