BAB 4
IMPLEMENTASI DAN PENGUJIAN

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. Pada bagian ini akan dijelaskan mengenai lingkungan implementasi dan pengujian aplikasi, yang mana terdiri dari implementasi perangkat keras, implementasi perangkat lunak, implementasi antarmukanya, pengujian alpha dan pengujian beta.

4.2 Implementasi Perangkat Keras

Perangkat keras yang dapat digunakan untuk menjalankan aplikasi *game* ini tidak harus memiliki spesifikasi *device* yang tinggi, seperti *processor* 1.5Ghz dan *memory* 2GB yang dimiliki oleh *Samsung galaxy S4*. Sebagai ilustrasi *system* minimum agar dapat menjalankan *game Say No To Drugs* adalah sebagai berikut :

1. *Processor* dengan kecepatan 800Mhz.
2. RAM dengan kapasitas 512Mb.

4.3 Implementasi Perangkat Lunak

Adapun spesifikasi perangkat lunak yang dibutuhkan untuk menjalankan aplikasi *Say No To Drugs* adalah Sistem Operasi Android v2.3 (*Gingerbread*) atau diatasnya dan *adobe air* sebagai aplikasi pendukungnya.

4.4 Implementasi Antarmuka

Pada bagian ini akan ditampilkan antar muka yang ada dalam *game Say No To Drugs* yaitu sebagai berikut :

4.4.1 Antarmuka Menu Utama

Saat pengguna membuka aplikasi pertama kali, maka antarmuka yang ditampilkan oleh sistem adalah antarmuka menu utama. Berikut ini merupakan antarmuka menu utama yang dapat dilihat pada Gambar 4.1.



Gambar 4. 1 Antarmuka Menu Permainan

4.4.2 Antarmuka Intro

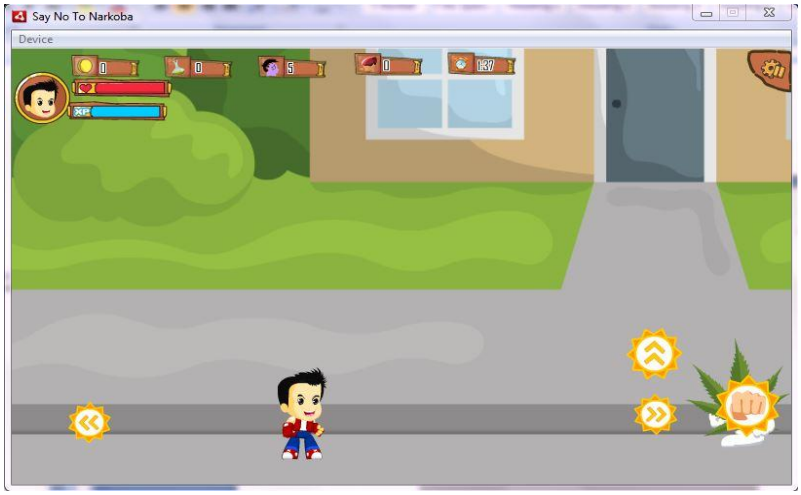
Tujuan dari antarmuka intro ini yaitu untuk memberikan gambaran alur cerita dalam permainan ini sehingga pemain mengetahui misinya dalam permainan. Antarmuka intro dapat dilihat pada Gambar 4.2.



Gambar 4. 2 Antarmuka Intro

4.4.3 Antarmuka Permainan

Antarmuka *Permainan* merupakan tampilan ketika pemain berada dalam permainan *Say No To Drugs*. Terdapat 3 (tiga) *stage* dalam *game* ini. Gambar 4.3 memperlihatkan antarmuka permainan *stage* 1, Gambar 4.4 memperlihatkan antarmuka permainan *stage* 2 dan Gambar 4.5 memperlihatkan antarmuka permainan *stage* 3.



Gambar 4. 3 Antarmuka Permainan *Stage* 1



Gambar 4. 4 Antarmuka Permainan *Stage* 2



Gambar 4. 5 Antarmuka Permainan *Stage* 3

4.4.4 Antarmuka Menu Permainan

Tampilan dari antarmuka menu permainan dapat dilihat pada gambar 4.6.



Gambar 4. 6 Antarmuka Menu Permainan

4.4.5 Antarmuka Outro

Setelah pemain berhasil melewati semua rintangan dan berhasil mendapatkan misinya, pemain akan disuguhkan pada ilustrasi atau gambaran akhir alur cerita permainan (*Outro*). Tampilan dari antarmuka outro dapat dilihat pada gambar 4.7.



Gambar 4. 7 Antarmuka Outro

4.4.6 Antarmuka Skor

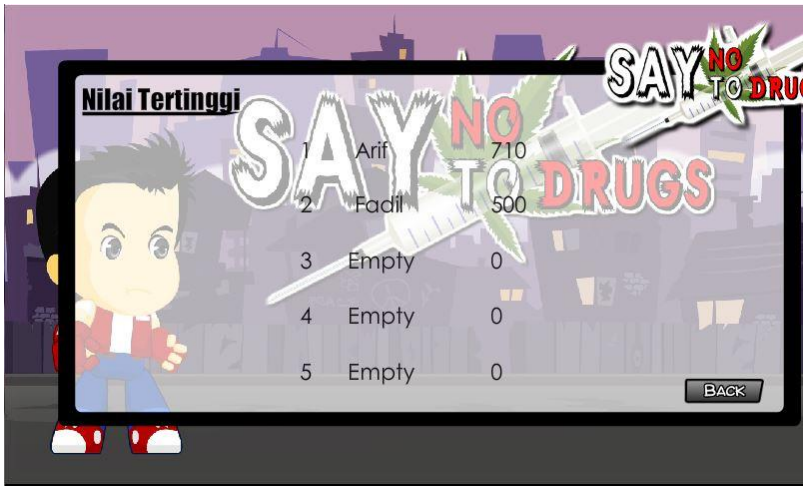
Saat pemain menyelesaikan permainan, maka antarmuka yang ditampilkan oleh sistem adalah informasi mengenai skor yang didapatkan pada permainan *Say No To Drugs* dan juga pemain diharuskan untuk memasukan sebuah nama pemain. Tampilan dari antarmuka skor dapat dilihat pada gambar 4.8.



Gambar 4. 8 Antarmuka Skor

4.4.7 Antarmuka Skor Tertinggi

Tampilan dari antarmuka skor tertinggi dapat dilihat pada gambar 4.9.



Gambar 4. 9 Antarmuka Skor Tertinggi

4.4.8 Antarmuka Cara Main

Antarmuka cara main merupakan tampilan ketika pemain memilih menu *Control* pada menu utama. Tampilan dari antarmuka cara main dapat dilihat pada Gambar 4.10.



Gambar 4. 10 Antarmuka Cara Main

4.4.9 Implementasi Pesan

Implementasi pesan adalah bagian yang menunjukan bentuk tampilan setiap pesan pada aplikasi yang dibangun.

Terdapat pesan pada aplikasi *game* ini, yaitu pesan *Keluar*. Pesan keluar terjadi ketika pengguna memilih menu *Keluar* dan berfungsi sebagai validasi jika pemain ingin keluar dari aplikasi permainan *Say No To Drugs*. Tampilan dari pesan keluar dapat dilihat pada gambar 4.11.



Gambar 4. 11 Pesan Keluar

4.5 Pengujian

Dalam melakukan pengujian terhadap aplikasi *game Say No To Drugs* 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 *beta*.

4.5.1 Pengujian Alpha

Pada tahap ini, pengujian dilakukan untuk memastikan aplikasi *game Say No To Drugs* telah berjalan dengan benar dan sesuai dengan kebutuhan yang diinginkan. Pengujian *alpha* menitikberatkan kepada persyaratan fungsional perangkat lunak.

4.5.1.1 Skenario Pengujian Aplikasi

Skenario pengujian menjelaskan pengujian terhadap sistem yang ada pada aplikasi *game Say No To Drugs*. Skenario pengujian dari aplikasi *game Say No To Drugs* dapat dilihat pada Tabel 4.1.

Tabel 4. 1 Skenario Pengujian Aplikasi

No	Komponen yang di uji	Skenario Pengujian	Jenis Pengujian
1	Permainan	Menampilkan permainan level 1	*Black box*
		Menampilkan permainan level 2	*Black box*
		Menampilkan permainan level 3	*Black box*
		Mengendalikan Karakter	*Black box*
		Mengambil herbal	*Black box*
		Mengambil hati	*Black box*
		Melawan musuh	*Black box*
		Melawan Raja	*Black box*
2	Skor	Tampil informasi skor	*Black box*
		Sentuh tombol Simpan	*Black box*
3	*Control*	Tampil informasi cara main	*Black box*
		Sentuh tombol *Back*	*Black box*

4.5.1.2 Kasus dan Hasil Pengujian (*Black Box*)

Berikut ini merupakan kasus untuk menguji perangkat lunak yang sudah dibangun dengan menggunakan metode *black box* berdasarkan skenario pengujian aplikasi yang terdapat pada Tabel 4.1 yang sudah dijabarkan. Hasil pengujian yang akan dilakukan pada aplikasi ini selengkapnya dapat dilihat pada Tabel 4.2.

Tabel 4. 2 Kasus dan Hasil Pengujian (*Black Box*)

No	Kasus yang di uji	Skenario dan Hasil Uji		
		Skenario pengujian	Hasil yang diharapkan	Kesimpulan
1	Permainan	Tampil Permainan *level* 1	Sistem menampilkan arena permainan *level* 1	[√] Berhasil [ ] Gagal
		Tampil Permainan *level* 2	Sistem menampilkan arena permainan *level* 2	[√] Berhasil [ ] Gagal
		Tampil Permainan *level* 3	Sistem menampilkan arena permainan *level* 3	[√] Berhasil [ ] Gagal
		Mengendalikan Karakter	Karakter utama dapat bergerak	[√] Berhasil [ ] Gagal
		Mengambil herbal	Karakter utama dapat mengambil herbal	[√] Berhasil [ ] Gagal
		Mengambil hati	Karakter utama dapat mengambil hati dan darah bertambah	[√] Berhasil [ ] Gagal
		Melawan musuh	Karakter utama melawan musuh dan musuh menghilang	[√] Berhasil [ ] Gagal
		Melawan Raja	Karakter utama melawan raja dan raja dapat dikalahkan	[√] Berhasil [ ] Gagal
2	Skor	Tampil informasi skor	Sistem menampilkan informasi skor	[√] Berhasil [ ] Gagal
		Sentuh tombol Oke	Sistem menampilkan halaman Menu Permainan	[√] Berhasil [ ] Gagal
3	*Control*	Tampil informasi cara main	Sistem menampilkan informasi cara main	[√] Berhasil [ ] Gagal
		Sentuh tombol Kembali	Sistem menampilkan halaman Menu Permainan	[√] Berhasil [ ] Gagal

4.5.1.3 Kasus dan Hasil 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.

1. Pseudo Algoritma A*

Pseudocode algoritma A* pada raja yang penerapannya menggunakan bahasa pemrograman *Action Script 3.0* dapat dilihat pada Tabel 4.3.

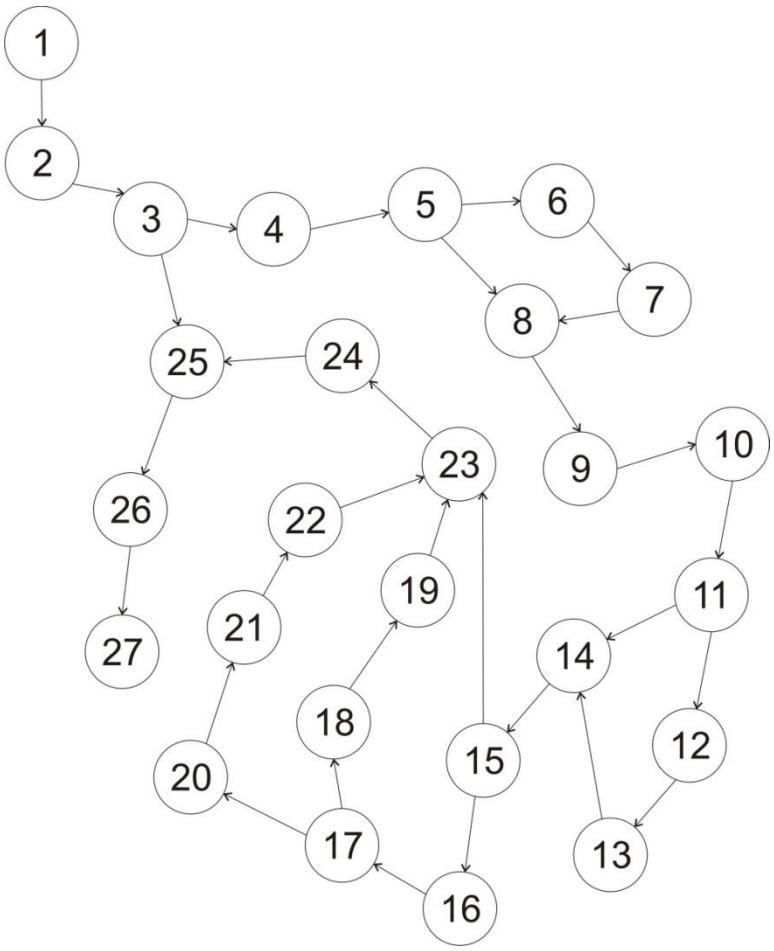
Tabel 4. 3 Pseudo Algoritma A*

```
1  function pathFinder() {
2      var startNode:Nodes = varHolder[cur];
       startNode.g = 0;
       startNode.f = 0;
       push(startNode);
       startNode.opened = true;
       var tahap = 0;
3      while (opened.length > 0) {
4       var gh,gg;
        var node = pop();
        node.closes = true;
5       if (node.id == des) {
6         return backtrace(des);
7       }
8       getTetangga(node.id);
        var ttgs = node.tetangga;
9       for (var d = 0; d < ttgs.length; d++)  {
10        var ttg = ttgs[d];
11        if (ttg.closes) {
12          continue;
13        }
```

```
14          var cost:Number = (ttg.xs - node.xs === 0 || ttg.ys - node.ys
    === 0 ? 1:Math.SQRT2);
            var ng:Number = node.g + cost;
15          if (! ttg.opened || ng < ttg.g) {
16            ttg.g = ng;
              ttg.h = ttg.h || heuristic (Math.abs(ttg.xs -
    endN.xs),Math.abs(ttg.ys - endN.ys));
              ttg.f = ttg.g + ttg.h;
              ttg.par = node;
17            if (! ttg.opened) {
18              push(ttg);
                ttg.opened = true;
19            }
20            else {
21              listUpdate(ttg);
22            }
23        }
24      }
25    }
26    return [];
27  }
```

2. *Flow Graph*

Berdasarkan *pseudocode* di atas maka *flow graph* algoritma A* dapat dilihat pada Gambar 4.12.



Gambar 4. 12 *Flow Graph* Algoritma A*

Keterangan :

= Menggambarkan kondisi

= Menggambarkan aksi

3. *Cyclomatic Complexity* V(G)

V(G) = E – N + 2

V(G) = 31 – 27 + 2

V(G) = 4 + 2

V(G) = 6

Dimana :

E = Jumlah edge pada grafik alir

N = Jumlah node pada grafik alir

4. *Independent Path*

Path 1 = 1-2-3-25-26-27

Path 2 = 1-2-3-4-5-8-9-10-11-14-15-23-24-25-26-27

Path 3 = 1-2-3-4-5-8-9-10-11-12-13-14-15-16-17-18-19-23-24-25-26-27

Path 4 = 1-2-3-4-5-6-7-8-9-10-11-12-13-14-15-17-18-19-23-24-25-26-27

Path 5 = 1-2-3-4-5-6-7-8-9-10-11-12-13-14-15-16-17-18-19-23-24-25-26-27

Path 6 =1-2-3-4-5-6-7-8-9-10-11-12-13-14-15-16-17-20-21-22-23-24-25-26-27

5. *Graph* Matriks

Berikut ini adalah *graph* matriks dari algoritma A* yang dapat dilihat pada Tabel 4.4.

Tabel 4. 4 Graph Matriks A*

N	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	N(E)-1
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																	0
11												1		1														1
12													1															0
13														1														0
14															1													0
15																1							1					1
16																	1											0
17																		1		1								1
18																			1									0
19																							1					0
20																					1							0
21																						1						0
22																							1					0
23																								1				0
24																									1			0
25																										1		0
26																											1	0
27																												0
SUM(E)+1																												5+1=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.

4.5.1.4 Kesimpulan Pengujian Alpha

Dari hasil pengujian yang dilakukan, dapat disimpulkan bahwa sistem sudah dapat berjalan sesuai dengan yang diharapkan beserta proses penanganan kesalahannya. Jadi, secara fungsional sistem sudah dapat menghasilkan keluaran yang diharapkan.

4.5.2 Pengujian Beta

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

4.5.2.1 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 30 orang dari usia 15 tahun hingga 24 tahun, dari hasil kuesioner tersebut akan dilakukan perhitungan agar dapat diambil kesimpulan terhadap penilaian penerapan sistem yang telah dibangun. Kuesioner terdiri dari 5 pertanyaan dengan menggunakan skala 1 sampai 5 (skala likert).

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

$$\boldsymbol{P = \frac{S}{Skor\ Ideal} \times 100\%}$$

Keterangan :

P = Nilai persentase yang dicari.

S = Jumlah frekuensi dikalikan dengan skor yang di tetapkan jawaban.

Skor Ideal = Nilai tertinggi dikalikan dengan jumlah sampel.

Nilai tertinggi = 5.

Jumlah sampel= 30.

Skor Ideal = (5 x 30) = 150.

Tabel 4. 5 Interpretasi Skor

No	Range	Keterangan	
		Pertanyaan Positif	Pertanyaan Negatif
1	0% - 19%	Sangat Tidak Setuju	Sangat Setuju
2	20% - 39%	Tidak Setuju	Setuju
3	40% - 59%	Cukup Setuju	Cukup Setuju
4	60% - 79%	Setuju	Tidak Setuju
5	80% - 100%	Sangat Setuju	Sangat Tidak Setuju

Berikut ini adalah hasil presentase masing-masing jawaban yang sudah dihitung nilainya dengan menggunakan rumus diatas.

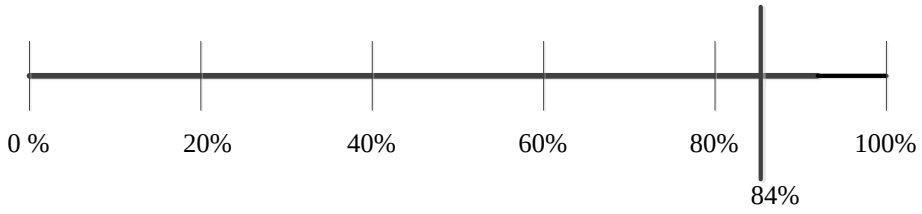
1. Apakah *game Say No To Drugs* sudah membantu anda dalam mengetahui tentang bahaya narkoba ?

Tabel 4. 6 Perhitungan persentase pertanyaan ke-1

Pertanyaan	Keterangan	Skor	Frekuensi	S
1	Sangat Setuju	5	6	30
	Setuju	4	24	96
	Cukup Setuju	3	0	0
	Tidak Setuju	2	0	0
	Sangat Tidak Setuju	1	0	0
Jumlah			30	126

$$\boldsymbol{P = \frac{126}{150} \times 100\% = 84\%}$$

Berdasarkan hasil persentase diatas, maka dapat disimpulkan bahwa penilaian terhadap bahaya narkoba yang dimasukan kedalam *game* ini adalah 84% dari yang diharapkan 100%, atau dapat dikategorikan sebagai sangat setuju. Berikut kriteria interpretasi skor yang dapat dilihat pada Gambar 4.13.



Gambar 4. 13 Interpretasi skor pertanyaan ke-1

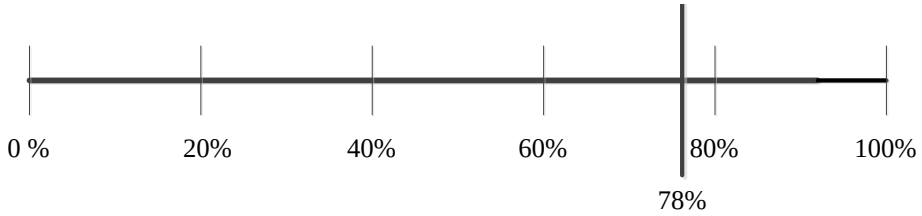
2. Apakah setelah memainkan *game Say No To Drugs* anda menjadi tahu bahwa narkoba itu harus dihindari ?

Tabel 4. 7 Perhitungan persentase pertanyaan ke-2

Pertanyaan	Keterangan	Skor	Frekuensi	S
2	Sangat Setuju	5	5	25
	Setuju	4	17	68
	Cukup Setuju	3	8	24
	Tidak Setuju	2	0	0
	Sangat Tidak Setuju	1	0	0
Jumlah			30	117

$$\boldsymbol{P = \frac{117}{150} \times 100\% = 78\%}$$

Berdasarkan hasil persentase diatas, maka dapat disimpulkan bahwa penilaian terhadap narkoba itu harus dihindari setelah memainkan *game* ini adalah 78% dari yang diharapkan 100%, atau dapat dikategorikan sebagai setuju. Berikut kriteria interpretasi skor yang dapat dilihat pada Gambar 4.14.



Gambar 4. 14 Interpretasi skor pertanyaan ke-2

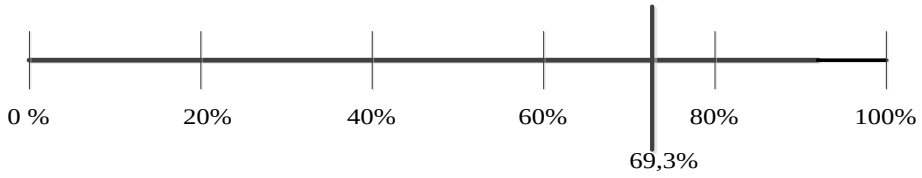
3. Apakah anda merasa kesulitan dalam memainkan game ini ?

Tabel 4. 8 Perhitungan persentase pertanyaan ke-3

Pertanyaan	Keterangan	Skor	Frekuensi	S
3	Sangat Setuju	1	0	0
	Setuju	2	3	6
	Cukup Setuju	3	10	30
	Tidak Setuju	4	17	68
	Sangat Tidak Setuju	5	0	0
Jumlah			30	104

$$\boldsymbol{P = \frac{104}{150} \times 100\% = 69,3\%}$$

Berdasarkan hasil persentase diatas, maka dapat disimpulkan bahwa penilaian terhadap kesulitan dalam memainkan *game* ini adalah 69,3% dari yang diharapkan 100%, atau dapat dikategorikan sebagai tidak setuju. Berikut kriteria interpretasi skor yang dapat dilihat pada Gambar 4.15.



Gambar 4. 15 Interpretasi skor pertanyaan ke-3

4. Apakah permainan pada *game Say No To Drugs* ini menyenangkan?

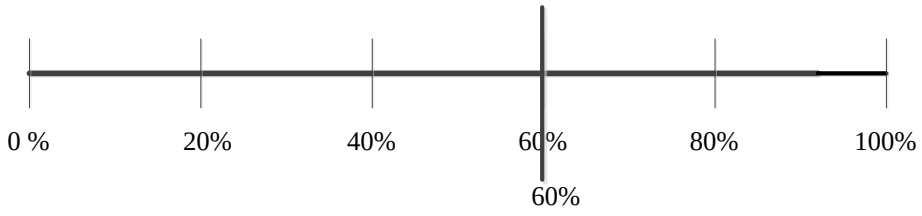
Tabel 4. 9 Perhitungan persentase pertanyaan ke-4

Pertanyaan	Keterangan	Skor	Frekuensi	S
4	Sangat Setuju	5	0	0
	Setuju	4	7	28

	Cukup Setuju	3	16	48
	Tidak Setuju	2	7	14
	Sangat Tidak Setuju	1	0	0
Jumlah			30	90

$$\boldsymbol{P = \frac{90}{150} \times 100\% = 60\%}$$

Berdasarkan hasil persentase diatas, maka dapat disimpulkan bahwa penilaian terhadap kesenangan dalam memainkan *game* ini adalah 60% dari yang diharapkan 100%, atau dapat dikategorikan sebagai cukup setuju. Berikut kriteria interpretasi skor yang dapat dilihat pada Gambar 4.16.



Gambar 4. 16 Interpretasi skor pertanyaan ke-4

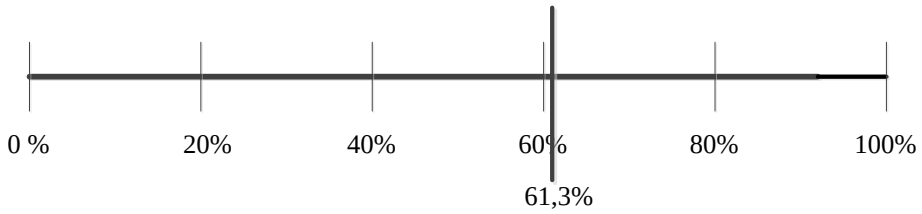
5. Apakah *game Say No To Drugs* ini membuat anda tertarik untuk memainkannya kembali ?

Tabel 4. 10 Perhitungan persentase pertanyaan ke-5

Pertanyaan	Keterangan	Skor	Frekuensi	S
5	Sangat Setuju	5	0	0
	Setuju	4	9	36
	Cukup Setuju	3	14	42
	Tidak Setuju	2	7	14
	Sangat Tidak Setuju	1	0	0
Jumlah			30	92

$$\boldsymbol{P = \frac{92}{150} \times 100\% = 61,3\%}$$

Berdasarkan hasil persentase diatas, maka dapat disimpulkan bahwa penilaian ketertarikan *user* terhadap *game* ini adalah 61,3% dari yang diharapkan 100%, atau dapat dikategorikan sebagai setuju. Berikut kriteria interpretasi skor yang dapat dilihat pada gambar 4.17.



Gambar 4. 17 Interpretasi skor pertanyaan ke-5

4.5.2.2 Kesimpulan Pengujian Beta

Dari hasil pengujian yang dilakukan, dengan cara menyebarkan kuisioner dapat disimpulkan bahwa penerapan sistem yang telah dibangun yaitu aplikasi game *Say No To Drugs* sesuai dengan tujuan yang ingin dicapai yaitu menyampaikan informasi tentang bahaya narkoba.