BAB 4

IMPLEMENTASI DAN PENGUJIAN

4.1 Implementasi

Proses implementasi merupakan tahapan dimana aplikasi yang sudah dibuat dan di rancang lalu di uji kelayakannya. Pemograman dalam sistem ini menggunakan Greenfoot version 2.1.2. Selanjutnya dioperasikan sebagaimana mestinya sesuai fungsi dan kelayakannya untuk digunakan.

4.1.1 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.

4.1.1.1. Perangkat Keras (*Hardware*)

Perangkat keras yang digunakan selama pengujian aplikasi ini memiliki spesifikasi sebagai berikut:

1. Prosesor dengan kecepatan 1.46 Ghz
2. Memori 256 Mb
3. Harddisk 10 GB
4. VGA card 256 Mb
5. Monitor
6. Mouse dan Keyboard

4.1.1.2 Perangkat Lunak (*Software*)

Perangkat lunak yang digunakan untuk menjalankan *game* edukasi 2Duntuk anak-anak prasekolah ini adalah sebagai berikut:

1. Sistem Operasi Windows XP, Windows Vista, Windows 7
2. JDK (*Java Development Kit*) 6 Minimal

4.2 Implementasi Aplikasi

File yang ada pada *game* edukasi 2D untuk anak-anak prasekolah ini, dibungkus menjadi satu dalam bentuk *Game* Membuang Sampah.Jar, berikut langkah-langkah untuk menjalankan *game* edukasi Membuang Sampah ini:

1. *Double* Klik pada file *Game* Membuang Sampah.Jar yang ada di *folder Game* Membuang Sampah.
2. Akan tampil *window Game* Membuang Sampah, dan itu menandakan bahwa *game* sukses dijalankan.

4.2.1 Implementasi Class

Pada tahap pembangunan game dibutuhkan class yang sesuai dengan pemodelan class diagram pada analisis sistem. Implementasi class pada *game* Membuang Sampah Bersama Si Unyil adalah sebagai berikut :

1. Class WasteWorld

Tabel 4.1 Implementasi class WasteWorld

```
public static WasteWorld getWasteWorld() {
        return WasteWorld.wasteWorld;
    }
```

2. Class Si Unyil

Tabel 4.2 Implementasi class Si Unyil

```
public Boy getBoy() {
        return this.boy;
    }
```

3. Class Beaddie

Tabel 4.3 Implementasi class Beaddie

```
public Baddie(int c, int r, int d,
Class<?>... cls) {
          this.imgs = new
GreenfootImage[4];
          for (int i = 0; i < 4; i++) {
               this.imgs[i] = new
GreenfootImage("baddie-" + i + "-0" +
".png");
          }
```

4. Class Sampah

Tabel 4.4 Implementasi class Sampah

```
public Waste(int type) {
          super(type);
     }
```

5. Class Waktu

Tabel 4.5 Implementasi class Waktu

```
public class Clock extends Actor {
     }
public static Clock getClock(int secs) {
          Clock.clock.maxTime = secs;
          Clock.clock.updateImage();
          return Clock.clock;
     }
```

6. Class Level

Tabel 4.6 Implementasi class Level

```
public static Level init(Item[][] cells) {
          // create the level objects
    }
```

7. Class SetLevelSelection

Tabel 4.7 Implementasi class SetLevelSelection

```
public class LevelSelection extends Level
{

     }
```

8. Class Halaman Rumah

Tabel 4.8 Implementasi class rumah

```
public void createLevel(Item[][] items) {
        this.numberOfWaste = 0;
        this.worldFilter = null; // no
world filter
        this.cellFilter = null; // no cell
filter

        // empty all cells
        for (int c = 0; c < items.length;
c++) {


}
```

9. Class Halaman Sekolah

Tabel 4.9 Implementasi class Halaman Sekolah

```
public void createLevel(Item[][] items) {
        this.numberOfWaste = 0;
        this.worldFilter = null; // no
world filter
        this.cellFilter = null; // no cell
filter
```

```
            // empty all cells
            for (int c = 0; c < items.length;
    c++) {

    }
```

10. Class Taman Bermain

Tabel 4.10 Implementasi class Taman Bermain

```
    public void createLevel(Item[][] items) {
              this.numberOfWaste = 0;
              this.cellFilter = null; // no
    cell filter

              // empty all cells
              for (int c = 0; c <
    items.length; c++) {
                   for (int r = 0; r <
    items[c].length; r++) {
                        items[c][r] = null;
                   }
              }
    public int getSeconds() {
              return 220;
         }
    }
```

4.2.2 Implementasi Antar Muka

1. Tampilan Utama



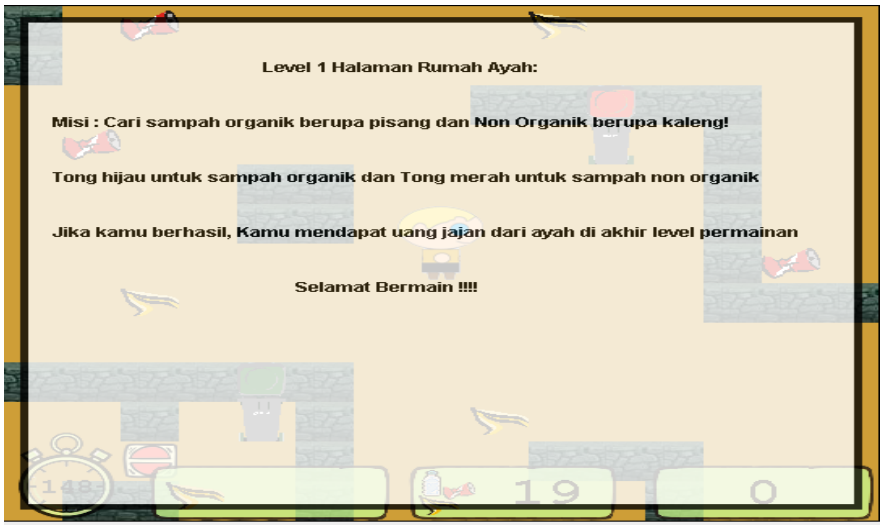
Gambar 4.1 Tampilan Menu Utama

Gambar 4.1 berfungsi menampilkan Tampilan utama atau tampilan awal *game* dimana didalamnya tersedia tiga pilihan menu *game*, yaitu Halaman Rumah, Halaman Sekolah dan Taman Bermain. Pada menu utama ini terdapat penjelasan atau petunjuk *game*, agar pemain dapat mengetahui bagaimana cara menggunakan *game* tersebut.

2. Tampilan Pemilihan Halaman Rumah



Gambar 4.2 Tampilan Pemilihan *Game*play Halaman Rumah



Gambar 4.3 Tampilan Misi Halaman Rumah

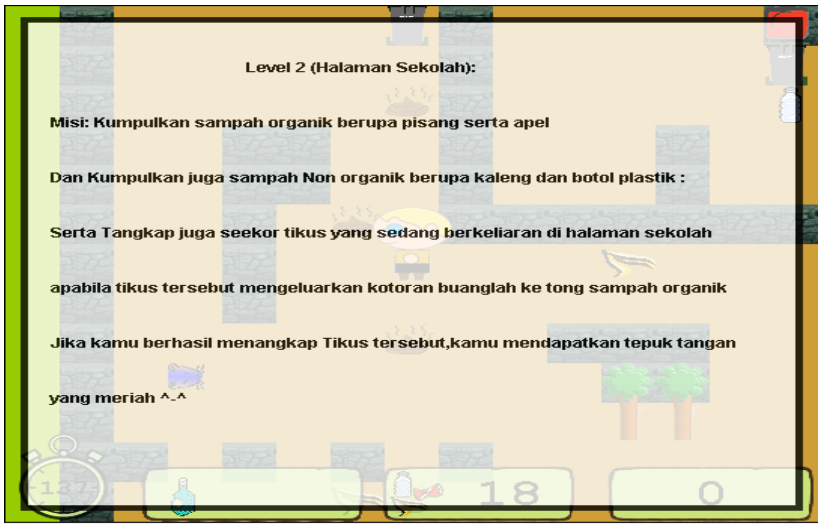
Gambar 4.2 berfungsi menampilkan pemilihan menu halaman rumah yang diinginkan oleh pemain, dimana pada bagian halaman rumah ini terdapat 2 jenis sampah yaitu pisang dalam jenis organik serta kaleng yang berjenis non organik. Permainan di halaman rumah akan berakhir jika misi yang diberikan

telah diselesaikan. Pada gambar 4.3 berfungsi untuk meliaht misi pada stage halaman rumah.

3. Tampilan Permainan Halaman Sekolah



Gambar 4.4 Tampilan *Game*play Halaman Sekolah



Gambar 4.5 Tampilan Misi pada Halaman Sekolah



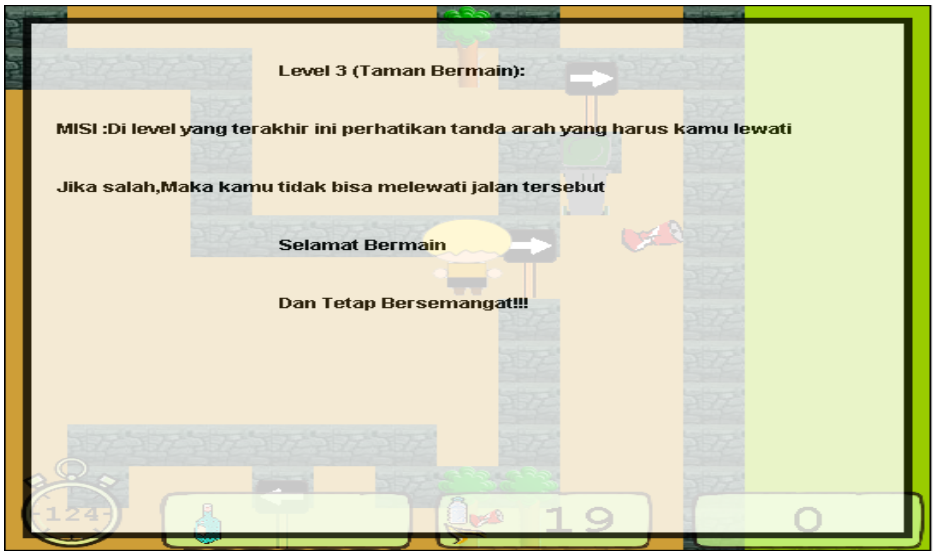
Gambar 4.6 Tampilan Tikus pada Halaman Sekolah

Gambar 4.4berfungsi menampilkan pemilihan menu halaman sekolah yang diinginkan oleh pemain.Pada gambar 4.5 berfungsi untuk meliaht misi pada stage halaman. Pada bagian halaman sekolah ini terdapat seekor tikus seperti pada gambar 4.6. Permainan di halaman rumah akan berakhir jika misi yang diberikan telah diselesaikan. Pada gambar 4.3 berfungsi untuk meliaht misi pada stage halaman rumah.

4. Tampilan Permainan Taman Bermain



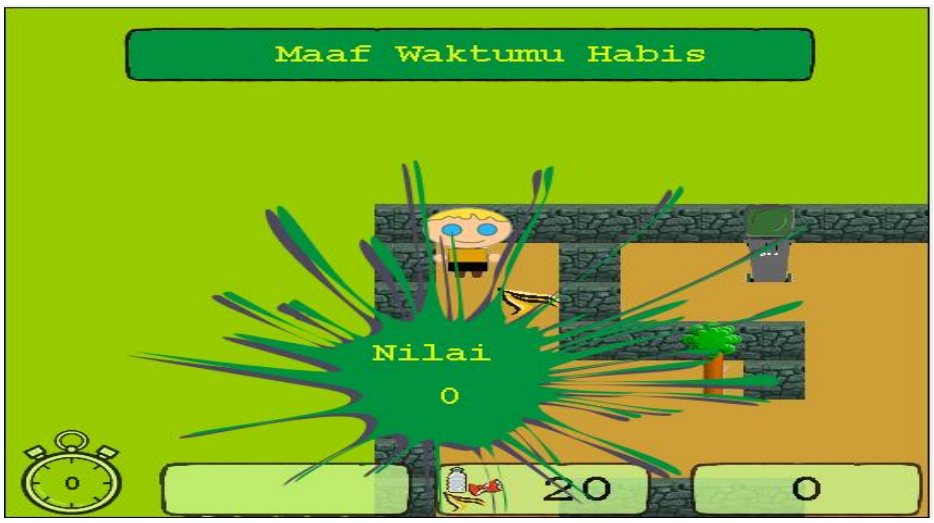
Gambar 4.7 Tampilan *Game*play Taman Bermain



Gambar 4.8 Tampilan Misi Pada Taman Bermain

Gambar 4.7berfungsi menampilkan pemilihan menu taman bermain yang diinginkan oleh pemain, dimana pada bagian taman bermain ini terdapat misi yang dapat dilihat seperti pada gambar 4.8 Permainan di taman bermain akan berakhir jika misi yang diberikan telah diselesaikan.

5. Tampilan Waktu Berakhir



Gambar 4.9 Tampilan waktu berakhir

Gambar 4.9 merupakan tampilan pada saat waktu permainan berakhir apabila pemain tidak berhasil menyelesaikan permainan.

6. Tampilan Score



Gambar 4.10 Tampilan Score

Gambar 4.10 merupakan tampilan score yang di dapat apabila pemain telah berhasil menyelesaikan permainan

4.3 Pengujian Sistem

Pengujian sistem merupakan hal terpenting yang dilakukan untuk menemukan kekurangan atau kesalahan pada perangkat lunak yang diuji.Pengujian bermaksud untuk mengetahui perangkat lunak yang dibuat sudah memenuhi kriteria yang sesuai dengan tujuan perancangan perangkat lunak tersebut. Pengujian yang dilakukan yaitu pengujian white box, pengujian alpha dan pengujian beta.

4.3.1 Pengujian White box

Pengujian white-box digunakan untuk meyakinkan semua perintah dan kondisi pada aplikasi dieksekusi secara minimal. Pengujian white-box menggunakan dua tools yaitu flow graph yang digunakan untuk menggambarkan alur dari algoritma dan graph matrix yang digunakan untuk menggenerasi flow graph. Adapun pengujian white-box nya adalah sebagai berikut :

1. Pengujian *generate and test pada gerak tikus*

   Berikut algoritma *generate and test pada gerak tikus*:

Tabel 4.11 algoritma *generate and test pada gerak tikus*:

No	Algoritma
1	boolean freeAhead :
2	switch this.direction :
3	case RIGHT: return !this.world.cellOccupied(this.col + 1, this.row);
4	case DOWN: return !this.world.cellOccupied(this.col, this.row + 1);
5	case LEFT: return !this.world.cellOccupied(this.col - 1, this.row);
6	default: return !this.world.cellOccupied(this.col, this.row - 1);

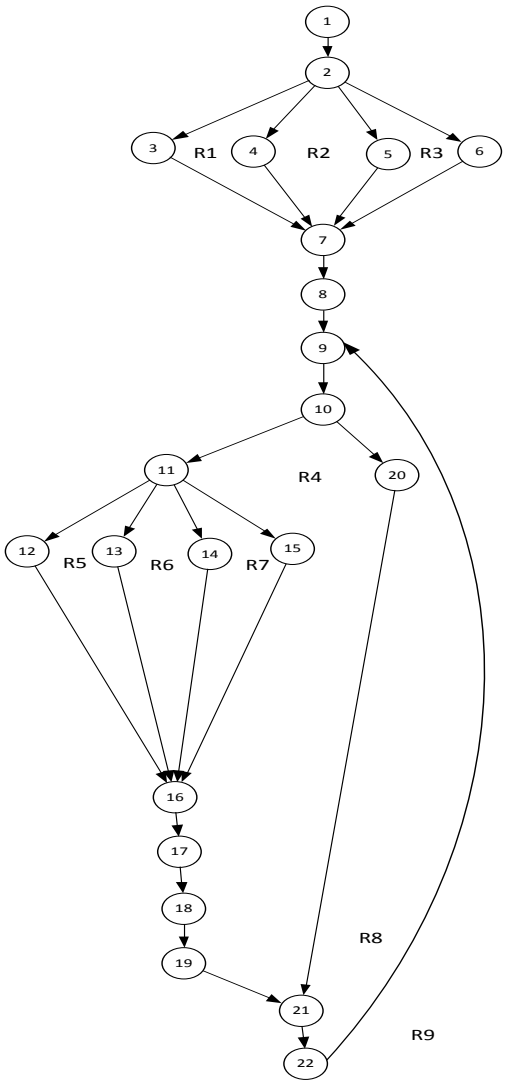
7	EndSwicth
8	boolean moved = false;
9	FOR i = 1 TO 4
10	IF this.freeAhead THEN
11	Swicth this.direction
12	case RIGHT: this.col = this.col + 1; break;
13	case DOWN: this.row = this.row + 1; break;
14	case LEFT: this.col = this.col - 1; break;
15	default: this.row = this.row - 1; break;
16	ENDSwicth
17	this.world.removeItem(this.oldCol, this.oldRow);
18	this.world.changeItem(this.col, this.row, this);
19	break;

20	ELSE
	this.direction = (this.direction + 1) % 4;
	this.setImage(this.imgs[this.direction]);

21	ENDIF
22	ENDFOR

a. *Flow Graphgenerate and test pada gerak tikus*

Berikut ini adalah *flow graph generate and test pada gerak tikus*



Gambar 4.11 *flow graph generate and test pada gerak tikus*

Keterangan :

 = Menggambarkan kondisi

 = Menggambarkan aksi

Dari Gambar x.x dapat dihitung *cyclomatic complexity* sebagai berikut :

Region = 9

V(G) = 29 *edge* - 22 *node* + 2 = 9

Jadi, *cyclomatic complexity* untuk gambar x.x adalah 12

Berdasarkan *cyclomatic complexity* tersebut, maka terdapat 12 *path*yang terdiri dari :

Path 1 1-2-3-7-8-9-10-11-12-16-17-18-19-21-22-9

Path 2 1-2-3-7-8-9-10-11-12-16-17-18-19-21-22

Path 3 1-2-3-7-8-9-10-11-13-16-17-18-19-21-22

Path 4 1-2-3-7-8-9-10-11-14-16-17-18-19-21-22

Path 5 1-2-3-7-8-9-10-11-15-16-17-18-19-21-22

Path 6 1-2-3-7-8-9-10-20-12-16-17-18-19-21-22

Path 7 1-2-4-7-8-9-10-11-12-16-17-18-19-21-22

Path 8 1-2-5-7-8-9-10-11-12-16-17-18-19-21-22

Path 9 1-2-6-7-8-9-10-11-12-16-17-18-19-21-22

b.*Graph Matrixgenerate and test pada gerak tikus*

Berikut ini adalah *graph matrix generate and test* pada gerak tikus

Tabel 4.12 *graph matrixgenerate and test* pada gerak tikus

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	E
1		1																					0
2			1	1	1	1																	3
3							1																0
4							1																0
5							1																0
6							1																0
7								1															0
8									1														0
9										1													0
10											1									1			1
11												1	1	1	1								3
12																1							0
13																1							0
14																1							0
15																1							0
16																	1						0
17																		1					0
18																			1				0
19																					1		0
20																					1		0
21																						1	0
22									1														0
SUM (E) + 1																							9

4.3.2 Pengujian Alpha (Fungsional)

Pengujian *fungsional* yang digunakan untuk menguji sistem yang baru adalah metode pengujian *alpha*. Pengujian *alpha* berfokus pada persyaratan fungsional perangkat lunak.

1. Rencana Pengujian

   Rencana pengujian yang akan dilakukan pada aplikasi ini selengkapnya terlihat pada Tabel 4.13

Tabel 4.13 Rencanan Pengujian Aplikasi

No	Komponen yang di Uji	Butir Uji	Jenis Uji
1.	Menu	Menekan Tombol Run	*Black Box*
		Memilih Menu Halaman Rumah	*Black Box*
		Memilih MenuHalaman Sekolah	*Black Box*
		Memilih MenuTaman Bermain	*Black Box*
		Memilih Menu Petunjuk	*Black Box*
		Menekan Menu Misi	*Black Box*
2.	Gerak Karakter	Bergerak ke Atas	*Black Box*
		Bergerak ke Bawah	*Black Box*
		Bergerak ke Kanan	*Black Box*
		Bergerak ke Kiri	*Black Box*
		Bergerak ke kanan	*Black Box*
3.	Gerak Tikus	Tikus bergerak	*White Box*
4.	Menampilkan Sampah	Menampilkan Sampah	*Black Box*
5.	Menampilkan Waktu	Menampilkan Waktu	*Black Box*

2. Kasus dan Hasil Pengujian

Berikut ini adalah kasus untuk menguji perangkat lunak yang sudah dibangun menggunakan metode *BlackBox* berdasarkan Tabel 4.13 Rencanapengujian yang telah dibuat sebelumnya.

Tabel 4.14 Kasus dan Hasil Uji

No	Kasus / Diuji	Skenario Uji	Hasil yang diharapkan	Hasil Pengujian
1	Pemilihan menu	Memilih Menu Halaman Rumah	Masuk ke permainan yang ada di halaman rumah	[√] Berhasil [ ] Tidak Berhasil
		Memilih MenuHalaman Sekolah	Masuk ke permainan yang ada di halaman sekolah	[√] Berhasil [ ] Tidak Berhasil
		Memilih MenuTaman Bermain	Masuk ke permainan yang ada di taman bermain	[√] Berhasil [ ] Tidak Berhasil
		Menekan Tombol Misi	Menampilkan misi yang harus diselesaikan	[√] Berhasil [ ] Tidak Berhasil
2.	Karakter Bergerak	Bergerak ke Atas	Ketika pemain menekan panah atas karakter pemain bergerak ke atas	[√] Berhasil [ ] Tidak Berhasil
		Bergerak ke kiri	Ketika pemain menekan panah kiri Karakter pemain bergerak ke arah kiri	[√] Berhasil [ ] Tidak Berhasil
		Bergerak ke kanan	Ketika pemain menekan panah kanan Karakter	[√] Berhasil [ ] Tidak Berhasil

			pemain bergerak ke arah kanan	
3	Gerak Tikus	Gerak Tikus	Ketika pemain masuk pada permainan halaman sekolah tikus langsung bergerak dan membuang kotoran	[√] Berhasil [ ] Tidak Berhasil
4	Menampilkan sampah	Menampilkan Sampah	Ketika pemain masuk pada permainan yang dipilih maka sampah-sampah ditampilkan	[√] Berhasil [ ] Tidak Berhasil
5	Menampilkan Waktu	Menampilkan Waktu	Ketika pemain masuk pada permainan yang dipilih maka waktu akan tampil	[√] Berhasil [ ] Tidak Berhasil

4.3.2.1 Kesimpulan Pengujian Alpha

Berdasarkan hasil pengujian alpha (fungsional) dengan kasus uji diatas dapat ditarik kesimpulan bahwa pembangunan *Game* Membuang Sampah Bersama Si Unyil tidak terdapat kesalahan proses dan secara fungsional mengeluarkan hasil yang sesuai dengan yang diharapkan.

4.3.3 Pengujian *Beta*

4.3.3.1 Evaluasi dari Pengguna

Untuk mengetahui tanggapan dan penilaian pemain terhadap *game* ini. Telah disebarkan kuesioner kepada 30 responden dengan usia yang berdasarkan pada sasaran user. Berikut adalah pertanyaan dan hasil kuesioner yang telah dibagikan dengan menggunakan rumus :

Y = P/Q * 100%

Keterangan :

P = Banyaknya jawaban responden tiap soal

Q = Jumlah responden

Y = Nilai prosentase

Hasil pengujian *Beta* adalah sebagai berikut:

1. Apakah *Game* Membuang Sampah Bersama Si Unyil ini menonjolkan sisi edukasi untuk pengguna ? (orang tua)

Kategori Jawaban	Ya	Tidak
Frekuensi Jawaban	18	12
Persentase Nilai	80%	20%

2. Apakah setelah bermain *Game* Membuang Sampah Bersama Si Unyil ini pengguna mengenal sampah organik dan non organik? (orang tua)

Kategori Jawaban	Ya	Tidak
Frekuensi Jawaban	17	13
Persentase Nilai	70%	30%

3. Apakah menurut pengguna penyampain cara membuang sampah pada *Game* Membuang Sampah Bersama Si Unyil ini cukup jelas ? (orang tua dan pengguna)

Kategori Jawaban	Ya	Tidak
Frekuensi Jawaban	16	14
Persentase Nilai	60%	40%

4. Apakah petunjuk yang terdapat pada *Game* Membuang Sampah Bersama Si Unyil membantu adik-adik dalam memainkan *game*? (pengguna)

Kategori Jawaban	Ya	Tidak
Frekuensi Jawaban	16	14
Persentase Nilai	60%	40%

5. Apakah pengguna kesulitan dalam menyelesaikan misi yang ada pada *Game* Membuang Sampah Bersama Si Unyil ini? (pengguna)

Kategori Jawaban	Ya	Tidak
Frekuensi Jawaban	18	12
Persentase Nilai	80%	20%

4.3.3.2 Kesimpulan Pengujian Beta

Berdasarkan jawaban hasil kuesioner yang diberikan kepada para responden, maka dapat diambil kesimpulan yaitu:

1. Setelah bermain *Game* Membuang Sampah Bersama Si Unyil tersebut pengguna dapat menerima sisi edukasi
2. Pengguna dapat mengenal apa yang di maksud sampah organik dan non organik.
3. Petunjuk yang diberikan untuk memainkan *game* ini sudah cukup membantu user dalam memainkan *game*
4. Misi-misi yang diberikan sebagai tantangan permainan pada game ini lebih disesuaikan dengan pengguna.