BAB 4
IMPLEMENTASI DAN PENGUJIAN SISTEM

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

4.1 *Implementasi*

Tahap implementasi merupakan tahap penciptaan perangkat lunak dan juga tahap kelanjutan dari kegiatan perancangan aplikasi berdasarkan hasil analisis pada bab sebelumnya.

4.1.1 Perangkat Pendukung yang digunakan

Dalam proses pembuatan aplikasi permainan *The Tale of Hunter*, sangat membutuhkan perangkat yaitu perangkat keras (*hardware*) dan perangkat lunak (*software*). Berikut merupakan penjelasan dari perangkat keras dan perangkat lunak yang digunakan.

4.1.1.1Perangkat Keras (*Hardware*)

Perangkat keras yang digunakan untuk menjalankan aplikasi permainan The Tale of Hunter sebagai berikut:

1. Prosesor : 1.0 GHz
2. Memory : 256 MB
3. VGA : 512 MB
4. Harddisk : minimal 10MB
5. Monitor dengan Resolusi 1366 x 768
6. Mouse dan Keyboard

4.1.1.2 Perangkat Lunak (*Software*)

Perangkat lunak yang digunakan untuk membangun aplikasi permainan The Tale of Hunter sebagai berikut :

1. Sistem Operasi Windows 7
2. Greenfoot + JDK
3. Adobe Photoshop
4. Adobe Illustrator
5.

4.1.2 Implementasi Aplikasi

The Tale of Hunter merupakan aplikasi permainan yang di fokuskan untuk mencari jalan keluar, mengoleksi *item,* serta mengalahkan semua musuh.

Permainan dikatakan *game over* apabila jumlah *lifepoint* berkurang menjadi 0 dan jika permainan dikatakan *finish* apabila pemain dapat menyelesaikan misi pada setiap *level*.

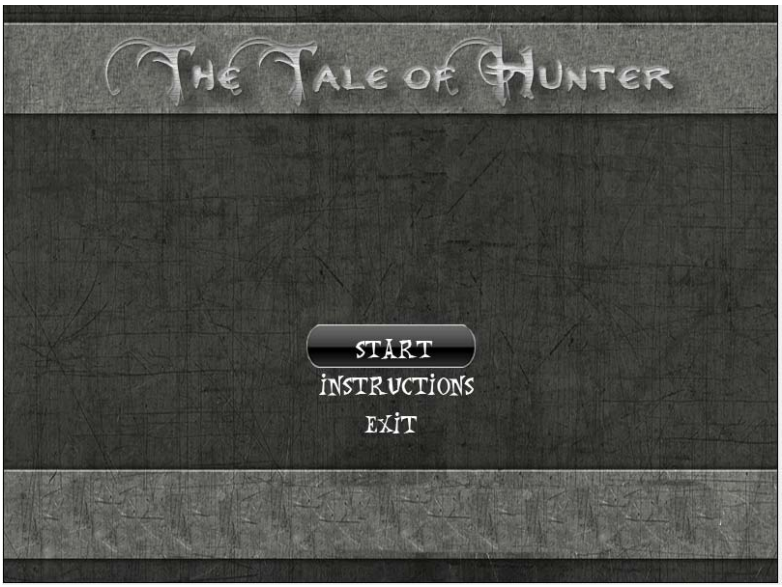
Aplikasi permainan *The Tale of Hunter* ini berbasis *desktop* dengan menggunakan bahasa pemrograman *Java*. Aplikasi ini dapat dijalankan apabila pemain telah menginstal JVM (*Java Virtual Machine*) pada perangkat komputernya.

4.1.3 Implementasi Antarmuka

Adapun implementasi antarmuka pada aplikasi The Tale of Hunter adalah sebagai berikut:

1. Antarmuka Menu *Main*

   Antarmuka Menu *Main* merupakan tampilan menu utama pada saat pertama kali pemain membuka aplikasi permainan. Tampilan dari antar muka meni *main* dapat dilihat pada gambar 4.1.



Gambar 4.1 Tampilan Antarmuka Menu *Main*

2. **Antarmuka Intro Permainan**

   Antarmuka intro permainan merupakan tampilan ketika pemain memilih menu *start* pada menu *main*. Antarmuka ini bercerita tentang misi yang

   harus di selesaikan oleh pemain. Tampilan dari antarmuka intro dapat dilihat pada gambar 4.2



Gambar 4.2 Tampilan Antarmuka Menu Intro

3. Antarmuka Mulai Permainan

Antarmuka mulai permainan merupakan tampilan ketika pemain memilih menu *next* pada intro, sehingga akan masuk ke *level* 1. Tampilan dari antarmuka mulai permainan *level* 1 dapat dilihat pada gambar 4.3.



Gambar 4.3 Tampilan Antarmuka *Level* 1

Tampilan antarmuka *level* 2 akan muncul, ketika pemain sudah menyelesaikan misi pada *level* 1. Tampilan antarmuka level 2 dapat dilihat pada gambar 4.4.



Gambar 4.4 Tampilan Antarmuka *Level* 2

Tampilan antarmuka *level* 3 akan muncul, ketika pemain sudah menyelesaikan misi pada *level* 2. Tampilan antarmuka level 3 dapat dilihat pada gambar 4.5.



Gambar 4.5 Tampilan Antarmuka Level 3

Tampilan antarmuka *level* 4 akan muncul, ketika pemain sudah menyelesaikan misi pada *level* 3. Tampilan antarmuka level 4 dapat dilihat pada gambar 4.5.



Gambar 4.6 Tampilan Antarmuka Level 4

4.2 Pengujian Sistem

Dalam melakukan pengujian terhadap aplikasi permainan *The Tale of Hunter* dilakukan dengan dua macam 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 pengujian ini dilakukan untuk menguji aplikasi permainan *The Tale of Hunter*. Pengujian aplha berfokus pada persyaratakan fungsional perangkat lunak.

4.2.1.1 Rencana Pengujian

Rencana pengujian yang akan dilakukan pada aplikasi permainan *The Tale of Hunter* selengkapnya terlihat pada tabel di bawah.

Tabel 4.1 Rencana Pengujian Aplikasi permainan *The Tale of Hunter*

No.	Komponen yang diuji	Skenario Pengujian	Jenis Pengujian
1	Menu Main	Pilih Menu *Start*	*Black Box*
		Pilih Menu *Instructions*	*Black Box*
		Pilih Menu *Exit*	*Black Box*
2	Menu Instructions	Pilih Menu *Start*	*Black Box*
		Pilih Menu *Back*	*Black Box*
4	Menu Intro	Pilih Menu *Back*	*Black Box*
		Pilih Menu *Exit*	*Black Box*
		Permainan *Level1*	*Black Box*
5	Menu Exit	Pilih Menu *Exit*	*Black Box*
6	Algoritma A* (A Star)	Algoritma A* pada permainan	*White Box*

4.2.1.2 Kasus dan Hasil Pengujian *Black Box*

Berdasarkan hasil pengujian yang telah disusun, maka dapat dilakukan pengujian sebagai berikut. Pengujian yang dilakukan meliputi pengujian tampilan pembuat.

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
Pilih *Next* (pada intro)	Menampilkan map *level* 1	Dapat menampilkan map *level* 1	[√] Berhasil [ ] Gagal
Menyelesaikan misi *level* 1	Menampilkan map *level* 2	Dapat menampilkan map *level* 2	[√] Berhasil [ ] Gagal
Menyelesaikan misi *level* 2	Menampilkan map *level* 3	Dapat menampilkan map *level* 3	[√] Berhasil [ ] Gagal
Menyesaikan misi *level* 3	Menampilkan keterangan *Game Finish*	Dapat Menampilkan keterangan *Game Finish*	[√] Berhasil [ ] Gagal

B. Pengujian Tampil *Instructions*

Pengujian tampil *instuction* dengan kasus dan hasil uji, kemudian diamati dan menghasilkan kesimpulan dapat dilihat pada tabel 4.3.

Tabel 4.3 Pengujian Tampil *Instruction*

Kasus dan Hasil Uji (Data benar)			
Masukan	Keluaran yang diharapkan	Pengamatan	Kesimpulan
Pilih Menu *Start*	Menampilkan Menu Intro	Dapat menampilkan menu intro	[√] Berhasil [ ] Gagal
Pilih Menu *Back*	Menampilkan menu *main*	Dapat menampilkan menu main	[√] Berhasil [ ] Gagal

C. Pengujian Tampil Intro

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

Tabel 4.4 Pengujian Tampil Intro

Kasus dan Hasil Uji (Data benar)			
Masukan	Keluaran yang diharapkan	Pengamatan	Kesimpulan
Pilih Menu *Next*	Menampilkan permainan *level* 1	Dapat menampilkan permainan *level* 1	[√] Berhasil [ ] Gagal
Pilih Menu *Back*	Menampilkan menu *main*	Dapat menampilkan menu *main*	[√] Berhasil [ ] Gagal

4.2.1.3 Kasus dan Pengujian *White Box*

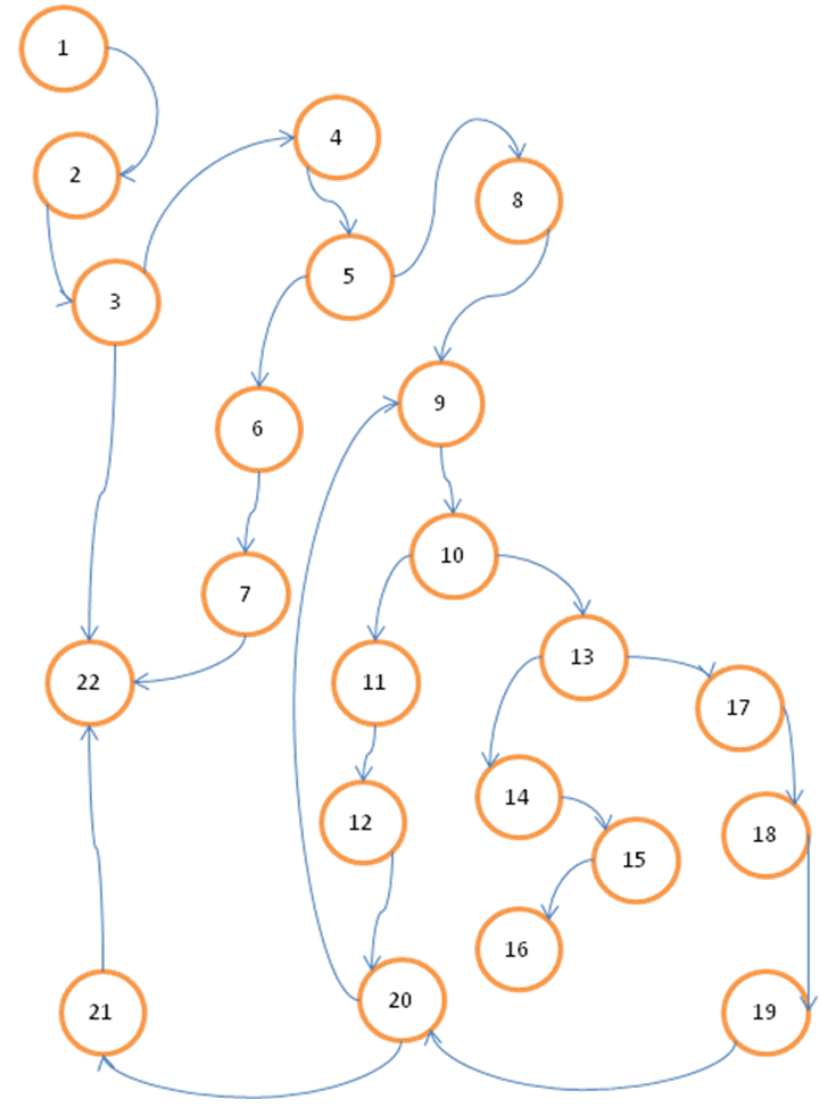
Dibawah ini merupakan kasus algoritma A* (A *star*) untuk menguji perangkat lunk yang sudah dibangun dengan menggunakan metode *white box,* berdasarkan skenario pengujian aplikasi yang dijabarkan.

A. Algoritma

```
1  create the open list of nodes, initially containing only our starting node
2  create the closed list of nodes, initially empty
3  while (we have not reached our goal) {
4          consider the best node in the open list (the node with the lowest f
           value)
5          if (this node is the goal) {
6                  then we're done
7          }
8          else {
9                  move the current node to the closed list and consider all
                   of its neighbors
```

```
10                  for (each neighbor) {
11                      if (this neighbor is in the closed list and our
                        current g value is lower) {
12                          update the neighbor with the new,
                            lower, g value
                            change the neighbor's parent to our
                            current node
13                      }
14                      else if (this neighbor is in the open list and our
                        current g value is lower) {
15                          update the neighbor with the new,
                            lower, g value
                            change the neighbor's parent to our
                            current node
16                      }
17                      else this neighbor is not in either the open or
                        closed list {
18                          add the neighbor to the open list and
                            set its g value
19                      }
20                  }
21              }
22  }
```

B. *Flow Graph*



Gambar 4.7 *Flow Graph* A*

Kerangan :

= Menggambarkan kondisi

= Menggambarkan aksi

C. *Cylomatic Complexity* V(G)

V(G) = E – N + 2

V(G) = 26 – 22 + 2

V(G) = 6

Di mana:

E = jumlah edge pada grafik alir

N = jumlah node pada grafik alir

D. *Independent Path*

Path 1 = 1-2-3-4-5-6-7-22

Path 2 = 1-2-3-22

Path 3 = 1-2-3-4-5-8-9-10-11-12-20-21-22

Path 4 = 1-2-3-4-5-6-7-8-9-10-13-14-15-16-20-21-22

Path 5 = 1-2-3-4-5-8-9-10-13-17-18-19-20-21-22

Path 6 = 1-2-3-4-5-8-9-10-11-12-20-9-10-11-12-20-21-22

E. *Graph Matrices*

Berikut in adalah *graph matrices* dari algoritma A* yang dapat dilihat pada tabel 4.5.

Tabel 4.5 *Graph Matrices* A*

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	Hasil
1		1																					0
2			1																				0
3				1																		1	1
4					1																		0
5						1		1															1
6							1																0
7																						1	0
8									1														0
9										1													0
10											1		1										1
11												1											0
12																				1			0
13														1			1						1
14															1								0
15																1							0
16																				1			0
17																		1					0
18																			1				0
19																				1			0

20								1											1		1
21																				1	0
22																					0
Hasil																					5

V(G) = Jumlah Graph Matrices + 1

V(G) = 5 + 1

V(G) = 6

Berdasarkan hasil pengujian dihasilkan nilai *Cyclomatic Complexity* yang sama yaitu 6, 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 dilakukan dengan mencoba aplikasi secara langsung ke kalangan yang menjadi target user aplikasi permainan The Tale of Hunter. Pengujian dilakukan dengan cara membagikan kuisioner kepada para penggunakan guna mendapatkan data tentang kepuasan pengguna terhadap game The Tale of Hunter.

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 25 orang , dari hasil kuesioner tersebut akan dilakukan perhitungan agar dapat diambil kesimpulan terhadap penilaian kepuasan terhadap aplikasi game *The Tale of Hunter*. Kuesioner terdiri dari 5 pernyataan dengan menggunakan skala 1 sampai 4 (skala likert)

Untuk mencari nilai persentase dari masing-masing pendapat kuesioner digunakan rumus skala likert sebagai berikut :

$$P = \frac{\sum s}{skorideal} \times 100\% \qquad (4.1)$$

Keterangan :

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

Berikut adalah kuesioner yang disebarkan ke target *user* dan harus di isi oleh responden :

1) Apakah musuh dalam *game* ini lebih pintar dari *game* yang di analisis.
    a) Sangat Setuju
    b) Setuju
    c) Kurang Setuju
    d) Tidak Setuju
2) Apakah memperbanyak penghalang (*map)* membuat game lebih menantang.
    a) Sangat Setuju
    b) Setuju
    c) Kurang Setuju
    d) Tidak Setuju

3) Apakah dengan 2 jenis permainan yang berbeda, membuat game lebih menarik.
    a) Sangat Setuju
    b) Setuju
    c) Kurang Setuju
    d) Tidak Setuju
4) Apakah secara umum kontrol dalam permainan sudah baik.
    a) Sangat Setuju
    b) Setuju
    c) Kurang Setuju
    d) Tidak Setuju
5) Apakah secara umum menurut anda game ini sudah menarik
    a) Sangat Setuju
    b) Setuju
    c) Kurang Setuju
    d) Tidak Setuju

4.2.2.1 Hasil pengujian Beta

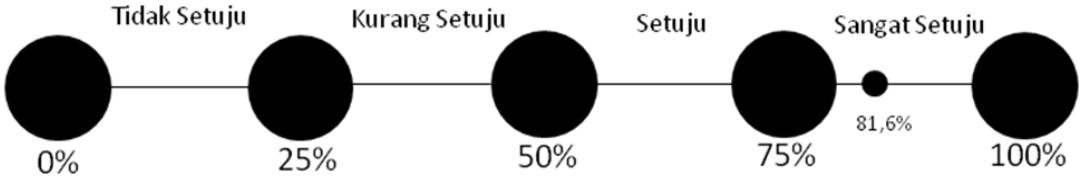
1. Apakah musuh dalam *game* ini lebih pintar dari *game* yang di analisis. Hasil kuesioner untuk pertanyaan ke-1 dapat dilihat pada tabel 4.6.

Tabel 4.6 Perhitungan Presentase Pernyataan ke-1

Pernyataan	Keterangan	Skor	Frekuensi	S
1	Sangat Setuju	4	10	40
	Setuju	3	18	54
	Kurang Setuju	2	2	4
	Tidak Setuju	1	0	0
Jumlah			30	98

$$P = \frac{98}{120} \times 100 = 81{,}6\%$$

Berdasarkan hasil presentase nilai diatas, maka dapat disimpulkan bahwa penilaian terhadap aplikasi permainan *The Tale of Hunter* adalah 81,6% dari yang diharapkan 100% atau dikatakan sebagai setuju. Berikut Kriteria skor yang dapat dilihat pada gambar 4.8 :



Gambar 4.8 Interpretasi Skor Pertanyaan ke-1

2. Apakah memperbanyak penghalang (*map)* membuat game lebih menantang.

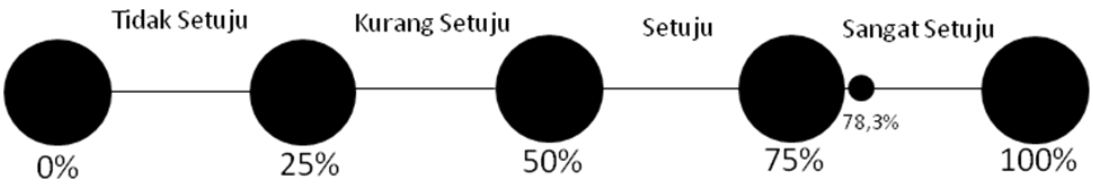
   Hasil kuesioner untuk pertanyaan ke-1 dapat dilihat pada tabel 4.7

Tabel 4.7 Perhitungan Presentase Pernyataan ke-2

Pernyataan	Keterangan	Skor	Frekuensi	S
2	Sangat Setuju	4	9	36
	Setuju	3	16	48
	Kurang Setuju	2	5	10
	Tidak Setuju	1	0	0
Jumlah			30	94

$$P = \frac{94}{120} \text{ X } 100 = 78{,}3\%$$

Berdasarkan hasil presentase nilai diatas, maka dapat disimpulkan bahwa penilaian terhadap aplikasi permainan *The Tale of Hunter* adalah 78,3 dari yang diharapkan 100% atau dikatakan sebagai setuju. Berikut Kriteria skor yang dapat dilihat pada gambar 4.9 :



Gambar 4.9 Interpretasi Skor Pertanyaan ke-2

3. Apakah dengan 2 jenis permainan yang berbeda, membuat game lebih menarik.

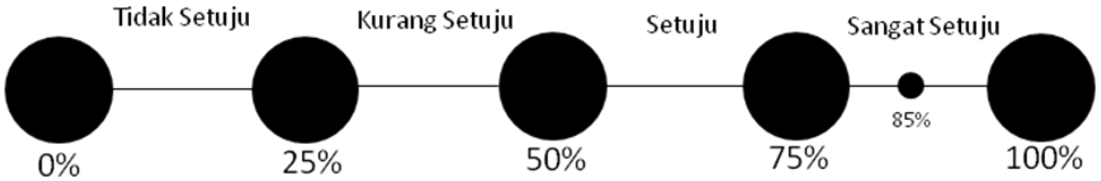
   Hasil kuesioner untuk pertanyaan ke-3 dapat dilihat pada tabel 4.8

Tabel 4.8 Perhitungan Presentase Pernyataan ke-3

Pernyataan	Keterangan	Skor	Frekuensi	S
3	Sangat Setuju	4	15	60
	Setuju	3	12	36
	Kurang Setuju	2	3	6
	Tidak Setuju	1	0	0
Jumlah			30	102

$$P = \frac{102}{120} \times 100 = 85\%$$

Berdasarkan hasil presentase nilai diatas, maka dapat disimpulkan bahwa penilaian terhadap aplikasi permainan *The Tale of Hunter* adalah 85% dari yang diharapkan 100% atau dikatakan sebagai setuju. Berikut Kriteria skor yang dapat dilihat pada gambar 4.10 :



Gambar 4.10 Interpretasi Skor Pertanyaan ke-3

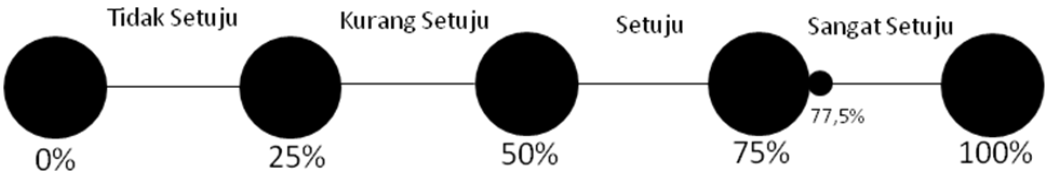
4. Apakah Apakah secara umum kontrol dalam permainan sudah baik.
   Hasil kuesioner untuk pertanyaan ke-4 dapat dilihat pada tabel 4.9

Tabel 4.9 Perhitungan Presentase Pernyataan ke-4

Pernyataan	Keterangan	Skor	Frekuensi	S
4	Sangat Setuju	4	7	28
	Setuju	3	19	57
	Kurang Setuju	2	4	8
	Tidak Setuju	1	0	0
Jumlah			30	93

$$P = \frac{93}{120} \text{ X } 100 = 77{,}5\%$$

Berdasarkan hasil presentase nilai diatas, maka dapat disimpulkan bahwa penilaian terhadap aplikasi permainan *The Tale of Hunter* adalah 77,5 dari yang diharapkan 100% atau dikatakan sebagai setuju. Berikut Kriteria skor yang dapat dilihat pada gambar 4.11 :



Gambar 4.11 Interpretasi Skor Pertanyaan ke-4

5. Apakah secara umum menurut anda game ini sudah menarik

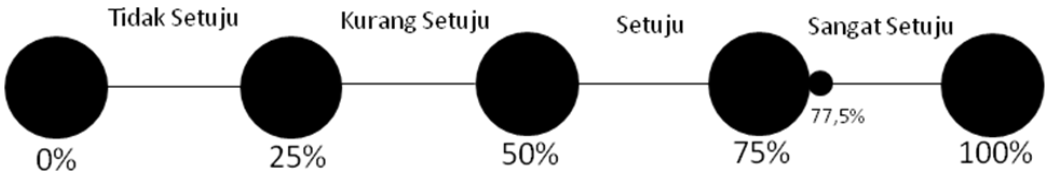
   Hasil kuesioner untuk pertanyaan ke-4 dapat dilihat pada tabel 4.10

Tabel 4.10 Perhitungan Presentase Pernyataan ke-5

Pernyataan	Keterangan	Skor	Frekuensi	S
5	Sangat Setuju	4	6	24
	Setuju	3	21	63
	Kurang Setuju	2	3	6
	Tidak Setuju	1	0	0
Jumlah			30	93

$$P = \frac{93}{120} \times 100 = 77{,}5\%$$

Berdasarkan hasil presentase nilai diatas, maka dapat disimpulkan bahwa penilaian terhadap aplikasi permainan *The Tale of Hunter* adalah 84,1% dari yang diharapkan 100% atau dikatakan sebagai setuju. Berikut Kriteria skor yang dapat dilihat pada gambar 4.12 :



Gambar 4.11 Interpretasi Skor Pertanyaan ke-5

4.2.2.2 Kesimpulan Pengujian *Beta*

Berdasarkan pengujian beta yang telah didapat dari hasil kuesioner yang telah dijawab oleh para responden, maka bisa ditarik kesimpulan bahwa aplikasi permainan *The Tale of Hunter* ini mempunyai musuh yang lebih sulit dari pada musuh dalam *game* yang dianalisis. Penghalang (*map*) yang banyak membuat game lebih menantang. Dua jenis permainan yang bebeda membuat game lebih bervariasi dan menarik, kontrol permainan yang baik sehingga secara keseluruhan aplikasi permainan *The Tale of Hunter* sudah menarik.