BAB 4
IMPLEMENTASI DAN PENGUJIAN SISTEM

4.1 Implementasi Sistem

Tahap implementasi sistem merupakan tahap penerjemahan perancangan berdasarkan hasil analisis ke dalam suatu bahasa pemrograman tertentu serta penerapan perangkat lunak yang dibangun pada lingkungan yang sesungguhnya. Adapun pembahasan implementasi terdiri dari perangkat lunak pembangun, perangkat keras pembangun dan implementasi antarmuka.

4.1.1 Implementasi Perangkat Keras

Berikut adalah spesifikasi perangkat keras yang digunakan untuk menguji *game* yang telah dibangun.

Tabel 4.1 Implementasi Perangkat Keras

Nama Perangkat	Spesifikasi
Processor	Intel Core 2 Duo T6600 2.20GHz
RAM	4096 MB
VGA	NVIDIA GeForce G210M
Harddisk	500 GB
Display	Resolusi 1366 x 768 pixels
Keyboard	Standar
Mouse	Standar

4.1.2 Implementasi Perangkat Lunak

Implementasi perangkat lunak digunakan untuk menjalankan aplikasi *game* yang telah dibangun dapat dilihat pada tabel berikut.

Tabel 4.2 Implementasi Perangkat Lunak

Nama Perangkat Lunak	Spesifikasi
Sistem Operasi	Microsoft Window 7 Home Premium
Game Engine	Unity 3D
Editor Gambar	Adobe Photoshop CS5
DirectX	DirectX 11

Sistem operasi harus sudah terpasang pada komputer untuk memainkan *game* yang telah dibangun. Untuk membuka *project* pada *game* yang dibangun maka pengguna harus menginstall Unity 3D dan komponen yang mendukungnya. Sedangkan untuk menjalankan *game* yang telah dibangun gunakan *file* Mygame.exe.

4.1.3 Implementasi Antarmuka

Adapun implementasi antarmuka pada aplikasi *game* yang telah dibangun adalah sebagai berikut.

4.1.3.1 Antarmuka Menu Utama

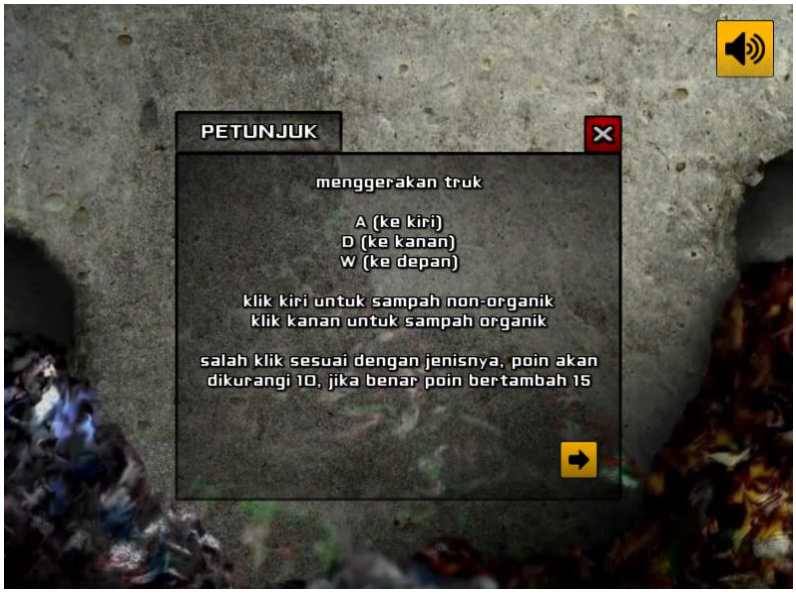
Antarmuka menu utama merupakan tampilan menu utama pada saat pengguna membuka aplikasi *game*. Implementasi antarmuka untuk perancangan menu utama pada *game* yang dibangun dapat dilihat pada gambar berikut.



Gambar 4.1 Antarmuka Menu Utama

4.1.3.2 Antarmuka Petunjuk

Pada antarmuka petunjuk akan ditampilkan informasi mengenai petunjuk cara bermain seperti tombol apa saja yang digunakan ketika bermain.



Gambar 4.2 Antarmuka Petunjuk

4.1.3.3 Antarmuka Menu Misi

Pada antarmuka menu misi berisi mengenai target yang harus dicapai saat permainan. Berikut adalah antarmuka menu misi.



Gambar 4.3 Antarmuka Menu Misi

4.1.3.4 Antarmuka Permainan

Berikut adalah tampilan antarmuka permainan mengumpulkan sampah. Terdapat tombol *pause*, *mute*, dan *recycle* serta waktu permainan poin yang didapat.



Gambar 4.4 Implementasi Permainan Mengumpulkan Sampah

4.1.3.5 Antarmuka *Pop-Up* Daur Ulang

Tampilan antarmuka pada permainan daur ulang dapat dilihat pada gambar berikut. Terdapat beberapa produk yang dapat dibuat dengan membutuhkan bahan – bahan dari sampah tertentu.



Gambar 4.5 Implementasi Permainan Daur Ulang

4.1.3.6 Antarmuka Permainan Gagal

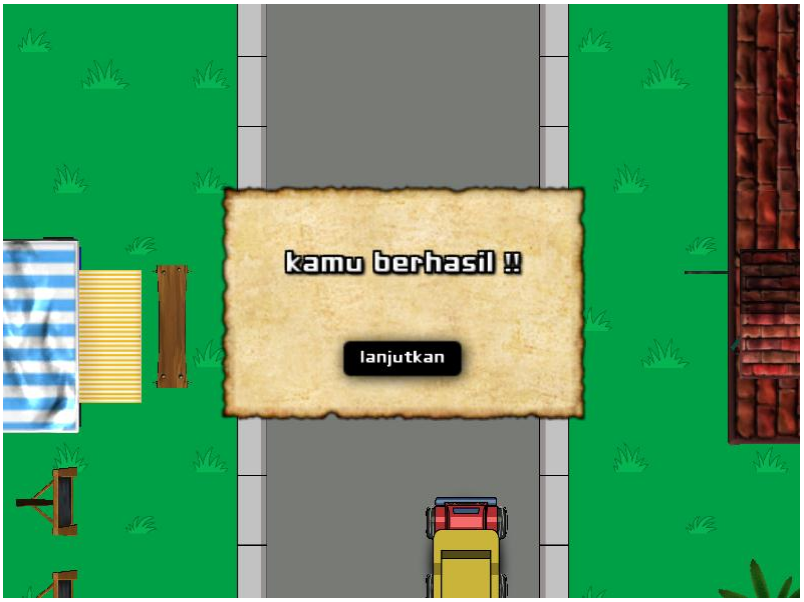
Tampilan antarmuka ketika permainan gagal diselesaikan dapat dilihat pada gambar berikut.



Gambar 4.6 Implementasi Antarmuka Permainan Gagal

4.1.3.7 Antarmuka Permainan Berhasil

Tampilan antarmuka ketika permainan berhasil diselesaikan dapat dilihat pada gambar berikut.



Gambar 4.7 Implementasi Antarmuka Permainan Berhasil

4.1.3.8 Antarmuka Pesan Keluar Aplikasi *Game*

Berikut adalah pesan konfirmasi saat pemain akan keluar dari aplikasi *game*.



Gambar 4.8 Implementasi Pesan Keluar Aplikasi *Game*

4.2 Pengujian Sistem

Dalam melakukan pengujian terhadap aplikasi *game* yang dibangun dilakukan dengan dua macam teknik, yaitu teknik pengujian *black box* dan teknik pengujian *white box*. 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* (fungsional) dan tahap pengujian *beta*.

4.2.1 Pengujian *Alpha*

Pada tahap ini, pengujian dilakukan untuk memastikan aplikasi *game* yang dibangun telah berjalan dengan benar dan sesuai dengan kebutuhan yang diinginkan. Pengujian *alpha* berfokus pada persyaratan fungsional perangkat lunak.

4.2.1.1 Skenario Pengujian Aplikasi

Skenario pengujian menjelaskan pengujian terhadap sistem yang ada pada aplikasi *game* yang dibangun. Skenario pengujian dari aplikasi *game* yang dibangun dapat dilihat pada tabel berikut.

Tabel 4.3 Skenario Pengujian Aplikasi

No	Komponen Yang Diuji	Skenario Pengujian	Jenis Pengujian
1.	Menu Utama	Pilih tombol Mulai	*Black Box*
		Pilih tombol Petunjuk	*Black Box*
		Pilih tombol Keluar	*Black Box*
2.	Menu Mulai Permainan	Pilih tombol Main	*Black Box*
		Pilih tombol *Pause*	*Black Box*
		Pilih tombol *Mute*	*Black Box*
		Pilih tombol *Resume*	*Black Box*
		Pilih tombol *Unmute*	*Black Box*
3.	Menu Petunjuk	Pilih tombol Selanjutnya	*Black Box*
		Pilih tombol Kembali	*Black Box*
		Pilih tombol X	*Black Box*
4.	Menu Keluar	Pilih tombol Ya	*Black Box*
		Pilih tombol Batal	*Black Box*
5.	Truk Sampah	Menekan tombol W	*Black Box*
		Menekan tombol A	*Black Box*
		Menekan tombol D	*Black Box*
		Klik tombol *mouse* kiri	*Black Box*
		Klik tombol *mouse* kanan	*Black Box*
		Menabrak mobil lain	*Black Box*
6.	Algoritma *A Star*	Algoritma *A Star* pada pergerakan mobil yang berdatangan	*White Box*

4.2.1.2 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:

1. **Pengujian Menu Utama**

Penguijian menu utama dengan kasus dan hasil uji, kemudian diamati dan menghasilkan kesimpulan dapat dilihat pada tabel berikut.

Tabel 4.4 Pengujian Menu Utama

Kasus dan Hasil Uji (Uji Data Benar)			
Masukan	**Keluaran yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Pilih menu mulai	Menampilkan layar permainan	Dapat menampilkan layar permainan	[√] Berhasil [ ] Tidak
Pilih menu petunjuk	Menampilkan petunjuk cara bermain	Dapat menampilkan petunjuk cara bermain	[√] Berhasil [ ] Tidak
Pilih menu keluar	Menampilkan pesan keluar dari aplikasi *game*	Dapat menampilkan pesan keluar dari aplikasi *game*	[√] Berhasil [ ] Tidak

2. Pengujian Mulai Permainan

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

Tabel 4.5 Pengujian Mulai Permainan

Kasus dan Hasil Uji (Uji Data Benar)			
Masukan	**Keluaran yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Pilih tombol main	Memainkan permainan	Dapat memainkan permainan	[√] Berhasil [ ] Tidak
Menyelesaikan level 1	Menampilkan pesan berhasil menyelesaikan permainan dan menampilkan tombol lanjutkan	Dapat menampilkan pesan berhasil menyelesaikan permainan dan menampilkan tombol lanjutkan	[√] Berhasil [ ] Tidak
Menyelesaikan level 2	Menampilkan pesan berhasil menyelesaikan permainan dan menampilkan tombol lanjutkan	Dapat menampilkan pesan berhasil menyelesaikan permainan dan menampilkan tombol lanjutkan	[√] Berhasil [ ] Tidak
Menyelesaikan level 3	Menampilkan pesan berhasil menyelesaikan permainan	Dapat menampilkan pesan berhasil menyelesaikan permainan	[√] Berhasil [ ] Tidak
Pilih menu *pause*	Memberhentikan permainan	Dapat memberhentikan permainan	[√] Berhasil [ ] Tidak

Pilih menu *resume*	Melanjutkan kembali permainan	Dapat melanjutkan kembali permainan	[√] Berhasil [ ] Tidak
Pilih menu *mute*	Mematikan suara permainan	Dapat mematikan suara permainan	[√] Berhasil [ ] Tidak
Pilih menu *unmute*	Menghidupkan suara permainan	Dapat menghidupkan suara permainan	[√] Berhasil [ ] Tidak
Pilih menu recycle	Menampilkan *pop-up* permainan *recycle*	Dapat menampilkan *pop-up* permainan *recycle*	[√] Berhasil [ ] Tidak
Waktu	Menampilkan waktu	Dapat menampilkan waktu	[√] Berhasil [ ] Tidak
Poin	Menampilkan poin	Dapat menampilkan poin	[√] Berhasil [ ] Tidak
Jumlah sampah	Menampilkan jumlah sampah yang telah didapat	Dapat menampilkan jumlah sampah yang telah didapat	[√] Berhasil [ ] Tidak
Jumlah sampah tidak mencukupi untuk membuat produk	Mematikan tombol produk	Dapat mematikan tombol produk	[√] Berhasil [ ] Tidak
Jumlah sampah mencukupi untuk membuat produk	Menghidupkan tombol produk	Dapat menghidupkan tombol produk	[√] Berhasil [ ] Tidak
Produk yang akan dibuat	Menampilkan produk yang dibuat	Dapat menampilkan produk yang dibuat	[√] Berhasil [ ] Tidak
Mengklik kiri atau kanan tombol *mouse* sesuai aturan	Poin bertambah	Poin bertambah	[√] Berhasil [ ] Tidak
Mengklik kiri atau kanan tombol *mouse* tidak sesuai aturan	Poin berkurang	Poin berkurang	[√] Berhasil [ ] Tidak
Gagal menyelesaikan permainan	Menampilkan pesan gagal dan animasi banjir	Dapat menampilkan pesan gagal dan animasi banjir	[√] Berhasil [ ] Tidak
Klik tombol lanjutkan	Menampilkan level selanjutnya	Dapat menampilkan level selanjutnya	[√] Berhasil [ ] Tidak

3. Pengujian Menu Petunjuk

Pengujian menu petunjuk dengan kasus dan hasil uji, kemudian diamati dan menghasilkan kesimpulan yang dapat dilihat pada tabel berikut.

Tabel 4.6 Pengujian Menu Petunjuk

Kasus dan Hasil Uji (Uji Data Benar)			
Masukan	**Keluaran yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Pilih menu petunjuk	Menampilkan petunjuk bermain	Dapat menampilkan petunjuk bermain	[√] Berhasil [ ] Tidak
Pilih panah selanjutnya	Menampilkan petunjuk selanjutnya	Dapat menampilkan petunjuk selanjutnya	[√] Berhasil [ ] Tidak
Pilih panah kembali	Kembali ke petunjuk sebelumnya	Dapat kembali ke petunjuk sebelumnya	[√] Berhasil [ ] Tidak
Klik tanda silang	Menutup tampilan petunjuk bermain dan kembali ke menu utama	Dapat menutup tampilan petunjuk bermain dan kembali ke menu utama	[√] Berhasil [ ] Tidak

4. Pengujian Menu Keluar

Pengujian menu keluar dengan kasus dan hasil uji, kemudian diamati dan menghasilkan kesimpulan yang dilihat pada tabel berikut.

Tabel 4.7 Pengujian Menu Keluar

Kasus dan Hasil Uji (Uji Data Benar)			
Masukan	**Keluaran yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Pilih menu Keluar	Menampilkan pesan keluar dari aplikasi *game*	Dapat menampilkan pesan keluar dari aplikasi *game*	[√] Berhasil [ ] Tidak
Pilih tombol ya	Menutup aplikasi *game*	Dapat menutup aplikasi *game*	[√] Berhasil [ ] Tidak
Pilih tombol batal	Kembali ke menu utama	Dapat kembali ke menu utama	[√] Berhasil [ ] Tidak

4.2.1.3 Kasus dan Hasil Pengujian *(White Box)*

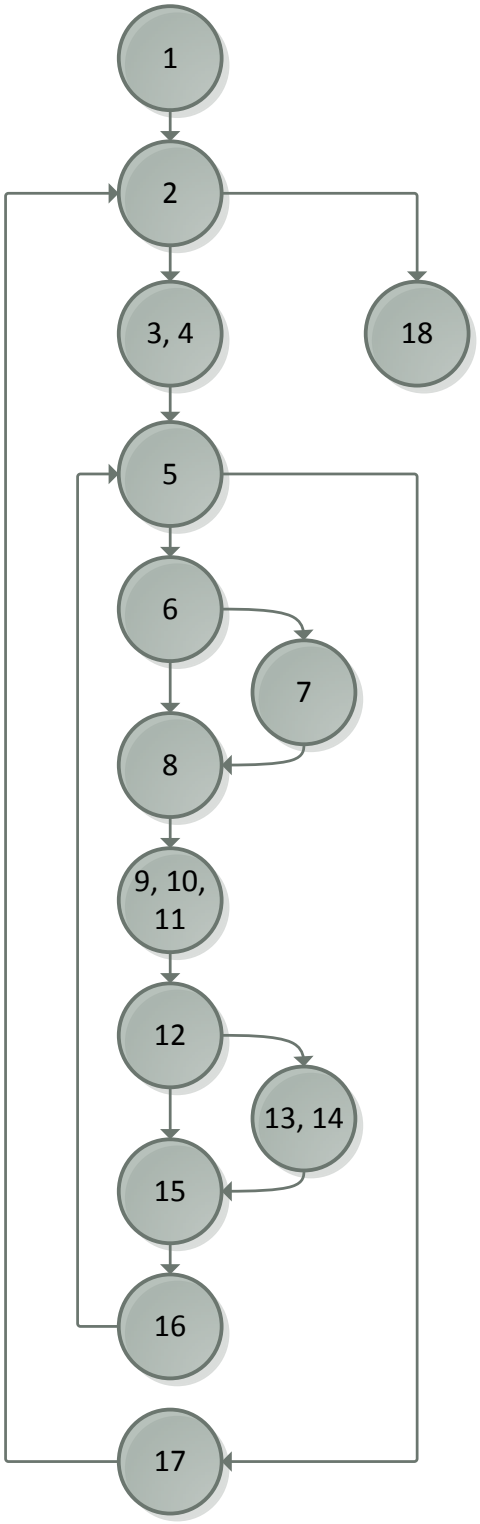
Berikut ini adalah kasus menguji perangkat lunak yang telah dibangun menggunakan metode *white box*. Pengujian *white box* menggunakan *flow graph* yang digunakan untuk menggambarkan alur dari algoritma dan *grap matrix* yang digunakan untuk menggenerasi *flow graph*. Adapun pengujian *white-box* adalah sebagai berikut.

1. Pengujian Algoritma A* (A *Star*)

Pengujian algoritma A* merupakan proses dimana akan diuji implementasi A* terhadap NPC mobil dalam melakukan pencarian tujuan.

```
1.  Masukkan startNode ke dalam openList
2.  while openList ≠ kosong do
3.      bestNode ← f terkecil dalam openList
4.      keluarkan bestNode dari openList
5.      for each bestNode
6.          if bestNode = goalNode then
7.              selesai
8.          endif
9.          g ← jarak antara startNode ke node sekitarnya
10.         h ← jarak dari bestNode ke goalNode
11.         f ← g + h
12.         if node baru di openList memiliki nilai f lebih kecil dari bestNode
13          then
14.             simpan node baru sebagai bestNode baru
15.         endif
16.     endfor
17.     keluarkan bestNode lama dari openList dan masukkan ke closedList
18. endwhile
```

2. *Flow Graph*



Gambar 4.9 *Flow Graph* Algoritma A*

Keterangan:

 = Menggambarkan kondisi

 = Menggambarkan aksi

3. *Cyclomatic Complexity* V(G)

V(G) = E – N + 2

V(G) = 17 – 14 + 2

V(G) = 5

Keterangan:

E = Jumlah busur pada *flowgraph*

N = Jumlah simpul pada *flowgraph*

V(G) = *Cyclomatic Complexity*

4. *Independent Path*

Path 1: 1-2-18

Path 2: 1-2-3-4-5-17-2-18

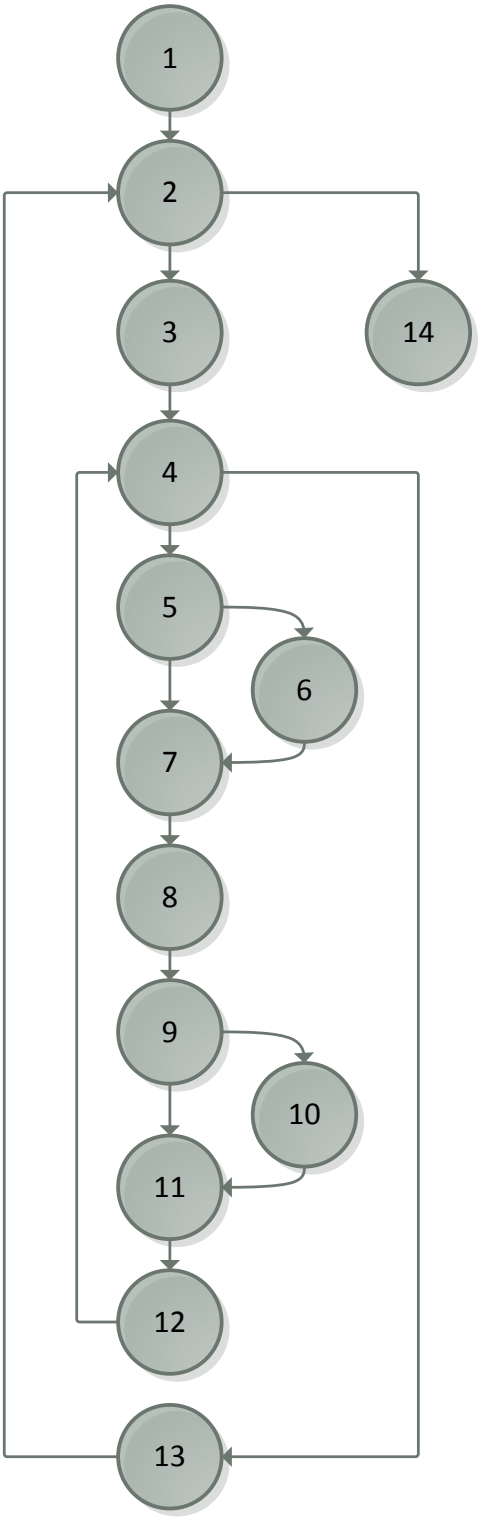
Path 3: 1-2-3-4-5-6-8-9-10-11-12-15-16-5-17-2-18

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

Path 5: 1-2-3-4-5-6-8-9-10-11-12-13-14-15-16-5-17-2-18

Path 6: 1-2-3-4-5-6-7-8-9-10-11-12-15-16-5-17-2-18

5. Penyederhanaan *Flow Graph*



Gambar 4.10 Penyederhanaan *Flow Graph*

6. *Graph Matrices Testing*

Tabel 4.8 *Graph Matrices Testing*

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**Hasil**
1		1													0
2			1											1	1
3				1											0
4					1								1		1
5						1	1								1
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															0
Jumlah *graph matrices*															4

V(G) = jumlah *graph matrices* + 1

V(G) = 4 + 1

V(G) = 5

7. Kesimpulan

Berdasarkan hasil pengujian dihasilkan nilai *cyclomatic complexity* yang sama yaitu 3, maka didapat kesimpulan bahwa pada proses algoritma A* berjalan dengan baik, karena hasil pengujian memiliki nilai yang sama.

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 kuesioner mengenai kepuasan para pengguna terhadap aplikasi permainan yang dibangun. Pengujian dilakukan terhadap 30 responden dengan usia minimal 9 tahun dan setiap responden diberikan 5 pertanyaan untuk kemudian dilakukan penilaian dengan memilih 5 pilihan yang telah disediakan untuk setiap pernyatannya.

Skala yang digunakan dalam pengukuran kuesioner adalah skala *guttman*, yaitu skala yang menginginkan tipe jawaban tegas, seperti jawaban benar – salah, ya – tidak, pernah – tidak pernah, positif – negative, dan seterusnya. Pada skala *guttman*, hanya ada dua interval, yaitu setuju dan tidak setuju. Pada kuesioner ini tipe skala *guttman* yang digunakan adalah ya – tidak.

4.2.2.1 Rumusan Pengujian Beta

Berdasarkan data hasil kuesioner tersebut, dicari persentase masing – masing jawaban dengan menggunakan rumus:

$$\mathbf{R = P/Q * 100\%} \qquad (4.1)$$

Keterangan:

P = Banyaknya jawaban responden tiap soal

Q = Jumlah Responden

R = Nilai persentase

4.2.2.2 Hasil Kuesioner

Berikut ini adalah daftar pertanyaan kuesioner yang diberikan, serta hasil yang didapat setelah nilainya dihitung.

1. Apakah *game* ini menarik untuk dimainkan?
    A. Ya
    B. Tidak

Kategori Jawaban	Ya	Tidak
Frekuensi jawaban	17	13
Persentase jawaban	56.67	43.33%

2. Apakah kamu kesulitan dalam memainkan *game* ini?
    A. Ya
    B. Tidak

Kategori Jawaban	Ya	Tidak
Frekuensi jawaban	10	20
Persentase jawaban	33.33%	66.67%

3. Apakah penyampaian mengenai sampah organik dan non – organik pada *game* ini cukup jelas?
    A. Ya
    B. Tidak

Kategori Jawaban	Ya	Tidak
Frekuensi jawaban	21	9
Persentase jawaban	70%	30%

4. Apakah penyampaian mengenai penanganan sampah dengan mendaur ulang pada *game* ini cukup jelas?
    A. Ya
    B. Tidak

Kategori Jawaban	Ya	Tidak
Frekuensi jawaban	14	16
Persentase jawaban	46.67%	53.33%

5. Apakah kamu tertarik untuk menggunakan sampah kembali setelah memainkan *game* ini?
   A. Ya
   B. Tidak

Kategori Jawaban	Ya	Tidak
Frekuensi jawaban	23	7
Persentase jawaban	76.67%	23.33%

Kriteria kesimpulan hasil tersebut adalah sebagai berikut:

1. Jika jawaban "Ya" 15 – 20 poin = *Game* sudah sangat sesuai
2. Jika jawaban "Ya" 7 – 15 poin = *Game* cukup baik
3. Jika jawaban "Ya" 4 – 6 poin = *Game* tidak disarankan untuk dimainkan
4. Jika jawaban "Ya" 1 – 3 poin = *Game* yang dibangun dilarang untuk dimainkan

4.2.2.3 Kesimpulan Pengujian *Beta*

Berdasarkan hasil pengujian *beta*, maka didapatkan kesimpulan sebagai berikut:

1. Aplikasi *game* yang dibangun menarik untuk dimainkan
2. Aplikasi *game* yang dibangun tidak sulit untuk dimainkan meskipun memiliki tingkat kesulitan yang berbeda pada setiap levelnya
3. Pemilahan jenis sampah organik dan non – organik sangat tersampaikan pada aplikasi *game* ini.
4. Penanganan sampah dengan mendaur ulang sudah tersampaikan dengan baik pada aplikasi *game* ini, meskipun 53.33% responden menyatakan penyampaian daur ulang tidak cukup jelas dalam permainan ini
5. Dengan *game* ini dapat menambah ketertarikan pengguna dalam menggunakan sampah kembali.