BAB IV
IMPLEMENTASI DAN PENGUJIAN SISTEM

IV.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.

IV.1.1 Implementasi Perangkat Keras

Berikut ini adalah spesifikasi perangkat keras yang digunakan untuk menguji aplikasi *game Save Our Forest.*jar terdapat pada tabel IV.1

Tabel IV.1 Implementasi Perangkat Keras

Nama Perangkat	Spesifikasi
Prosesor	1.0 Ghz
RAM	256 MB
Harddisk	Minimal 1GB
VGA	512 MB
Display	Resolusi 1366 x 768 pixels
Keyboard	Standar
Mouse	Standar
Speaker	Standar

IV.1.2 Perangkat Lunak (*Software*)

Implementasi perangkat lunak digunakan untuk menjalankan aplikasi *game Save Our Forest.*jar**,** terdapat pada tabel IV.2

Tabel IV.2 Implementasi Perangkat Lunak

Nama Perangkat Lunak	Spesifikasi
Sistem Operasi	Windows 7 Ultimate 32 Bit
Tools	JDK 1.7

IV.1.3 Implementasi Aplikasi

Save Our Forest merupakan aplikasi permainan berjenis *action games* dengan kategori *platform*. Pemain mendapatkan misi setiap *level* nya yaitu untuk mengumpulkan bibit tanaman yang tersebar di area permainan dan mengalahkan semua musuh. Selama dalam perjalanan, pemain dihadang oleh musuh-musuh yang menyerang pemain.

Permainan dikatakan berhasil apabila telah berhasil mengalahkan semua musuh yang ada di setiap *level* nya dan mengalahkan *boss* yang ada di *level* terakhir, sedangkan permainan dikatakan gagal apabila pemain terbunuh oleh musuh atau masuk ke dalam jurang.

Aplikasi permainan ini berbasis *Java desktop* yang dibangun dengan menggunakan bahasa pemograman *java* dan *tools greenfoot* sehingga permainan ini hanya bisa dijalankan apabila pengguna telah menginstall JDK (*Java Development Kit*) pada perangkat komputernya.

IV.1.4 Implementasi Antarmuka

Berikut ini adalah implementasi antarmuka dari *game Save Our Forest*.

IV.1.4.1 Tampilan Menu Utama *Game*

Pada tampilan menu utama *game* ini terdapat 3 menu yaitu mulai untuk memulai permainan, panduan untuk menampilkan panduan cara bermain *game* dan tentang untuk menampilkan informasi pembuat *game*. Tampilan menu utama *game* dapat dilihat pada gambar IV.1 :



Gambar IV.1 Tampilan Menu Utama *Game*

IV.1.4.2 Tampilan Petunjuk *Game*

Pada tampilan petunjuk *game* terdapat tampilan informasi tombol *keyboard* yang digunakan pada *game*. Tampilan petunjuk dapat dilihat pada gambar IV.2 :



Gambar IV.2 Tampilan Menu Petunjuk

IV.1.4.3 Tampilan Kredit *Game*

Pada tampilan menu kredit terdapat tampilan informasi pembuat *game*. Tampilan panduan dapat dilihat pada gambar IV.3 :



Gambar IV.3 Tampilan Menu Kredit

IV.1.4.4 Tampilan *Intro Game* 1

Pada tampilan *intro game* 1 berisi mengenai kisah *game* secara singkat. Tampilan *intro game* 1 dapat dilihat pada gambar IV.4 :



Gambar IV.4 Tampilan *Intro Game* 1

IV.1.4.5 Tampilan *Intro Game* 2

Pada tampilan *intro game* 2 berisi mengenai kisah *game* secara singkat. Tampilan *intro game* 2 dapat dilihat pada gambar IV.5 :



Gambar IV.5 Tampilan *Intro Game* 2

IV.1.4.6 Tampilan *Intro Game* 3

Pada tampilan *intro game* 3 berisi mengenai kisah *game* secara singkat. Tampilan *intro game* 3 dapat dilihat pada gambar IV.6 :



Gambar IV.5 Tampilan *Intro Game* 3

IV.1.4.6 Tampilan *Intro Game* 4

Pada tampilan *intro game* 4 berisi mengenai kisah *game* secara singkat. Tampilan *intro game* 4 dapat dilihat pada gambar IV.7 :



Gambar IV.7 Tampilan *Intro Game* 4

IV.1.4.6 Tampilan *Level* 1 *Game*

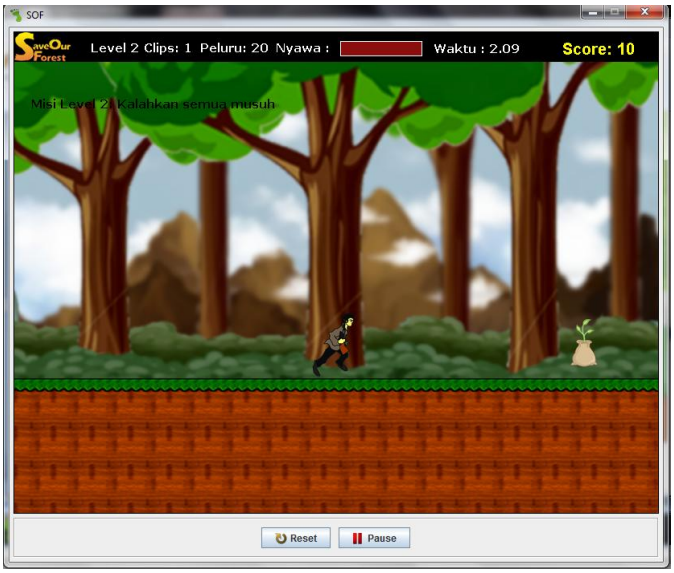
Pada tampilan *level* 1 *game* ini berisi mengenai tempat di *level* 1. Tampilan level 1 dapat dilihat pada gambar IV.8



Gambar IV.8 Tampilan *Level* 1 *Game*

IV.1.4.7 Tampilan *Level* 2 *Game*

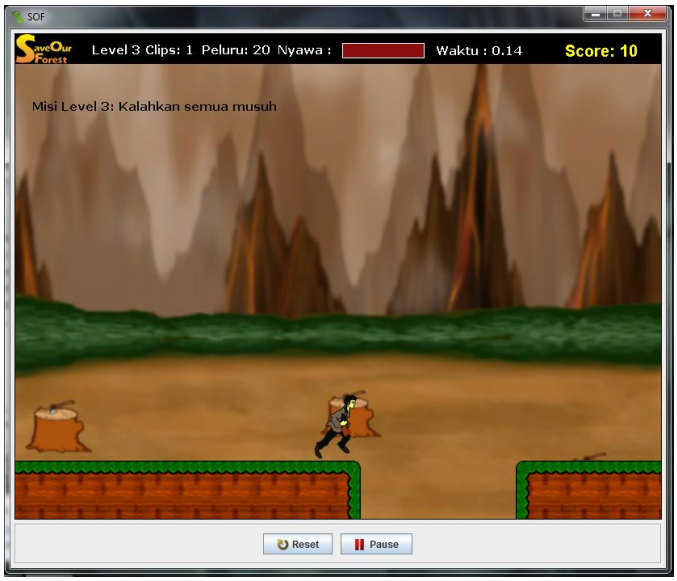
Pada tampilan *level* 2 *game* ini berisi mengenai tempat di level 2. Tampilan *level* 2 dapat dilihat pada gambar IV.9



Gambar IV.9 Tampilan *Level* 2 *Game*

IV.1.4.8 Tampilan *Level* 3 *Game*

Pada tampilan *level* 3 *game* ini berisi mengenai tempat di level 3. Tampilan level 3 dapat dilihat pada gambar IV.10



Gambar IV.10 Tampilan Level 3 *Game*

IV.2 Pengujian Sistem

Dalam melakukan pengujian terhadap aplikasi *game Save Our Forest* dilakukan dengan dua macam teknik, yaitu teknik pengujian *black box* dan teknik 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 melalui pengujian beta.

IV.2.1 Skenario Pengujian Aplikasi

Skenario pengujian menjelaskan pengujian terhadap sistem yang ada pada aplikasi permainan *Save Our Forest*. Skenario pengujian dari aplikasi permainan *Save Our Forest* dapat dilihat pada tabel IV.3.

Tabel IV.3 Skenario Pengujian Aplikasi

No	Komponen yang di uji	Skenario Pengujian	Jenis Pengujian
1	Menu Mulai	Pilih menu mulai	*Black Box*
		Pilih tombol lanjut *intro* 1	*Black Box*
		Pilih tombol lanjut *intro* 2	*Black Box*
		Permainan *level* 1	*Black Box*
		Pilih tombol lanjut pesan level selesai	*Black Box*
		Permainan *level* 2	*Black Box*
		Pilih tombol lanjut pesan level selesai	*Black Box*
		Permainan *level* 3	*Black Box*
		Ending game	*Black Box*
2	Menu Petunjuk	Pilih menu petunjuk	*Black Box*
		Pilih tombol kembali	*Black Box*
4	Menu Kredit	Pilih menu kredit	*Black Box*
		Pilih tombol kembali	*Black Box*
5	Karakter utama	Menekan tombol arah kanan	*Black Box*
		Menekan tombol arah kiri	*Black Box*
		Menekan tombol arah atas	*Black Box*
		Menekan tombol “F”	*Black Box*
		Menekan tombol “R”	*Black Box*
6	Algoritma A* (*A Star*)	Algoritma A* pada *boss*	*White Box*

IV.2.2 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 Mulai Permainan

   Pengujian mulai permainan dengan kasus dan hasil uji, kemudian diamati dan menghasilkan kesimpulan dapat dilihat pada tabel IV.4 :

Tabel IV.4 Pengujian Menu Mulai

Kasus dan Hasil Uji(Data benar)			
Masukan	Keluaran yang diharapkan	Pengamatan	Kesimpulan
Pilih mulai	Menampilkan *intro* 1	Dapat menampilkan *intro* 1	[√] Berhasil [ ] Tidak
Pilih lanjut *intro* 1	Menampilkan *intro* 2	Dapat menampilkan *intro* 2	[√] Berhasil [ ] Tidak
Pilih lanjut *intro* 2	Menampilkan *intro* 3	Dapat menampilkan *intro* 3	[√] Berhasil [ ] Tidak
Pilih lanjut *intro* 3	Menampilkan *intro* 4	Dapat menampilkan *intro* 4	[√] Berhasil [ ] Tidak
Pilih lanjut *intro* 4	Menampilkan *level* 1	Dapat menampilkan *level 1*	[√] Berhasil [ ] Tidak
Menyelesaikan *level* 1	Menampilkan tombol *level* selanjutnya	Dapat menampilkan tombol *level selanjutnya*	[√] Berhasil [ ] Tidak
Pilih *level selanjutnya*	Menampilkan *level* 2	Dapat menampilkan *level* 2	[√] Berhasil [ ] Tidak
Menyelesaikan *level* 2	Menampilkan tombol *level* selanjutnya	Dapat menampilkan tombol *level selanjutnya*	[√] Berhasil [ ] Tidak
Pilih *level selanjutnya*	Menampilkan *level* 3	Dapat menampilkan *level* 3	[√] Berhasil [ ] Tidak
Menyelesaikan *level* 3	Menampilkan *tombol selanjutnya*	Dapat menampilkan *tombol selanjutnya*	[√] Berhasil [ ] Tidak
Pilih *level selanjutnya*	Menampilkan *level boss*	Dapat menampilkan *level boss*	[√] Berhasil [ ] Tidak
Menyelesaikan *level* 3	Menampilkan *ending game*	Dapat menampilkan *ending game*	[√] Berhasil [ ] Tidak

2. Pengujian Menu Petunjuk

   Pengujian menu petunjuk dengan kasus dan hasil uji, kemudian diamati dan menghasilkan kesimpulan dapat dilihat pada tabel IV.5 :

Tabel IV.5 Pengujian Menu Petunjuk

Kasus dan Hasil Uji(Data benar)			
Masukan	Keluaran yang diharapkan	Pengamatan	Kesimpulan
Pilih petunjuk	Menampilkan informasi panduan bermain *game*	Dapat menampilkan informasi panduan bermain *game*	[√] Berhasil [ ] Tidak

Klik kembali	Menampilkan menu utama	Dapat menampilkan menu utama	[√] Berhasil [ ] Tidak

3. Pengujian Menu Kredit

   Pengujian menu kredit dengan kasus dan hasil uji, kemudian diamati dan menghasilkan kesimpulan dapat dilihat pada tabel IV.6 :

Tabel IV.6 Pengujian Menu Kredit

Kasus dan Hasil Uji(Data benar)			
Masukan	Keluaran yang diharapkan	Pengamatan	Kesimpulan
Pilih kredit	Menampilkan informasi pembuat *game*	Dapat menampilkan informasi pembuat *game*	[√] Berhasil [ ] Tidak
Klik kembali	Menampilkan menu utama	Dapat menampilkan menu utama	[√] Berhasil [ ] Tidak

4. Pengujian Karakter Utama

Tabel IV.7 Pengujian Karakter Utama

Hasil uji (data normal)			
Masukan	Keluaran yang diharapkan	Pengamatan	Kesimpulan
Tombol ←	Menggerakan karakter ke kiri.	Karakter utama bergerak ke kiri.	[√] Berhasil [ ] Tidak
Tombol →	Menggerakan karakter ke kanan.	Karakter utama bergerak ke kanan.	[√] Berhasil [ ] Tidak
Tombol ↑	Menggerakan karakter untuk meloncat	Karakter utama bergerak.	[√] Berhasil [ ] Tidak
Tombol F	Mengeluarkan tembakan.	Karakter mengeluarkan tembakan	[√] Berhasil [ ] Tidak
Tombol R	Karakter mengisi ulang peluru jika tersedia	Karakter mengisi ulang peluru jika tersedia	[√] Berhasil [ ] Tidak

IV.2.3 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 *graph matrix* yang digunakan untuk menggenerasi *flow graph*. Adapun pengujian *white-box* adalah sebagai berikut :

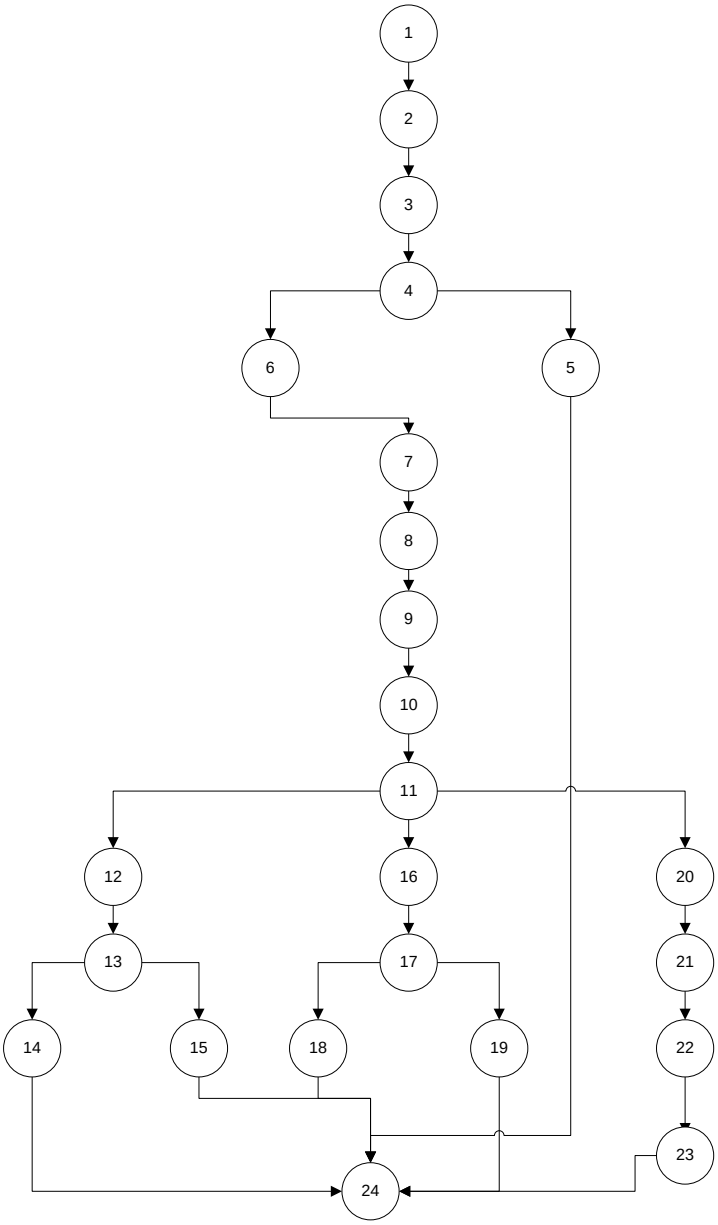
a. Algoritma

Tabel IV.8 Pengujian *White Box*

```
1   Public LinkedList openList = new LinkedList();
    Public LinkedList closeList = new LinkedList();
    openList.clear();
    closeList.clear();
2   while (!atGoal()){
       while (iter.hasNext())
       {
3         lastCheckedNode = iter.next();
          lastCheckedF = lastCheckedNode.F;
       }
4      if(currentCheckNode==world.getGoalNode()){
5         return true;}
       else
6         openList.remove(currentCheckNode);
7         closeList.add(currentCheckNode);
8         if(!closeList.contains(currentCheckNode)){
9         double currentGvalue = currentCheckNode.actualPath;
          List<Node> neighbours =
          currentCheckNode.returnNeighbours();
10        for (int j=0; j<neighbours.size(); j++)
          {
11        if
          (closeList.contains(neighbours.get(j))&&currentGvalue<neigh
          bours.get(j).actualPath)
          {
12             neighbours.get(j).examinedNode++;
13             If(currentGvalue<neighbours.get(j).actualPath)
                    {
14                  neighbours.get(j).changeG(currentGvalue+1);
                    }
15                  neighbours.get(j).parent = currentCheckNode;
               }else if
          (openList.contains(neighbours.get(j))&&currentGvalue<neighb
          ours.get(j).actualPath)
                    {
16                  neighbours.get(j).examinedNode++;
17                  if(currentGvalue<neighbours.get(j).actualPath){
18                     neighbours.get(j).changeG(currentGvalue+1);
                       }
19                  neighbours.get(j).parent = currentCheckNode;
          }else if
          (!(openList.contains(neighbours.get(j)))&&!(closeList.conta
          ins(neighbours.get(j))))
                    {
20                  neighbours.get(j).examinedNode++;
21                  openList.add(neighbours.get(j));
22                  if(currentGvalue<neighbours.get(j).actualPath){
23                     neighbours.get(j).changeG(currentGvalue+1);
                       }
           }
24      }
```

b. *Flow Graph*

Untuk menyederhanakan pseudocode dari tabel IV.8 dilkakukan konversi ke dalam *flow chart*, dapat dilihat pada gambar IV.11 berikut:



Gambar IV.11 *Flow Graph* Algoritma A*

Keterangan :

= Menggambarkan kondisi

= Menggambarkan aksi

c. *Cyclomatic Complexity*

Berdasarkan gambar IV.11 dapat dihitung *cyclomatic complexity* sebagai berikut:

V (G) = E – N + 1

V (G) = 28-24 + 1

V (G) = 4 + 1

V(G) = 5

Dimana :

E = jumlah busur pada *flow graph*

N = jumlah simpul pada *flow graph*

d. *Independent Path*

Path 1 : 1,2,3,4,5,24

Path 2 : 1,2,3,4,6,7,8,9,10,11,12,13,14,24

Path 3 : 1,2,3,4,6,7,8,9,10,11,12,13,15,24

Path 4 : 1,2,3,4,6,7,8,9,10,11,16,17,18,24

Path 5 : 1,2,3,4,6,7,8,9,10,11,16,17,19,24

Path 6 : 1,2,3,4,6,7,8,9,10,11,20,21,22,23,24

e. *Graph Matrix*

Tabel IV.9 Graph *Matrix*

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	E
1		1																							
2			1																						
3				1																					
4					1	1																			1
5																								1	
6							1																		
7								1																	
8									1																
9										1															
10											1														
11												1								1					1
12													1												
13														1	1										1
14																								1	
15																								1	
16																	1								
17																		1	1						1
18																								1	
19																								1	
20																					1				
21																						1			
22																							1		
23																								1	
24																									
Jumlah E + 1																									5

V(G) = Jumlah Graph Matrix + 1

V(G) = 4 + 1

V(G) = 5

Berdasarkan hasil pengujian dihasilkan nilai *Cyclomatic Complexity* yang sama yaitu 5, maka dapat disimpulkan bahwa pada proses pencarian dengan algoritma A* (A *star*) berjalan dengan baik, karena setiap pengujian menghasilkan nilai yang sama.

IV.2.4 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 dan menggunakan *pre test* dan *post test* untuk mengetahui perubahan pengetahuan dari pengguna aplikasi.

IV.2.5 Kuesioner

Kuesioner diberikan dengan teknik *Simple Random Sampling* dimana anggota *sample* dari populasi dilakukan secara acak tanpa memperhatikan strata yang ada dalam populasi kepada beberapa calon pengguna perangkat lunak yang dibangun dengan usia responden 13 tahun keatas. Sampel yang diambil masing-masing calon pengguna sebanyak 30 orang, dari hasil kuesioner tersebut akan dilakukan perhitungan agar dapat diambil kesimpulan penilaian dari para responden.

Berikut ini daftar pertanyaan berupa pertanyaan wawancara dan kuesioner pada calon pengguna dalam perangkat lunak yang dibangun.

1. Apakah aplikasi *game* ini menarik untuk dimainkan?
2. Apakah aplikasi *game* ini mempunyai tingkat kesulitan yang berbeda di setiap *level*?
3. Apakah perilaku musuh dalam *game* yang dapat mengejar pemain pada aplikasi *game* ini menambah tantangan ketika dimainkan?
4. Apakah aplikasi *game* ini menambah pengetahuan anda tentang pentingnya menjaga kelestarian lingkungan?
5. Apakah aplikasi *game* ini dapat membantu anda belajar tentang pentingnya menjaga kelestarian lingkungan hidup?
6. Apakah aplikasi *game* ini dapat menyampaikan informasi bahaya tidak menjaga kelestarian lingkungan hidup secara efektif?

Hasil presentase masing-masing nilai jawaban kuesioner yang diujikan kepada tiap responden dengan diberikan 5 skala menggunakan skala *likert*. Sistem skor yang diberikan untuk jawaban kuesioner dapat dilihat di tabel IV.10 :

Tabel IV.10 Skor Jawaban Pada Kuesioner

Skala Jawaban	Keterangan	Skor
SS	Sangat Setuju	5
ST	Setuju	4
BS	Biasa Saja	3
KS	Kurang Setuju	2
TS	Tidak Setuju	1

Berikut kriteria interpretasi dari skor :

1. Nilai persentase 0% – 20% = Tidak setuju terhadap kuesioner
2. Nilai persentase 21% – 40% = Kurang setuju terhadap kuesioner
3. Nilai persentase 41% – 60% = Biasa Saja terhadap kuesioner
4. Nilai persentase 61% – 80% = Setuju terhadap kuesioner
5. Nilai persentase 81% – 100% = Sangat setuju terhadap kuesioner.

IV.2.6 Hasil dan Evaluasi Pengujian Kuesioner

Hasil persentase kuesioner masing-masing jawaban yang sudah dihitung dengan menggunakan rumus II.2, kuesioner ini ditujukan kepada 30 orang responden yang berusia diatas 13 tahun.

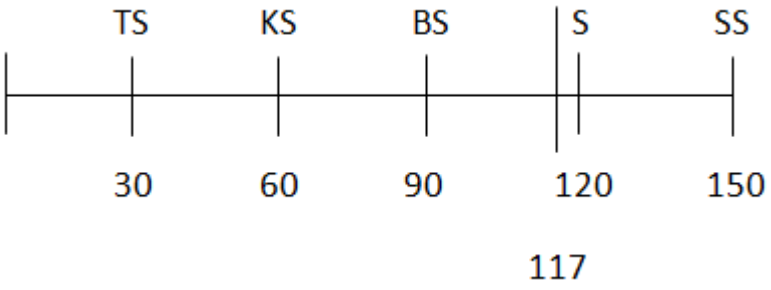
Hasil perhitungan kuesioner yang telah diujikan kepada 30 responden dapat dilihat pada tabel IV.11 sampai dengan tabel IV.16

1. Apakah aplikasi *game* ini menarik untuk dimainkan?

Tabel IV.11 Hasil Pengujian Kuesioner Soal Nomor 1

Kategori Jawaban	Skor	Frekuensi Jawaban Responden	Jumlah Skor	Nilai Presentase (%)
A.Sangat setuju	5	5	25	117/(30x5)x100%=78%
B.Setuju	4	17	68	
C.Biasa Saja	3	8	24	
D.Kurang setuju	2	0	0	
E.Tidak setuju	1	0	0	
Jumlah		30	117	

Hasil nilai persentase pada tabel 4.10 adalah 78% atau 117 dari total 150. Maka dapat disimpulkan bahwa aplikasi *game Save Our Forest* ini menarik untuk dimainkan. Secara kontinum dapat digambarkan seperti pada gambar 4.12



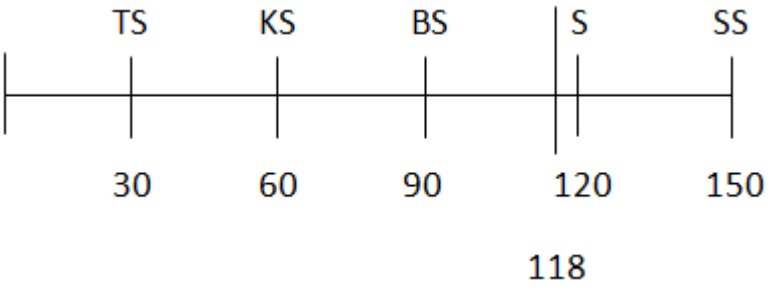
Gambar IV.12 Hasil Pengujian Kuesioner Soal Nomor 1

2. Apakah aplikasi *game* mempunyai tingkat kesulitan yang berbeda di setiap *level*?

Tabel IV.12 Hasil Pengujian Kuesioner Soal Nomor 2

Kategori Jawaban	Skor	Frekuensi Jawaban Responden	Jumlah Skor	Nilai Presentase (%)
A.Sangat setuju	5	3	15	118/(30x5)x100%=78,67 %
B.Setuju	4	22	88	
C.Biasa Saja	3	5	15	
D.Kurang setuju	2	0	0	
E.Tidak setuju	1	0	0	
Jumlah		30	118	

Hasil nilai persentase pada tabel IV.11 adalah 78,67% atau 118 dari total 150. Maka dapat disimpulkan aplikasi *game Save Our Forest* ini memiliki tingkat kesulitan yang berbeda-beda. Secara kontinum dapat digambarkan seperti pada gambar IV.13



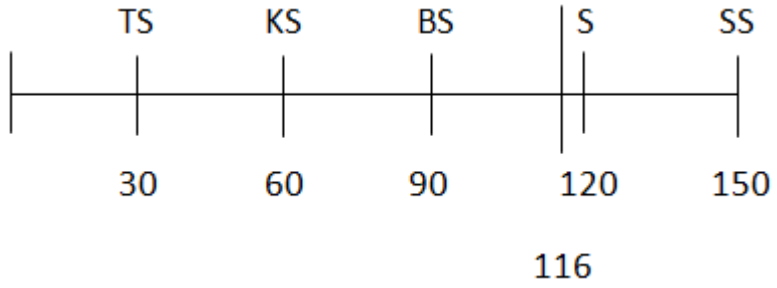
Gambar IV.13 Hasil Pengujian Kuesioner Soal Nomor 2

3. Apakah perilaku musuh dalam *game* yang dapat mengejar pemain pada aplikasi *game* ini menambah tantangan ketika dimainkan?

Tabel IV.13 Hasil Pengujian Kuesioner Soal Nomor 3

Kategori Jawaban	Skor	Frekuensi Jawaban Responden	Jumlah Skor	Nilai Presentase (%)
A.Sangat setuju	5	5	25	116/(30x5)x100%=77,34 %
B.Setuju	4	16	64	
C.Biasa Saja	3	9	27	
D.Kurang setuju	2	0	0	
E.Tidak setuju	1	0	0	
Jumlah		30	116	

Hasil nilai persentase pada tabel IV.12 adalah 77,34% atau 116 dari total 150. Maka dapat disimpulkan bahwa kemampuan musuh pada aplikasi *game Save Our Forest* ini memberikan tantangan yang lebih ketika dimainkan. Secara kontinum dapat digambarkan seperti pada gambar IV.14



Gambar IV.14 Hasil Pengujian Kuesioner Soal Nomor 3

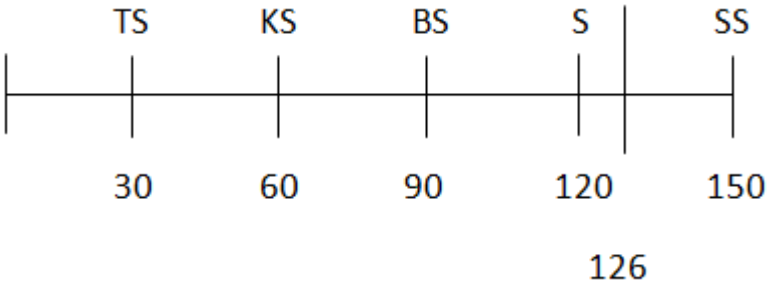
4. Apakah aplikasi *game* ini menambah pengetahuan anda tentang pentingnya menjaga kelestarian lingkungan?

Tabel IV.14 Hasil Pengujian Kuesioner Soal Nomor 4

Kategori Jawaban	Skor	Frekuensi Jawaban Responden	Jumlah Skor	Nilai Presentase (%)
A.Sangat setuju	5	6	30	126/(30x5)x100=84%
B.Setuju	4	24	96	
C.Biasa Saja	3	0	6	
D.Kurang setuju	2	0	0	
E.Tidak setuju	1	0	0	
Jumlah		30	126	

Hasil nilai persentase pada tabel IV.13 adalah 84% atau 126 dari total 150. Maka dapat disimpulkan bahwa aplikasi *game Save Our Forest* ini menambah

pengetahuan *user* tentang menjaga kelestarian lingkungan. Secara kontinum dapat digambarkan seperti pada gambar IV.14.5



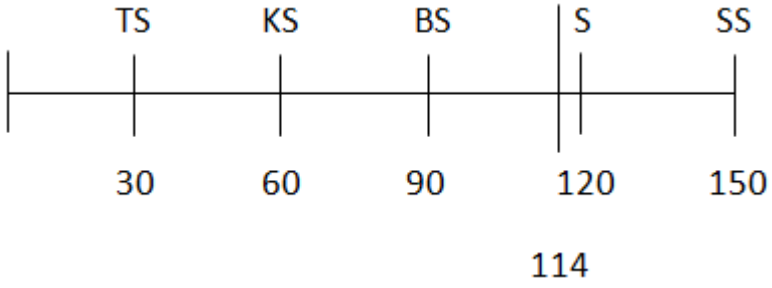
Gambar IV.15 Hasil Pengujian Kuesioner Soal Nomor 4

5. Apakah aplikasi *game* ini dapat membantu anda belajar tentang pentingnya menjaga kelestarian lingkungan hidup ?

Tabel IV.15 Hasil Pengujian Kuesioner Soal Nomor 5

Kategori Jawaban	Skor	Frekuensi Jawaban Responden	Jumlah Skor	Nilai Presentase (%)
A.Sangat setuju	5	2	10	114/(30x5)x100%=76%
B.Setuju	4	20	80	
C.Biasa Saja	3	8	24	
D.Kurang setuju	2	0	0	
E.Tidak setuju	1	0	0	
Jumlah		30	114	

Hasil nilai persentase pada tabel IV.14 adalah 76% atau 114 dari total 150. Maka dapat disimpulkan bahwa aplikasi *game Save Our Forest* ini membantu *user* belajar tentang pentingnya menjaga kelestarian lingkungan. Secara kontinum dapat digambarkan seperti pada gambar IV.16



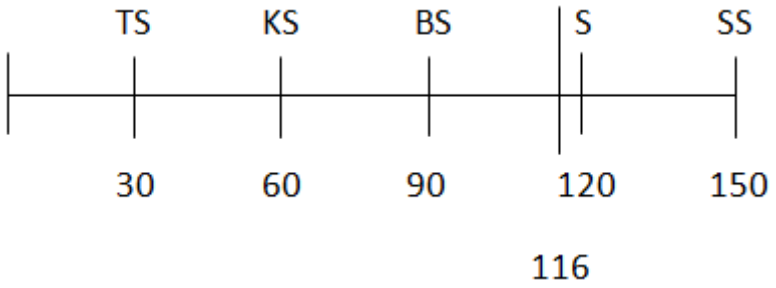
Gambar IV.16 Hasil Pengujian Kuesioner Soal Nomor 5

6. Apakah aplikasi *game* ini dapat menyampaikan informasi bahaya tidak menjaga kelestarian lingkungan hidup secara efektif?

Tabel IV.16 Hasil Pengujian Kuesioner Soal Nomor 6

Kategori Jawaban	Skor	Frekuensi Jawaban Responden	Jumlah Skor	Nilai Presentase (%)
A.Sangat setuju	5	2	10	116/(30x5)x100%=77,34%
B.Setuju	4	22	88	
C.Biasa Saja	3	6	18	
D.Kurang setuju	2	0	0	
E.Tidak setuju	1	0	0	
Jumlah		30	116	

Hasil nilai persentase pada tabel 4.15 adalah 77,34%. Maka dapat disimpulkan bahwa aplikasi *game Save Our Forest* ini menyampaikan informasi bahaya tidak menjaga kelestarian lingkungan hidup secara efektif Secara kontinum dapat digambarkan seperti pada gambar IV.17



Gambar IV.17 Hasil Pengujian Kuesioner Soal Nomor 6

IV.2.7 Kesimpulan Kuesioner

Berdasarkan hasil kuesioner, maka dapat diambil beberapa kesimpulan yaitu :

1. Aplikasi *game Save Our Forest* ini menarik untuk di mainkan.
2. Aplikasi *game Save Our Forest* ini memiliki tingkat kesulitan yang berbeda di setiap *level* nya.
3. Musuh pada aplikasi *game Save Our Forest* yang mempunyai kemampuan untuk mengejar pemain menambah tantangan ketika dimainkan
4. Aplikasi *game Save Our Forest* ini menambah pengetahuan anda tentang pentingnya menjaga kelestarian lingkungan.

5. Aplikasi *game Save Our Forest* ini dapat membantu *user* untuk belajar tentang kelestarian lingkungan hidup.
6. Aplikasi *game Save Our Forest* ini dapat menyampaikan informasi tentang kelestarian lingkungan hidup secara efektif.

IV.2.8 Hipotesis

Pre Test dan *Post Test* diberikan kepada 30 responden dengan usia 13 tahun ke atas, menggunakan metode pengujian Kolmogorov-Smirnov dan pengujian statistik non parametrik Wilcoxon. Hasil dari *pre test* dan *post test* ini akan membuktikan hipotesis adanya perubahan pengetahuan responden setelah memainkan *game Save Our Forest*. Soal terdiri dari 5 pertanyaan, setiap nomor memiliki bobot 20, nilai maksimal keseluruhan adalah 100. Uji normalisasi dilakukan untuk mengetahui data tersebut normal atau tidak. Tabel IV.17 merupakan hasil pengujian *pretest* dan *posttest.*

IV.2.9 Hasil dan Evaluasi *Pre Test* dan *Post Test*

Berikut ini adalah nilai responden sebelum dan sesudah bermain *game* dapat dilihat pada tabel IV.17

Tabel IV.17 Hasil Pengujian *Pretest* dan *Posttest*

N	Sebelum Bermain *Game*	Sesudah Bermain *Game*
1	60	100
2	80	100
3	60	80
4	80	100
5	60	80
6	60	100
7	60	80
8	60	100
9	100	100
10	80	80
11	60	100
12	80	100
13	60	100

14	100	100
15	100	100
16	80	80
17	80	100
18	80	80
19	80	80
20	60	80
21	60	80
22	100	100
23	80	80
24	80	80
25	80	100
26	60	80
27	80	100
28	80	80
29	100	100
30	60	100

Hasil tersebut perlu diuji untuk mengetahui data tersebut normal atau tidak. berikut ini adalah hasil data uji normalisasi menggunakan metode Kolmogorov-Smirnov dapat dilihat pada tabel IV.18

Tabel IV.18 Hasil *Case Processing Summary*

Case Processing Summary

	Cases					
	Valid		Missing		Total	
	N	Percent	N	Percent	N	Percent
Sebelum	30	100.0%	0	.0%	30	100.0%
Sesudah	30	100.0%	0	.0%	30	100.0%

Tabel IV.19 ***Output Descriptives***

Descriptives

			Statistic	Std. Error
Sebelum	Mean		75.33	2.658
	95% Confidence Interval for Mean	Lower Bound	69.90	
		Upper Bound	80.77	
	5% Trimmed Mean		74.81	
	Median		80.00	
	Variance		211.954	
	Std. Deviation		14.559	
	Minimum		60	
	Maximum		100	
	Range		40	
	Interquartile Range		20	
	Skewness		.396	.427
	Kurtosis		-.957	.833
Sesudah	Mean		92.00	1.819
	95% Confidence Interval for Mean	Lower Bound	88.28	
		Upper Bound	95.72	
	5% Trimmed Mean		92.22	
	Median		100.00	
	Variance		99.310	
	Std. Deviation		9.965	
	Minimum		80	
	Maximum		100	
	Range		20	
	Interquartile Range		20	
	Skewness		-.430	.427
	Kurtosis		-1.950	.833

Tabel IV.20 ***Test of Normality***

Tests of Normality

	Kolmogorov-Smirnov[a]			Shapiro-Wilk		
	Statistic	df	Sig.	Statistic	df	Sig.
Sebelum	.254	30	.000	.793	30	.000
Sesudah	.389	30	.000	.624	30	.000

a. Lilliefors Significance Correction

Berdasarkan tabel IV.20 hasil *pretest* dan *posttest* belum normal, terlihat dari nilai signifikansi atau nilai probabilitas. Data disebut dinyatakan tidak *normal* jika nilai signifikan < 0,05 dan sebaliknya jika nilai signifikan > 0,05 maka data tersebut dikatakan *normal*.

Berdasarkan data diatas, alat uji kenormalan distribusi data yang digunakan adalah *Kolmogorov Smirnov* dengan keterangan sama dengan uji *Liliefors*. Nilai Sig. untuk variabel adalah sebagai berikut:

Sebelum = 0.000

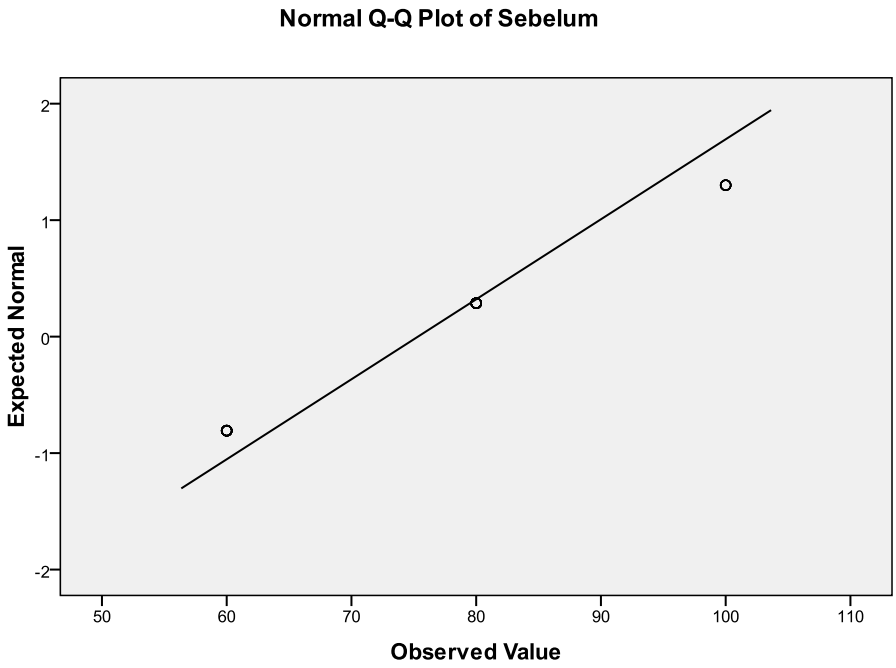
Sesudah = 0.000

Keterangan :

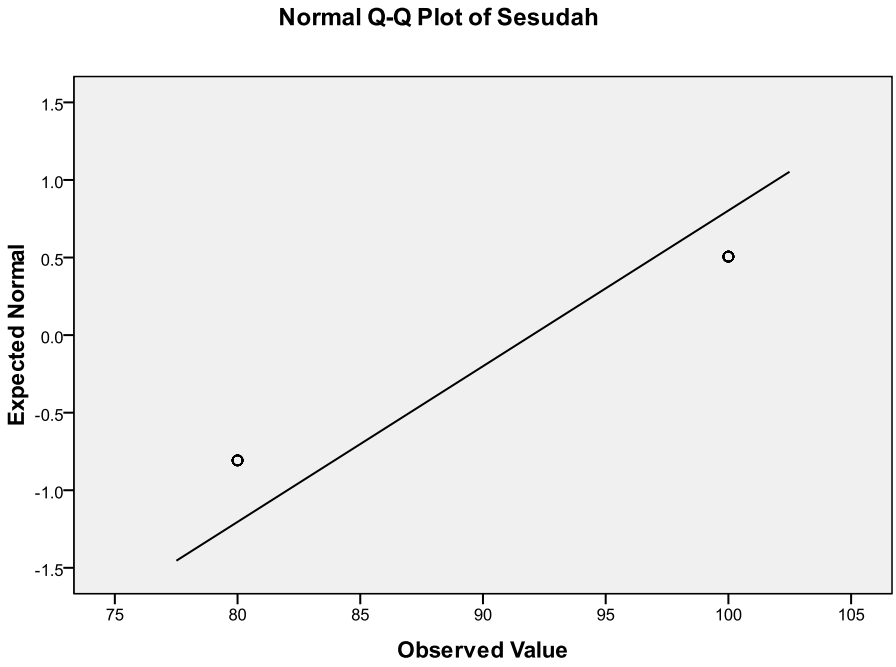
Sebelum = Kemampuan responden sebelum bermain *game*.

Sesudah = Kemampuan responden setelah bermain *game*.

Nilai *sig* yang didapat dari data diatas memiliki nilai < 0,05, maka data tersebut tidak normal atau tidak simetri. Jika digambarkan pada grafik sebaran data tersebut tidak membentuk garis lurus, terlihat pada gambar IV.18 dan gambar IV.19 dibawah ini :



Gambar IV.18 Distribusi Data Normal Sebelum Bermain *Game*



Gambar IV.19 Distribusi Data Normal Sesudah Bermain *Game*

Dikarenakan data tersebut tidak normal, maka selanjutnya pengujian dilakukan dengan menggunakan statistik non-parametrik, jenis hipotesa untuk pengujian ini adalah hipotesa komparatif dengan skala data ordinal, dan memiliki sampel yang relatif. Berdasarkan kriteria tersebut, maka pengujian selanjutnya akan menggunakan metode Wilcoxon. Sebelum melakukan pengujian data nilai dengan menggunakan metode Wilcoxon, terlebih dahulu mendefinisikan hipotesis untuk mengetahui hasil pengujian diterima atau ditolak. Berikut ini adalah pengujian hipotesisnya:

a. H_0 : Tidak terdapat perubahan pengetahuan setelah bermain *game*.
b. H_1 : Terdapat perubahan pengetahuan setelah bermain *game*.

Berikut ini adalah hasil pengujian metode Mann-Whitney dapat dilihat pada tabel IV.21 – IV.23

Tabel IV.21 *Descriptive Statistics*

Descriptive Statistics

	N	Minimum	Maximum	Mean	Std. Deviation
Sebelum	30	60	100	75.33	14.559
Sesudah	30	80	100	92.00	9.965

Keterangan:

a. Mean adalah nilai rata-rata terukur suatu data.
b. Minimum adalah hasil terkecil dari nilai suatu data.
c. Maximum adalah hasil terbesar dari nilai suatu data.
d. Std. Deviation adalah rata-rata jarak penyimpangan titik-titik data diukur dari nilai rata-rata data tersebut

Tabel IV.22 *Ranks*

Ranks

		N	Mean Rank	Sum of Ranks
Posttest - Pretest	Negative Ranks	**0**[a]	**.00**	**.00**
	Positive Ranks	**18**[b]	**9.50**	**171.00**
	Ties	**12**[c]		
	Total	**30**		

a. Posttest < Pretest

b. Posttest > Pretest

c. Posttest = Pretest

Berdasarkan hasil diatas, maka dapat diperoleh:

a. *Negative Ranks* atau selisih antara variabel sebelum dan sesudah yang negatif. Dalam kasus ini tidak ada variabel yang negatif
b. *Positive Ranks* atau selisih antara variabel sebelum dan sesudah yang positif. Terdapat 18 observasi pada variabel sesudah yang lebih dari variabel sebelum dengan rata-rata 10,00 dan jumlah ranking positif sebanyak 190,00
c. *Ties* atau tidak adanya perbedaan antara variabel sebelum dan sesudah bermain *game*.

Berikut ini adalah hasil uji statistik menggunakan metode Mann-Whitney dapat dilihat pada tabel IV.23:

Tabel IV.23 *Test Statistics*

Test Statistics[a]

	Posttest - Pretest
Z	**-3.874**[b]
Asymp. Sig. (2-tailed)	**.000**
Exact Sig. (2-tailed)	**.000**
Exact Sig. (1-tailed)	**.000**
Point Probability	**.000**

a. Wilcoxon Signed Ranks Test

b. Based on negative ranks.

Kriteria Pengujian:

1. α = 0,005 = 5%
2. H_0 diterima dan H_1 ditolak apabila probabilitas sampel >= α.
3. H_0 ditolak dan H_1 diterima apabila probabilitas sampel < α.

Kesimpulan Pengujian:

Karena α > probabilitas sampel, maka H_0 ditolak dan H_1 diterima, berarti ada perbedaan nilai antara *pre test* dan *post test*. Hal ini menunjukkan bahwa ada peningkatan pengetahuan dimana hasil *post test* memiliki hasil yang lebih tinggi dilihat dari *mean* yang lebih tinggi.