BAB IV
IMPLEMENTASI DAN PENGUJIAN

IV.1 Implementasi

Setelah melakukan analisis dan perancangan secara terperinci maka selanjutnya akan dilakukan tahap implementasi. Implementasi sistem merupakan tahap peletakan sistem sehingga dapat dioperasikan. Implementasi bertujuan untuk mengkonfirmasi modul-modul perancangan sehingga pengguna sistem dapat memberikan masukan-masukan terhadap pengembangan sistem.

IV.1.1 Implementasi Perangkat Keras

Perangkat keras yang digunakan untuk implementasi *game* edukasi yang dibuat tertera pada tabel IV.1.

Tabel IV.1 Spesifikasi Implementasi Perangkat Keras

Nama Perangkat	Spesifikasi
Processor	Dual Core T4400, 220GHz
Memory	2 GB
HDD	320 GB
VGA	512 MB
Mouse	1 Unit
Keyboard	1 Unit

IV.1.2 Implementasi Perangkat Lunak

Perangkat lunak yang digunakan untuk implementasi *game* edukasi yang dibuat tertera pada tabel IV.2.

Tabel IV.2 Spesifikasi Implementasi Perangkat Lunak

Nama Perangkat Lunak	Jenis
Sistem Operasi	Windows 7 Premium
Tools Pembangun	Construct2-r140
Web Browser	Google Chrome
Plugin	*Node*-Webkit

IV.1.3 Implementasi Aplikasi

Game Edukasi Lindungi Kebersihan Laut merupakan aplikasi *game* edukasi yang dibangun dengan menggambarkan lingkungan laut sebagai lingkungan permainannya dengan tujuan agar mengenalkan keindahan dan keanekaragaman ekosistem bawah laut. *Game* ini memiliki konten edukasi peduli terhadap kebersihan diantaranya penyimpanan sampah pada tempatnya sesuai jenis sampah, mengetahui kategori jenis sampah serta pesan bahaya sampah terhadap kehidupan laut khususnya bagi kehidupan kura-kura atau penyu.

Pada *game* ini terdapat 3 *level* dan pada setiap akhir masing-masing *level* terdapat *Minigame* yang harus diselesaikan. Pada *level* kedua dan ketiga terdapat musuh yang harus dihindari yang akan mengurangi point ataupun nyawa. Pada *game* ini pemain diberikan jumlah awal 3 nyawa, jika pemain terkena dengan hiu maka nyawa akan berkurang 1. Permainan dikatakan *game over* apabila jumlah nyawa pemain berkurang menjadi 0, dan sebaliknya permainan dikatakan berhasil apabila pemain dapat menyelesaikan misi disetiap *level*-nya dengan baik.

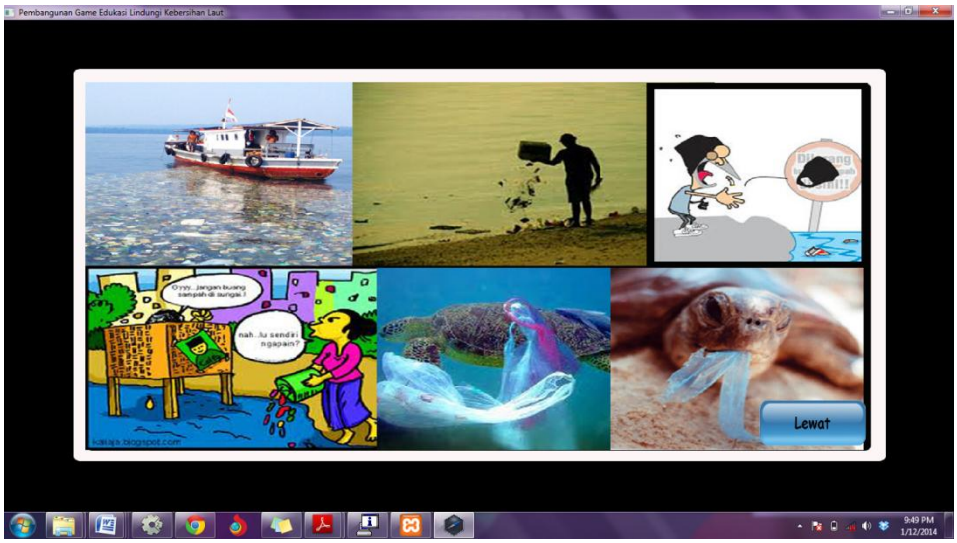
Game ini menggunakan algoritma Minimax yang akan digunakan untuk pencarian solusi pada "*Minigame* Kalahkan AI" dan penggunaan algoritma *Fisher–Yates* pada objek sampah yang akan muncul tak menentu (*random*).

IV.1.4 Implementasi Antarmuka

Implementasi antarmuka dilakukan dengan setiap tampilan program yang dibangun. Antarmuka yang disajikan berikut merupakan antarmuka yang mewakili keseluruhan program. Berikut ini merupakan implementasi antarmuka yang ada pada pembangunan *game* edukasi lindungi kebersihan laut:

1. Implementasi Antarmuka Perkenalan

Gambar IV.1 menunjukkan antarmuka perkenalan pada *game* dimana halaman tersebut menceritakan latar belakang dari pembuatan *game* edukasi lindungi kebersihan laut.



Gambar IV.1 Implementasi Antarmuka Perkenalan

2. Implementasi Antarmuka Menu Utama

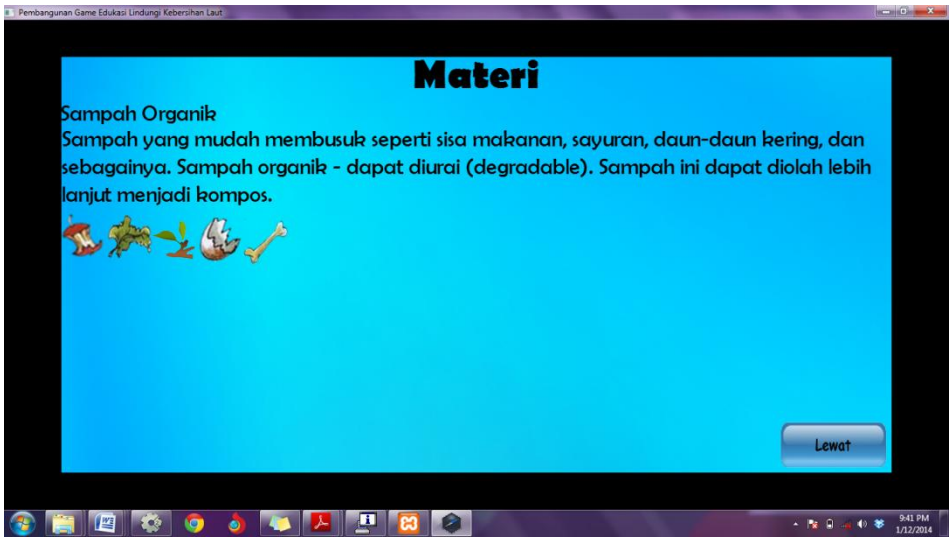
Gambar IV.2 merupakan implementasi antarmuka menu utama. Pada halaman ini terdiri dari empat menu diantaranya menu mulai, nilai tertinggi, bantuan, dan keluar.



Gambar IV.2 Implementasi Antarmuka Menu Utama

3. Implementasi Antarmuka Materi Sampah Organik

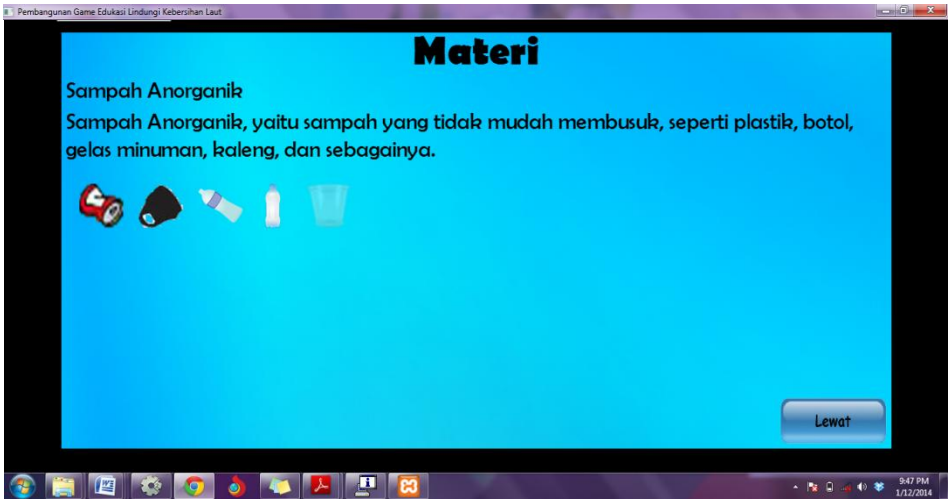
Gambar IV.3 merupakan implementasi antarmuka materi sampah organik. Pada halaman ini memberikan materi mengenai pengertian sampah organik dan contoh sampah organik yang digambarkan dengan beberapa gambar.



Gambar IV.3 Implementasi Antarmuka Materi Sampah Organik

4. Implementasi Antarmuka Materi Sampah Anorganik

Gambar IV.4 merupakan implementasi antarmuka materi sampah anorganik. Pada halaman ini memberikan materi mengenai pengertian sampah anorganik dan contoh sampah anorganik yang digambarkan dengan beberapa gambar.



Gambar IV.4 Implementasi Antarmuka Materi Sampah Anorganik

5. Implementasi Antarmuka *Level* 1

Gambar IV.5 merupakan implementasi antarmuka *level* 1. Pada halaman ini digambarkan seorang anak yang harus mengumpulkan sampah-sampah yang ada di laut dari atas perahu.



Gambar IV.5 Implementasi Antarmuka *Level* 1

6. Implementasi Antarmuka *Minigame* Simpan Sampah Pada Tempatnya

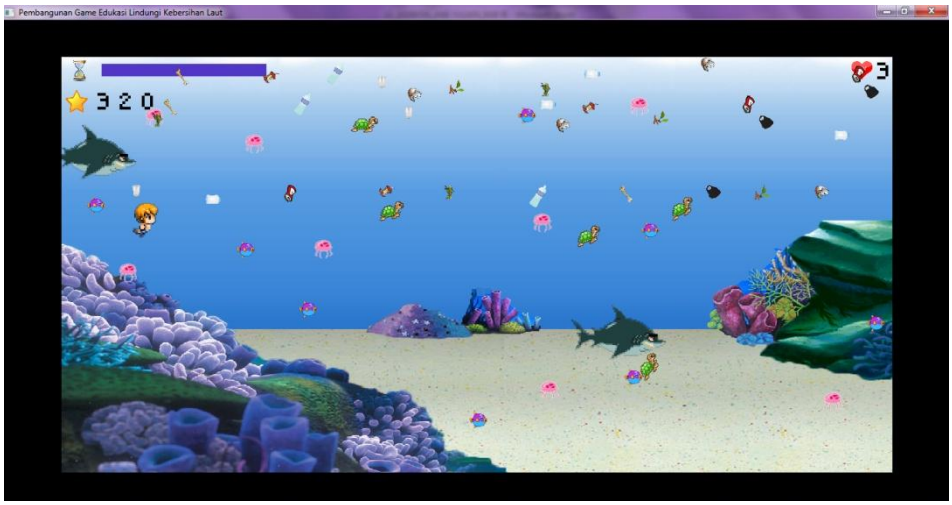
Gambar IV.6 merupakan implementasi antarmuka *minigame* simpan sampah pada tempatnya. Pada *minigame* ini pemain diharuskan memasukkan sampah-sampah yang ada pada tempat sampah sesuai dengan jenis sampah.



Gambar IV.6 Implementasi Antarmuka *Minigame* Simpan Sampah Pada Tempatnya

7. Implementasi Antarmuka *Level* 2

Gambar IV.7 merupakan implementasi antarmuka *level* 2. Pada *level* 2 ini digambarkan seorang anak didalam laut yang berenang untuk mengumpulkan berbagai macam sampah dan menyelamatkan kehidupan kura-kura yang ada dilaut.



Gambar IV.7 Implementasi Antarmuka *Level* 2

8. Implementasi Antarmuka Minigame Kalahkan AI

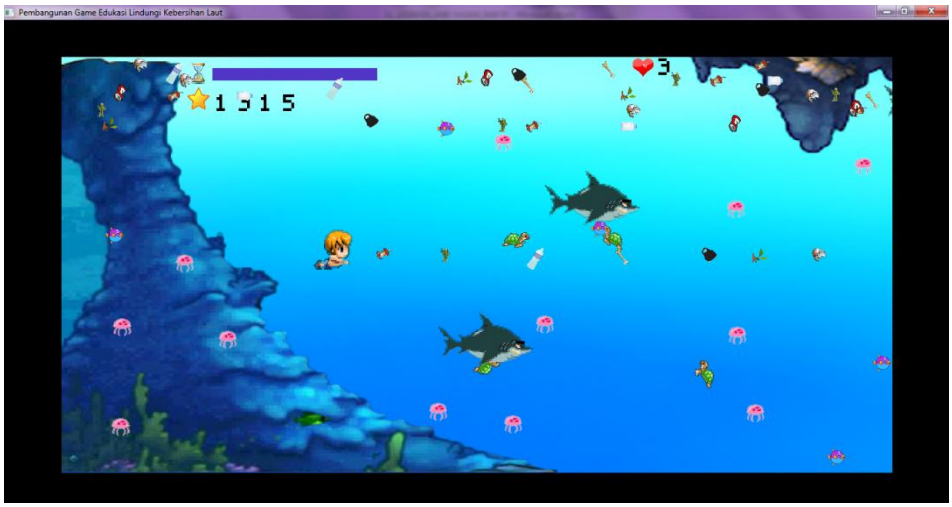
Gambar IV.8 merupakan implementasi antarmuka *minigame* kalahkan AI, dimana pemain harus membuat garis *vertical*, *horizontal*, atau *diagonal* agar bisa menang dan mengalahkan AI.



Gambar IV.8 Implementasi Antarmuka *Minigame* Kalahkan AI

9. Implementasi Antarmuka *Level* 3

Gambar IV.9 merupakan implementasi antarmuka *level* 3. Pada *level* 3 ini digambarkan seorang anak didalam laut yang berenang untuk mengumpulkan berbagai macam sampah dan menyelamatkan kehidupan kura-kura yang ada dilaut.



Gambar IV.9 Implementasi Antarmuka *Level* 3

10. Implementasi Antarmuka Minigame Pilih Jenis Sampah

Gambar IV.10 merupakan implementasi antarmuka *minigame* pilih jenis sampah, dimana pemain harus mengategorikan sampah-sampah sesuai dengan jenisnya.



Gambar IV.10 Implementasi Antarmuka *Minigame* Pilih Jenis Sampah

11. Implementasi Antarmuka Nilai Tertinggi

Gambar IV.11 merupakan implementasi antarmuka nilai tertinggi. Pada halaman ini diberikan informasi mengenai nama pemain dan score tertinggi yang pernah dimainkan.



Gambar IV.11 Implementasi Antarmuka Nilai Tertinggi

12. Implementasi Antarmuka Bantuan

Gambar IV.12 merupakan implementasi antarmuka bantuan, dimana pada halaman ini memberikan informasi aturan permainan dan cara bermain pada *game* ini.



Gambar IV.12 Implementasi Antarmuka Bantuan

IV.2 Pengujian

Pengujian perangkat lunak merupakan hal terpenting yang dilakukan untuk menemukan kekurangan atau kesalahan pada perangkat lunak yang diuji. Pengujian bermaksud untuk mengetahui perangkat lunak yang dibuat telah memenuhi kriteria yang sesuai dengan tujuan perancangan perangkat lunak. Adapun metode pengujian yang digunakan pada perangkat lunak ini adalah metode pengujian *black box* dan *white box*. Pengujian *black box* berfokus pada persyaratan fungsional perangkat lunak, sedangkan *white box* pengujian dengan melihat ke dalam modul untuk meneliti kode-kode program yang ada, dan menganalisis apakah ada kesalahan atau tidak.

IV.2.1 Pengujian Alpha

Pengujian dilakukan terhadap aplikasi untuk memastikan bahwa aplikasi dapat berjalan dengan benar sesuai dengan kebutuhan dan tujuan yang diharapkan. Pengujian *alpha* berfokus pada persyaratan fungsional perangkat lunak.

IV.2.1.1 Skenario Pengujian Aplikasi

Skenario pengujian menjelaskan pengujian terhadap sistem yang ada pada aplikasi *game* edukasi lindungi kebersihan laut. Skenario pengujian yang akan dilakukan pada aplikasi ini selengkapnya dapat dilihat pada tabel IV.3.

Tabel IV.3 Skenario Pengujian Alpha

No	Komponen yang Diuji	Skenario Pengujian	Jenis Pengujian
1.	Perkenalan	Memilih Tombol Lewat	*Black box*
2.	Menu Utama	Memilih Tombol Mulai	*Black box*
		Memilih Tombol Nilai Tertinggi	*Black box*
		Memilih Tombol Bantuan	*Black box*
		Memilih Tombol Keluar	*Black box*
3.	Materi	Memilih Tombol Lewat	*Black box*
4.	Menu Nilai Tertinggi	Memilih Tombol Kembali	*Black box*
5.	Menu Bantuan	Memilih Tombol Kembali	*Black box*
6.	Menu Keluar	Memilih Tombol Ya	*Black box*
		Memilih Tombol Tidak	*Black box*
7.	Gerak Karakter *Player*	Bergerak Ke Kiri	*Black box*
		Bergerak Ke Kanan	*Black box*
		Bergerak Ke Atas	*Black box*
		Bergerak Ke Bawah	*Black box*

No	Komponen yang Diuji	Skenario Pengujian	Jenis Pengujian
8.	Gerak Karakter Ikan	Bergerak Ke Kanan	*Black box*
9.	Gerak Karakter Ubur-ubur	Bergerak Ke Kanan	*Black box*
10.	Gerak Karakter Hiu	Bergerak Ke Kanan	*Black box*
11.	Gerak Karakter Kura-Kura	Bergerak menuju sampah anorganik	*Black box*
12.	Kemunculan Sampah	Penelusuran Algoritma Fisher Yates	*White box*
13.	Menentukan posisi bermain	Penelusuran Algoritma Minimax	*White box*

IV.2.1.2 Kasus dan Hasil Pengujian Alpha

Kasus dan hasil pengujian berisi pemaparan dari rencana pengujian yang telah disusun pada skenario pengujian. Pengujian ini dilakukan secara *black box* dengan hanya memperhatikan masukan ke dalam sistem dan keluaran dari masukan tersebut. Berikut ini pemaparan dari setiap butir pengujian yang terdapat pada skenario pengujian.

1. **Pengujian Perkenalan**

Pengujian menu utama dengan kasus dan hasil pengujian yang menghasilkan kesimpulan dapat dilihat pada tabel IV.4 .

Tabel IV.4 Pengujian Perkenalan

Kasus dan Pengujian			
Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Tombol Lewat	Menampilkan Menu Utama	Tombol dapat berfungsi sesuai dengan yang diharapkan	(✓) Diterima ( ) Ditolak

2. **Pengujian Menu Utama**

Pengujian menu utama dengan kasus dan hasil pengujian yang menghasilkan kesimpulan dapat dilihat pada Tabel IV.5 .

Tabel IV.5 Pengujian Menu Utama

Kasus dan Pengujian			
Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Tombol Mulai	Menampilkan Materi	Tombol dapat berfungsi sesuai dengan yang diharapkan	(✓) Diterima ( ) Ditolak

Kasus dan Pengujian			
Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Tombol Nilai Tertinggi	Menampilkan Nilai Tertinggi	Tombol dapat berfungsi sesuai dengan yang diharapkan	(✓) Diterima ( ) Ditolak
Tombol Bantuan	Menampilkan Bantuan	Tombol dapat berfungsi sesuai dengan yang diharapkan	(✓) Diterima ( ) Ditolak
Tombol Keluar	Keluar dari aplikasi	Tombol dapat berfungsi sesuai dengan yang diharapkan	(✓) Diterima ( ) Ditolak

3. Pengujian Materi

Pengujian materi dengan kasus dan hasil pengujian yang menghasilkan kesimpulan dapat dilihat pada tabel IV.6 .

Tabel IV.6 Pengujian Materi

Kasus dan Pengujian			
Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Tombol Lewat	Menampilkan Permainan *level* 1	Tombol dapat berfungsi sesuai dengan yang diharapkan	(✓) Diterima ( ) Ditolak

4. Pengujian Menu Nilai Tertinggi

Pengujian menu nilai tertinggi dengan kasus dan hasil pengujian yang menghasilkan kesimpulan dapat dilihat pada tabel IV.7 .

Tabel IV.7 Pengujian Menu Nilai Tertinggi

Kasus dan Pengujian			
Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Tombol Kembali	Menampilkan menu utama	Tombol dapat berfungsi sesuai dengan yang diharapkan	(✓) Diterima ( ) Ditolak

5. Pengujian Menu Bantuan

Pengujian menu bantuan dengan kasus dan hasil pengujian yang menghasilkan kesimpulan dapat dilihat pada tabel IV.8 .

Tabel IV.8 Pengujian Menu Bantuan

Kasus dan Pengujian			
Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Tombol Kembali	Menampilkan menu utama	Tombol dapat berfungsi sesuai dengan yang diharapkan	(✓) Diterima ( ) Ditolak

6. Pengujian Menu Keluar

Pengujian menu keluar dengan kasus dan hasil pengujian yang menghasilkan kesimpulan dapat dilihat pada tabel IV.9 .

Tabel IV.9 Pengujian Menu Keluar

Kasus dan Pengujian			
Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Tombol Ya	Keluar dari aplikasi	Tombol dapat berfungsi sesuai dengan yang diharapkan	(✓) Diterima ( ) Ditolak
Tombol Tidak	Menampilkan menu utama	Tombol dapat berfungsi sesuai dengan yang diharapkan	(✓) Diterima ( ) Ditolak

7. Pengujian Gerak Karakter *Player*

Pengujian gerak karakter *player* dengan kasus dan hasil pengujian yang menghasilkan kesimpulan dapat dilihat pada tabel IV.10 .

Tabel IV.10 Pengujian Gerak Karakter *Player*

Kasus dan Pengujian			
Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Kiri	Karakter *player* bergerak ke arah kiri	Karakter *player* bergerak ke arah kiri sesuai dengan yang diharapkan	(✓) Diterima ( ) Ditolak
Kanan	Karakter *player* bergerak ke arah kanan	Karakter *player* bergerak ke arah kanan sesuai dengan yang diharapkan	(✓) Diterima ( ) Ditolak
Atas	Karakter *player* bergerak ke arah atas	Karakter *player* bergerak ke arah atas sesuai dengan yang diharapkan	(✓) Diterima ( ) Ditolak
Bawah	Karakter *player* bergerak ke arah bawah	Karakter *player* bergerak ke arah bawah sesuai dengan yang diharapkan	(✓) Diterima ( ) Ditolak

8. Pengujian Gerak Karakter Ikan

Pengujian gerak karakter ikan dengan kasus dan hasil pengujian yang menghasilkan kesimpulan dapat dilihat pada tabel IV.11 .

Tabel IV.11 Pengujian Gerak Karakter Ikan

Kasus dan Pengujian			
Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Kanan	Karakter ikan bergerak ke arah kanan	Karakter ikan bergerak ke arah kanan sesuai dengan yang diharapkan	(✓) Diterima ( ) Ditolak

9. Pengujian Gerak Karakter Ubur-ubur

Pengujian gerak karakter ubur-ubur dengan kasus dan hasil pengujian yang menghasilkan kesimpulan dapat dilihat pada tabel IV.12 .

Tabel IV.12 Pengujian Gerak Karakter Ubur-Ubur

Kasus dan Pengujian			
Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Kanan	Karakter ubur-ubur bergerak ke arah kanan	Karakter ubur-ubur bergerak ke arah kanan sesuai dengan yang diharapkan	(✓) Diterima ( ) Ditolak

10. Pengujian Gerak Karakter Hiu

Pengujian gerak karakter hiu dengan kasus dan hasil pengujian yang menghasilkan kesimpulan dapat dilihat pada tabel IV.13 .

Tabel IV.13 Pengujian Gerak Karakter Hiu

Kasus dan Pengujian			
Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Kanan	Karakter hiu bergerak ke arah kanan	Karakter hiu bergerak ke arah kanan sesuai dengan yang diharapkan	(✓) Diterima ( ) Ditolak

11. Pengujian Gerak Karakter Kura-Kura

Pengujian gerak karakter kura-kura dengan kasus dan hasil pengujian yang menghasilkan kesimpulan dapat dilihat pada tabel IV.14 .

Tabel IV.14 Pengujian Gerak Karakter Kura-Kura

Kasus dan Pengujian			
Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Bergerak menuju sampah anorganik	Karakter kura-kura bergerak menuju sampah anorganik	Karakter kura-kura bergerak menuju sampah anorganik sesuai dengan yang diharapkan	(✓) Diterima ( ) Ditolak

IV.2.1.3 Kesimpulan Hasil Pengujian Alpha

Berdasarkan hasil pengujian yang dilakukan, dapat disimpulkan bahwa sistem sudah berjalan seperti yang diharapkan dan secara fungsional sudah dapat menghasilkan keluaran yang diharapkan.

IV.2.1.4 Kasus dan Hasil Pengujian (White Box)

Berikut ini adalah kasus menguji perangkat lunak yang telah dibangun menggunakan metode *white-box* berdasarkan tabel IV.3. Pengujian *white-box* menggunakan *flowgraph* yang digunakan untuk menggambarkan alur dari algoritma dan *graph matrix* yang digunakan untuk menggeneralisasi *flowgraph*. Berikut pengujian *white-box* Algoritma Fisher Yates dan Minimax:

IV.2.1.4.1 Pengujian White box Algoritma Fisher Yates

Pengujian dilakukan berdasarkan *pseudocode* pada tabel III.8. Berikut ini merupakan pengujian *White box* Algoritma Fisher Yates.

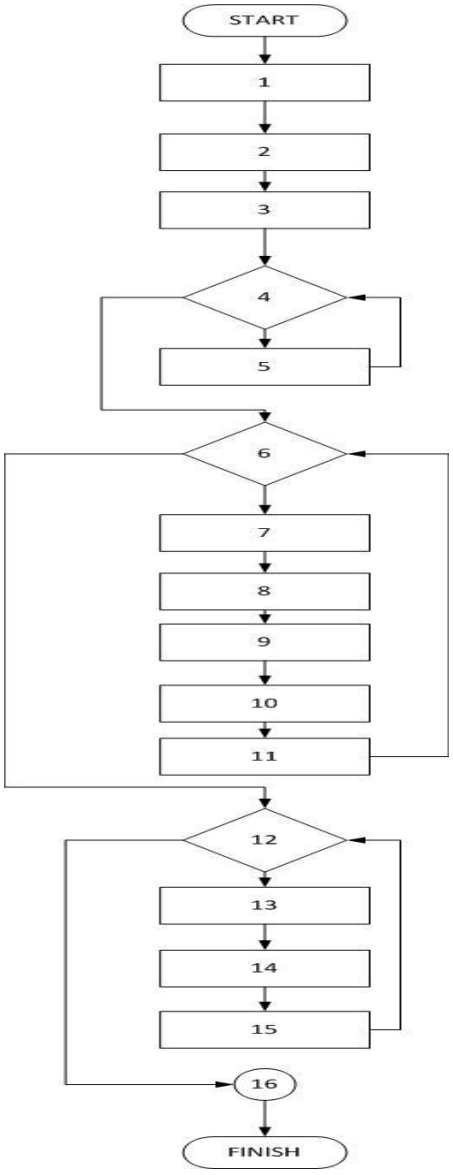
Tabel IV.15 Pengujian *White-box* Algoritma Fisher Yates

Source Code	
No	***Parameter For***
1.	System Set **n** to 5
2.	ArrayShuffle Set size to (**n**, 1, 1)
3.	System Substract 1 from **i**
4.	System For "**FillLoop**" from 0 to n-1
5.	ArrayShuffle Set value at loopindex("FillLoop") to loopindex("FillLoop")+1
6.	System While **i** ≥ 0
7.	System Set **k** to int(floor(random(i+1)))
8.	System Set **temp** to ArrayShuffle.At(k)
9.	ArrayShuffle Set value at **k** to ArrayShuffle.At(i)
10.	ArrayShuffle Set value at **i** to **temp**
11.	System Substract 1 from **I**

Source Code	
No	***Parameter For***
	//End while
12.	System Every 0.5 seconds
13.	System Create spSampah on layer 0 at (random(0,600),-10)
14.	spSampah Set animation frame to ArrayShuffle.At(gCount)
15.	System Add 1 to gCount
16.	//End

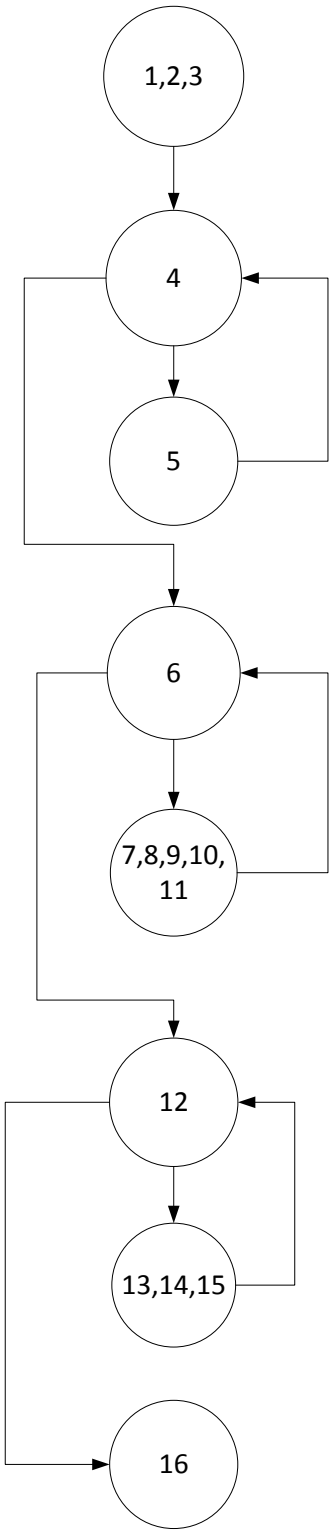
White box testing

1. ***Flow chart:***



Gambar IV.13 *Flowchart* Algoritma Fisher Yates

2. *Flowgraph*



Gambar IV.14 *Flowgraph* Algoritma Fisher Yates

***Cyclomatic Complexity* (kompleksitas kondisional)**

Cyclomatic Complexity adalah suatu sistem pengukuran yang menyediakan ukuran kuantitatif dari kompleksitas logika suatu program. *Cyclomatic Complexity* digunakan untuk mencari jumlah *Path* dalam satu *flowgraph*. Berikut ini merupakan rumus untuk menghitung *Cyclomatic Complexity:*

$$V(G) = E - N + 2$$

Keterangan :

E : Jumlah busur pada *flowgraph*

N : Jumlah simpul pada *flowgraph*

$$\begin{aligned} V(G) &= E - N + 2 \\ &= 10 - 8 + 2 \\ &= 4 \end{aligned}$$

a. Jumlah Region (Jumlah Wilayah)

Jumlah region adalah 4.

b. *Independent Path* (Jalur independen)

Independent Path adalah jalur yang melintasi atau melalui program dimana sekurang-kurangnya terdapat proses perintah yang baru atau kondisi yang baru. Berikut ini merupakan *Independent Path* dari algoritma fisher Yates diatas:

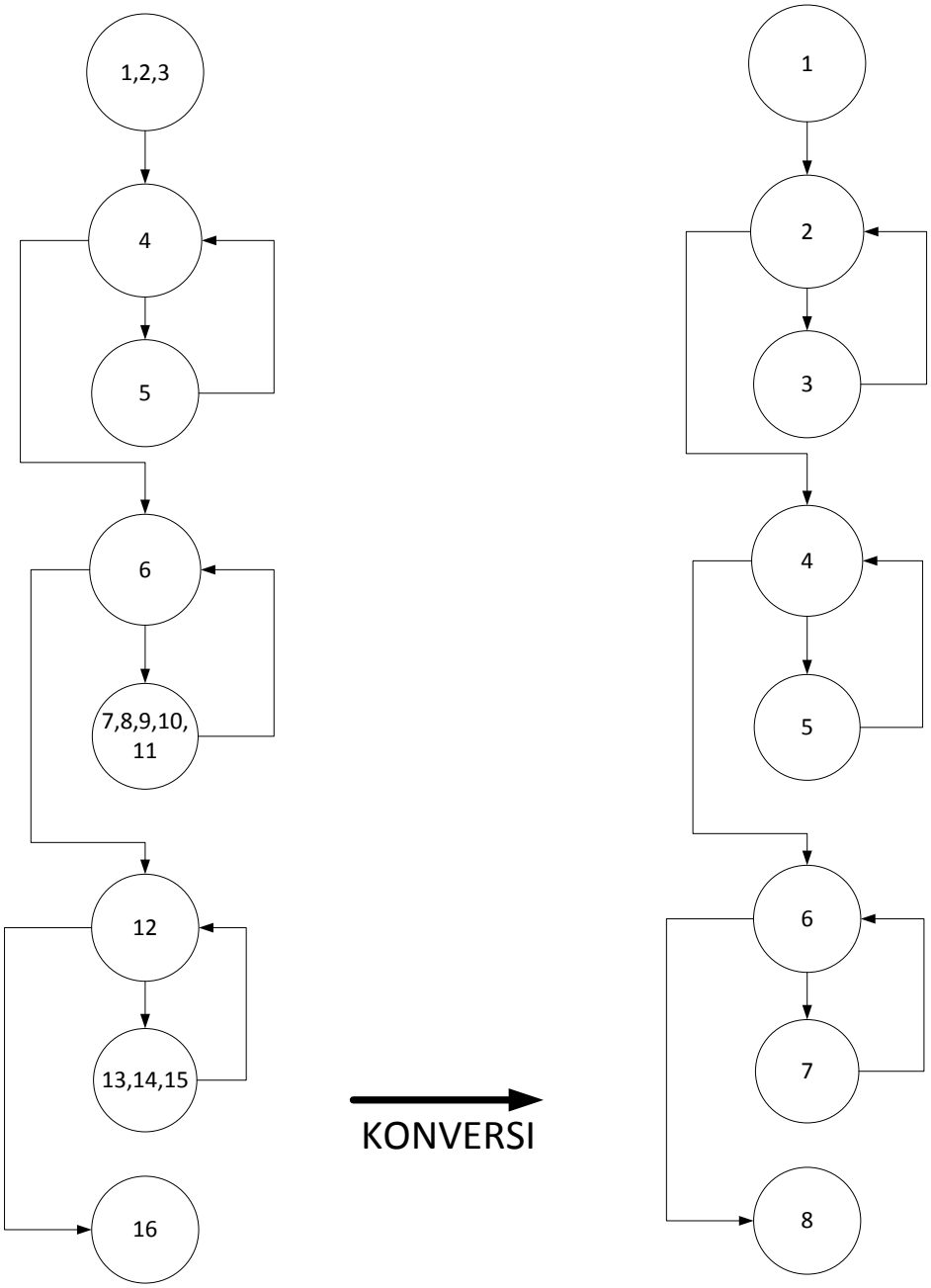
Path 1 : 1-2-3-4-5-4-6-7-8-9-10-11-6-12-13-14-15-12-16

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

Path 3 : 1-2-3-4-6-12-13-14-15-12-16

Path 4 : 1-2-3-4-6-12-16

3. *Graph Matriks*



Gambar IV.15 *Graph Matrix* Algoritma Fisher Yates

Tabel IV.11 *Graph Matriks* Algoritma Fisher Yattes

Node	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**Sum**
1		1							0
2			1	1					1
3		1							0
4					1	1			1
5				1					0
6							1	1	1
7									0
8									0
									3

a. $V_{(G)}$ = Jumlah Graph Matriks + 1

= 3 + 1

= 4

b. *PredicateNode* (*node* predikat)

Adalah branch *node* / proses yang membentuk dua cabang atau lebih.

PredicateNode= 2,4,6

$V_{(G)}$ = P (Jumlah *node* yang memiliki jalur lebih dari 1) + 1

= 3 + 1

= 4

Kesimpulan

Berdasarkan pengujian pada setiap metode, dihasilkan nilai *Cyclomatic Complexity* yang sama yaitu 4 termasuk kategori *low risk – simple procedure*. Maka dapat disimpulkan bahwa pengujian *white box* pada proses Fisher Yattes berjalan dengan baik, karena setiap pengujian menghasilkan nilai yang sama.

IV.2.1.4.2 Pengujian White box Algoritma Minimax

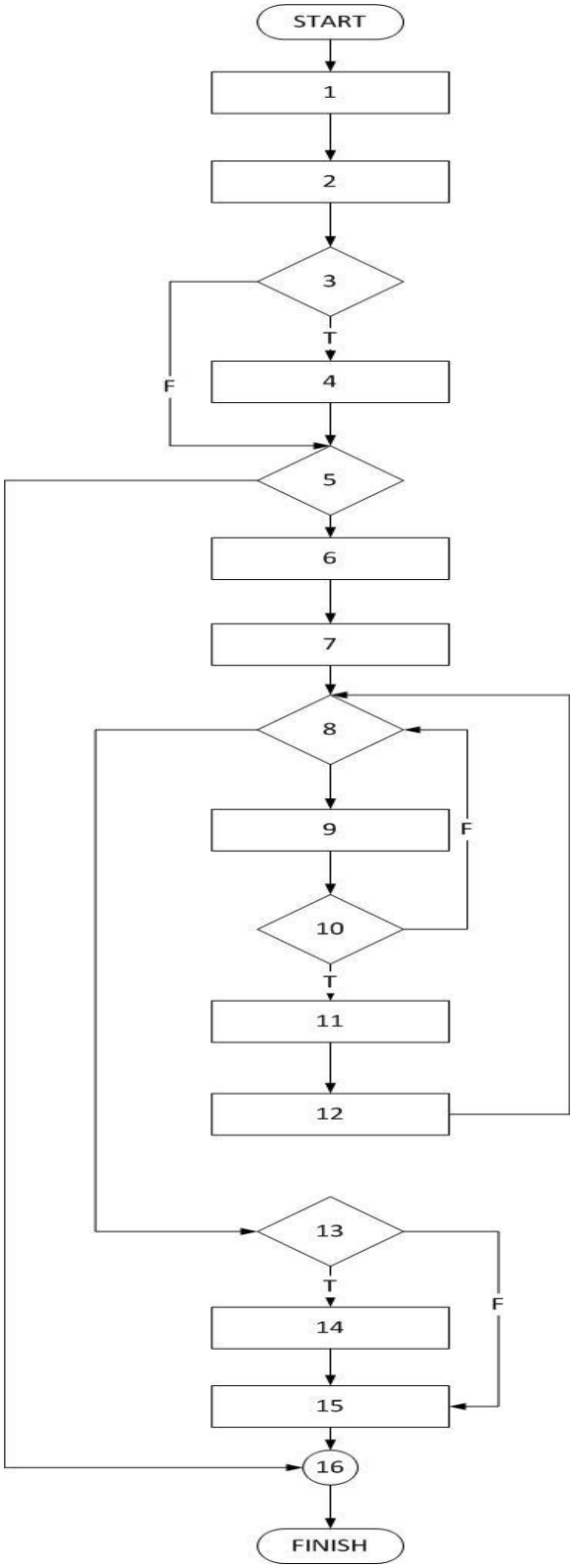
Pengujian dilakukan berdasarkan *pseudocode* pada tabel III.11. Berikut ini merupakan pengujian *White box* Algoritma Minimax.

Tabel IV. 17 Pengujian *White-box* Algoritma Minimax

Coding	
	function BuildPossibleWinnerLines(board, winners, isPlayerX)
	{
1	var winnerLines = winners.concat(new Array());
2	var opponentValue = CONST_PlayerX;
3	if(isPlayerX)
	{
4	opponentValue = CONST_PlayerO;
	}
5	for(var i = winnerLines.length-1; i>=0; i--)
	{
6	var remove = false;
7	var line = winnerLines[i];
8	for(var j=0; j<3; j++)
	{
9	var pos = line[j];
10	if(board[pos] == opponentValue)
	{
11	remove = true;
12	break;
	}
	}
13	if(remove)
	{
14	winnerLines.splice(i,1);
	}
	}
15/16	return winnerLines; //dianalogikan ada endfor
	}

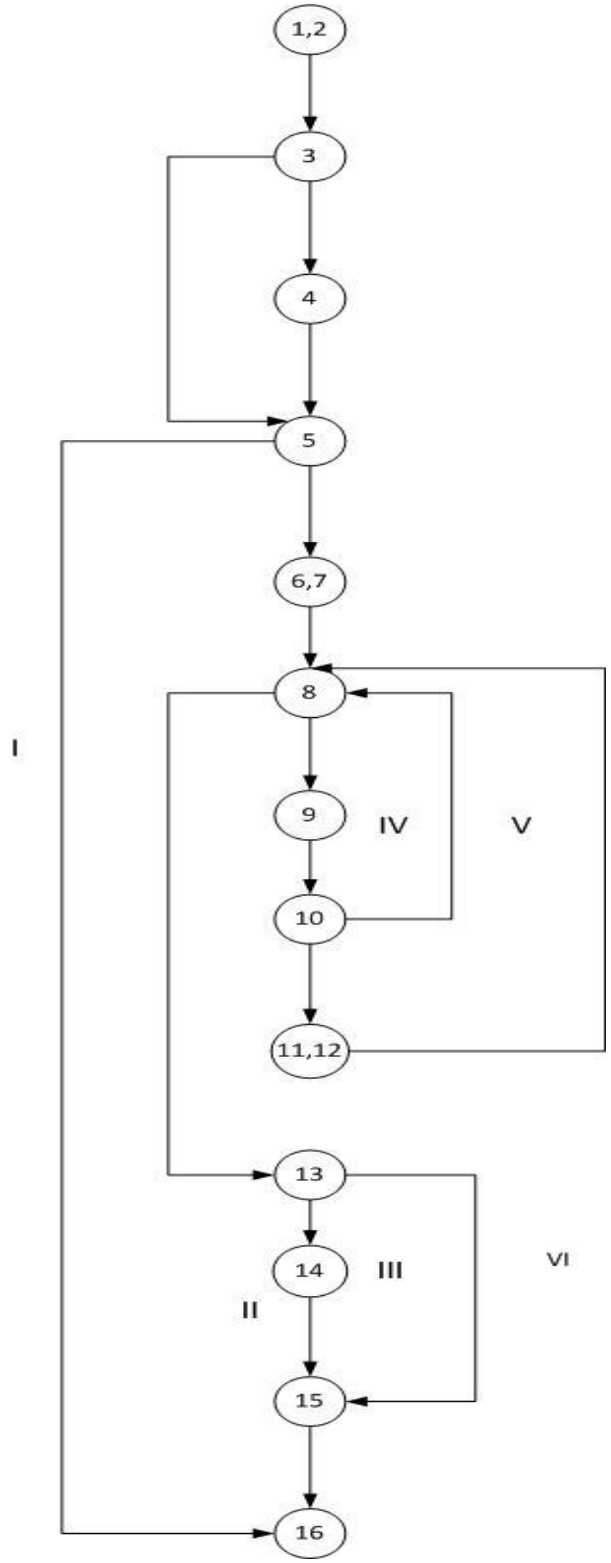
White box testing

1. ***Flowchart***



Gambar IV.16 Flowchart Algoritma Minimax

2. ***Flowgraph***



Gambar IV.17 Flowgraph Algoritma Minimax

***Cyclomatic Complexity* (kompleksitas kondisional)**

$$V(G) = E - N + 2$$

Keterangan :

E : Jumlah busur pada *flowgraph*

N : Jumlah simpul pada *flowgraph*

$$V(G) = E - N + 2$$
$$= 17 - 13 + 2$$
$$= 6$$

a. Jumlah Region (jumlah wilayah)

Jumlah region adalah 6.

***b. Independent Path* (Jalur independen)**

Path 1 : 1-2-3-5-16

Path 2 : 1-2-3-4-5-6-7-8-13-14-15-16

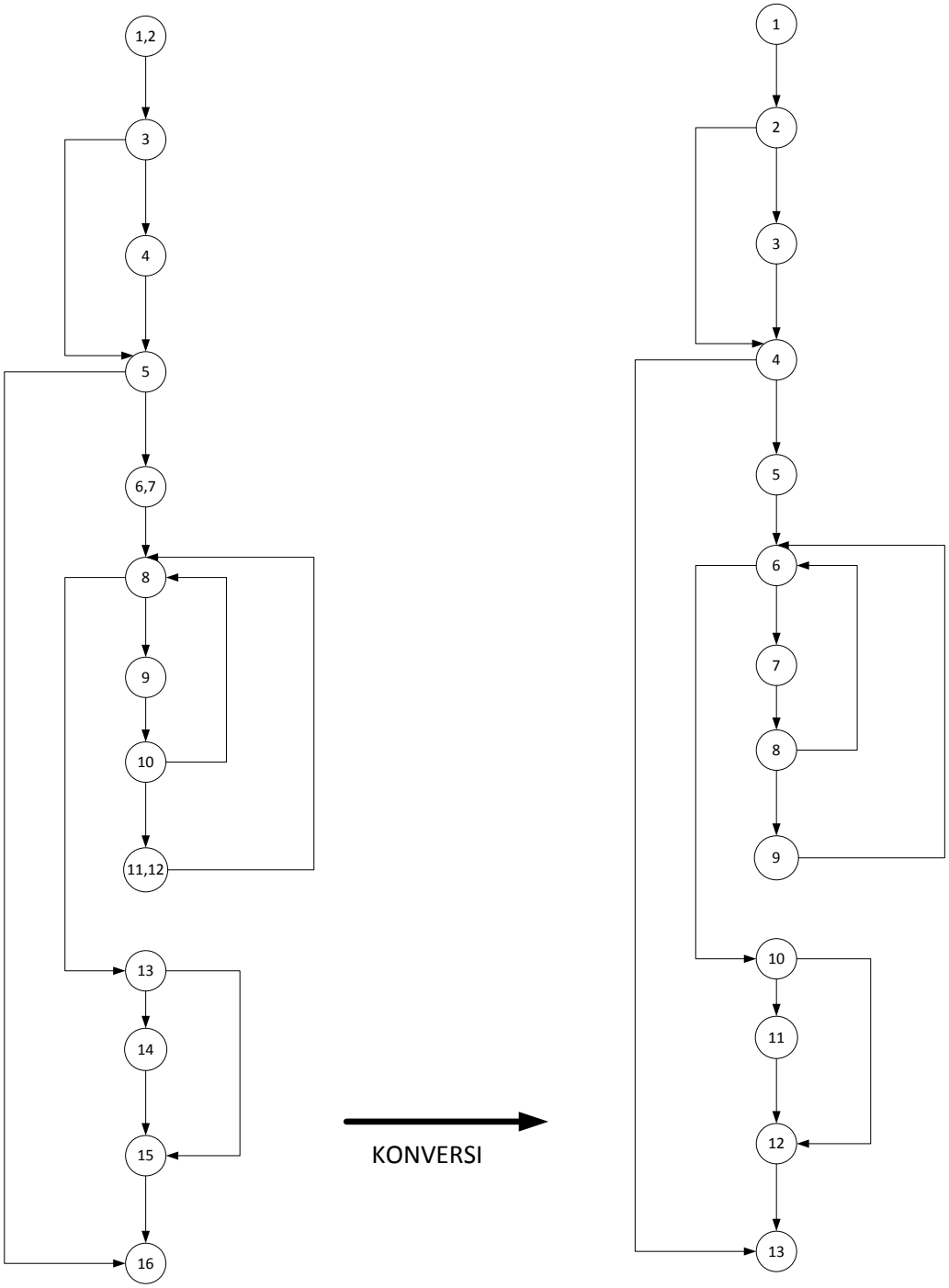
Path 3 : 1-2-3-4-5-6-7-8-13-15-16

Path 4 : 1-2-3-4-5-6-7-8-9-10-8-13-15-16

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

Path 6 : 1-2-3-4-5-6-7-8-13-15-16

3. *Graph Matrix*



Gambar IV.18 *Graph Matrix* Algoritma Minimax

Tabel IV.18 *Graph Matriks* Algoritma Minimax

Node	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**Sum**
1		1												0
2			1	1										1
3				1										0
4					1								1	1
5						1								0
6							1			1				1
7								1						0
8						1			1					1
9						1								0
10											1	1		1
11												1		0
12													1	0
13														0
														5

a. $V_{(G)}$ = Jumlah Graph Matriks + 1

= 5 + 1

= 6

b. *PredicateNode* (*Node* Predikat)

PredicateNode= 2,4,6,8,10

$V_{(G)}$ = P (Jumlah *node* yang memiliki jalur lebih dari 1) + 1

= 5 + 1

= 6

Kesimpulan

Berdasarkan pengujian pada setiap metode, dihasilkan nilai *Cyclomatic Complexity* yang sama yaitu 6 termasuk kategori *low risk – simple procedure*. Maka dapat disimpulkan bahwa pengujian *white box* pada proses Minimax berjalan dengan baik, karena setiap pengujian menghasilkan nilai yang sama.

IV.2.2 Pengujian Beta

Pengujian beta merupakan pengujian yang dilakukan secara objektif, dimana pengujian langsung melibatkan para pengguna aplikasi *game* secara langsung dengan menggunakan kuesioner mengenai kepuasan para pengguna

terhadap aplikasi *game* yang dibangun. Metode penilaian pengujian yang digunakan adalah metode kuantitatif berdasarkan data dari pengguna.

IV.2.2.1 Kuesioner

Data hasil kuesioner yang telah diberikan dapat dicari presentasenya dari masing-masing jawaban dengan menggunakan rumus:

$$P = \frac{\sum s}{skor\ ideal} \times 100\%$$

dimana,

P = Nilai Persentase yang dicari

$\sum s$ = Jumlah skor yang didapatkan dari setiap pilihan jawaban dikalikan dengan nilai yang telah ditetapkan sebelumnya.

Skor ideal = Nilai tertinggi pada point pilihan dikalikan dengan jumlah responden

Data yang didapat dari hasil kuesioner terhadap 30 orang responden adalah sebagai berikut:

1. Apakah *game* lindungi kebersihan laut ini menarik untuk dimainkan?

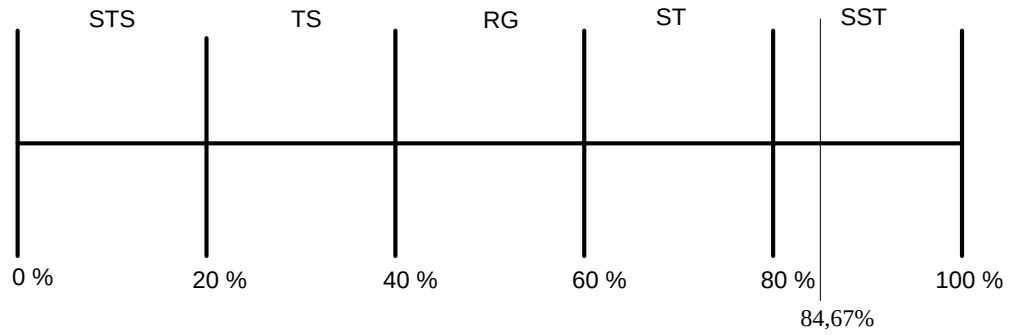
   Hasil kuisioner untuk pertanyaan ke-1 dapat dilihat pada tabel IV.19.

Tabel IV.19 Hasil Kuisioner Pertanyaan 1

Pertanyaan	Jawaban	Skor	Responden	S
1	Sangat Setuju	5	10	50
	Setuju	4	17	68
	Ragu-Ragu	3	3	9
	Tidak Setuju	2	0	0
	Sangat Tidak Setuju	1	0	0
Jumlah			**30**	**127**

$$P = \frac{127}{150} \times 100\% = 84{,}67\ \%$$

Dapat disimpulkan bahwa penilaian terhadap menarik tidaknya untuk dimainkan pada aplikasi *Game* Edukasi Lindungi Kebersihan Laut adalah 84,67% dari yang diharapkan 100%, atau dapat dikategorikan sebagai sangat setuju.



Gambar IV.19 Pengukuran Garis Kontinum Pertanyaan 1

2. Apakah secara umum tampilan *game* ini sudah menarik?

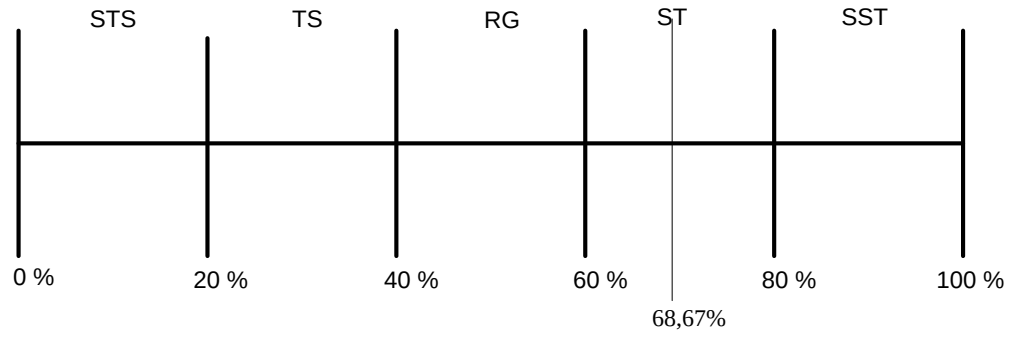
Hasil kuisioner untuk pertanyaan ke-2 dapat dilihat pada tabel IV.20.

Tabel IV.20 Hasil Kuisioner Pertanyaan 2

Pertanyaan	Jawaban	Skor	Responden	S
2	Sangat Setuju	5	5	25
	Setuju	4	10	40
	Ragu-Ragu	3	8	24
	Tidak Setuju	2	7	14
	Sangat Tidak Setuju	1	0	0
Jumlah			30	103

$$P = \frac{103}{150} \times 100\% = 84{,}67\ \%$$

Dapat disimpulkan bahwa penilaian terhadap menarik tidaknya tampilan pada aplikasi *Game* Edukasi Lindungi Kebersihan Laut adalah 68,67% dari yang diharapkan 100%, atau dapat dikategorikan sebagai setuju.



Gambar IV.20 Pengukuran Garis Kontinum Pertanyaan 2

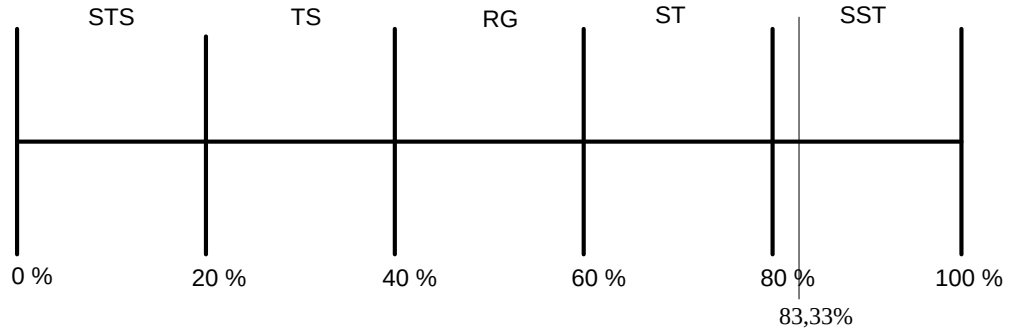
3. Apakah dengan adanya *game* ini menjadikan pembelajaran mengenai kebersihan menjadi lebih menyenangkan?
   Hasil kuisioner untuk pertanyaan ke-4 dapat dilihat pada tabel IV.21.

Tabel IV.21 Hasil Kuisioner Pertanyaan 3

Pertanyaan	Jawaban	Skor	Responden	S
3	Sangat Setuju	5	7	35
	Setuju	4	21	84
	Ragu-Ragu	3	2	6
	Tidak Setuju	2	0	0
	Sangat Tidak Setuju	1	0	0
Jumlah			**30**	**125**

$$P = \frac{125}{150} \times 100\% = 83{,}33\ \%$$

Dapat disimpulkan bahwa penilaian terhadap menyenangkan atau tidaknya pembelajaran mengenai kebersihan yang dilakukan dengan adanya *Game* Edukasi Lindungi Kebersihan Laut adalah 83,33% dari yang diharapkan 100%, atau dapat dikategorikan sebagai sangat setuju.



Gambar IV.21 Pengukuran Garis Kontinum Pertanyaan 3

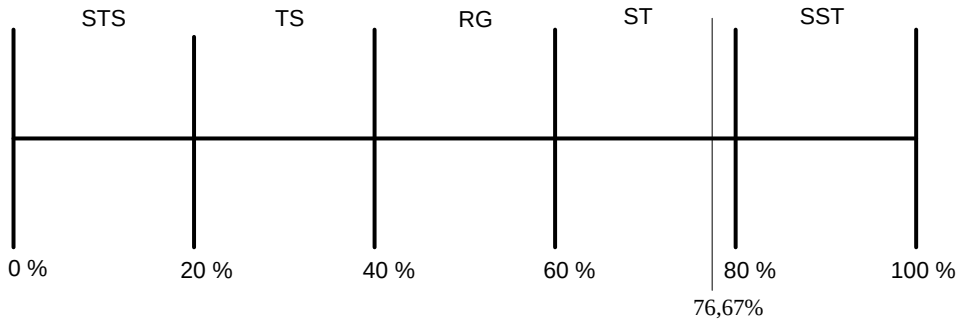
4. Apakah pembelajaran mengenai kebersihan dengan menggunakan game ini dapat sering anda mainkan dibanding pembelajaran menggunakan buku?
   Hasil kuisioner untuk pertanyaan ke-4 dapat dilihat pada tabel IV.22.

Tabel IV.22 Hasil Kuisioner Pertanyaan 4

Pertanyaan	Jawaban	Skor	Responden	S
4	Sangat Setuju	5	4	20
	Setuju	4	18	72
	Ragu-Ragu	3	7	21
	Tidak Setuju	2	1	2
	Sangat Tidak Setuju	1	0	0
Jumlah			**30**	**115**

$$P = \frac{115}{150} \times 100\% = 76{,}67\%$$

Dapat disimpulkan bahwa penilaian terhadap seberapa sering *Game* Edukasi Lindungi Kebersihan Laut dapat digunakan sebagai media pembelajaran mengenai kebersihan dibanding media lain adalah 76,67% dari yang diharapkan 100%, atau dapat dikategorikan sebagai setuju.



Gambar IV.22 Pengukuran Garis Kontinum Pertanyaan 4

5. Apakah kita perlu peduli terhadap kebersihan lingkungan laut?

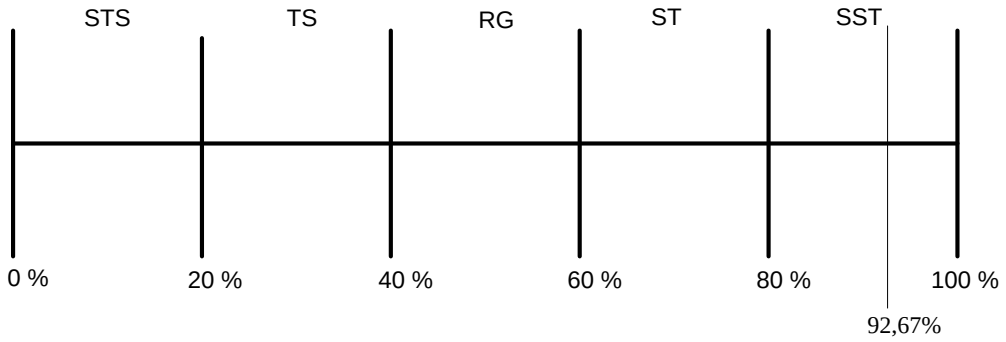
Hasil kuisioner untuk pertanyaan ke-5 dapat dilihat pada tabel IV.23.

Tabel IV.23 Hasil Kuisioner Pertanyaan 5

Pertanyaan	Jawaban	Skor	Responden	S
5	Sangat Setuju	5	21	105
	Setuju	4	7	28
	Ragu-Ragu	3	2	6
	Tidak Setuju	2	0	0
	Sangat Tidak Setuju	1	0	0
Jumlah			**30**	**139**

$$P = \frac{139}{150} \times 100\% = 92{,}67\ \%$$

Dapat disimpulkan bahwa penilaian terhadap kepedulian terhadap kebersihan laut yang ada di *Game* Edukasi Lindungi Kebersihan Laut adalah 92,67% dari yang diharapkan 100%, atau dapat dikategorikan sebagai sangat setuju.



Gambar IV.23 Pengukuran Garis Kontinum Pertanyaan 5

6. Apakah dengan adanya *game* ini dapat meningkatkan kepedulian anda terhadap kebersihan lingkungan?

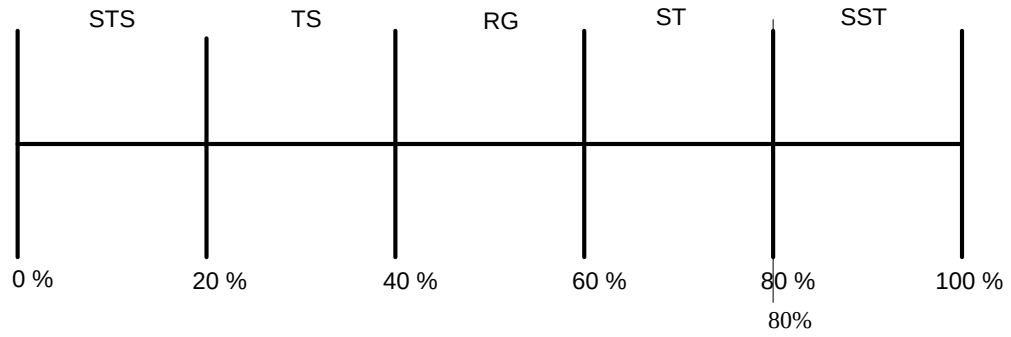
   Hasil kuisioner untuk pertanyaan ke-6 dapat dilihat pada tabel IV.24.

Tabel IV.24 Hasil Kuisioner Pertanyaan 6

Pertanyaan	Jawaban	Skor	Responden	S
6	Sangat Setuju	5	7	35
	Setuju	4	16	64
	Ragu-Ragu	3	7	21
	Tidak Setuju	2	0	0
	Sangat Tidak Setuju	1	0	0
Jumlah			**30**	**120**

$$P = \frac{120}{150} \times 100\% = 80\ \%$$

Dapat disimpulkan bahwa tersampaikannya atau tidak pesan pesan jenis-jenis sampah pada *Game* Edukasi Lindungi Kebersihan Laut adalah 80% dari yang diharapkan 100%, atau dapat dikategorikan sebagai sangat setuju.



Gambar IV.24 Pengukuran Garis Kontinum Pertanyaan 6

7. Apakah dengan adanya *game* ini dapat meningkatkan pengetahuan anda mengenai bahaya sampah terhadap kehidupan laut?

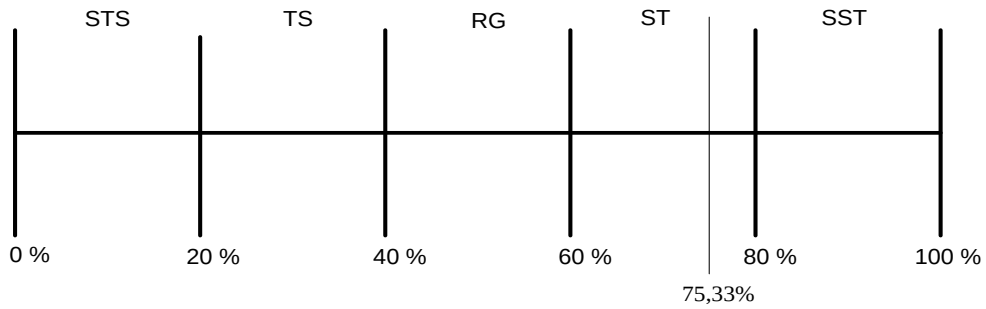
   Hasil kuisioner untuk pertanyaan ke-7 dapat dilihat pada tabel IV.25.

Tabel IV.25 Hasil Kuisioner Pertanyaan 7

Pertanyaan	Jawaban	Skor	Responden	S
7	Sangat Setuju	5	3	15
	Setuju	4	17	68
	Ragu-Ragu	3	10	30
	Tidak Setuju	2	0	0
	Sangat Tidak Setuju	1	0	0
Jumlah			**30**	**113**

$$P = \frac{113}{150} \times 100\% = 75{,}33\ \%$$

Dapat disimpulkan bahwa penilaian terhadap peningkatan pengetahuan mengenai bahaya sampah terhadap kehidupan laut di *Game* Edukasi Lindungi Kebersihan Laut adalah 75,33% dari yang diharapkan 100%, atau dapat dikategorikan sebagai setuju.



Gambar IV.25 Pengukuran Garis Kontinum Pertanyaan 7

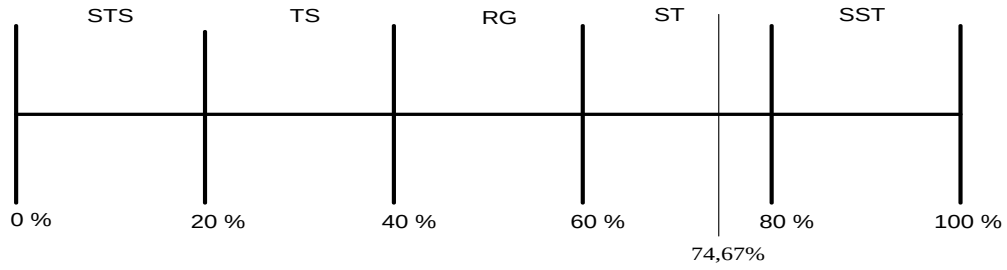
8. Apakah konten edukasi mengenai kebersihan yang dimiliki *game ini* membantu meningkatkan minat anda untuk menjaga kebersihan lingkungan? Hasil kuisioner untuk pertanyaan ke-8 dapat dilihat pada Tabel IV.26.

Tabel IV.26 Hasil Kuisioner Pertanyaan 8

Pertanyaan	Jawaban	Skor	Responden	S
8	Sangat Setuju	5	3	15
	Setuju	4	17	68
	Ragu-Ragu	3	9	27
	Tidak Setuju	2	1	2
	Sangat Tidak Setuju	1	0	0
Jumlah			**30**	**112**

$$P = \frac{112}{150} \times 100\% = 74{,}67\ \%$$

Dapat disimpulkan bahwa penilaian terhadap peningkatan minat terhadap menjaga kebersihan lingkungan yang dimiliki *Game* Edukasi Lindungi Kebersihan Laut adalah 74,67% dari yang diharapkan 100%, atau dapat dikategorikan sebagai setuju.



Gambar IV.26 Pengukuran Garis Kontinum Pertanyaan 8