BAB 4
IMPLEMENTASI DAN PENGUJIAN

4.1. Implementasi

Tahap implementasi merupakan tahap penciptaan perangkat lunak kedalam bahasa pemograman yang dimengerti oleh komputer, serta merupakan tahap kelanjutan dari kegiatan perancangan sistem.

4.1.1. Implementasi Perangkat Keras

Perangkat keras yang akan digunakan untuk menjalankan aplikasi *game Save Our Planet from Extinction* ini tidak harus komputer yang berspesifikasi tinggi. Adapun spesifikasi perangkat keras yang digunakan saat implementasi dan pengujian aplikasi *game Save Our Planet from Extinction* dapat dilihat pada tabel 4.1.

Tabel 4.1 Spesifikasi Perangkat Keras

Spesifikasi Perangkat Keras	
Processor	Intel Core i3 2.4Ghz
RAM	Kingston 6GB
Harddisk	Seagate 500GB
VGA	NVIDIA GFORCE 610M 2GB
Monitor	Resolusi 1366x768
Mouse	Optic 1 buah
Keyboard	1 buah
Speaker	1 pasang

4.1.2. Implementasi Perangkat Lunak

Spesifikasi perangkat lunak yang dibutuhkan untuk menjalankan aplikasi *game Save Our Planet from Extinction* yaitu sistem operasi windows XP SP2 dan directX9 untuk mendukung performa grafik untuk menjalankan aplikasi.

4.1.3. Implementasi Aplikasi

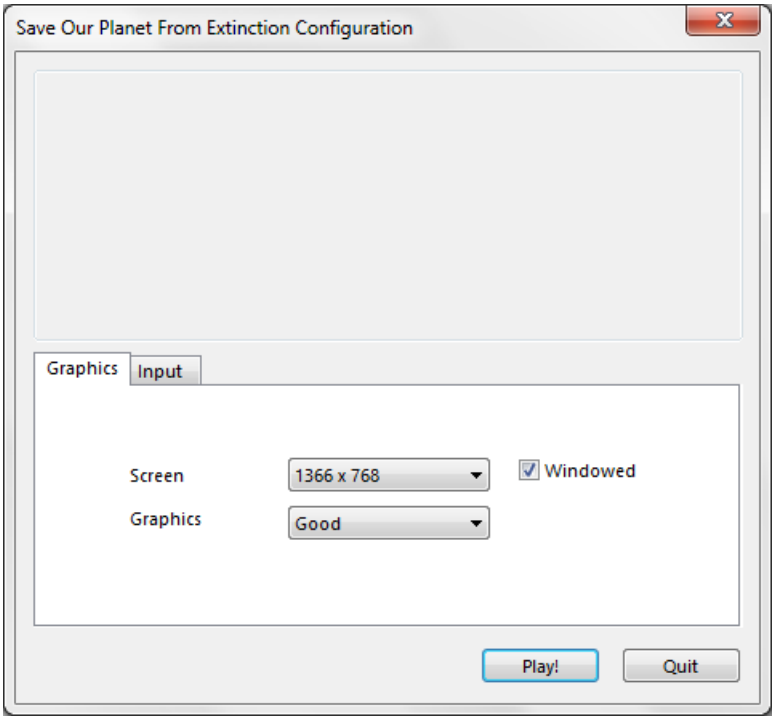
Aplikasi *game Save Our Planet from Extinction* merupakan sebuah *game* ber-genre TPS yang terdiri dari beberapa *Level* yang dimana pemain harus menyelesaikan misi di setiap *Level*nya untuk dapat melanjutkan ke *Level-Level* berikutnya. Aplikasi *game* ini tidak melakukan proses instalasi, cukup dengan melakukan *double*-klik pada aplikasi ini.

4.1.4. Implementasi Antarmuka

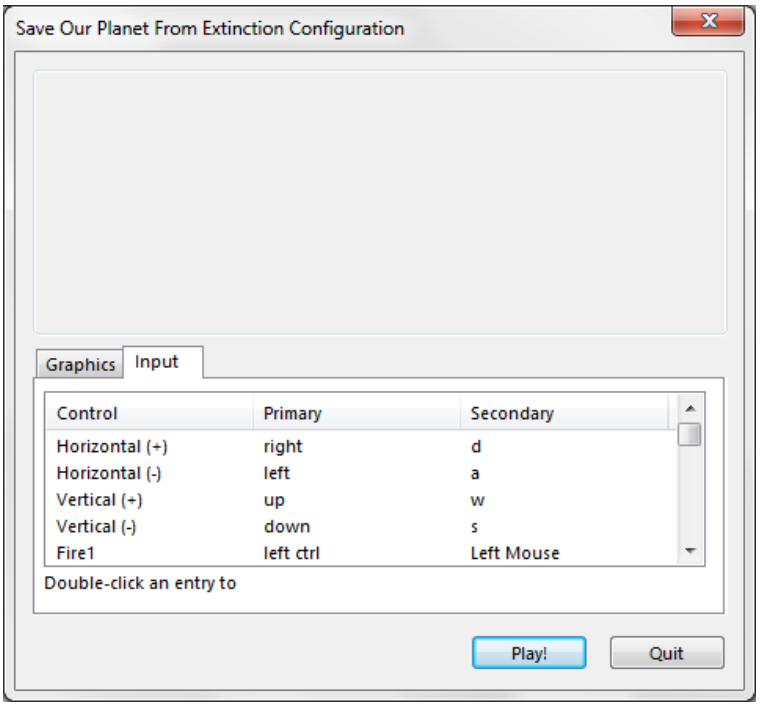
Adapun implementasi antarmuka pada aplikasi *game Save Our Planet from Extinction* adalah sebagai berikut :

4.1.4.1. Antarmuka Konfigurasi Default

Antarmuka konfigurasi default merupakan tampilan awal pada saat aplikasi dijalankan, tampilan tersebut merupakan tampilan *default* yang diberikan untuk semua *project* yang dibangun menggunakan Unity. Terdapat 2 menu untuk men-*setting graphic* seperti pada gambar 4.1 dan input dapat dilihat pada gambar 4.2 .



Gambar 4.1 Menu *Setting Graphic*



Gambar 4.2 Menu *Input Keyboard*

4.1.4.2. Antarmuka Menu Utama

Antarmuka menu utama merupakan tampilan menu utama aplikasi *game* setelah pengguna mengklik *play* pada tampilan konfigurasi *default*. Terdapat 4 menu dapat dilihat pada gambar 4.3.



Gambar 4.3 Tampilan Antarmuka Menu Utama

Pada tampilan menu utama jika pengguna mengklik menu pengaturan maka akan tampil seperti pada gambar 4.4.



Gambar 4.4 Tampilan Antarmuka Pengaturan Suara

Jika pengguna mengklik tab grafis maka akan tampil seperti pada gambar 4.5.



Gambar 4.5 Tampilan Antarmuka Pengaturan Grafis

Pada tampilan menu utama jika pengguna mengklik menu petunjuk maka akan tampil seperti pada gambar 4.6.



Gambar 4.6 Tampilan Antarmuka Petunjuk

4.1.4.3. Antarmuka Mulai Permainan

Antarmuka mulai permainan merupakan tampilan yang akan ditampilkan ketika pengguna mengklik tombol mulai pada tampilan menu utama, pengguna akan berada di *Level* 1 permainan yang berlatar tempat di pedesaan, dimana pengguna dapat mulai menggerakan karakter menggunakan *keyboard* dan *mouse* untuk mencari portal agar dapat menuju ke *level* berikutnya. berikut adalah tampilan *level 1* permainan yang dapat dilihat pada gambar 4.7.



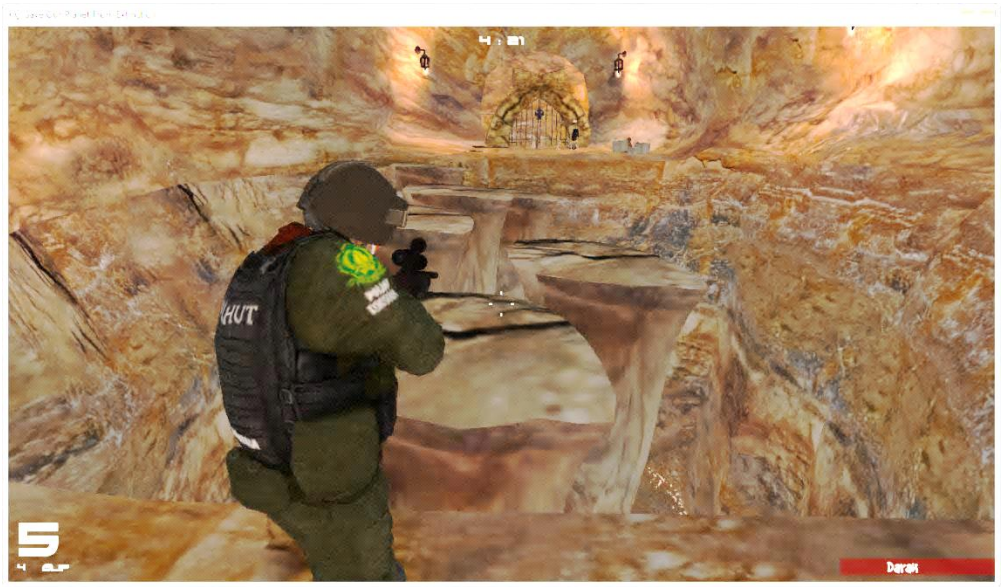
Gambar 4.7 Tampilan Antarmuka *Level* 1 Permainan

Setelah berhasil menyelesaikan *Level* 1 pengguna dapat melanjukan ke *Level* 2 dan dapat dilihat pada gambar 4.8.



Gambar 4.8 Tampilan Antarmuka *Level* 2 Permainan

Setelah berhasil menyelesaikan *Level* 2 pengguna dapat melanjukan ke *Level* 3 dan dapat dilihat pada gambar 4.9.



Gambar 4.9 Tampilan Antarmuka *Level* 3 Permainan

Setelah berhasil menyelesaikan *Level* 3 pengguna dapat melanjukan ke *Level* 4 dan dapat dilihat pada gambar 4.10.



Gambar 4.10 Tampilan Antarmuka *Level* 4 Permainan

Setelah berhasil menyelesaikan *Level* 4 pengguna dapat melanjukan ke *Level* 5 dan dapat dilihat pada gambar 4.11.



Gambar 4.11 Tampilan Antarmuka *Level* 5 Permainan

4.2. Pengujian

Pengujian dilakukan betujuan untuk menemukan kesalahan kekurangan pada perangkat lunak yang diuji. Pengujian tersebut dimaksudkan untuk mengetahui perangkat lunak yang dibuat sudah memenuhi kriteria yang sesuai dengan tujuan perancangan perangkat lunak.

Pengujian dilakukan dengan dua macam teknik yaitu pengujian *white box* dan Pengujian *black box*. Adapun pengujian *black box* lebih berfokus pada menemukan kesalahan program, sedangkan pengujian *white box* berfokus pada struktur internal (*source code*) program, dimana setiap baris kode yang akan di-*compile* dicek satu persatu untuk mengetahui apakah masih terjadi *error*. Cara pengujian dilakukan melalui dua tahap, yaitu tahap pengujian *alpha* dan tahap pengujian *beta.*

4.2.1. Pengujian *Alpha*

Pada tahap *alpha*, pengujian dilakukan untuk memastikan aplikasi *game Save Our Planet from Extinction* dapat berjalan sesuai dengan kebutuhan dan tujuan yang diharapkan. Pendekatan perancangan kasus uji menggunakan dua macam teknik yaitu *white box* dan *black box*.

4.2.1.1. Skenario Pengujian Aplikasi

Skenario pengujian menjelaskan pengujian terhadap sistem yang ada pada aplikasi *game Save Our Planet from Extintction*. Berikut ini merupakan skenario pengujian dari aplikasi *game* dapat dilihat pada tabel 4.2.

Tabel 4.2 Skenario Pengujian

No	Komponen yang diuji	Butir Uji	Jenis Pengujian
1	Menu Utama	Klik Menu Mulai Main	*Black Box*
		Klik Menu Pengaturan	*Black Box*
		Klik Menu Petunjuk	*Black Box*
		Klik Menu Keluar	*Black Box*
2	Menu Mulai	Klik Tombol Mulai	*Black Box*
		Menampilkan permainan	*Black Box*
3	Menu Pengaturan	Klik Menu Pengaturan	*Black Box*
		Klik Tab Suara	*Black Box*
		Klik Tab Grafik	*Black Box*
		Klik Tombol Kembali	*Black Box*
4	Menu Pengaturan Suara	Menggeser Tombol *Volume*	*Black Box*
		Klik Tombol Kembali	*Black Box*
5	Menu Pengaturan Grafis	Klik Tombol Resolusi Layar	*Black Box*

		Klik Tombol Mode Layar	*Black Box*
		Klik Tombol Turunkan	*Black Box*
		Klik Tombol Naikan	*Black Box*
		Klik Tombol Kembali	*Black Box*
6	Pause Menu	Tampil Pause Menu	*Black Box*
		Klik Tombol *Restart Level*	*Black Box*
		Klik Tombol Main Menu	*Black Box*
		Klik Tombol Pengaturan	*Black Box*
		Klik Tombol Petunjuk	*Black Box*
		Klik Tombol Keluar	*Black Box*
7	Kontrol Karakter Pemain	Karakter Bergerak Kedepan	*Black Box*
		Karakter Bergerak Kekanan	*Black Box*
		Karakter Bergerak Kekiri	*Black Box*
		Karakter Bergerak Kebelakang	*Black Box*
		Karakter Bergerak Melompat	*Black Box*

		Karakter Menyerang	*Black Box*
		Karakter Me-*Reload* Senjata	*Black Box*
		Karakter Menunduk	*Black Box*
8	Algoritma A*(A Star)	Implementasi Algoritma A* Pada Musuh Untuk Melakukan Pencarian Terhadap Karakter Pemain	*White Box*

4.2.1.2. Kasus dan Hasil Pengujian (*Black Box*)

Kasus dan hasil uji perangkat lunak yang sudah dibangun dengan menggunakan teknik *black box* berdasarkan pengamatan yang dilakukan dan menghasilkan kesimpulan dari pengujian tersebut adalah sebagai berikut:

1. Pengujian Menu Utama

Pengujian menu utama merupakan proses dimana pengguna menyentuh menu-menu yang ada pada tampilan menu utama. Adapun hasil dari proses pengujian dapat dilihat pada tabel 4.3.

Tabel 4.3 Pengujian Menu Utama

Kasus dan Hasil Uji			
Masukan	**Keluaran yang diharapkan**	**Pengamatan**	**Kesimpulan**
Klik tombol Mulai	Sistem melakukan proses untuk	Sistem menampilkan *level*	[√] Berhasil

	menampilkan *level* 1 permainan	1 permainan	[ ] Tidak Berhasil
Klik tombol Pengaturan	Sistem melakukan proses untuk menampilkan halaman pengaturan	Sistem menampilkan halaman pengaturan	[√] Berhasil [ ] Tidak Berhasil
Klik tombol Petunjuk	Sistem melakukan proses untuk menampilkan halam petunjuk	Sistem menampilkan halaman petunjuk	[√] Berhasil [ ] Tidak Berhasil
Klik tombol Keluar	Sistem melakukan proses untuk menutup aplikasi	Sistem menutup aplikasi	[√] Berhasil [ ] Tidak Berhasil

2. Pengujian Menu Mulai

Pengujian menu mulai merupakan proses setelah pengguna mengklik tombol mulai pada menu utama yang selanjutkan akan menampilkan *level* 1 permainan. Adapun hasil dari proses pengujian dapat dilihat pada tabel 4.4.

Tabel 4.4 Pengujian Menu Mulai Main

Kasus dan Hasil Uji			
Masukan	**Keluaran yang diharapkan**	**Pengamatan**	**Kesimpulan**
Klik Tombol Mulai	Sistem melakukan proses untuk menampilkan halaman *level* 1 permainan	Sistem menampilkan halaman *level* 1 permainan	[√] Berhasil [ ] Tidak Berhasil

3. Pengujian Menu Pengaturan

Pengujian menu pengaturan merupakan proses dimana pengguna dapat melihat dan menkofigurasi pengaturan yang tersedia pada *game*. Adapun hasil dari proses pengujian dapat dilihat pada tabel 4.5.

Tabel 4.5 Pengujian Menu Pengaturan

Kasus dan Hasil Uji			
Masukan	**Keluaran yang diharapkan**	**Pengamatan**	**Kesimpulan**
Klik tombol Pengaturan	Sistem melakukan proses untuk menampilkan menu pengaturan permainan	Sistem menampilkan menu pengaturan permainan	[√] Berhasil [ ] Tidak Berhasil
Klik tombol Suara	Sistem melakukan proses untuk menampilkan konfigurasi suara permainan	Sistem menampilkan konfigurasi suara permainan	[√] Berhasil [ ] Tidak Berhasil
Klik tombol Grafik	Sistem melakukan proses untuk menampilkan konfigurasi grafik permainan	Sistem menampilkan konfigurasi grafik permainan	[√] Berhasil [ ] Tidak Berhasil
Klik Tombol Kembali	Sistem melakukan proses untuk kembali menampilkan menu utama	Sistem menampilkan halaman menu utama	[√] Berhasil [ ] Tidak Berhasil

4. Pengujian Menu Pengaturan Suara

Tabel 4.6 Pengujian Menu Pengaturan Suara

Kasus dan Hasil Uji			
Masukan	**Keluaran yang diharapkan**	**Pengamatan**	**Kesimpulan**
Menggeser Tombol Volume ke Kiri	Sistem melakukan proses untuk memperkecil suara	Sistem mengecilkan suara permainan	[√] Berhasil [ ] Tidak Berhasil
Menggeser Tombol Volume ke Kanan	Sistem melakukan proses untuk memperbesar suara	Sistem memperbesar suara permainan	[√] Berhasil [ ] Tidak Berhasil
Klik Tombol Kembali	Sistem melakukan proses untuk kembali menampilkan menu utama	Sistem menampilkan halaman menu utama	[√] Berhasil [ ] Tidak Berhasil

5. Pengujian Menu Pengaturan Grafis

Tabel 4.7 Pengujian Menu Pengaturan Grafis

Kasus dan Hasil Uji			
Masukan	**Keluaran yang diharapkan**	**Pengamatan**	**Kesimpulan**
Klik dan pilih tombol resolusi layar	Sistem melakukan proses untuk merubah resolusi layar	Sistem merubah resolusi layar	[√] Berhasil [ ] Tidak Berhasil
Klik tombol ganti mode layar	Sistem melakukan proses untuk merubah mode	Sistem merubah mode layar	[√] Berhasil [ ] Tidak Berhasil

	layar		
Klik tombol Turunkan	Sistem melakukan proses untuk menurunkan kualitas gambar	Sistem menurunkan kualitas gambar	[√] Berhasil [ ] Tidak Berhasil
Klik tombol Naikan	Sistem melakukan proses untuk menaikan kualitas gambar	Sistem menaikan kualitas gambar	[√] Berhasil [ ] Tidak Berhasil
Klik Tombol Kembali	Sistem melakukan proses untuk kembali menampilkan menu utama	Sistem menampilkan halaman menu utama	[√] Berhasil [ ] Tidak Berhasil

6. Pengujian Menu Petunjuk

Pengujian menu petunjuk merupakan proses dimana pengguna dapat melihat petunjuk permainan. Adapun hasil dari proses pengujian dapat dilihat pada tabel 4.8.

Tabel 4.8 Pengujian Menu Petunjuk

Kasus dan Hasil Uji			
Masukan	**Keluaran yang diharapkan**	**Pengamatan**	**Kesimpulan**
Klik tombol Petunjuk	Sistem melakukan proses untuk menampilkan Petunjuk permainan	Sistem menampilkan petunjuk permainan	[√] Berhasil [ ] Tidak Berhasil
Klik Tombol Kembali	Sistem melakukan proses untuk	Sistem menampilkan	[√] Berhasil

	menampilkan kembali ke menu utama	menu utama	[ ] Tidak Berhasil

7. Pengujian *Pause Menu*

Pengujian *pause menu* merupakan proses dimana pengguna dapat menghentikan sementara jalannya permainan sewaktu-waktu atau ingin keluar ditengah-tengah jalannya permainan. Adapun hasil dari proses pengujian dapat dilihat pada tabel 4.9.

Tabel 4.9 Pengujian Pause Menu

Kasus dan Hasil Uji			
Masukan	**Keluaran yang diharapkan**	**Pengamatan**	**Kesimpulan**
Tekan Tombol Esc pada keyboard	Sistem melakukan proses untuk menampilkan halaman pause menu	Sistem menampilkan halaman pause menu	[√] Berhasil [ ] Tidak Berhasil
Klik Tombol Lanjutkan	Sistem melakukan proses untuk menampilkan *Level* 1/2/3/4 /5 permainan	Sistem menampilkan *Level* 1/2/3/4/5 permainan	[√] Berhasil [ ] Tidak Berhasil
Klik Tombol *Restart Level*	Sistem melakukan proses untuk mengulangi *level* yang dimainkan	Sistem menampilkan area permainan yang dimainkan	[√] Berhasil [ ] Tidak Berhasil
Klik Tombol Main Menu	Sistem melakukan proses untuk kembali ke menu	Sistem menampilkan menu utama	[√] Berhasil [ ] Tidak Berhasil

	utama		
Klik Tombol Pengaturan	Sistem melakukan proses untuk menampilkan menu pengaturan permainan	Sistem menampilkan menu pengaturan permainan	[√] Berhasil [ ] Tidak Berhasil
Klik Tombol Petunjuk	Sistem melakukan proses untuk menampilkan Petunjuk permainan	Sistem menampilkan petunjuk permainan	[√] Berhasil [ ] Tidak Berhasil
Tekan Tombol Keluar	Sistem melakukan proses untuk menampilkan kembali menu utama	Sistem menampilkan menu utama	[√] Berhasil [ ] Tidak Berhasil

8. Pengujian Kontrol Karakter Permainan

Pengujian kontrol karakter pemain merupakan proses yang dilakukan oleh pemain untuk menggerakkan karakter pemain pada permainan. Adapun hasil dari proses pengujian dapat dilihat pada tabel 4.10.

Tabel 4.10 Pengujian Kontrol Karakter Pemain

Kasus dan Hasil Uji			
Masukan	**Keluaran yang diharapkan**	**Pengamatan**	**Kesimpulan**
Tekan Tombol W	Karakter akan bergerak kedepan	Karakter pemain bergerak kedepan	[√] Berhasil [ ] Tidak Berhasil
Tekan Tombol D	Karakter pemain akan bergerak	Karakter pemain bergerak kekanan	[√] Berhasil

	kekanan		[ ] Tidak Berhasil
Tekan Tombol A	Karakter pemain akan bergerak kekiri	Karakter pemain bergerak kekiri	[√] Berhasil [ ] Tidak Berhasil
Tekan Tombol S	Karakter pemain akan bergerak kebelakang	Karakter pemain bergerak kebelakang	[√] Berhasil [ ] Tidak Berhasil
Tekan Tombol Space	Karakter pemain akan bergerak melompat	Karakter pemain melompat	[√] Berhasil [ ] Tidak Berhasil
Klik Kanan pada Mouse	Karakter pemain akan membidik	Karakter pemain membidik	[√] Berhasil [ ] Tidak Berhasil
Tekan Tombol V	Karakter pemain akan menunduk	Karakter pemain menunduk	[√] Berhasil [ ] Tidak Berhasil
Klik Kiri pada Mouse	Karakter pemain akan melakukan tembakan	Karakter pemain melakukan tembakan	[√] Berhasil [ ] Tidak Berhasil
Tekan Tombol R	Karakter pemain akan mengisi ulang peluru senjata	Karakter pemain melakukan isi ulang peluru senjata	[√] Berhasil [ ] Tidak Berhasil
Tekan Tombol 1	Senjata karakter pemain akan berganti	Senjata karakter pemain berganti	[√] Berhasil [ ] Tidak Berhasil

4.2.2. Kasus dan Hasil Pengujian (*White Box Testing)*

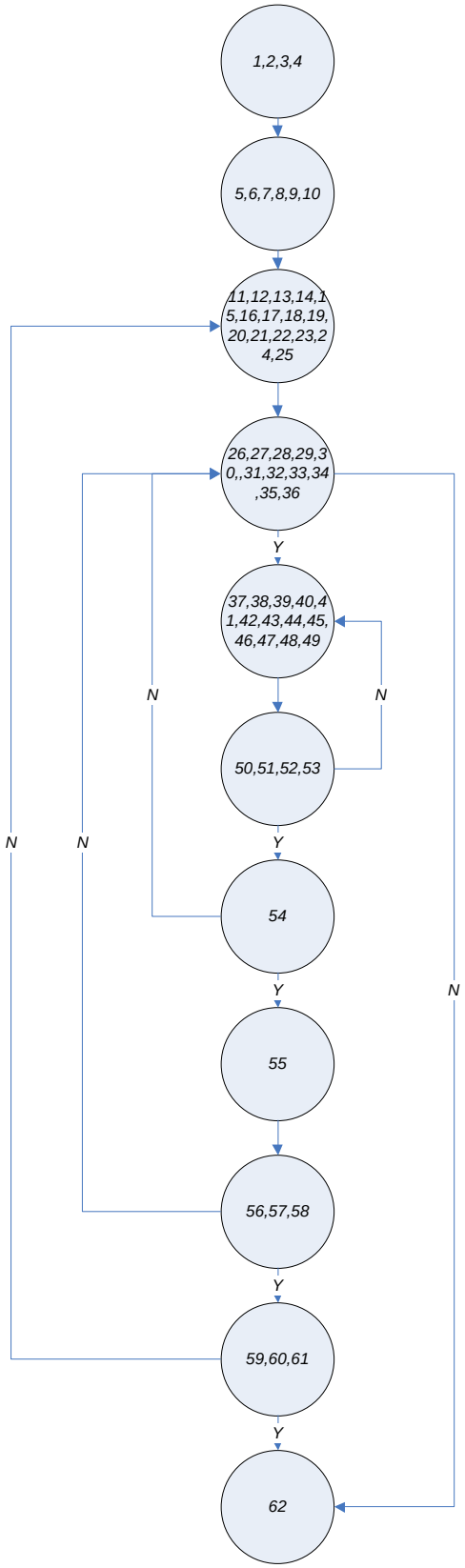
Berikut ini merupakan algoritma A* untuk *level* 4 yang penerapannya menggunakan bahasa pemrograman C#:

```
1.  public System.Type[] graphTypes{
2.  get {
3.  return astarData.graphTypes;}
4.  }
5.  public Matrix translate (float _X, float _Y, float _Z){
6.  m[12] = _X;
7.  m[13] = _Y;
8.  m[14] = _Z;
9.  return this;
10. }
11. public virtual int[] InitialOpen (BinaryHeapM open,
    Int3
    targetPosition, Int3 position, Path path, bool doOpen)
    {
12. return BaseInitialOpen
13. (open,targetPosition,position,path,doOpen);
14. }
15. public void Close () {
16. if (readerStream != null) {
17. readerStream.Close ();
18. }
19. if (writerStream != null) {
20. if (anchors != null) {
21. SerializeAnchors ();
22. }
23. writerStream.Close ();
24. }
25. }
26. if (doOpen) {
27. connectionCosts = costs;
28. }
29. if (connectionCosts == null) {
30. return null;
31. }
32. int[] costs = connectionCosts;
33. connectionCosts = new int[connectionCosts.Length];
34. for (int i=0;i<connectionCosts.Length;i++) {
35. connectionCosts[i] = (connections[i].position-
    position).costMagnitude;
36. }
37. if (!doOpen) {
38. for (int i=0;i<connectionCosts.Length;i++) {
39. Node other = connections[i];
40. if (other.connections != null) {
41. for (int q = 0;q < other.connections.Length;q++) {
42. if (other.connections[q] == this) {
```

```
43.  other.connectionCosts[q] = connectionCosts[i];
44.  break;
45.  }
46.  }
47.  }
48.  }
49.  }
50.  if (connections[i].walkable && connections[i].area !=
     area) {
51.  stack.Push (connections[i]);
52.  connections[i].area = area;
53.  }
54.  if (doOpen) {
55.  nodeR.g = nodeR.parent.g+nodeR.cost+penalty +
     nodeRunData.path.GetTagPenalty(tags);
56.  public virtual void UpdateAllG (NodeRun nodeR,
     NodeRunData nodeRunData) {
57.  BaseUpdateAllG (nodeR, nodeRunData);
58.  }
59.  if (Open == null){
60.  return null;
61.  }
62.  return path;
```

a. *Flow Graph* Algoritma A*

Berdasarkan *pseudocode* di atas maka *flow graph* algoritma A* adalah sebagai berikut:



Gambar 4.12 *Flow Graph* Algoritma A*

Keterangan:

 = Menggambarkan kondisi

 = Menggambarkan aksi

b. *Cyclomatic Complexity* V(G)

E = 15, N = 11

V(G) = E – N + 2

V(G) = 15 – 11 + 2

V(G) = 4 + 2 = 6

Keterangan

E = Jumlah aksi

N = Jumlah Kondisi

c. *Independent Path*

Path 1 = 1-2-3-4-5-6-7-8-9-10-11-12-13-14-15-16-17-18-19-20-21-22-23-24-25-26-27-28-29-30-31-32-33-34-35-36-37-38-39-40-41-42-43-44-45-46-47-48-49-50-51-52-53-54-55-56-57-58-59-60-61-62

Path 2 = 1-2-3-4-5-6-7-8-9-10-11-12-13-14-15-16-17-18-19-20-21-22-23-24-25-26-27-28-29-30-31-32-33-34-35-36-62

Path 3 = 1-2-3-4-5-6-7-8-9-10-11-12-13-14-15-16-17-18-19-20-21-22-23-24-25-26-27-28-29-30-31-32-33-34-35-36-37-38-39-40-41-42-43-44-45-46-47-48-49-50-51-52-53-37-38-39-40-41-42-43-44-45-46-47-48-49-50-51-52-53-54-55-56-57-58-59-60-61-62

Path 4 = 1-2-3-4-5-6-7-8-9-10-11-12-13-14-15-16-17-18-19-20-21-22-23-24-25-26-27-28-29-30-31-32-33-34-35-36-37-38-39-40-41-42-43-44-45-46-47-48-49-50-51-52-53-54-26-27-28-29-30-31-32-33-34-35-36-37-38-39-40-41-42-43-44-45-46-47-48-49-50-51-52-53-54-55-56-57-58-59-60-61-62

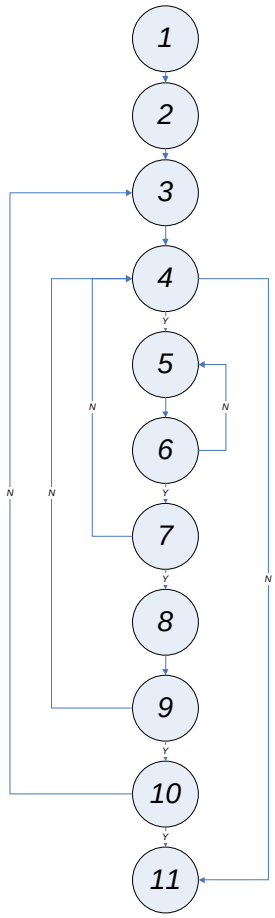
Path 5 = 1-2-3-4-5-6-7-8-9-10-11-12-13-14-15-16-17-18-19-20-21-22-23-24-25-26-27-28-29-30-31-32-33-34-35-36-37-38-39-40-

41-42-43-44-45-46-47-48-49-50-51-52-53-54-55-56-57-58-26-27-28-29-30-31-32-33-34-35-36-37-38-39-40-41-42-43-44-45-46-47-48-49-50-51-52-53-54-55-56-57-58-59-60-61-62

Path 6 = 1-2-3-4-5-6-7-8-9-10-11-12-13-14-15-16-17-18-19-20-21-22-23-24-25-26-27-28-29-30-31-32-33-34-35-36-37-38-39-40-41-42-43-44-45-46-47-48-49-50-51-52-53-54-55-56-57-58-59-60-61-11-12-13-14-15-16-17-18-19-20-21-22-23-24-25-26-27-28-29-30-31-32-33-34-35-36-37-38-39-40-41-42-43-44-45-46-47-48-49-50-51-52-53-54-55-56-57-58-59-60-61-62

d. *Graph Matriks* Algoritma A*

Penomoran ulang *flowgraph* atau penyederhanaan berdasarkan kondisi atau simpul sebagai berikut:



Gambar 4.13 Penyederhanaan *Flow Graph* Algoritma A*

Tabel 4.11 *Graph Matriks* Algoritma A*

N	1	2	3	4	5	6	7	8	9	10	11	N(E) -1
1		1										1-1=0
2			1									1-1=0
3				1								1-1=0
4					1						1	2-1=1
5						1						1-1=0
6					1		1					2-1=1
7				1				1				2-1=1
8									1			1-1=0
9				1						1		2-1=1

10			1								1	2-1=1
11												1-1=0
											Sum(E)+1	5+1=6

Keterangan:

1. Baris dan kolom merepresentasikan simpul
1. Nilai 1 merepresentasikan adanya keterhubungan antar simpul

e. Hasil Uji Algoritma A*

Tabel 4.12 Hasil Uji Algoritma A*

No	Kasus uji	Yang diharapkan	Pengamatan	Keterangan
1	- readerStream != null - writerStream != null - anchors != null - doOpen - connectionCosts == null - int i=0; i<connectionCosts.Length; i++ - !doOpen - int i=0;i<connectionCosts.Length;i++ - other.connections != null - int q = 0;q < other.connections.Length;q++ - other.connections[q] == this - connections[i].walkable && connections[i].area != area - doOpen	Rute = null, rute tidak ada	Rute tidak ditemukan	[√] Alur terlewati [ ] Alur tidak terlewati
2	- readerStream != null - writerStream != null - anchors != null - doOpen - connectionCosts == null - int i=0; i<connectionCosts.Length; i++	Current Node = Goal. Rute ditemukan	Rute ditemukan	[√] Alur terlewati [ ] Alur tidak terlewati
3	- readerStream != null - writerStream != null - anchors != null - doOpen - connectionCosts ==	Node tidak dapat dilalui, pengujian collision atau	Node menjadi unwalkable	[√] Alur terlewati [ ] Alur tidak terlewati

No	Kasus uji	Yang diharapkan	Pengamatan	Keterangan
	null - int i=0; i<connectionCosts.Length; i++ - !doOpen - int i=0;i<connectionCosts.Length;i++ - other.connections != null - int q = 0;q < other.connections.Length;q++ - other.connections[q] == this - connections[i].walkable && connections[i].area != area - !doOpen - int i=0;i<connectionCosts.Length;i++ - other.connections != null - int q = 0;q < other.connections.Length;q++	unwalkable node		
4	- readerStream != null - writerStream != null - anchors != null - doOpen - connectionCosts == null - int i=0; i<connectionCosts.Length; i++ - !doOpen - int i=0;i<connectionCosts.Length;i++ - other.connections != null - int q = 0;q < other.connections.Length;q++ - other.connections[q] == this - connections[i].walkable && connections[i].area != area - doOpen - connectionCosts ==	Masukkan semua *neighbor node* ke dalam open dan hitung nilai F,G,H. Set *current node*	*Current node* dan nilai F,G,H masuk ke dalam open	[√] Alur terlewati [ ] Alur tidak terlewati

No	Kasus uji	Yang diharapkan	Pengamatan	Keterangan
	null - int i=0; i<connectionCosts.Length; i++			
5	- readerStream != null - writerStream != null - anchors != null - doOpen - connectionCosts == null - int i=0; i<connectionCosts.Length; i++ - !doOpen - int i=0;i<connectionCosts.Length;i++ - other.connections != null - int q = 0;q < other.connections.Length;q++ - other.connections[q] == this - connections[i].walkable && connections[i].area != area - doOpen - connectionCosts == null - int i=0; i<connectionCosts.Length; i++ - !doOpen - int i=0;i<connectionCosts.Length;i++ - other.connections != null - int q = 0;q < other.connections.Length;q++ - other.connections[q] == this - connections[i].walkable && connections[i].area != area	Nilai G yang didapat lebih kecil dari nilai G sebelumnya. Kalkulasi ulang nilai F,G,H	Nilai F,G,H, dikalkulasi ulang	[√] Alur terlewati [ ] Alur tidak terlewati
6	- readerStream != null - writerStream != null - anchors != null - doOpen - connectionCosts ==	Open = null, tidak ada *node* di dalam open	Tidak ada *node* di dalam open	[√] Alur terlewati [ ] Alur tidak terlewati

No	Kasus uji	Yang diharapkan	Pengamatan	Keterangan
	null - int i=0; i<connectionCosts.Length; i++ - !doOpen - int i=0;i<connectionCosts.Length;i++ - other.connections != null - int q = 0;q < other.connections.Length;q++ - other.connections[q] == this - connections[i].walkable && connections[i].area != area - doOpen - Open == null			

f. Kesimpulan

Dari hasil pengujian algoritma A* yang dilakukan, maka didapat kesimpulan bahwa seluruh *node* yang ada pada setiap *independent path* telah dikerjakan, serta tidak terjadi pengulangan tak terhingga.

4.2.3. Pengujian Beta

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

4.2.3.1. Kuesioner

Kuesioner disebarkan kepada 30 orang responden minimal dari umur 15 tahun, dari hasil kuesioner tersebut akan dilakukan perhitungan agar dapat diambil kesimpulan terhadap penilaian penerapan sistem yang telah dibangun. Kuesioner terdiri dari 10 pernyataan dengan menggunakan sistem penskoran dengan skala

pengukuran yang digunakan adalah skala likert, Berikut skor untuk jawaban kuesioner.

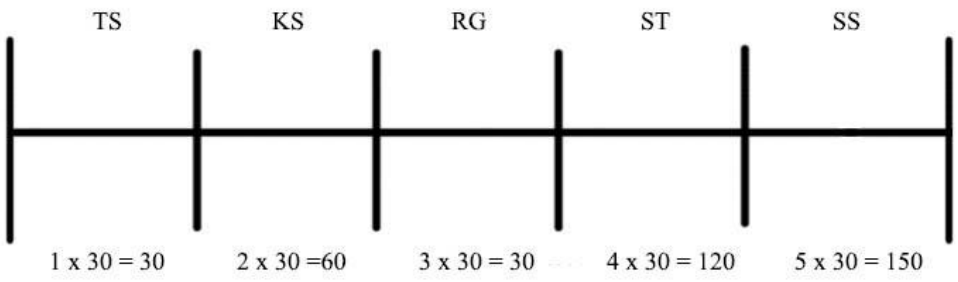
Tabel 4.13 Skor Jawaban Pada Kuesioner

Skala Jawaban	Keterangan	Skor
SS	Sangat Setuju	5
ST	Setuju	4
RG	Ragu-ragu	3
KS	Kurang Setuju	2
TS	Tidak Setuju	1

Data yang diperoleh dari pemberian kuesioner kepada responden dapat dianalisis dengan menghitung rata-rata jawaban berdasarkan *scoring* setiap jawaban dari responden. Analisis yang dapat dilakukan yaitu sebagai berikut.

a. Jumlah skor ideal (kriterium) untuk seluruh item = 5 × jumlah responden
b. Rata-rata skor dalam persentase = (jumlah skor : jumlah skor ideal) × 100%
c. Jumlah skor ideal pada penelitian ini yaitu : Jumlah skor ideal = 5×30 = 150

Secara kontinum dapat digambarkan seperti berikut:



Gambar 4.14 Penilaian

Adapun pertanyaan yang diajukan dalam kuesioner untuk pengguna aplikasi ini adalah sebagai berikut :

1. Apakah anda setuju *game* ini bisa menjadi sarana hiburan yang menarik?
2. Apakah anda setuju *game* ini bisa memberikan unsur pembelajaran yang interaktif bagaimana menjaga hutan?
3. Apakah *game* ini bisa melatih pola pikir *user* dalam menentukan strategi dalam permainan?
4. Apakah *game* ini bisa menjadi sarana dalam menyampaikan informasi tentang kebudayaan yang ada di Kalimantan?
5. Apakah anada setuju tampilan grafis aplikasi *game Save Our Planet from extinction* secara umum sudah baik?
6. Apakah anda setuju AI musuh yang terdapat pada aplikasi *game* ini sudah berjalan dengan baik?
7. Apakah anda setuju jalan cerita aplikasi *game* ini menarik ?
8. Apakah Anda setuju tingkat kesulitan dari aplikasi *game* ini sulit?
9. Apakah anda setuju karakter pemain mudah untuk dikendalikan?
10. Apakah anda terpacu untuk memainkan *game* ini kembali?

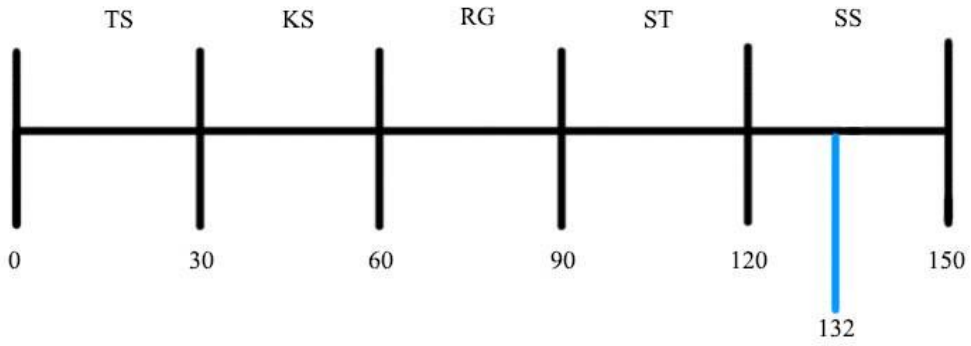
Berikut adalah hasil kuisioner masing-masing jawaban yang sudah dihitung dengan menggunakan persamaan skala kontinum (persamaan 2.2). Kuisioner ini ditujukan kepada 30 responden.

1. Apakah anda setuju *game* ini bisa menjadi sarana hiburan yang menarik

Tabel 4.14 Hasil Pengujian Kuesioner No 1

Pertanyaan	**Jawaban**	**Skor**	**Responden**	**S**
1	Sangat Setuju	5	12	60
	Setuju	4	18	72
	Ragu-ragu	3	0	0
	Kurang Setuju	2	0	0
	Tidak Setuju	1	0	0
Jumlah			**30**	**132**

Berdasarkan hasil skor nilai di atas, maka dapat disimpulkan bahwa penilaian terhadap *game* ini adalah 132 dari yang diharapkan 150, atau dapat dikategorikan sebagai sangat setuju dan dapat disimpulkan bahwa *game* ini bisa menjadi sarana hiburan yang menarik.



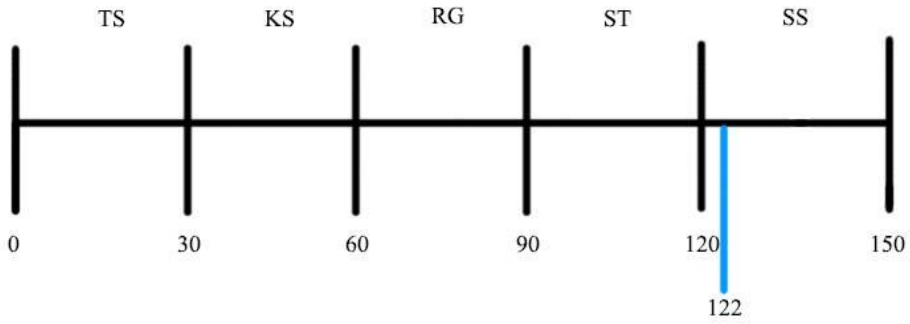
Gambar 4.15 *Interprestasi* Skor Pertanyaan No 1

2. Apakah anda setuju *game* ini bisa memberikan unsur pembelajaran yang interaktif tentang bagaimana menjaga hutan.

Tabel 4.15 Hasil Pengujian Kuesioner No 2

Pertanyaan	Jawaban	Skor	Responden	S
2	Sangat Setuju	5	8	40
	Setuju	4	16	64
	Ragu-ragu	3	6	18
	Kurang Setuju	2	0	0
	Tidak Setuju	1	0	0
Jumlah			**30**	**122**

Berdasarkan hasil skor nilai di atas, maka dapat disimpulkan bahwa penilaian terhadap *game* ini adalah 122 dari yang diharapkan 150, atau dapat dikategorikan sebagai sangat setuju dan dapat disimpulkan bahwa *game* ini bisa memberikan unsur pembelajaran yang interaktif tentang bagaimana menjaga hutan.



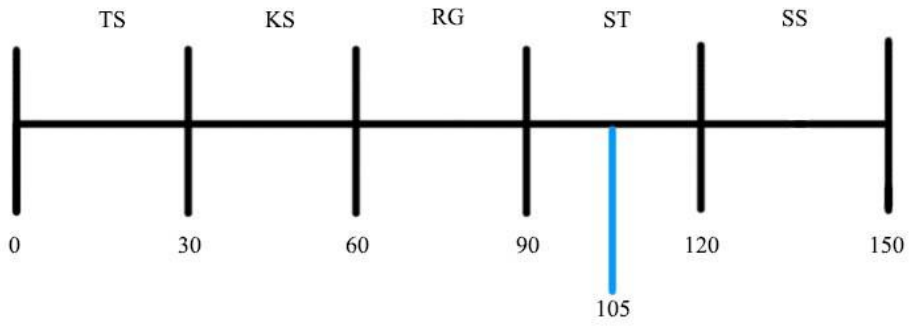
Gambar 4.16 *Interprestasi* Skor Pertanyaan No 2

3. Apakah anda setuju *game* ini bisa melatih pola pikir *user* dalam menentukan strategi dalam permainan.

Tabel 4.16 Hasil Pengujian Kuesioner No 3

Pertanyaan	Jawaban	Skor	Responden	S
3	Sangat Setuju	5	5	25
	Setuju	4	9	36
	Ragu-ragu	3	12	36
	Kurang Setuju	2	4	8
	Tidak Setuju	1	0	0
Jumlah			**30**	**105**

Berdasarkan hasil skor nilai di atas, maka dapat disimpulkan bahwa penilaian terhadap *game* ini adalah 105 dari yang diharapkan 150, atau dapat dikategorikan sebagai setuju dan dapat disimpulkan bahwa *game* ini bisa melatih pola pikir *user* dalam menentukan strategi dalam permainan.



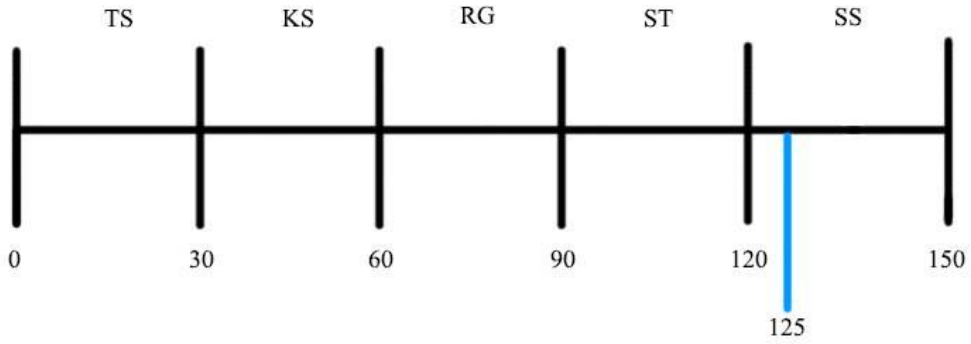
Gambar 4.17 *Interprestasi* Skor Pertanyaan No 3

4. Apakah anda setuju *game* ini bisa menjadi sarana dalam menyampaikan informasi tentang kebudayaan yang ada di Kalimantan.

Tabel 4.17 Hasil Pengujian Kuesioner No 4

Pertanyaan	Jawaban	Skor	Responden	S
4	Sangat Setuju	5	10	50
	Setuju	4	15	60
	Ragu-ragu	3	5	15
	Kurang Setuju	2	0	0
	Tidak Setuju	1	0	0
Jumlah			**30**	**125**

Berdasarkan hasil skor nilai di atas, maka dapat disimpulkan bahwa penilaian terhadap *game* ini adalah 125 dari yang diharapkan 150, atau dapat dikategorikan sebagai sangat setuju dan dapat disimpulkan bahwa *game* ini bisa menjadi sarana dalam menyampaikan informasi tentang kebudayaan yang ada di Kalimantan.



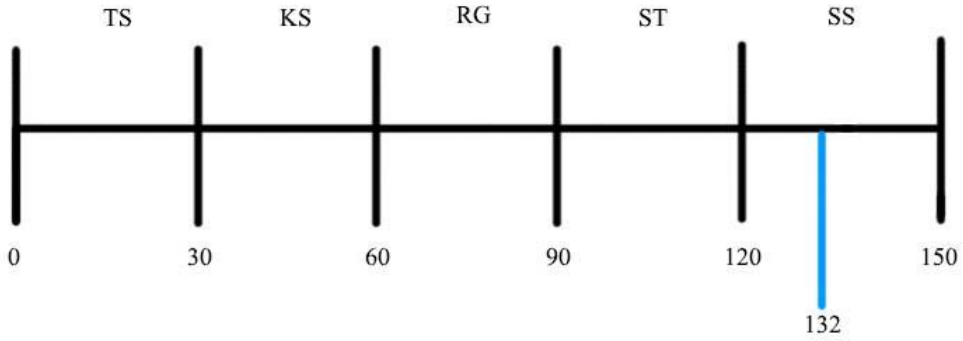
Gambar 4.18 *Interprestasi* Skor Pertanyaan No 4

5. Apakah anda setuju tampilan grafis aplikasi *game Save Our Planet from extinction* secara umum sudah baik.

Tabel 4.18 Hasil Pengujian Kuesioner No 5

Pertanyaan	Jawaban	Skor	Responden	S
5	Sangat Setuju	5	15	75
	Setuju	4	12	48
	Ragu-ragu	3	3	9
	Kurang Setuju	2	0	0
	Tidak Setuju	1	0	0
Jumlah			**30**	**132**

Berdasarkan hasil skor nilai di atas, maka dapat disimpulkan bahwa penilaian terhadap *game* ini adalah 132 dari yang diharapkan 150, atau dapat dikategorikan sebagai sangat setuju dan dapat disimpulkan bahwa *game* ini tampilan grafis aplikasi *game Save Our Planet from extinction* secara umum sudah baik.



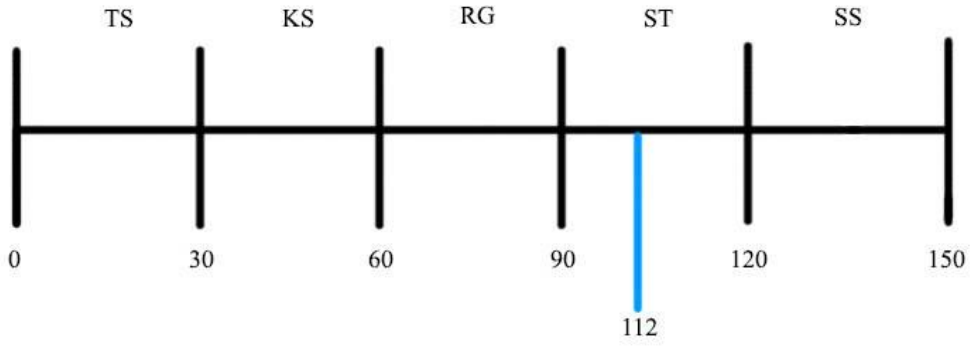
Gambar 4.19 *Interprestasi* Skor Pertanyaan No 5

6. Apakah anda setuju AI musuh yang terdapat pada aplikasi *game* ini sudah berjalan dengan baik.

Tabel 4.19 Hasil Pengujian Kuesioner No 6

Pertanyaan	Jawaban	Skor	Responden	S
6	Sangat Setuju	5	8	40
	Setuju	4	10	40
	Ragu-ragu	3	8	24
	Kurang Setuju	2	4	8
	Tidak Setuju	1	0	0
Jumlah			**30**	**112**

Berdasarkan hasil skor nilai di atas, maka dapat disimpulkan bahwa penilaian terhadap *game* ini adalah 112 dari yang diharapkan 150, atau dapat dikategorikan sebagai setuju dan dapat disimpulkan bahwa AI musuh yang terdapat pada aplikasi *game* ini sudah berjalan dengan baik.



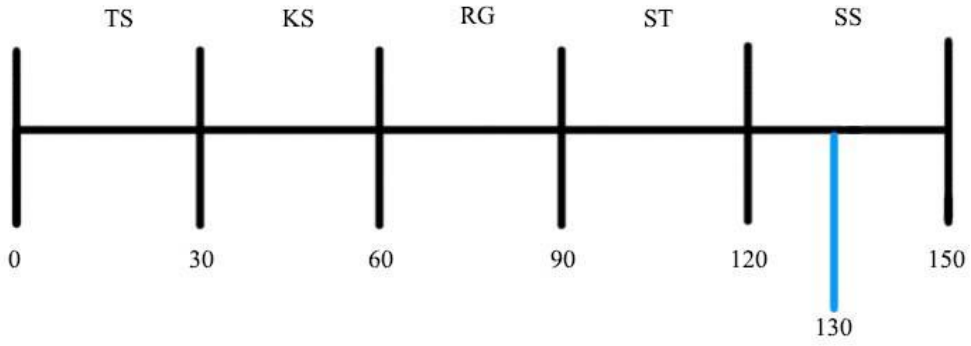
Gambar 4.20 *Interprestasi* Skor Pertanyaan No 6

7. Apakah anda setuju jalan cerita aplikasi *game* ini menarik

Tabel 4.20 Hasil Pengujian Kuesioner No 7

Pertanyaan	Jawaban	Skor	Responden	S
7	Sangat Setuju	5	15	75
	Setuju	4	10	40
	Ragu-ragu	3	5	15
	Kurang Setuju	2	0	0
	Tidak Setuju	1	0	0
Jumlah			**30**	**130**

Berdasarkan hasil skor nilai di atas, maka dapat disimpulkan bahwa penilaian terhadap *game* ini adalah 130 dari yang diharapkan 150, atau dapat dikategorikan sebagai sangat setuju dan dapat disimpulkan bahwa jalan cerita aplikasi *game* ini menarik.



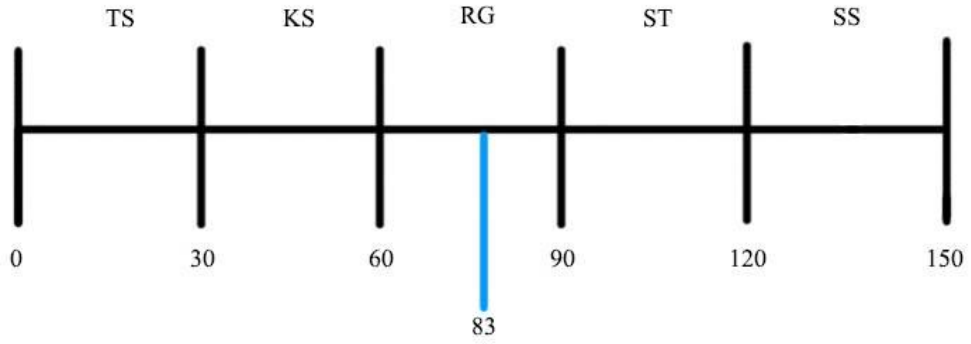
Gambar 4.21 *Interprestasi* Skor Pertanyaan No 7

8. Menurut anda setuju tingkat kesulitan dari aplikasi *game* ini sulit.

Tabel 4.21 Hasil Pengujian Kuesioner No 8

Pertanyaan	Jawaban	Skor	Responden	S
8	Sangat Setuju	5	2	10
	Setuju	4	5	20
	Ragu-ragu	3	10	30
	Kurang Setuju	2	10	20
	Tidak Setuju	1	3	3
Jumlah			**30**	**83**

Berdasarkan hasil skor nilai di atas, maka dapat disimpulkan bahwa penilaian terhadap *game* ini adalah 83 dari yang diharapkan 150, atau dapat dikategorikan sebagai ragu-ragu terhadap tingkat kesulitan dari aplikasi *game* ini.



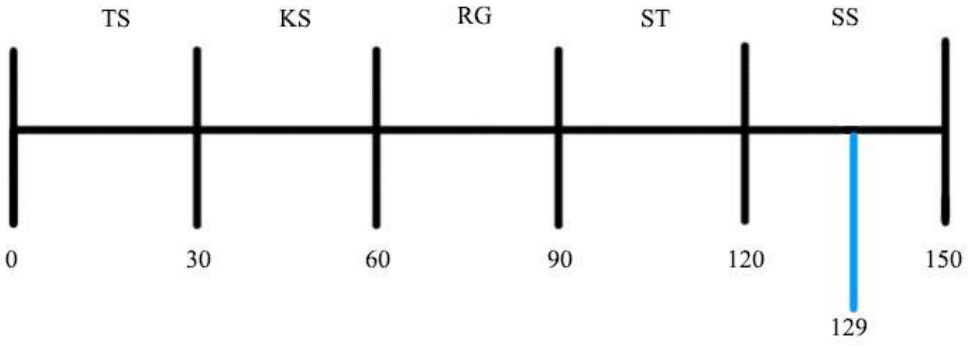
Gambar 4.22 *Interprestasi* Skor Pertanyaan No 8

9. Apakah anda setuju karakter pemain mudah untuk dikendalikan?

Tabel 4.22 Hasil Pengujian Kuesioner No 9

Pertanyaan	Jawaban	Skor	Responden	S
9	Sangat Setuju	5	15	75
	Setuju	4	10	40
	Ragu-ragu	3	4	12
	Kurang Setuju	2	1	2
	Tidak Setuju	1	0	0
Jumlah			**30**	**129**

Berdasarkan hasil skor nilai di atas, maka dapat disimpulkan bahwa penilaian terhadap *game* ini adalah 129 dari yang diharapkan 150, atau dapat dikategorikan sebagai sangat setuju dan dapat disimpulkan bahwa karakter pemain mudah untuk dikendalikan.



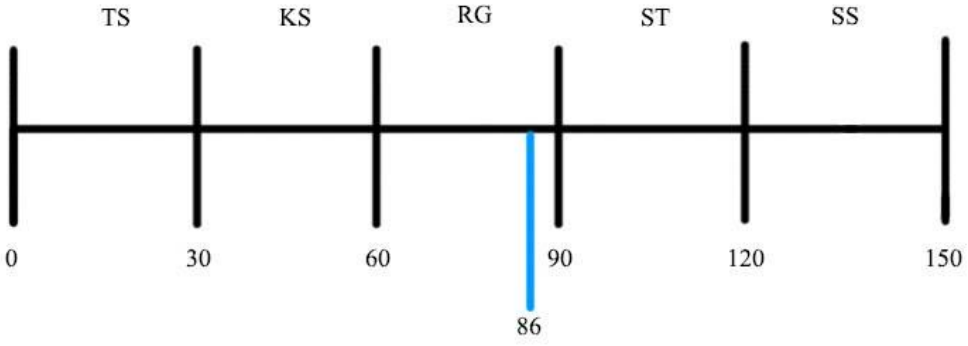
Gambar 4.23 *Interprestasi* Skor Pertanyaan No 9

10. Apakah anda tertarik untuk memainkan *game* ini kembali?

Tabel 4.23 Hasil Pengujian Kuesioner No 10

Pertanyaan	Jawaban	Skor	Responden	S
10	Sangat Setuju	5	2	10
	Setuju	4	8	32
	Ragu-ragu	3	10	30
	Kurang Setuju	2	6	12
	Tidak Setuju	1	4	4
Jumlah			**30**	**86**

Berdasarkan hasil skor nilai di atas, maka dapat disimpulkan bahwa penilaian terhadap *game* ini adalah 86 dari yang diharapkan 150, atau dapat dikategorikan sebagai ragu-ragu untuk memainkan *game* ini kembali.



Gambar 4.24 *Interprestasi* Skor Pertanyaan No 10

4.2.3.2. Hipotesis

Soal disebarkan kepada 30 orang responden minimal dari umur 15 tahun, dari hasil nilai tersebut akan diambil jawaban sementara terhadap masalah penelitian, yang kebenarannya akan diuji terlebih dahulu. Soal diberikan kepada responden sebelum dan sesudah memainkan *game Save Our Planet from Extinction.* Setelah pengujian maka akan terlihat peningkatan hasil nilai antara responden sebelum dan sesudah memainkan *game* tersebut. Soal terdiri dari 5 pertanyaan dengan sistem penilaian bila benar responden akan mendapatkan 20 poin dan jika salah responden mendapatkan nilai 0, dengan nilai maksimal keseluruhan 100, dalam hal ini pengujian terbagi menjadi dua bagian yaitu untuk menghitung hasil dari pertanyaan yang berhubungan tentang hutan dan pertanyaan yang berhubungan tentang kebudayaan suku dayak. Uji normalisasi dilakukan untuk mengetahui data tersebut normal atau tidak. Tabel 4.24 dan tabel 4.25 merupakan hasil nilai dari soal yang telah diberikan kepada responden sebelum dan sesudah pengujian mengenai informasi tentang hutan dan kebudayaan suku dayak :

Tabel 4.24 Data Nilai Pertanyaan Informasi tentang Hutan

Sebelum	Sesudah
40	40
20	40
40	40
20	40
40	40
20	40
0	20
40	40
40	40
0	20
40	40
40	40
20	40
20	20
40	40
40	40

0	40
20	40
20	40
20	20
20	40
20	40
20	40
40	40
0	40
0	40
40	40
40	40
20	40
20	40

Tabel 4.25 Data Nilai Pertanyaan Tentang Kebudayaan Suku Dayak

Sebelum	Sesudah
40	60
40	60
0	40
20	60
20	60
0	60
0	40
40	60
0	60
20	60
40	60
40	60
0	40
0	60
20	40
20	40
40	60
0	60
20	40
20	60
40	60

20	60
0	40
0	40
0	40
20	40
20	40
0	40
20	60
0	40

Dari hasil ini akan diuji data tersebut normal apa tidak dengan menggunakan *software* aplikasi SPSS. Berikut hasil data untuk pertanyaan pada tabel 4.24.

Tabel 4.26 *Output Case Processing Summary*

Case Processing Summary						
	Cases					
	Valid		Missing		Total	
	N	Percent	N	Percent	N	Percent
Sebelum Bermain Game	30	100.0%	0	0.0%	30	100.0%
Sesudah Bermain Game	30	100.0%	0	0.0%	30	100.0%

Tabel 4.27 *Output Descriptives*

Descriptives				
			Statistic	Std. Error
Sebelum Bermain Game	Mean		24.67	2.658
	95% Confidence Interval for Mean	Lower Bound	19.23	
		Upper Bound	30.10	
	5% Trimmed Mean		25.19	
	Median		20.00	
	Variance		211.954	
	Std. Deviation		14.559	
	Minimum		0	
	Maximum		40	
	Range		40	
	Interquartile Range		20	
	Skewness		-.396	.427
	Kurtosis		-.957	.833

Sesudah Bermain Game	*Mean*		*37.33*	*1.262*
	95% Confidence Interval for Mean	*Lower Bound*	*34.75*	
		Upper Bound	*39.92*	
	5% Trimmed Mean		*38.15*	
	Median		*40.00*	
	Variance		*47.816*	
	Std. Deviation		*6.915*	
	Minimum		*20*	
	Maximum		*40*	
	Range		*20*	
	Interquartile Range		*0*	
	Skewness		*-2.273*	*.427*
	Kurtosis		*3.386*	*.833*

Tabel 4.28 ***Tests of Normality***

Tests of Normality						
	Kolmogorov-Smirnov[a]			*Shapiro-Wilk*		
	Statistic	*df*	*Sig.*	*Statistic*	*df*	*Sig.*
Sebelum Bermain Game	*.254*	*30*	*.000*	*.793*	*30*	*.000*
Sesudah Bermain Game	*.517*	*30*	*.000*	*.404*	*30*	*.000*

a. Lilliefors Significance Correction

Dari Tabel 4.28 di atas dijelaskan bahwa data tidak terdistribusi normal, yang dapat dilihat dari nilai signifikansi atau nilai probabilitas. Pedoman pengambilan keputusan adalah jika nilai signifikan < 0,05 data tidak normal dan sebaliknya jika nilai signifikansi > 0,05 data dikatakan normal.

Dari data di atas alat kenormalan uji kenormalan distribusi data yang digunakan uji *Kolmogorov smirnov* dengan keterangan sama dengan uji Liliefors. Nilai sig untuk variabel sebagai berikut :

Sebelum Bermain *Game* = 0.000

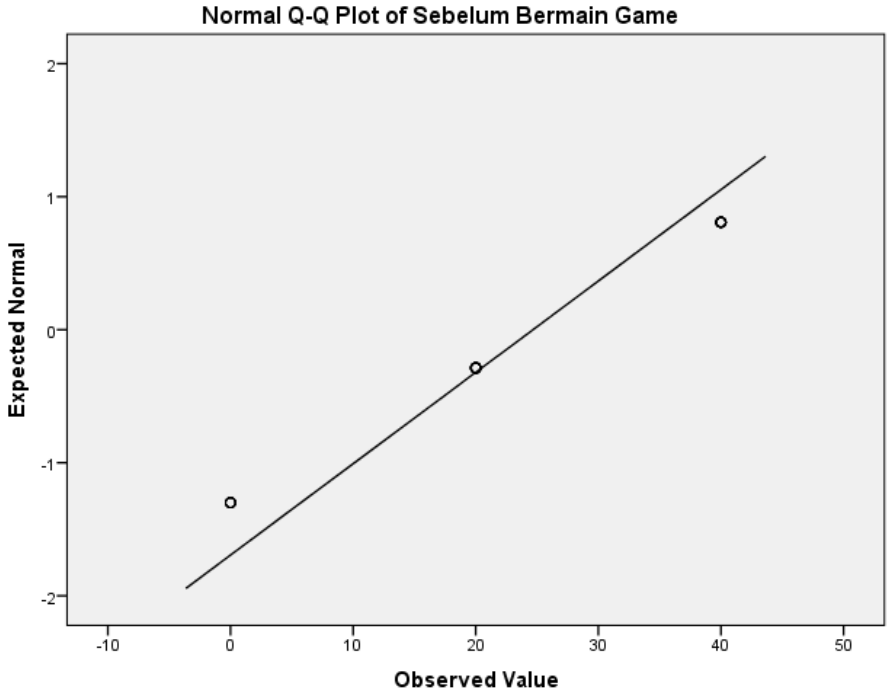
Sesudah Bermain *Game* = 0.000

Keterangan :

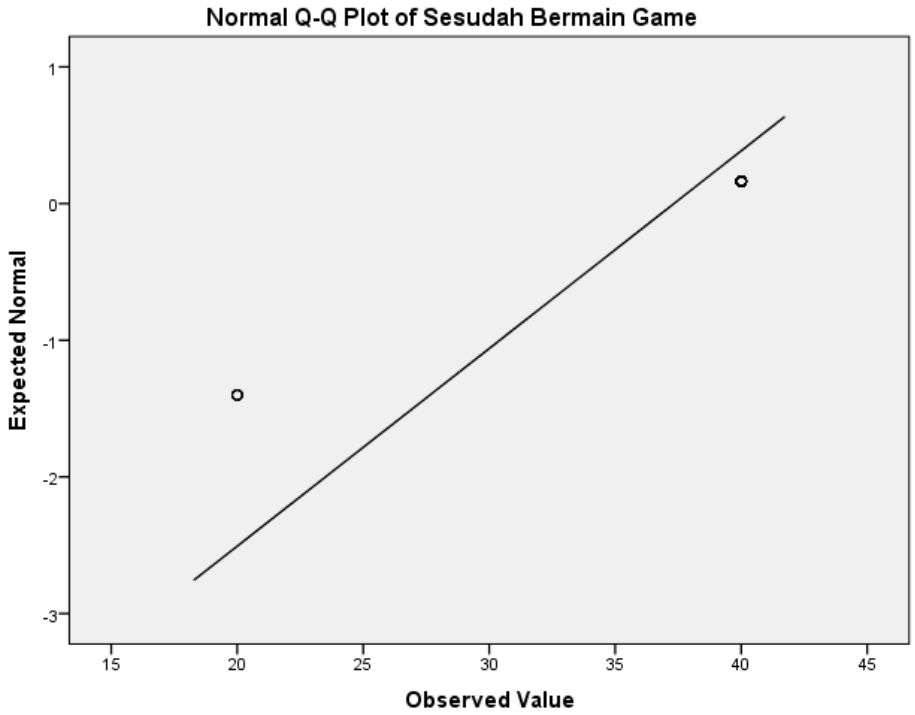
Sebelum Bermain *Game* = Kemampuan responden memiliki nilai rendah sebelum bermain *game.*

Sesudah Bermain *Game* = Kemampuan responden memiliki nilai tinggi setelah bermain *game*.

Nilai probabititas atau nilai *sig* yang didapat dari kedua kelompok data < 0,05, maka kedua di atas terdistribusi tidak normal atau tidak simetri. Jika, digambarkan dalam grafik sebaran data tidak membentuk garis lurus, terlihat seperti gambar dibawah ini :



Gambar 4.25 Plot Distribusi Data Normal Sebelum Bermain *Game*



Gambar 4.26 Plot Distribusi Data Normal Sesudah Bermain *Game*

Karena data pada tabel 4.24 tidak normal, maka selanjutnya pengujian dilakukan dengan menggunakan statistik *Non*-Paramterik dengan menggunakan metode Wilcoxon. Sebelum melakukan pengujian data nilai dengan menggunakan metode Wilcoxon, terlebih dahulu dilakukan pengujian hipotesis untuk mengetahui keputusan diterima atau ditolak. Dalam kasus ini pengujian dilakukan dengan teknik satu arah (*One Tailed*), karena akan dihitung peningkatan nilai dari setiap responden. Berikut adalah pengujian hipotesis-nya :

a. H_o : p(+) = p(-) (Nilai dari responden tidak meningkat sebelum bermain *game*)
b. H_1 : p(+) > p(-) (Nilai dari responden meningkat setelah bermain *game*)

Keterangan :

P(+) = Nilai Sesudah responden bermain *game*

P(-) = Nilai sebelum responden bermain *game*

Dengan $\alpha = 0{,}01 = 1\%$.

Berikut hasil *output* dari tabel 4.24 dengan menggunakan metode Wilcoxon.

Tabel 4.29 *Output Descriptive Statistics*

Descriptive Statistics					
	N	*Mean*	*Std. Deviation*	*Minimum*	*Maximum*
Sebelum Bermain Game	*30*	*24.67*	*14.559*	*0*	*40*
Sesudah Bermain Game	*30*	*37.33*	*6.915*	*20*	*40*

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

Tabel 4.30 *Ranks*

Ranks				
		N	*Mean Rank*	*Sum of Ranks*
Sesudah Bermain Game - Sebelum Bermain Game	*Negative Ranks*	*0*[a]	*.00*	*.00*
	Positive Ranks	*16*[b]	*8.50*	*136.00*
	Ties	*14*[c]		
	Total	*30*		

a. Sesudah Bermain Game > Sebelum Bermain Game
b. Sesudah Bermain Game = Sebelum Bermain Game

Dari *output* tersebut diperoleh:

a. *Negative Ranks* atau selisih antara variabel sebelum dan sesudah. Dalam kasus ini yang negatif tidak ada atau dengan kata lain tidak ada satupun observasi pada variabel sesudah yang kurang dari observasi pada variabel sebelum dan rata-rata rangkingnya = 0 dengan jumlah ranking negatif = 0.
b. *Positive Ranks* atau selisih variabel sebelum dan sesudah yang positif sebanyak 16 observasi atau dengan kata lain terdapat 16 observasi pada variabel sesudah yang lebih dari observasi pad avariabel sebelum dengan rata-rata rangkingnya = 8,50 dan jumlah rangking positif = 136,00.

c. *Ties* atau tidak ada perbedaan antara variabel sebelum dan sesudah. Dalam kasus ini terdapat 14 observasi.

Tabel 4.31 *Test Statistics*

Test Statistics[a]	
	Sesudah Bermain Game - Sebelum Bermain Game
Z	*-3.755[b]*
Asymp. Sig. (1-tailed)	*.000*

a. Wilcoxon Signed Ranks Test

b. Based on negative ranks.

Kriteria Pengujian :

Tolak Ho jika nilai sig ≤ α, terima jika sebaliknya. Terlihat pada tabel di atas nilai sig = 0,000 < 0,01, maka sig ≤ α, artinya H_o ditolak dan terima H_1 berarti dapat disimpulkan bahwa setelah responden setelah bermain *game* bisa meningkatkan nilai dan wawasan tentang informasi yang berhubungan dengan hutan.

Berikut hasil data *output* untuk pertanyaan pada tabel 4.25.

Tabel 4.32 *Output Case Processing Summary*

Case Processing Summary						
	Cases					
	Valid		*Missing*		*Total*	
	N	*Percent*	*N*	*Percent*	*N*	*Percent*
Sebelum Bermain Game	*30*	*100.0%*	*0*	*0.0%*	*30*	*100.0%*
Sesudah Bermain Game	*30*	*100.0%*	*0*	*0.0%*	*30*	*100.0%*

Tabel 4.33 *Output Descriptives*

Descriptives				
			Statistic	*Std. Error*
Sebelum Bermain Game	*Mean*		*16.67*	*2.890*
	95% Confidence Interval for	*Lower Bound*	*10.76*	

	Mean	*Upper Bound*	*22.58*	
	5% Trimmed Mean		*16.30*	
	Median		*20.00*	
	Variance		*250.575*	
	Std. Deviation		*15.830*	
	Minimum		*0*	
	Maximum		*40*	
	Range		*40*	
	Interquartile Range		*25*	
	Skewness		*.315*	*.427*
	Kurtosis		*-1.309*	*.833*
Sesudah Bermain Game	*Mean*		*51.33*	*1.840*
	95% Confidence Interval for Mean	*Lower Bound*	*47.57*	
		Upper Bound	*55.10*	
	5% Trimmed Mean		*51.48*	
	Median		*60.00*	
	Variance		*101.609*	
	Std. Deviation		*10.080*	
	Minimum		*40*	
	Maximum		*60*	
	Range		*20*	
	Interquartile Range		*20*	
	Skewness		*-.283*	*.427*
	Kurtosis		*-2.062*	*.833*

Tabel 4.34 ***Output Tests of Normality***

Tests of Normality						
	Kolmogorov-Smirnov[a]			*Shapiro-Wilk*		
	Statistic	*df*	*Sig.*	*Statistic*	*df*	*Sig.*
Sebelum Bermain Game	*.254*	*30*	*.000*	*.794*	*30*	*.000*
Sesudah Bermain Game	*.372*	*30*	*.000*	*.632*	*30*	*.000*
a. Lilliefors Significance Correction						

Dari Tabel 4.31 di atas dijelaskan bahwa data tidak terdistribusi normal, yang dapat dilihat dari nilai signifikansi atau nilai probabilitas. Pedoman pengambilan keputusan adalah jika nilai signifikan < 0,05 data tidak normal dan sebaliknya jika nilai signifikansi > 0,05 data dikatakan normal.

Dari data di atas alat kenormalan uji kenormalan distribusi data yang digunakan uji *Kolmogorov smirnov* dengan keterangan sama dengan uji Liliefors. Nilai sig untuk variabel sebagai berikut :

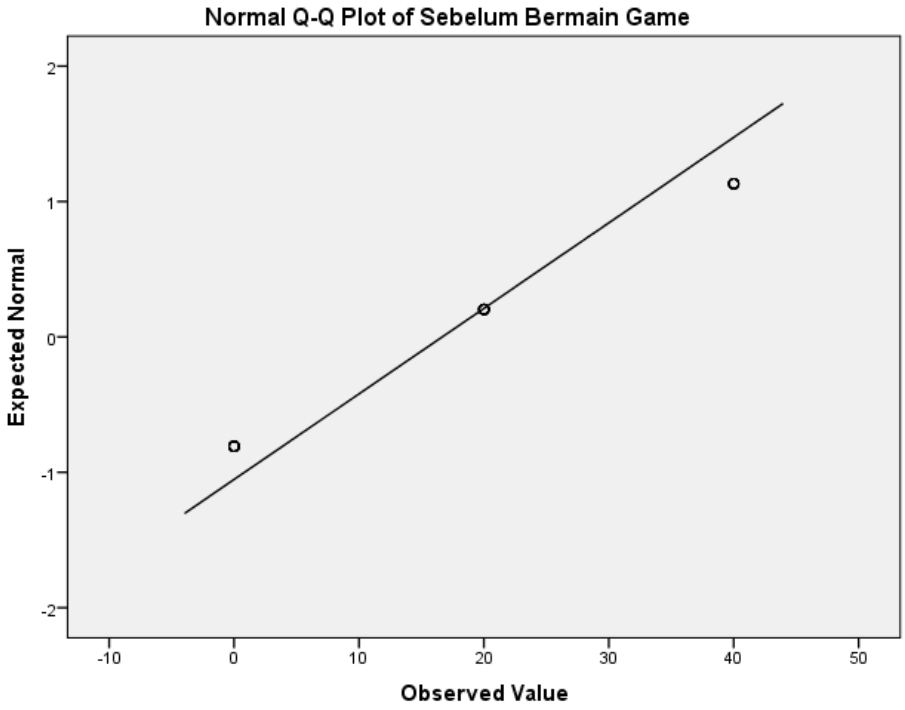
Sebelum Bermain *Game* = 0.000
Sesudah Bermain *Game* = 0.000

Keterangan :

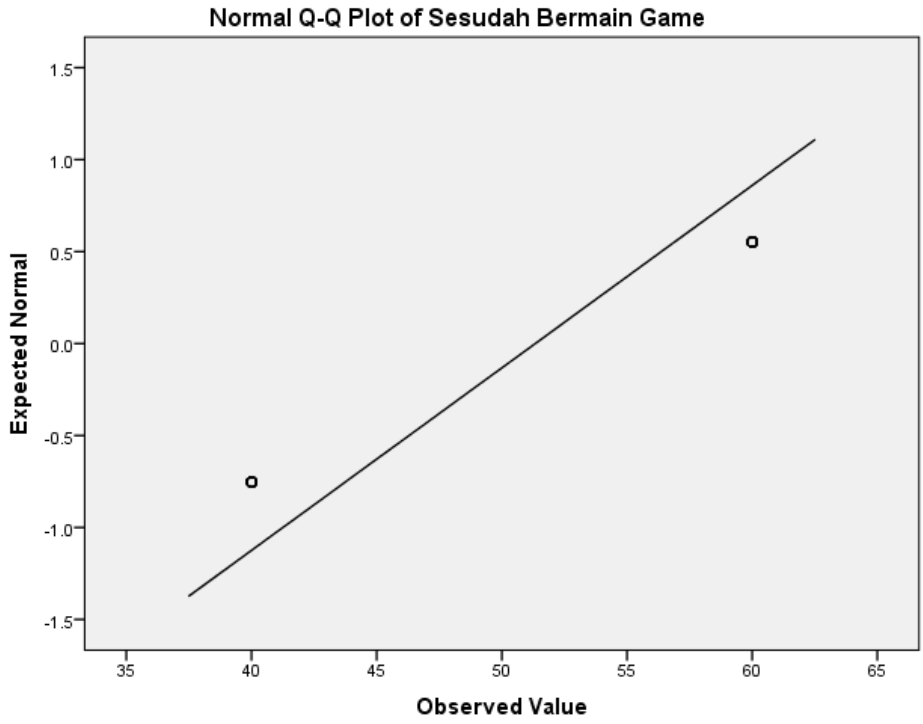
Sebelum Bermain *Game* = Kemampuan responden memiliki nilai rendah sebelum bermain *game.*

Sesudah Bermain *Game* = Kemampuan responden memiliki nilai tinggi setelah bermain *game*.

Nilai probabititas atau nilai *sig* yang didapat dari kedua kelompok data < 0,05, maka kedua di atas terdistribusi tidak normal atau tidak simetri. Jika, digambarkan dalam grafik sebaran data tidak membentuk garis lurus, terlihat seperti gambar dibawah ini :



Gambar 4.27 Plot Distribusi Data Normal Sebelum Bermain *Game*



Gambar 4.28 Plot Distribusi Data Normal Sesudah Bermain *Game*

Karena data kedua juga tidak normal, maka selanjutnya pengujian dilakukan dengan menggunakan statistik *Non*-Paramterik dengan menggunakan metode Wilcoxon. Sebelum melakukan pengujian data nilai dengan menggunakan metode Wilcoxon, terlebih dahulu dilakukan pengujian hipotesis untuk mengetahui keputusan diterima atau ditolak. Dalam kasus ini pengujian dilakukan dengan teknik satu arah (*One Tailed*), karena akan dihitung peningkatan nilai dari setiap responden. Berikut adalah pengujian hipotesis-nya :

c. H_o : p(+) = p(-) (Nilai dari responden tidak meningkat sebelum bermain *game*)
d. H_1 : p(+) > p(-) (Nilai dari responden meningkat sesudah bermain *game*)

Keterangan :

P(+) = Nilai Sesudah responden bermain *game*

P(-) = Nilai sebelum responden bermain *game*

Dengan $\alpha = 0{,}01 = 1\%$.

Berikut hasil *output* dari tabel 4.25 dengan menggunakan metode Wilcoxon.

Tabel 4.35 *Descriptive Statistics*

Descriptive Statistics					
	N	*Mean*	*Std. Deviation*	*Minimum*	*Maximum*
Sebelum Bermain Game	*30*	*16.67*	*15.830*	*0*	*40*
Sesudah Bermain Game	*30*	*51.33*	*10.080*	*40*	*60*

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

Tabel 4.36 *Ranks*

Ranks				
		N	*Mean Rank*	*Sum of Ranks*
Sesudah Bermain Game - Sebelum Bermain Game	*Negative Ranks*	*0[a]*	*.00*	*.00*
	Positive Ranks	*30[b]*	*15.50*	*465.00*
	Ties	*0[c]*		
	Total	*30*		
a. Sesudah Bermain Game > Sebelum Bermain Game				
b. Sesudah Bermain Game = Sebelum Bermain Game				

a. *Negative Ranks* atau selisih antara variabel sebelum dan sesudah. Dalam kasus ini yang negatif tidak ada atau dengan kata lain tidak ada satupun observasi pada variabel sesudah yang kurang dari observasi pada variabel sebelum. Dan rata-rata rangkingnya = 0 dengan jumlah ranking negatif = 0.
b. *Positive Ranks* atau selisih variabel sebelum dan sesudah yang positif sebanyak 30 observasi atau dengan kata lain terdapat 30 observasi pada variabel sesudah yang lebih dari observasi pad avariabel sebelum dengan rata-rata rangkingnya = 15,50 dan jumlah rangking positif = 465,00.
c. *Ties* atau tidak ada perbedaan antara variabel sebelum dan sesudah. Dalam kasus ini tidak ada observasi yang dilakukan.

Tabel 4.37 *Test Statistics*

Test Statistics[a]	
	Sesudah Bermain Game - Sebelum Bermain Game
Z	*-4.880[b]*
Asymp. Sig. (1-tailed)	*.000*

a. Wilcoxon Signed Ranks Test

b. Based on negative ranks.

Kriteria Pengujian :

Tolak Ho jika nilai sig $\leq \alpha$, terima jika sebaliknya. Terlihat pada tabel di atas nilai sig = 0,000 < 0,01, maka sig $\leq \alpha$, artinya H_o ditolak dan terima H_1 berarti dapat disimpulkan bahwa setelah responden setelah bermain *game* bisa meningkatkan nilai dan wawasan tentang informasi kebudayaan suku dayak.