BAB 4

IMPLEMENTASI DAN PENGUJIAN SISTEM

Pada bab ini akan dilakukan implementasi dan pengujian terhadap aplikasi permainan yang dibangun, yaitu *Game* Edukasi Pertolongan Pertama. Hasil perancangan pada tahap sebelumnya kemudian diimplementasikan ke dalam bahasa pemrograman. Setelah tahap implementasi selesai, maka dilakukan tahap pengujian terhadap aplikasi *Game* Edukasi Pertolongan Pertama.

4.1 Implementasi

Tahap implementasi merupakan tahap penciptaan perangkat lunak dan juga tahap kelanjutan dari kegiatan perancangan aplikasi.Tahap ini merupakan tahap dimana aplikasi siap untuk dioperasikan, yang terdiri dari penjelasan mengenai lingkungan implementasi, baik itu lingkungan perangkat keras,maupun lingkungan perangkat lunak, serta implementasi program.

4.1.1 Implementasi Perangkat Keras

Perangkat keras yang digunakan untuk menjalankan aplikasi permainan *Game* Edukasi Pertolongan Pertama tidak harus yang berspesifikasi tinggi. Spesifikasi minimum yang dapat digunakan untuk menjalankan aplikasi permainan *Game* Edukasi Pertolongan Pertama adalah sebagai berikut :

1. *Processor dengan kecepatan 1.8 Ghz*
2. *RAM 1 GB*
3. *Hardisk* dengan ruang kosong 500 MB
4. *Vga Card 512MB*
5. *Monitor* (resolusi 1024 x 768)
6. *Mouse*
7. *Keyboard*
8. *Speaker*

4.1.2 Implementasi Perangkat Lunak

Adapun spesifikasi perangkat lunak yang digunakan untuk menjalankan aplikasi *Game* Edukasi Pertolongan Pertama adalah perangkat komputer yang telah terinstal *Flash Player.*

4.1.3 Implementasi Aplikasi

Aplikasi permainan ini menggunakan Flash dan teknik dengan grafik 2D yang dibangun dengan menggunakan bahasa pemrograman *Action Script 3.0.*

4.1.4 Implementasi *Class*

Implementasi *class* merupakan implementasi dari perancangan *class diagram* yang terbentuk dari *use case* diagram ke dalam bentuk file fisik dengan ekstensi ".as".

Tabel 4.1 Implementasi *class*

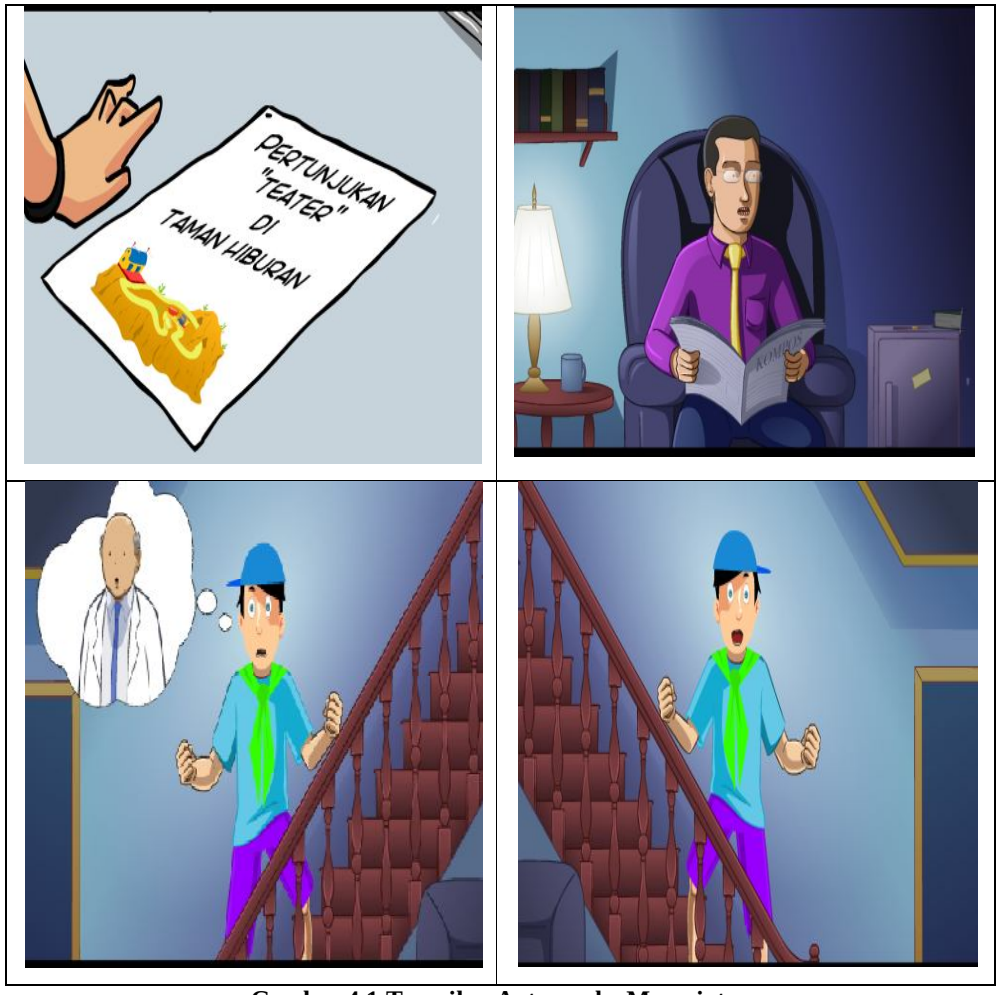
No	Nama *class*	Nama *class* fisik
1.	Ambulance	Ambulance.as
2.	Astar	Astar.as
3.	Benda	Benda.as
4.	Box	Box.as
5.	Cukur	Cukur.as
6.	Hud	Hud.as
7.	Hudquiz	Hudquiz.as
8.	Jangka	Jangka.as
9.	Korban	Korban.as
10.	Level	Level.as
11.	Main	Main.as
12.	*Node*s	*Node*s.as
13.	Palu	Palu.as
14.	Sounds	Sounds.as
15.	Suny	Suny.as
16.	Tandapanah	Tandapanah.as
17.	Tank	Tank.as
18.	TinySuny	TinySuny.as

4.1.5 Implementasi Antarmuka

Adapun implementasi antarmuka pada aplikasi *Game* Edukasi Pertolongan Pertama akan dijelaskan sebagai berikut :

4.1.5.1 Antarmuka Intro

Tujuan dari pembuatan antarmuka intro ini digunakan untuk memberikan gambaran alur cerita pada *Game* Pertolongan Pertama, sehingga pemain dapat mengetahui misi permainan yang akan dimainkan oleh pemain. Antarmuka intro dapat dilihat pada gambar 4.1 dibawah ini.



Gambar 4.1 Tampilan Antarmuka Menu intro

4.1.5.2 Antarmuka Menu Utama

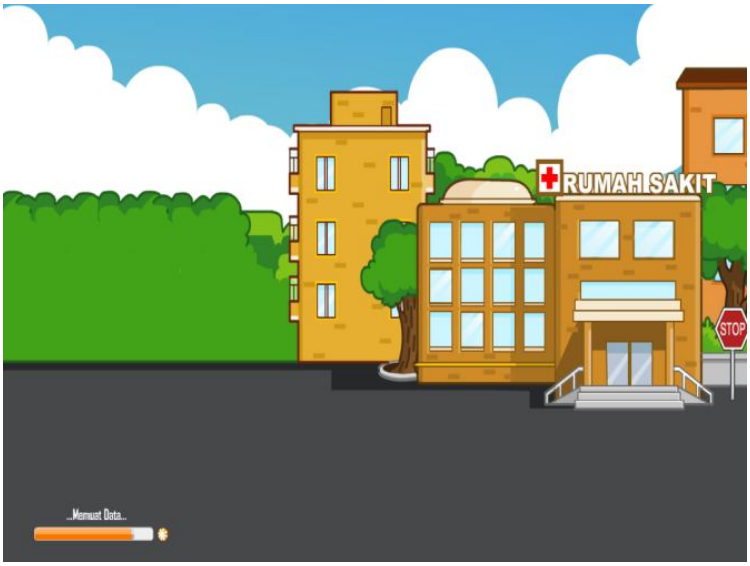
Setelah menu intro selesai tampil, maka antarmuka selanjutnya yang akan terlihat oleh pemain pertama sekali adalah menu utama seperti gambar 4.2 berikut ini :



Gambar 4.2 Antarmuka Menu Utama

4.1.5.3 Antarmuka Menu Memulai Permainan (Memuat Data)

Antarmuka pada menu ini digunakan untuk menamilkan tampilan memuat data.



Gambar 4.3 Antarmuka Memuat Data

4.1.5.4 Antarmuka Menu Permainan level 1

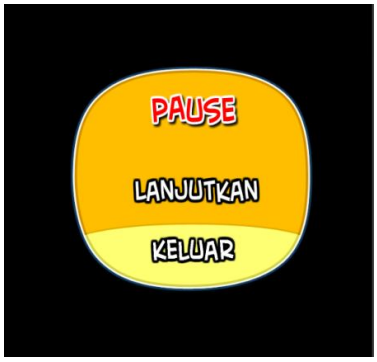
Gambar 4.4 - 4.9 tampilan Antarmuka dalam menu permainan yang dimainkan pemain pada level 1 adalah sebagai berikut :



Gambar 4.4 Antarmuka Menu Instuksi 1 (Level 1)



Gambar 4.5 Antarmuka Pengambilan Alat dan Informasi Alat yang Terambil (level 1)



Gambar 4.6 Antarmuka Menu Pause atau Jeda (level 1)



Gambar 4.7 Antarmuka Koin Informasi (level 1)



Gambar 4.8 Antarmuka Stage Pengobatan (level 1)



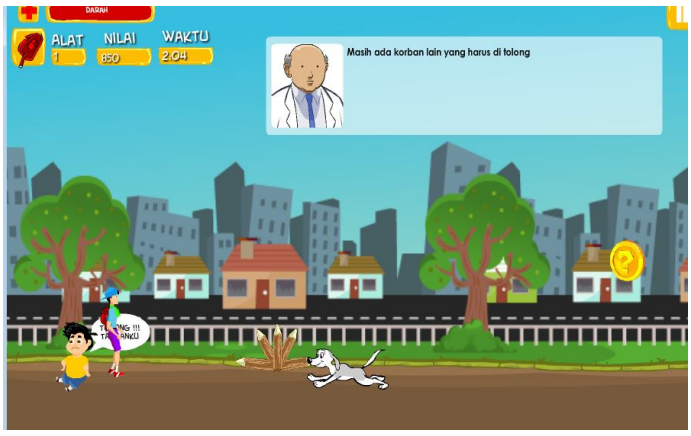
Gambar 4.9 Antarmuka Score (level 1)

4.1.5.5 Antarmuka Permainan Level 2

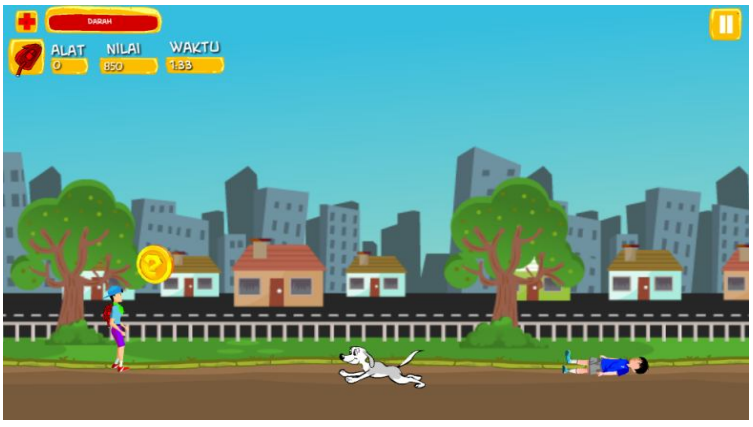
Tampilan gambar 4.10 – gambar 4.17 dibawah ini menggambarkan antarmuka yang ada pada *Game* pertolongan pertama.



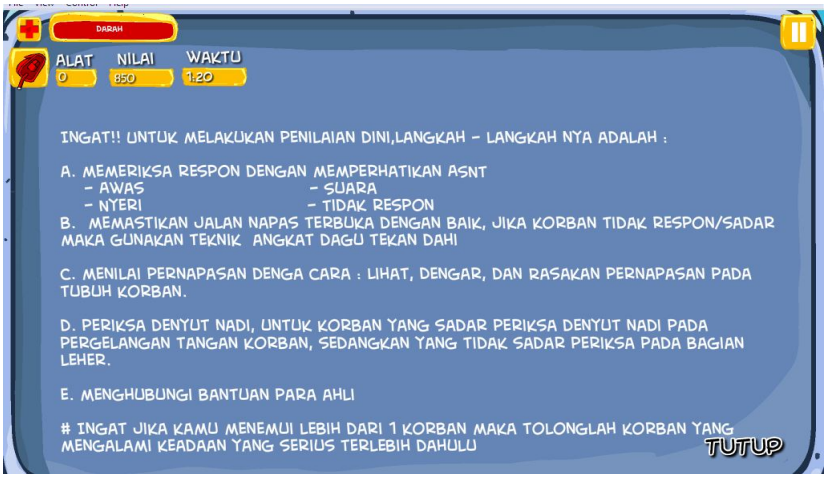
Gambar 4.10 Antarmuka Permainan Level 2



Gambar 4.11 Antarmuka permainan Level 2 – Saat Mengenai Korban Pertama



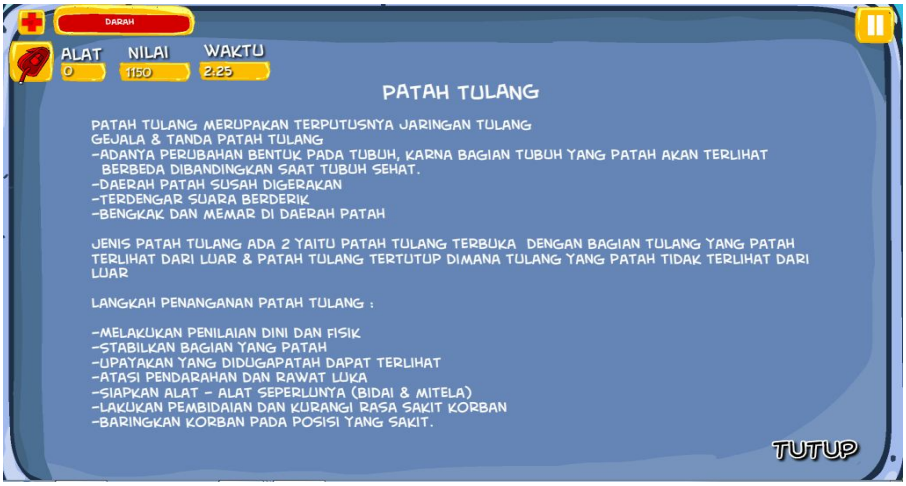
Gambar 4.12 Antarmuka Permainan Level 2 – Mengambil Koin Informasi



Gambar 4.13 Antarmuka Permainan Level 2 – Tampilan isi Koin Informasi



Gambar 4.14 Antarmuka Permainan Level 2 – Stage 1 Pengobatan Korban ke-1



Gambar 4.15 Antarmuka permainan Level 2 – Koin Informasi ke-2



Gambar 4.16 Antarmuka Permainan Level 2 – Stage 1 Pengobatan Korban ke-2



Gambar 4.17 Antarmuka Permainan Level 2 – Stage 5 Pengobatan Korban ke - 2

4.1.5.6 Antarmuka Permainan Level 3

Tampilan gambar 4.18 – gambar 4.24 dibawah ini menggambarkan antarmuka yang ada pada *Game* pertolongan pertama



Gambar 4.18 Antarmuka permainan Level 3 – Intruksi Permainan



Gambar 4.19 Antarmuka Permainan Level 3



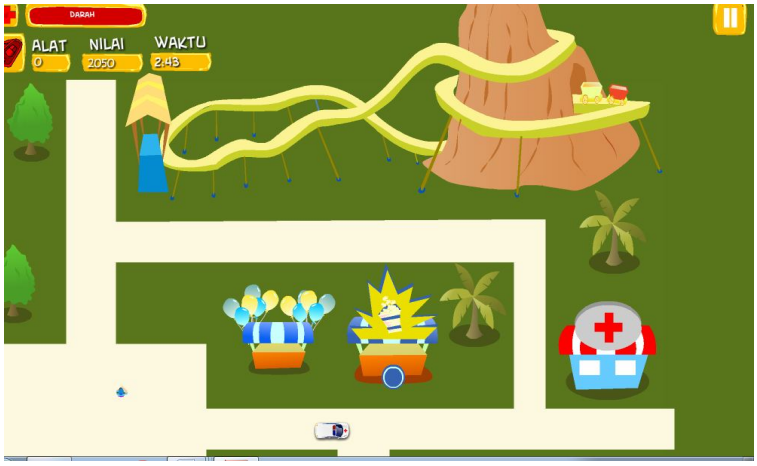
Gambar 4.20 Antarmuka Permainan Level 3 – Pengambilan Peta



Gambar 4.21 Antarmuka Permainan Level 3 – Stage Menghubungi Bantuan



Gambar 4.22 Antarmuka Permainan Level 3 – Bantuan Ambulance Jemput Korban



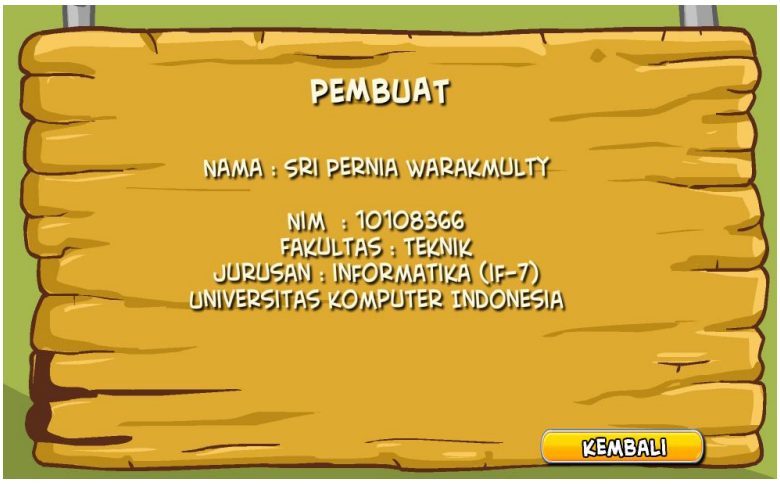
Gambar 4.23 Antarmuka Permainan Level 3 – Ambulance Kembali Kerumah Sakit Membawa Korban



Gambar 4.24 Antarmuka Level Selesai dan Input Nama

4.1.5.7 Antarmuka Menu Pembuat

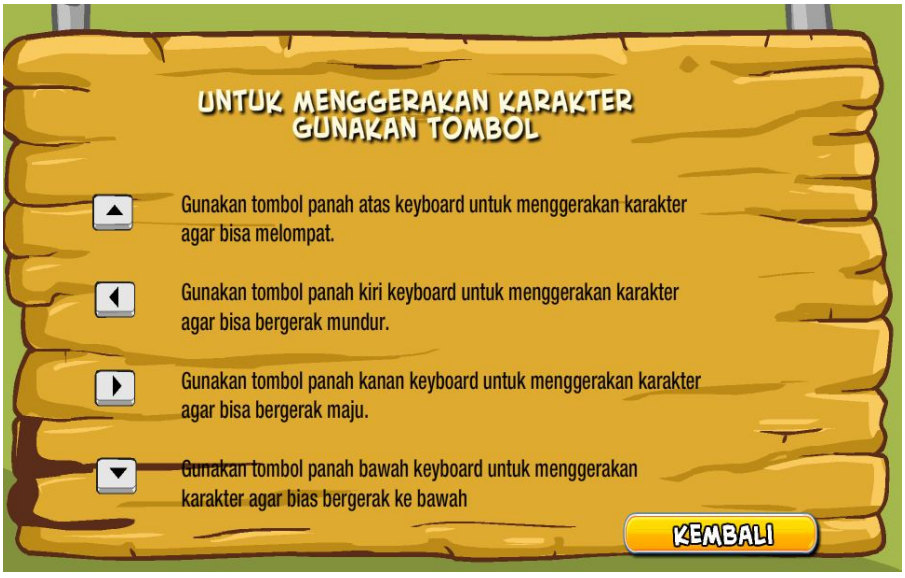
Berikut adalah tampilan antarmuka menu pembuat yang berada dimenu utama permainan



Gambar 4.25 Antarmuka Menu Pembuat

4.1.5.8 Antarmuka Menu Kontrol Permainan

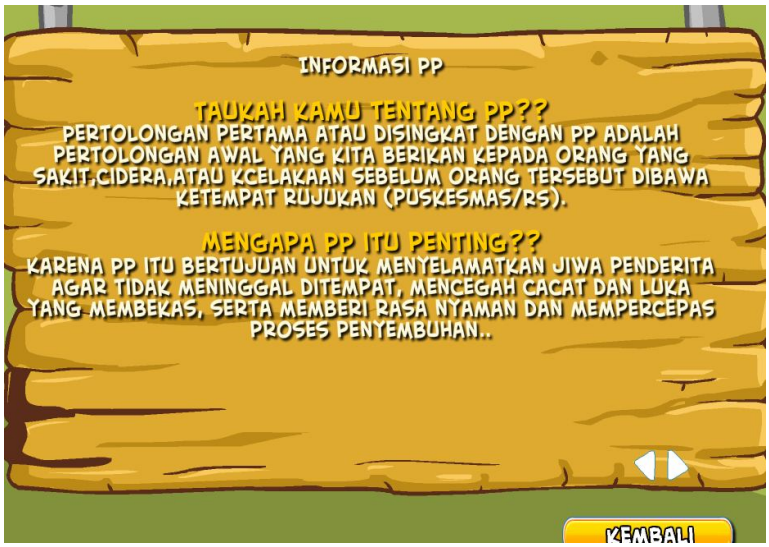
Berikut adalah tampilan antarmuka menu control Permainan yang berada dimenu utama permainan



Gambar 4.26 Antarmuka Menu Kontrol Permainan

4.1.5.9 Antarmuka Menu Informasi PP

Berikut adalah tampilan antarmuka menu Informasi PP yang berada dimenu utama permainan



Gambar 4.27 Antarmuka Menu Informasi PP

4.1.5.10 Antarmuka Menu Nilai Tertinggi

Berikut adalah tampilan antarmuka menu nilai tertinggi yang berada dimenu utama permainan



Gambar 4.28 Antarmuka Menu Nilai Tertinggi

4.1.5.11 Antarmuka Menu Keluar (aplikasi)

Berikut adalah tampilan antarmuka menu pesan keluar yang berada dimenu utama permainan



Gambar 4.29 Antarmuka Menu Keluar (aplikasi)

4.2 Pengujian

Pengujian terhadap aplikasi *Game* Edukasi Pertolongan Pertama dilakukan dengan dua macam teknik, yaitu pengujian *black-box* dan 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 dengan dua tahap, yaitu tahap pengujian *alpha* dan tahap pengujian *bet*a.

4.2.1 Pengujian *alpha*

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

4.2.2 Skenario pengujian Aplikasi

Skenario pengujian aplikasi diperlukan agar dalam pengujian terlihat jelas rincian tujuan utama dari pengujian, seingga hasil yang didapatkan dari pengujian menjadi jelas dan mudah dipahami.

4.2.3 Pengujian *Black Box*

Pengujian dilakukan untuk memastikan aplikasi *Game* Edukasi Pertolongan Pertama telah berjalan dengan benar dan sesuai dengan kebutuhan yang diinginkan. Pengujian *blackbox* menitik beratkan kepada persyaratan fungsional perangkat lunak.

Rencana pengujian yang akan dilakukan pada pengujian *Black Box* ini selengkapnya terlihat pada tabel 4.2 berikut ini

Tabel 4.2Tabel Skenario Pengujian Aplikasi

No	Kelas Uji	Detail Pengujian	Jenis Uji
1	*Skip* intro	Men*skip* /melewatkan intro	*Black Box*
2	Permainan	Memulai permainan	*Black Box*
3	Informasi pembuat	Menampilkan tampilan petunjuk pembuat	*Black Box*
4	informasi PP	Menampilkan tampilan petunjuk informasi	*Black Box*
5	kontrol permainan	Menampilkan tampilan petunjuk control permainan	*Black Box*
6	informasi Nilai Tertinggi	Menampilkan tampilan petunjuk nilai tertinggi permainan	*Black Box*
7	Karakter bergerak	Pergerakan karakter dengan mengunakan masukan dari keyboard	*Black Box*
8	Pengambilan alat	Menampilkan pengambilan alat	*Black Box*
9	Pengambilan dan penolongan korban	Menampilkan proses pengobatan korban	*Black Box*
10	Pengambilan koin	Menampilkan informasi isi koin	*Black Box*
11	Pengambilan Peta	Menampilkan proses pengambilan peta	*Black Box*
12	Pegerakan ambulance	Menampilkan pegerakan ambulance	*Black Box*
13	Input nama *high score* / nilai tertinggi	Penginputan nama untuk nilai tertinggi	*Black Box*
14	Pause	Meampilkan pilihan menu tombol pause	*Black Box*
15	Keluar	Memilih tombol menu utama – keluar	*Black Box*

4.2.3.1 Kasus dan Hasil Pengujian (*Black Box*)

Berikut ini merupakan kasus untuk menguji perangkat lunak yang sudah dibangun dengan menggunakan metode *black box* berdasarkan skenario pengujian aplikasi yang terdapat pada Tabel 4.1 yang sudah dijabarkan. Hasil pengujian yang akan dilakukan pada aplikasi ini selengkapnya dapat dilihat pada Tabel 4.3 hingga tabel 4.17 berikuti ini.

1. *Skip* intro

Tabel 4.3 Hasil Pengujian Skip Intro

Data Masukan	Yang diharapkan	Pengamatan	Hasil pengujian
Pilih tombol *Skip*	Meloncat tampilan intro *Game*	Tampilan langsung menuju menu utama	[√] Berhasil []Tidak berhasil

2. Permainan

Tabel 4.4 Hasil Pengujian Permainan

Data Masukan	Yang diharapkan	Pengamatan	Hasil pengujian
Pilih button menu mulai permainan	Menampilkan loading data	Menampilkan loading data dan halaman intruksi	[√] Berhasil []Tidak berhasil
Pilih button panah / main	Menampilkan halaman intruksi selanjutnya dan arena permainan	Menampilkan halaman instruksi dan arena permainan	[√] Berhasil []Tidak berhasil

3. Informasi Pembuat

Tabel 4.5 Hasil Pengujian Informasi Pembuat

Data Masukan	Yang diharapkan	Pengamatan	Hasil pengujian
Pilih button menu informasi Pembuat	Tampilan halaman informasi pembuat	Menampilkan halaman informasi pembuat	[√] Berhasil []Tidak berhasil
Pilih button keluar	Tampilan halaman kembali ke menu utama	Menampilkan halaman menu utama	[√] Berhasil []Tidak berhasil

4. Informasi PP

Tabel 4.6 Hasil Pengujian Informasi PP

Data Masukan	Yang diharapkan	Pengamatan	Hasil pengujian
Pilih button menu informasi PP	Tampilan halaman informasi PP	Menampilkan halaman informasi PP	[√] Berhasil []Tidak berhasil
Pilih button keluar	Tampilan halaman kembali ke menu utama	Menampilkan halaman menu utama	[√] Berhasil []Tidak berhasil

5. Kontrol Permainan

Tabel 4.7 Hasil Pengujian Kontrol Permainan

Data Masukan	Yang diharapkan	Pengamatan	Hasil pengujian
Pilih button menu kontrol permainan	Tampilan halaman informasi kontrol permainan	Menampilkan halaman informasi kontrol permainan	[√] Berhasil []Tidak berhasil
Pilih button keluar	Tampilan halaman kembali ke menu utama	Menampilkan halaman menu utama	[√] Berhasil []Tidak berhasil

6. Informasi Nilai Tertinggi

Tabel 4.8 Hasil Pengujian Nilai Tertinggi

Data Masukan	Yang diharapkan	Pengamatan	Hasil pengujian
Pilih button menu nilai Tertinggi	Tampilan halaman informasi nilai tertinggi	Menampilkan halaman informasi nilai tertinggi	[√] Berhasil []Tidak berhasil
Pilih button keluar	Tampilan halaman kembali ke menu utama	Menampilkan halaman menu utama	[√] Berhasil []Tidak berhasil

7. Karakter Bergerak

Tabel 4.9 Hasil Pengujian Karakter Bergerak

Data Masukan	Yang diharapkan	Pengamatan	Hasil pengujian
Input simbol keyboard ↑	Karakter bergerak lompat / berjalan ke arah atas.	Karakter bergerak lompat / berjalan ke atas	[√] Berhasil []Tidak berhasil
Input simbol keyboard ↓	Karakter berjalan ke arah bawah.	Karakter bergerak ke bawah	[√] Berhasil []Tidak berhasil
Input simbol keyboard ←	Karakter bergerak mundur atau berjalan kekiri	Karakter bergerak mundur / berjalan ke kiri	[√] Berhasil []Tidak berhasil
Input simbol keyboard →	Karakter bergerak maju atau berjalan ke kanan	Karakter bergerak maju atau berjalan ke kanan	[√] Berhasil []Tidak berhasil

8. Pengambilan alat

Tabel 4.10 Pngambilan Alat

Data Masukan	Yang diharapkan	Pengamatan	Hasil pengujian
Menggerakan Karakter untuk mendekati alat	Tampilan alat menghilang	Menampilkan alat hilang	[√] Berhasil []Tidak berhasil
	Tampilan informasi alat yang diambil	Menampilkan informasi alat	[√] Berhasil []Tidak berhasil

9. Pengambilan dan penolongan korban

Tabel 4.11 Hasil Pengujian Pengambilan dan Penolongan Korban

Data Masukan	Yang diharapkan	Pengamatan	Hasil pengujian
Mengerakan karakter untuk mendekati korban	Tampilan halaman pengobatan	Menampilkan halaman pengobatan	[√] Berhasil []Tidak berhasil
Memilih button jawaban quiz yang benar	Tampilan proses animasi penolongan yang benar	Menampilkan proses pertolongan	[√] Berhasil []Tidak berhasil

10. Pengambilan koin

Tabel 4.12 Hasil Pengujian Pengambilan Koin

Data Masukan	Yang diharapkan	Pengamatan	Hasil pengujian
Mengerakan karakter untuk mendekati koin	Tampilan halaman informasi koin	Menampilkan halaman informasi koin	[√] Berhasil []Tidak berhasil
Pilih button keluar	Tampilan halaman kembali ke halaman permainan	Menampilkan halaman permainan	[√] Berhasil []Tidak berhasil

11. Pengambilan peta

Tabel 4.13 Hasil Pengujian Pengambilan Peta

Data Masukan	Yang diharapkan	Pengamatan	Hasil pengujian
Mengerakan karakter untuk mendekati peta	Tampilan potonan peta hilang	Potongan peta hilang	[√] Berhasil []Tidak berhasil

12. Pergerakan Ambulance

Tabel 4.14 Hasil Pengujian Pegerakan Ambulance

Data Masukan	Yang diharapkan	Pengamatan	Hasil pengujian
Mengerakan karakter untuk mendekati korban	Tampilan animasi penjemputan korban olh ambulance	Pegerakan ambulance menjemput korban	[√] Berhasil []Tidak berhasil

13. Input nama *high score* / nilai Tertinggi

Tabel 4.15 Hasil Pengujian Input Nama high score / nilai Tertinggi

Data Masukan	Yang diharapkan	Pengamatan	Hasil pengujian
Menginputkan nama dengan meneka alphabet pada keyboard	Tampilan nama inputan	Tampilan nama inputan	[√] Berhasil []Tidak berhasil

14. Pause / jeda Permainan

Tabel 4.16 Hasil Pengujian pause / jeda Permainan

Data Masukan	Yang diharapkan	Pengamatan	Hasil pengujian
Memilih button pause (‖)	Tampilan menu Pause	Tampilan Menu pause	[√] Berhasil []Tidak berhasil
Memilih button lanjutkan	Tampilan halaman permainan	Kembali ke halaman permainan	[√] Berhasil []Tidak berhasil
Memilih button keluar	Tampilan menu utama	Menuju menu utama	[√] Berhasil []Tidak berhasil

15. Keluar

Tabel 4.17 Hasil Pengujian Keluar

Data Masukan	Yang diharapkan	Pengamatan	Hasil pengujian
Memilih button menu keluar	Tampil informasi pesan yakin keluar, ya atau tidak	Tampil pilihan ya atau tidak	[√] Berhasil []Tidak berhasil
Memilih button tidak	Tampilan halaman menu utama	Kembali ke halaman menu utama	[√] Berhasil []Tidak berhasil
Memilih button ya	Tampilan menutup aplikasi	Menutup aplikasi	[√] Berhasil []Tidak berhasil

4.2.3.2 Pengujian White Box

Berikut ini adalah kasus menguji perangkat lunak yang telah dibangun menggunakan metode *white box.* Pengujian *whitebox* menggunakan *flow graph* yang digunakan untuk menggambarkan alur dari algoritma Astar yang diterapkan pada ambulance dalam *Game* Pertolongan Pertama dan *grap matrix* yang digunakan untuk menggenerasi *flow graph*. Adapun pengujian *white-box* adalah sebagai berikut.

1. **Pengujian algoritma Astar**

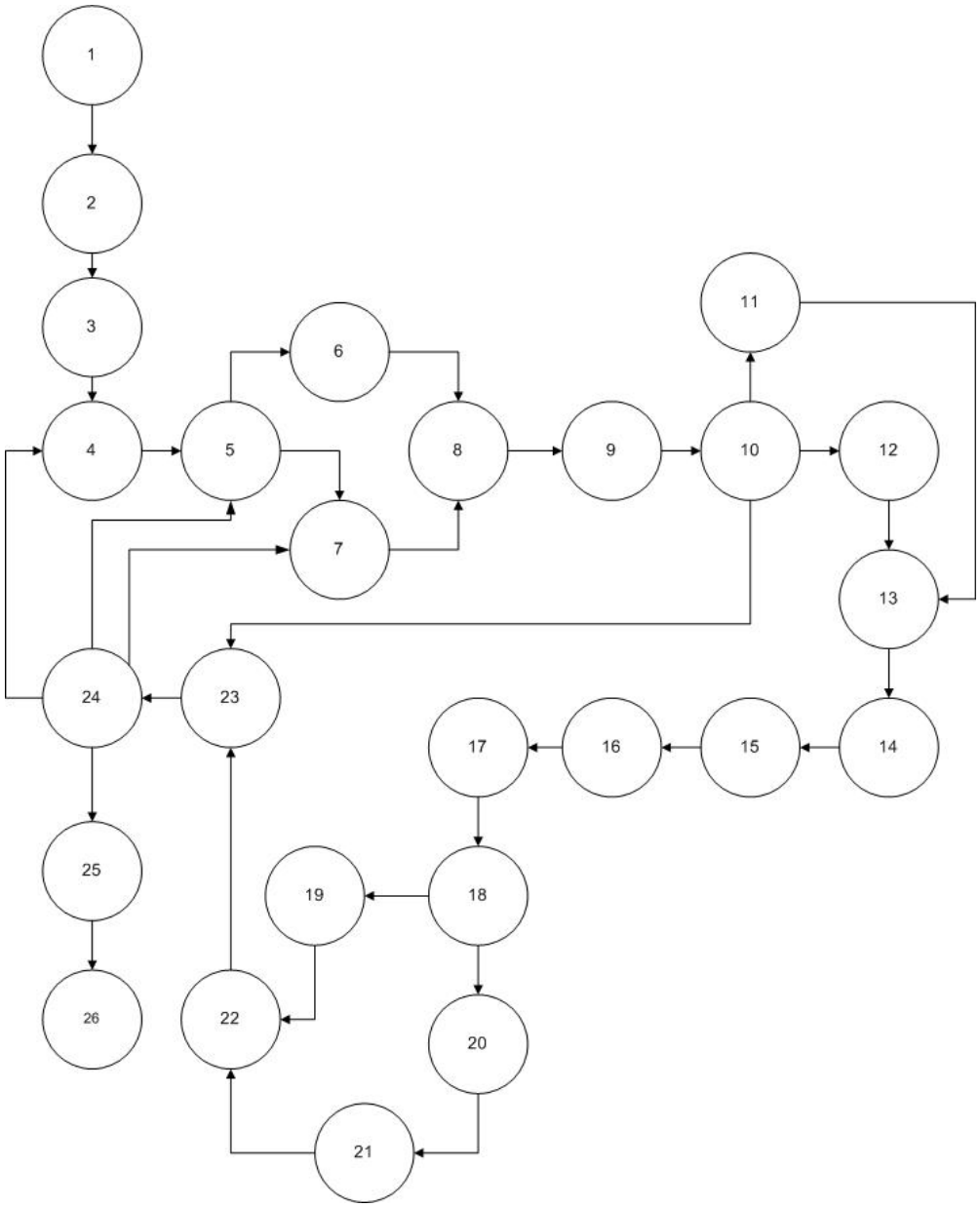
Pengujian algoritma Astar merupakan proses dimana akan diuji implementasi Astar terhadap NPC atau ambulance dalam menentukan jalur tercepat untuk mencapai letak posisi korban yang akan dijemput.

Tabel 4.18 Pseudocode Astar

No	Pseudocode
1	function pathFinder()
2	{
3	var start*Node*:*Node*s = varHolder[cur];
	start*Node*.g = 0;
	start*Node*.f = 0;
	push(start*Node*);
	start*Node*.opened = true;
	var tahap = 0;
4	while (opened.length > 0)
5	{
	var gh,gg;
	var *node* = pop();
	node.closes = true;
6	if (*node*.id == des)
7	{
	return backtrace(des);
	}
8	getTetangga(*node*.id);
	var ttgs = *node*.tetangga;
9	for (var d = 0; d < ttgs.length; d++)
10	{

11	var ttg = ttgs[d];
	if (ttg.closes)
12	{
	continue;
	}
13	var cost:Number = (ttg.xs - *node*.xs === 0 || ttg.ys - *node*.ys === 0 ? 1:Math.SQRT2);
14	var ng:Number = *node*.g + cost;
15	if (! ttg.opened || ng < ttg.g)
16	{
17	ttg.g = ng;
	ttg.h = ttg.h || heuristic(Math.abs(ttg.xs - endN.xs),Math.abs(ttg.ys - endN.ys));
	ttg.f = ttg.g + ttg.h;
	ttg.par = *node*;
18	if (! ttg.opened)
19	{
	push(ttg);
	ttg.opened = true;
	}
20	Else
21	{
	listUpdate(ttg);
	}
22	}
23	}
24	}
25	return [];
26	}

2. ***Flowgraph***



Gambar 4.30 Flowgraph

Keterangan :

= Menggambarkan Kondisi (N)

= Mengambarkan Aksi (E)

3. ***Cyclomatic complexity* (VG)**

   Dari gambar 4.43 diatas dapat dihitung nilai *Cyclomatic complexity (V*G) menggunakan rumus 2.3 dengan perhitungan sebagai berikut :

   E = 30 N = 26

   V(G) = E – N + 2

   V(G) = 32 – 26 + 2

   V(G) = (6) + 2 = 8

4. ***Independent path***

   Path 1 =1-2-3-4-5-6-8-9-10-11-13-14-15-16-17-18-19-22-23-24-25-26

   Path 2 =1-2-3-4-5-6-8-9-10-11-13-14-15-16-17-18-20-21-22-23-24-25-26

   Path 3 =1-2-3-4-5-6-8-9-10-12-13-14-15-16-17-18-20-21-22-23-24-25-26

   Path 4 =1-2-3-4-5-7-8-9-10-12-13-14-15-16-17-18-20-21-22-23-24-25-26

   Path 5 =1-2-3-4-5-7-8-9-10-12-13-14-15-16-17-18-20-21-22-23-24-5

   Path 6 = 1-2-3-4-5-6-8-9-10-12-13-14-15-16-17-18-20-21-22-23-24- 4

   Path 7 =1-2-3-4-5-7-8-9-10-12-13-14-15-16-17-18-20-21-22-23-24-7

   Path 8 = 1-2-3-4-5-7-8-9-10-23-24-25-26

5. **Graph Matriks**

	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	0
1		1																									0
2			1																								0
3				1																							0
4					1																						0
5						1	1																				1
6								1																			0
7								1																			0
8									1																		0
9										1																	0
10											1	1											1				2
11													1														0
12													1														0
13														1													0
14															1												0
15																1											0
16																	1										0
17																		1									0
18																			1	1							1
19																						1					0
20																					1						0
21																						1					0
22																							1				0
23																								1			0
24				1	1		1																		1		3
25																										1	0
26																											
																									hasil		7

V(G) = jumlah Graph Matrik + 1

V(G) = 7+1

V(G)=8

6. Kesimpulan

Setelah dilakukan pengujianterhadapan penerapan algoritma A* maka hasil yang didapat dapat dilihat pada tabel 4.19 berikut :

Tabel 4.19 Hasil Uji White Box Penerapan Algoritma A*

Path No	*Node*(n)	Hasil yang Diharapkan	Hasil Sesuai Uji Kasus	Keterangan
1	1-2-3-4-5-6-8-9-10-11-13-14-15-16-17-18-19-22-23-24-25-26	Menghasilkan *rute* terpendek dari A*	Menghasilkan *rute* terpendek dari A*	[√] alur terlewati [ ] alur tidak terlewati
2	1-2-3-4-5-6-8-9-10-11-13-14-15-16-17-18-20-21-22-23-24-25- 26	Memeriksa *node* tetangga dari *node* yang diperiksa	Memeriksa *node* tetangga dari *node* yang diperiksa	[√] alur terlewati [ ] alur tidak terlewati
3	1-2-3-4-5-6-8-9-10-12-13-14-15-16-17-18-20-21-22-23-24- 25-26	Memeriksa apakah *node* tetangga dari *node* yang diperiksa masuk pada *closed*	Memeriksa apakah *node* tetangga dari *node* yang diperiksa masuk pada *closed*	[√] alur terlewati [ ] alur tidak terlewati
4	1-2-3-4-5-7-8-9-10-12-13-14-15-16-17-18-20-21-22-23-24-25-26	Memeriksa F(n) terkecil pada setiap *node* tetangga dari *node* yang diperiksa, sehingga menghasilkan dugaan sementara terhadap *node* yang terpilih dari *node* tetangga untuk	Memeriksa F(n) terkecil pada setiap *node* tetangga dari *node* yang diperiksa, sehingga menghasilkan dugaan sementara terhadap *node* yang terpilih dari	[√] alur terlewati [ ] alur tidak terlewati

		dijadikan *node* yang selanjutnya akan menjadi *parent*	*node* tetangga untuk dijadikan *node* yang selanjutnya akan menjadi *parent*	
5	1-2-3-4-5-7-8-9-10-12-13-14-15-16-17-18-20-21-22-23-24-5	Memeriksa *openlist*	Memeriksa *openlist*	[√] alur terlewati [ ] alur tidak terlewati
6	1-2-3-4-5-6-8-9-10-12-13-14-15-16-17-18-20-21-22-23-24- 4	Memeriksa atau membandingan jarak terkecil dari *node* tetangga yang telah diperiksa dan menjadi nilai terkecil dengan *node* yang perada pada *openlist*	Memeriksa atau membandingan jarak terkecil dari *node* tetangga yang telah diperiksa dan menjadi nilai terkecil dengan *node* yang perada pada *openlist*	[√] alur terlewati [ ] alur tidak terlewati
7	1-2-3-4-5-7-8-9-10-12-13-14-15-16-17-18-20-21-22-23-24-7	Memasukan *node* yang tidak terpakai kedalam *openlist*	Memasukan *node* yang tidak terpakai kedalam *openlist*	[√] alur terlewati [ ] alur tidak terlewati
8	1-2-3-4-5-7-8-9-10-23-24-25-26	*Node* yang diperiksa sama dengan *node* tujuan	*Node* yang diperiksa sama dengan *node* tujuan	[√] alur terlewati [ ] alur tidak terlewati

Berdasarkan tabel diatas maka dapat disimpulkan hasil uji kasus setiap *node* pada *independent pathA** berjalan sesuai hasil yang diharapkan yaitu semua alur sudah terlewati 1 kali. Serta Berdasarkan hasil pengujian dihasilkan nilai cyclomatic complexity yang sama yaitu 8, maka dapat disimpulkan bahwa proses algoritma A star berjalan denga baik, karena menghasilkan nilai yang sama.

4.2.3.3 Kesimpulan Pengujian Alpha

Dari hasil pengujian yang dilakukan, dapat disimpulkan bahwa sistem sudah dapat berjalan sesuai dengan yang diharapkan beserta proses penanganan kesalahannya. Jadi, secara fungsional sistem sudah dapat menghasilkan keluaran yang diharapkan

4.2.4 Pengujian Beta

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

Pengisian kuesioner dilakukan untuk mengetahui ketertarikan siswa terhadap *Game* yang dibangun. Pembuatan kuesioner menggunakan *metode* skala *likert*, dimana masing-masing pertanyaan mempunyai pilihan sangat setuju, setuju, biasa saja, tidak setuju, dan sangat tidak setuju. Masing-masing pilihan tersebut diberi poin (5 sampai 1). Berikut ini adalah daftar pertanyaan *Kuesioner* yang diberikan, serta hasil yang didapat setelah nilainya dihitung.

Untuk mencari nilai persentase dari masing-masing jawaban kuesioner digunakan rumus skala *likert* sebagai berikut :

P = (S/Skorideal) x 100%

Keterangan :

P : Nilai persentase yang dicari

S : Jumlah frekuensi dikalikan dengan nilai yang ditetapkan jawaban

Skorideal : Nilai tertinggi dikalikan dengan jumlah sampel (dimana jumlah sampel yang digunakan pada hasil kuesioner pertolongan pertama ini sejumlah 30 sampel yang berarti nilai tertinggi = 5 x 30 = 150)

Berikut ini adalah hasil persentase masing-masing jawaban yang sudah dihitung nilainya dengan menggunakan rumus diatas.

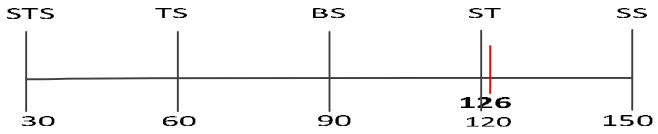
1. Apakah *Game* ini memberikan anda informasi mengenai pertolongan pertama?

Tabel 4.20 Perhitungan Persentase Pertanyaan ke-1

Pertanyaan	Keterangan	Skor	Frekuensi	S
1	Sangat Setuju (SS)	5	9	45
	Setuju (S)	4	18	72
	Biasa Saja (BS)	3	3	9
	Tidak Setuju (TS)	2	0	0
	Sangat Tidak Setuju (STS)	1	0	0
Jumlah			30	126

P = (126/150) x100% = 84%

Dapat disimpulkan bahwa *Game* yang dibangun dapat memberikan informasi mengenai perolongan pertama, terhitung 84% dari yang diharapkan 100%, Secara kontinum dapat digambarkan seperti berikut :



Jadi berdasarkan data yang diperoleh dari 30 responden maka rata – rata 126 terletak pada daerah setuju.

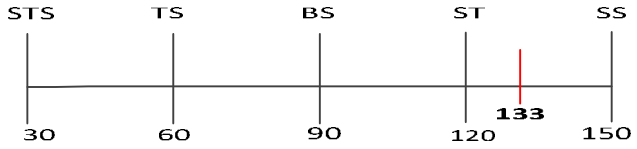
2. Apakah *Game* ini memberikan anda informasi mengenai peralatan yang dapat digunakan saat melakukan pertolongan pertama?

Tabel 4.21 Perhitungan Persentase Pertanyaan ke-2

Pertanyaan	Keterangan	Skor	Frekuensi	S
2	Sangat Setuju (SS)	5	18	90
	Setuju (S)	4	7	28
	Biasa Saja (BS)	3	5	15
	Tidak Setuju (TS)	2	0	0
	Sangat Tidak Setuju (STS)	1	0	0
Jumlah			30	133

P = (133/150) x100% = 88,6%

Dapat disimpulkan bahwa *Game* yang dibangun dapat memberikan informasi mengenai peralatan yang dapat digunakan saat melakukan pertolongan pertama, terhitung 88,6% dari yang diharapkan 100%, Secara kontinum dapat digambarkan seperti berikut :



Jadi berdasarkan data yang diperoleh dari 30 responden maka rata – rata 133 terletak pada daerah sangat setuju.

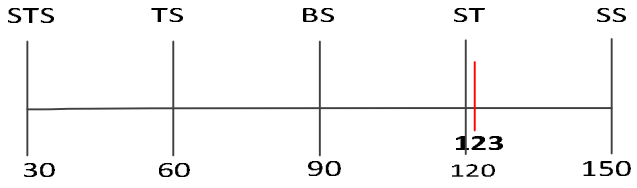
3. Apakah *Game* pertolongan pertama memberikan anda informasi mengenai macam - macam luka?

Tabel 4.22 Perhitungan Persentase Pertanyaan ke-3

Pertanyaan	Keterangan	Skor	Frekuensi	S
3	Sangat Setuju (SS)	5	10	50
	Setuju (S)	4	13	52
	Biasa Saja (BS)	3	7	21
	Tidak Setuju (TS)	2	0	0
	Sangat Tidak Setuju (STS)	1	0	0
Jumlah			30	123

P = (123/150) x100% = 82%

Dapat disimpulkan bahwa *Game* yang dibangun dapat memberikan informasi mengenai macam – macam luka, terhitung 82% dari yang diharapkan 100%, Secara kontinum dapat digambarkan seperti berikut :



Jadi berdasarkan data yang diperoleh dari 30 responden maka rata – rata 123 terletak pada daerah setuju.

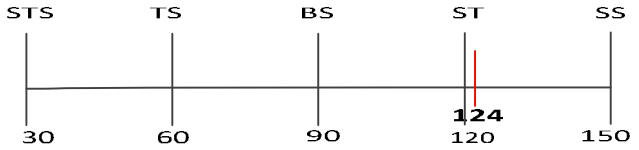
4. Apakah *Game* pertolongan pertama memberikan anda informasi mengenai penangganan luka?

Tabel 4.23 Perhitungan Persentase Pertanyaan ke-4

Pertanyaan	Keterangan	Skor	Frekuensi	S
4	Sangat Setuju (SS)	5	10	50
	Setuju (S)	4	14	56
	Biasa Saja (BS)	3	6	18
	Tidak Setuju (TS)	2	0	0
	Sangat Tidak Setuju (STS)	1	0	0
Jumlah			30	124

P = (124/150)x100% = 82,6%

Dapat disimpulkan bahwa *Game* yang dibangun dapat memberikan informasi mengenai penangganan luka, terhitung 82,6% dari yang diharapkan 100%, Secara kontinum dapat digambarkan seperti berikut :



Jadi berdasarkan data yang diperoleh dari 30 responden maka rata – rata 124 terletak pada daerah setuju.

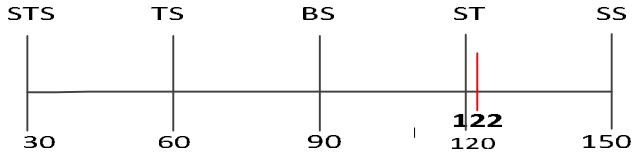
5. Apakah penunjuk permainan dalam *Game* ini, memudahkan anda selama memainkan permainan?

Tabel 4.22Perhitungan persentase pertanyaan ke-5

Pertanyaan	Keterangan	Skor	Frekuensi	S
5	Sangat Setuju (SS)	5	10	50
	Setuju (S)	4	12	48
	Biasa Saja (BS)	3	8	24
	Tidak Setuju (TS)	2	0	0
	Sangat Tidak Setuju (STS)	1	0	0
Jumlah			30	122

P = (122/150)x100% = 81.3%

Dapat disimpulkan bahwa *Game* yang dibangun dapat memberikan informasi mengenai macam – macam luka, terhitung 81.3 % dari yang diharapkan 100%, Secara kontinum dapat digambarkan seperti berikut :



Jadi berdasarkan data yang diperoleh dari 30 responden maka rata – rata 122 terletak pada daerah setuju.

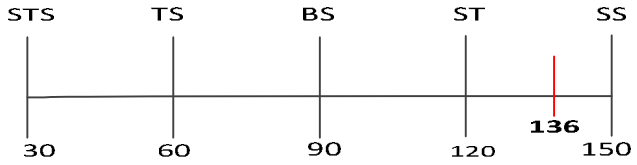
6. Apakah anda tertarik untuk memainkan *Game* pertolongan pertama ini lagi?

Tabel 4.24 Perhitungan persentase pertanyaan ke-6

Pertanyaan	Keterangan	Skor	Frekuensi	S
6	Sangat Setuju (SS)	5	18	90
	Setuju (S)	4	10	40
	Biasa Saja (BS)	3	2	6
	Tidak Setuju (TS)	2	0	0
	Sangat Tidak Setuju (STS)	1	0	0
Jumlah			30	136

P = (136/150)x100% = 90,6 %

Dapat disimpulkan bahwa *Game* yang dibangun dapat menarik minat pengguna untuk memainkan *Game* ini lagi, terhitung 90,6% dari yang diharapkan 100%, Secara kontinum dapat digambarkan seperti berikut :



Jadi berdasarkan data yang diperoleh dari 30 responden maka rata – rata 136 terletak pada daerah sangat setuju.

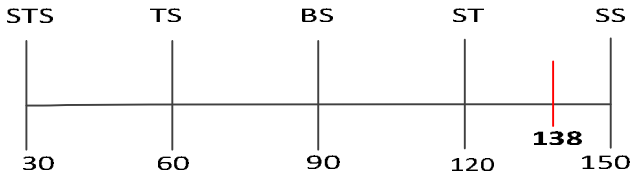
7. Apakah menurut anda pengetahuan pertolongan pertama penting diketahui dan membuat anda tertarik untuk mempelajarinya setelah anda memainkan *Game* ini ?

Tabel 4.25 Perhitungan persentase pertanyaan ke-7

Pertanyaan	Keterangan	Skor	Frekuensi	S
7	Sangat Setuju (SS)	5	20	100
	Setuju (S)	4	8	32
	Biasa Saja (BS)	3	2	6
	Tidak Setuju (TS)	2	0	0
	Sangat Tidak Setuju (STS)	1	0	0
Jumlah			30	138

P = (138/150)x100% = 92 %

Dapat disimpulkan bahwa pengetahuan pertolongan pertama penting untuk diketahui dan membuat pemain merasa terarik untuk mempelajarinya setelah memainkan *Game* ini lagi, terhitung 92% dari yang diharapkan 100%, Secara kontinum dapat digambarkan seperti berikut :



Jadi berdasarkan data yang diperoleh dari 30 responden maka rata – rata 138 terletak pada daerah sangat setuju.