BAB IV
IMPLEMENTASI DAN PENGUJIAN

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 permainan Selamatkan Masakan Nusantara Indonesia.jar terdapat pada tabel IV.1.

Tabel IV.1 Implementasi Perangkat Keras

Nama Perangkat	Spesifikasi
Prosesor	1.8 Ghz
RAM	1 GB
Harddisk	220 GB
VGA	512 MB
Display	Resolusi 800 x 600 pixels
Keyboard	Standar
Mouse	Standar
Speaker	Standar

IV.1.2 Perangkat Lunak (*Software*)

Implementasi perangkat lunak digunakan untuk menjalankan aplikasi permainan Selamatkan Masakan Nusantara Indonesia.jar**,** terdapat pada tabel IV.2.

Tabel IV.2 Implementasi Perangkat Lunak

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

IV.1.3 Implementasi Aplikasi

Selamatkan Masakan Nusantara Indonesia merupakan aplikasi permainan berjenis *action games* dan edukasi *games* dengan kategori *platform*. Pemain mendapatkan misi setiap tingkat nya yaitu dengan mengumpulkan bahan masakan 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 mengumpulkan semua bahan masakan dan menjadikannya sebuah masakan pada setiap tingkatnya, sedangkan permainan dikatakan gagal apabila pemain terbunuh oleh musuh atau darah musuh sama dengan 0.

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.2 Pengujian Sistem

Dalam melakukan pengujian terhadap aplikasi permainan Selamatkan Masakan Nusantara Indonesia 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 Selamatkan Masakan Nusantara Indonesia. Skenario pengujian dari aplikasi permainan Selamatkan Masakan Nusantara Indonesia 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 lanjutkan	*Black Box*
		Permainan tingkat 1	*Black Box*
		Pilih tombol mulai masak halus	*Black Box*
		Pilih tombol selesai masak halus	*Black Box*
		Pilih tombol mulai masak jadi	*Black Box*
		Pilih tombol selesai masak jadi	*Black Box*
		Pilih tombol lanjutkan	*Black Box*
		Permainan tingkat 2	*Black Box*
		Pilih tombol mulai masak halus	*Black Box*
		Pilih tombol selesai masak halus	*Black Box*
		Pilih tombol mulai masak aduk	*Black Box*
		Pilih tombol selesai masak aduk	*Black Box*
		Pilih tombol mulai masak jadi	*Black Box*
		Pilih tombol selesai masak jadi	*Black Box*
		Pilih tombol lanjutkan	*Black Box*
		Permainan tingkat 3	*Black Box*
		Pilih tombol mulai masak halus	*Black Box*
		Pilih tombol selesai masak halus	*Black Box*
		Pilih tombol mulai masak potong	*Black Box*
		Pilih tombol selesai masak potong	*Black Box*
		Pilih tombol mulai masak jadi	*Black Box*
		Pilih tombol selesai masak jadi	*Black Box*
		Pilih tombol menu utama	*Black Box*
2	Menu Petunjuk	Pilih menu petunjuk	*Black Box*
		Pilih tombol kembali	*Black Box*
3	Menu Kredit	Pilih menu kredit	*Black Box*
		Pilih tombol kembali	*Black Box*
4	Karakter utama	Menekan tombol “A”	*Black Box*
		Menekan tombol “D”	*Black Box*
		Menekan tombol “Spasi”	*Black Box*
		Menekan tombol “S”	*Black Box*
5	Pause	Menekan tombol “Esc”	*Black Box*

		Pilih tombol menu utama	*Black Box*
6	Algoritma A* (*A Star*)	Algoritma A* pada Burung dan kelelawar	*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 komik nasi tutug oncom	Dapat menampilkan komik nasi tutug oncom	[√] Berhasil [ ] Tidak
Pilih lanjutkan	Menampilkan tingkat 1	Dapat menampilkan tingkat 1	[√] Berhasil [ ] Tidak
Pilih mulai masak halus	Menampilkan waktu masak halus dan memproses masak halus	Dapat menampilkan waktu masak halus dan memproses masak halus	[√] Berhasil [ ] Tidak
Pilih selesai masak halus	Menampilkan bahan masakan bahan halus	Dapat menampilkan bahan masakan bahan halus	[√] Berhasil [ ] Tidak
Pilih mulai masak jadi	Menampilkan waktu masak jadi dan memproses masak jadi	Dapat menampilkan waktu masak jadi dan memproses masak jadi	[√] Berhasil [ ] Tidak
Pilih selesai masak jadi	Menampilkan komik mireng udang	Dapat menampilkan komik mireng udang	[√] Berhasil [ ] Tidak
Pilih lanjutkan	Menampilkan tingkat 2	Dapat menampilkan tingkat 2	[√] Berhasil [ ] Tidak
Pilih mulai masak halus	Menampilkan waktu masak halus dan memproses masak halus	Dapat menampilkan waktu masak halus dan memproses masak halus	[√] Berhasil [ ] Tidak

Pilih selesai masak halus	Menampilkan bahan masakan bahan halus	Dapat menampilkan bahan masakan bahan halus	[√] Berhasil [ ] Tidak
Pilih mulai masak aduk	Menampilkan waktu masak aduk dan memproses masak aduk	Dapat menampilkan waktu masak aduk dan memproses masak aduk	[√] Berhasil [ ] Tidak
Pilih selesai masak aduk	Menampilkan bahan masakan bahan aduk	Dapat menampilkan bahan masakan bahan aduk	[√] Berhasil [ ] Tidak
Pilih mulai masak jadi	Menampilkan waktu masak jadi dan memproses masak jadi	Dapat menampilkan waktu masak jadi dan memproses masak jadi	[√] Berhasil [ ] Tidak
Pilih selesai masak jadi	Menampilkan komik ayam santan	Dapat menampilkan ayam santan	[√] Berhasil [ ] Tidak
Pilih lanjutkan	Menampilkan tingkat 3	Dapat menampilkan tingkat 3	[√] Berhasil [ ] Tidak
Pilih mulai masak halus	Menampilkan waktu masak halus dan memproses masak halus	Dapat menampilkan waktu masak halus dan memproses masak halus	[√] Berhasil [ ] Tidak
Pilih selesai masak halus	Menampilkan bahan masakan bahan halus	Dapat menampilkan bahan masakan bahan halus	[√] Berhasil [ ] Tidak
Pilih mulai masak potong	Menampilkan waktu masak potong dan memproses masak potong	Dapat menampilkan waktu masak potong dan memproses masak potong	[√] Berhasil [ ] Tidak
Pilih selesai masak potong	Menampilkan bahan masakan bahan potong	Dapat menampilkan bahan masakan bahan potong	[√] Berhasil [ ] Tidak
Pilih mulai masak jadi	Menampilkan waktu masak jadi dan memproses masak jadi	Dapat menampilkan waktu masak jadi dan memproses masak jadi	[√] Berhasil [ ] Tidak
Pilih selesai masak jadi	Menampilkan komik ayam santan	Dapat menampilkan ayam santan	[√] Berhasil [ ] Tidak
Menyelesaikan tingkat 3	Menampilkan terima kasih	Dapat menampilkan terima kasih	[√] Berhasil [ ] Tidak
Pilih menu utama	Menampilkan menu utama	Dapat menampilkan utama	[√] 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 Menu Petunjuk

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

4. Pengujian Karakter Utama

Pengujian karakter utama dengan kasus dan hasil uji, kemudian diamati dan menghasilkan kesimpulan dapat dilihat pada tabel IV.7 :

Tabel IV.7 Pengujian Karakter Utama

Hasil uji (data normal)			
Masukan	Keluaran yang diharapkan	Pengamatan	Kesimpulan
Tombol A	Menggerakan karakter ke kiri.	Karakter utama bergerak ke kiri.	[√] Berhasil [ ] Tidak
Tombol D	Menggerakan	Karakter utama	[√] Berhasil

	karakter ke kanan.	bergerak ke kanan.	[ ] Tidak
Tombol Spasi	Menggerakan karakter untuk meloncat	Karakter utama bergerak.	[√] Berhasil [ ] Tidak
Tombol S	Mengeluarkan tembakan.	Karakter mengeluarkan pisau	[√] Berhasil [ ] Tidak

5. Pengujian Pause

Pengujian pause dengan kasus dan hasil uji, kemudian diamati dan menghasilkan kesimpulan dapat dilihat pada tabel IV.8 :

Tabel IV.8 Pengujian Pause

Hasil uji (data normal)			
Masukan	Keluaran yang diharapkan	Pengamatan	Kesimpulan
Tombol Esc	Menampilkan kembali pada tampilan permainan	Dapat kembali pada tampilan permainan	[√] Berhasil [ ] Tidak
Pilih menu utama	Menampilkan menu utama	Dapat menampilkan utama	[√] Berhasil [ ] Tidak

IV.2.3 Kesimpulan Pengujian (*Black Box*)

Berdasarkan hasil pengujian *black box* dengan kasus yang telah di uji maka dapat diambil kesimpulan sebagai berikut :

1. Jika inputan yang dilakukan sesuai maka sistem secara fungsional akan mengeluarkan keluaran sesuai dengan harapan.
2. Jika inputan tidak sesuai maka sistem tidak akan mengeluarkan keluaran tidak akan sesuai dengan harapan. Dengan kata lain bahwa membangun aplikasi *game* selamatkan masakan nusantara indonesia bebas kesalahan dan secara fungsional mengeluarkan hasil yang sesuai dengan yang diharapkan.

IV.2.4 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.9 Pengujian *White Box*

No	Algoritma A*

1	private ArrayList<PathFinding> PathList;
	private ArrayList<PathFinding> Rute;
	private int ruteIndex;
	private int parent;
	private boolean target;
	private boolean stopPatrol = false;
	boolean Cari_Pemain = false;
	private Point targetDestination;
	Pemain pemain;
	targetDestination = pPosition;
	target = false;
	parent = 0;

2	if ((Math.abs(getX() - pemain.getX()) < 150) && (Math.abs(getY() - pemain.getY()) < 150)) {
3	for (int q = 0; q < 80 && !target; q++) {

4	if (!AWorld.isGround(PathList.get(parent).getX() - getImage().getHeight()/2, PathList.get(parent).getY()))
	{
	if (!check(parent, 50, new Point(-1, 0)))
	{
	PathList.add(new PathFinding(new Point(PathList.get(parent).getX() -1, PathList.get(parent).getY()), parent, PathList.get(parent).getG(), 50, targetDestination));
	}

5	}
	if (!AWorld.isGround(PathList.get(parent).getX() - getImage().getHeight()/2, PathList.get(parent).getY() - getImage().getWidth()/2))
	{
	if (!check(parent, 70, new Point(-1, -1)))

No	Algoritma A*

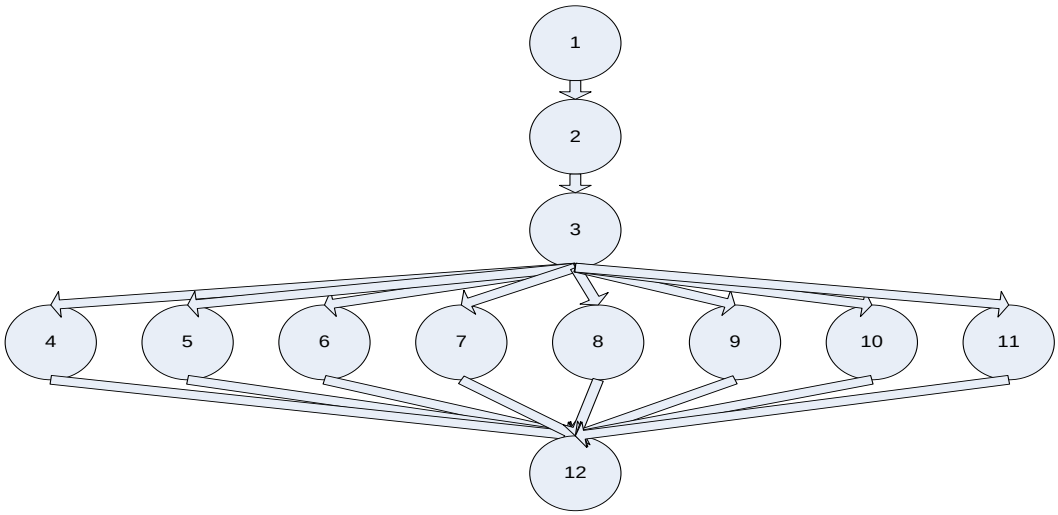
	{
	PathList.add(new PathFinding(new Point(PathList.get(parent).getX() -1, PathList.get(parent).getY() -1), parent, PathList.get(parent).getG(), 70, targetDestination));
	}

	}
6	if (!AWorld.isGround(PathList.get(parent).getX(), PathList.get(parent).getY() - getImage().getWidth()/2))
	{
	if (!check(parent, 50, new Point(0, -1)))
	{
	PathList.add(new PathFinding(new Point(PathList.get(parent).getX(), PathList.get(parent).getY() -1), parent, PathList.get(parent).getG(), 50, targetDestination));
	}
	}
7	if (!AWorld.isGround(PathList.get(parent).getX() + getImage().getHeight()/2, PathList.get(parent).getY() - getImage().getWidth()/2))
	{
	if (!check(parent, 70, new Point(1, -1)))
	{
	PathList.add(new PathFinding(new Point(PathList.get(parent).getX() +1, PathList.get(parent).getY() -1), parent, PathList.get(parent).getG(), 70, targetDestination));
	}
	}
8	if (!AWorld.isGround(PathList.get(parent).getX() + getImage().getHeight()/2, PathList.get(parent).getY()))
	{
	if (!check(parent, 50, new Point(1, 0)))
	{
	PathList.add(new PathFinding(new Point(PathList.get(parent).getX() +1, PathList.get(parent).getY()), parent, PathList.get(parent).getG(), 50, targetDestination));
	}
	}
9	if (!AWorld.isGround(PathList.get(parent).getX() + getImage().getHeight()/2, PathList.get(parent).getY() + getImage().getWidth()/2))
	{
	if (!check(parent, 70, new Point(1, 1)))
	{
	PathList.add(new PathFinding(new Point(PathList.get(parent).getX() +1,

No	Algoritma A*
	PathList.get(parent).getY() +1), parent, PathList.get(parent).getG(), 70, targetDestination));
	}
	}
10	if (!AWorld.isGround(PathList.get(parent).getX(), PathList.get(parent).getY() + getImage().getWidth()/2))
	{
	if (!check(parent, 50, new Point(0, 1)))
	{
	PathList.add(new PathFinding(new Point(PathList.get(parent).getX(), PathList.get(parent).getY() +1), parent, PathList.get(parent).getG(), 50, targetDestination));
	}
	}
11	if (!AWorld.isGround(PathList.get(parent).getX() - getImage().getHeight()/2, PathList.get(parent).getY() + getImage().getWidth()/2))
	{
	if (!check(parent, 70, new Point(-1, 1)))
	{
	PathList.add(new PathFinding(new Point(PathList.get(parent).getX() -1, PathList.get(parent).getY() +1), parent, PathList.get(parent).getG(), 70, targetDestination));
	}
	}
12	PathList.get(parent).close();

b. *Flow Graph*

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



Gambar IV.1 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 + 2

V (G) = 18-12 + 2

V (G) = 6 + 2

V(G) = 8

Dimana :

E = jumlah busur pada *flow graph*

N = jumlah simpul pada *flow graph*

d. *Independent Path*

Path 1 : 1,2,3,4,12

Path 2 : 1,2,3,5,12

Path 3 : 1,2,3,6,12

Path 4 : 1,2,3,7,12

Path 5 : 1,2,3,8,12

Path 6 : 1,2,3,9,12

Path 7 : 1,2,3,10,12

Path 8 : 1,2,3,11,12

e. *Graph Matrix*

Tabel IV.10 Graph *Matrix*

N	1	2	3	4	5	6	7	8	9	10	11	12	E
1		1											
2			1										
3				1	1	1	1	1	1	1	1		7
4													
5													
6													
7													
8													
9													
10													
11												1	
12													
Jumlah E+1													8

V(G) = Jumlah Graph Matrix + 1

V(G) = 7 + 1

V(G) = 8

Berdasarkan hasil pengujian dihasilkan nilai *Cyclomatic Complexity* yang sama yaitu 8, maka dapat disimpulkan bahwa pada proses pencarian dengan algoritma A* (A *star*) berjalan dengan baik, karena setiap pengujian menghasilkan nilai yang sama.

IV.2.5 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.

IV.2.6 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. Sampel yang diambil masing-masing calon pengguna sebanyak 40 orang, dari hasil kuesioner tersebut akan dilakukan perhitungan agar dapat diambil kesimpulan penilaian dari para responden.

Berikut ini adalah beberapa pertanyaan yang diajukan kepada calon pengguna dalam perangkat lunak yang dibangun adalah sebagai berikut :

Tabel IV.11 Pertanyaan Calon Pengguna

NO	PERTANYAAN	JAWABAN				
		SS	S	BS	KS	TS
1	Apa anda setuju aplikasi permainan ini menarik untuk dimainkan?	12	26	2	0	0
2	Apa anda setuju aplikasi permainan ini mempunyai tingkat kesulitan yang berbeda di setiap tingkat?	13	24	3	0	0
3	Apa anda setuju perilaku musuh dalam permainan yang dapat mengejar pemain pada aplikasi permainan ini menambah tantangan ketika dimainkan?	14	20	6	0	0
4	Apa anda setuju aplikasi permainan ini menambah pengetahuan anda tentang masakan nusantara indonesia?	13	21	6	0	0

5	Apakah aplikasi permainan ini dapat membantu anda belajar tentang memasak masakan nusantara indonesia?	7	22	11	0	0

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 adalah sebagai berikut :

Tabel IV.12 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

IV.2.7 Kesimpulan Pengujian Kuesioner

Kuesioner ini ditujukan kepada 40 orang responden. Hasil persentase kuesioner masing-masing jawaban yang sudah dihitung dengan menggunakan rumus :

$$Y = P/Q*100\%$$

Keterangan : Y : Nilai Prosentase

P : Total Skor

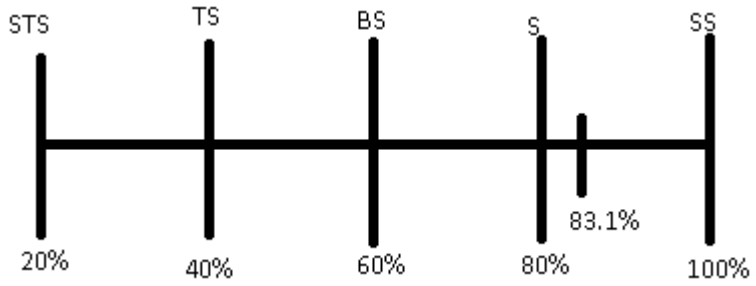
Q : Skor Tertinggi

Berikut ini adalah hasil prosentase masing-masing jawaban yang sudah dihitung nilainya dengan menggunakan rumus yang telah didefinisikan. Kuesioner ini diujikan kepada 40 orang calon pengguna. Oleh karena itu maka nilai Q adalah 40*5= 200.

Tabel IV.13 Hasil Pengujian Kuesioner

No	Pertanyaan	Skor	Frekuensi Jawaban Responden	Jumlah Skor	Presentase (%)
1	Apa anda setuju aplikasi permainan ini menarik untuk dimainkan?	5	12	60	170/200*100%=85%
		4	26	104	
		3	2	6	
		2	0	0	
		1	0	0	
	Jumlah		40	170	
2	Apa anda setuju aplikasi permainan ini mempunyai tingkat kesulitan yang berbeda di setiap tingkat?	5	13	65	170/200*100%=85%
		4	24	96	
		3	3	9	
		2	0	0	
		1	0	0	
	Jumlah		40	170	
3	Apa anda setuju perilaku musuh dalam permainan yang dapat mengejar pemain pada aplikasi permainan ini menambah tantangan ketika dimainkan?	5	14	70	168/200*100%=84%
		4	20	80	
		3	6	18	
		2	0	0	
		1	0	0	
	Jumlah		40	168	
4	Apa anda setuju aplikasi permainan ini menambah pengetahuan anda tentang masakan nusantara indonesia?	5	13	65	167/200*100%=83,5%
		4	21	84	
		3	6	18	
		2	0	0	
		1	0	0	
	Jumlah		40	167	
5	Apakah aplikasi permainan ini dapat membantu anda belajar tentang memasak masakan nusantara indonesia?	5	7	35	156/200*100%=78%
		4	22	88	
		3	11	33	
		2	0	0	
		1	0	0	
	Jumlah		40	156	

Hasil presentase dari semua pertanyaan yang diajukan kepada beberapa calon pengguna didapat rata-rata sebanyak 83,1%. Dengan demikian nilai perhitungan tersebut berarti pengguna setuju dengan permainan selamatkan masakan nusantara indonesia untuk dimainkan. Secara kontinum dapat digambarkan sebargai berikut:



Gambar IV.2 Hasil Pengujian Kuisioner