BAB 4
IMPLEMENTASI DAN PENGUJIAN

4.1 Implementasi

Pada bagian tahap implementasi ini merupakan terjemahan perancangan yang berdasarkan hasil analisis pada bab sebelumnya kedalam bahasa pemograman yang dapat dimengerti oleh komputer. Bahasa pemogroman dalam sistem ini menggunakan bahasa Java, kemudian Adobe Ilustrator sebagai tools untuk desain gambar.

4.1.1 Perangkat Keras

Perangkat keras yang dapat digunakan untuk menjalankan aplikasi *game* ini harus memiliki spesifikasi yang cukup tinggi. Sebagai ilustrasi sistem minimum agar dapat menjalankan *game Indonesia for Gaza* dalah sebagai berikut :

Tabel 4.1. Perangkat Keras

Nama Perangkat	Spesifikasi
Prosesor	Dual Core 1,80 Ghz
RAM	1GB
VGA	1GB
Harddisk	120GB
Display	Resolusi 800 x 700
Keyboard	Standar
Mouse	Standar

4.1.2 Perangkat Lunak

Perangkat Lunak yang digunakan untuk mengoperasikan *game Indonesia for Gaza* adalah sebagai berikut :

Tabel 4.2. Perangkat Lunak

Nama Perangkat Lunak	Spesifikasi
Sistem Operasi	Microsoft Windows XP,Seven
Tools	Java, JDK 1.6.0 , NetBeans IDE 6.9.1

Game *Indonesia for Gaza* dapat dimainkan, maka langkah-langkah yang harus dilakukan adalah :

1. Download JDK 1.6.0 version atau ke atasnya (Jika anda belum mempunyai Software Java Development Kit/Software pendukung).
2. Install JDK 1.6.0.
3. Install NetBeans IDE 6.9.1 version atau keatasnya
4. Klik file jar(exe) game *Indonesia for Gaza*.
5. Game sudah langsung bisa dimainkan.

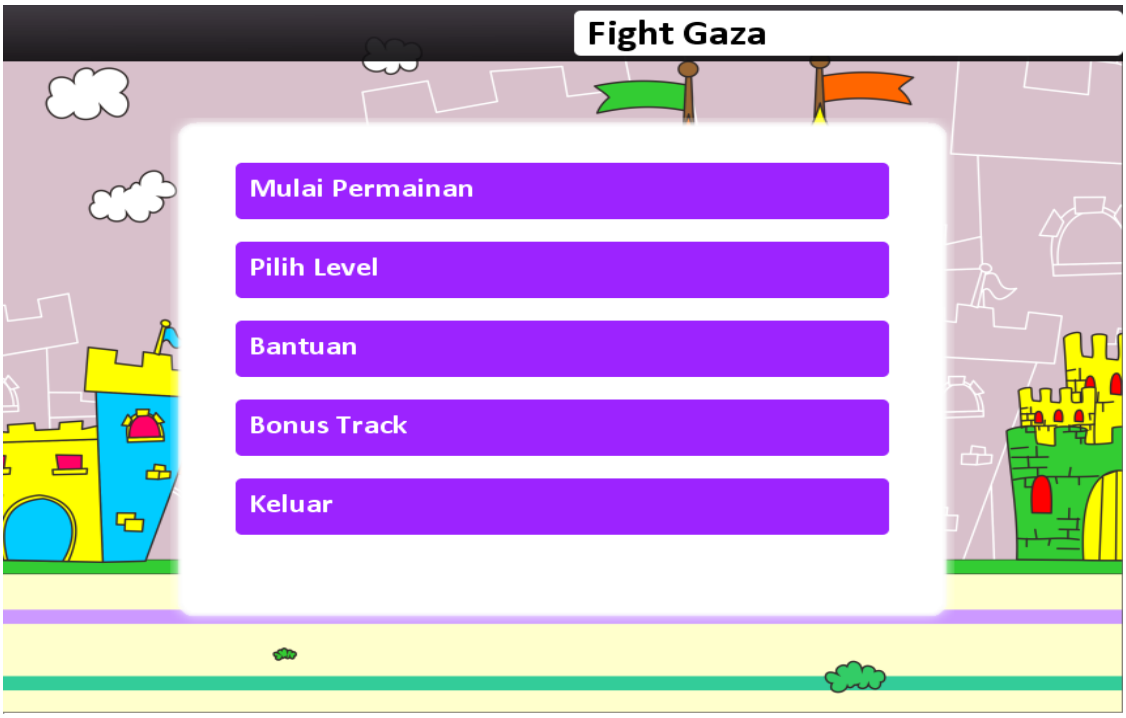
4.2 Implementasi Aplikasi antar muka

Berikut ini merupakan Implementasi aplikasi antar muka Indonesia for Gaza

4.2.1 Tampilan utama

Tampilan anatar muka dari sisi user Indonesia for gaza

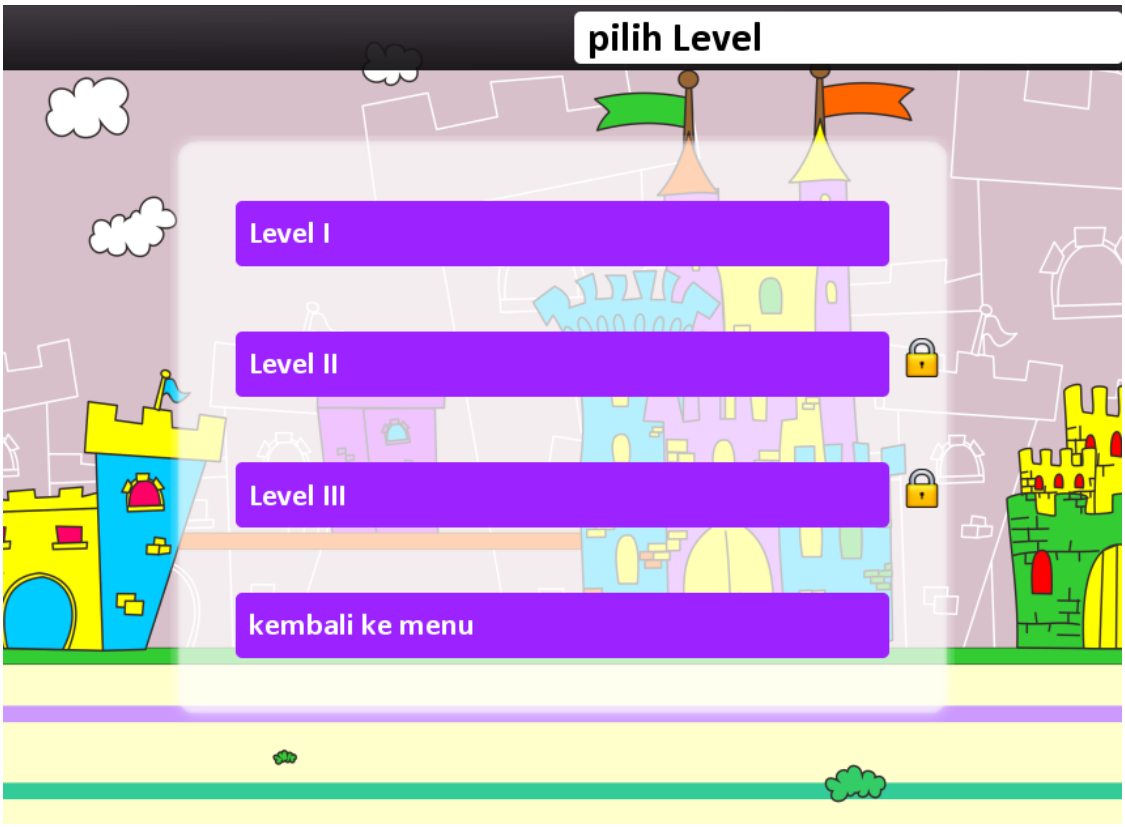
- **A. TampilanUtama**



Gambar 4.1 Tampilan utama

Layar ini berfungsi menampilkan Tampilan utama atau tampilan awal game dimana didalamnya tersidia empat pilihan, yaitu pilihan mulai permainan, pilih level, bantuan dan keluar. Untuk memulai permainan maka langsung pilih pilihan mulai permainan.

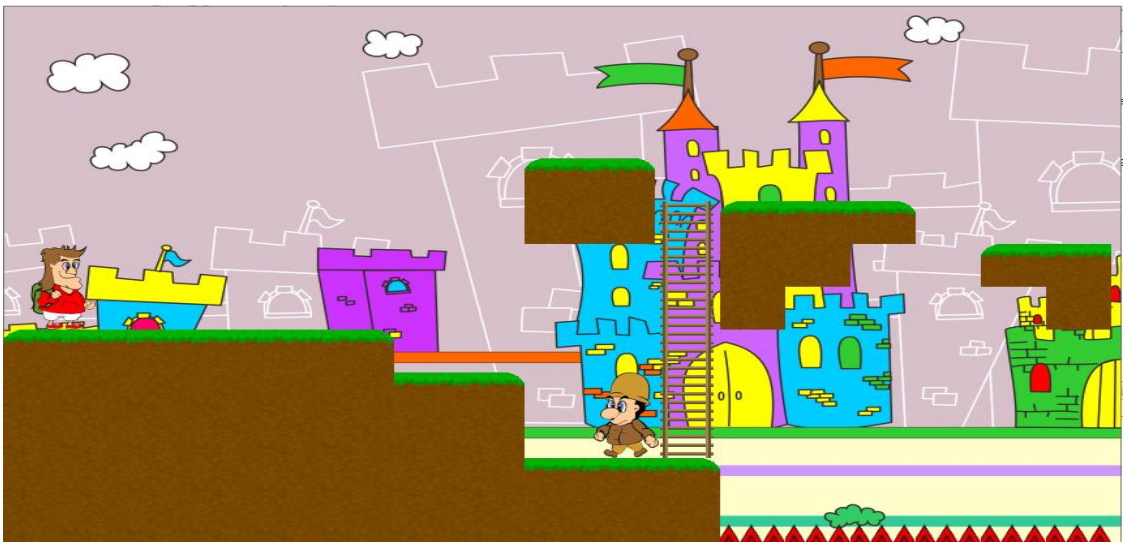
- **B. Tampil pilih level**



Gambar 4.2 Tampilan pilih level

Layar ini berfungsi untuk menampilkan pilihan level, dimana sudah disediakan 3 pilihan level. Pemain bisa memilih pilihan level 1 atau bisa memilih pilihan level ke 2 dan 3 jika sudah berhasil menyelesaikan permainan pada setiap level yang diselesaikan.

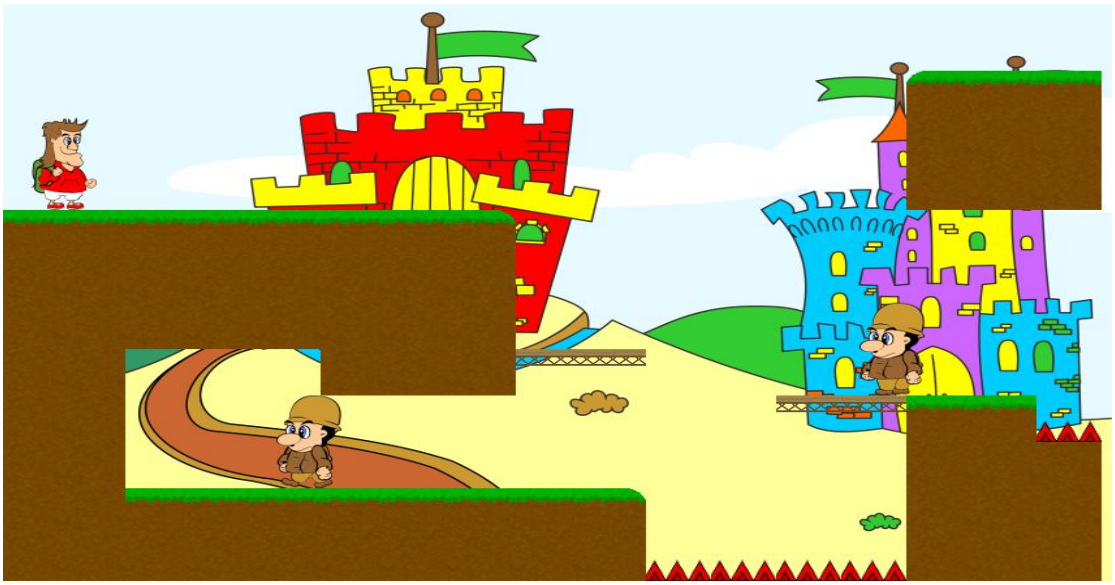
- **C. Tampil level 1**



Gambar 4.3 Tampilan level 1

Layar ini berfungsi untuk menampilkan level 1.

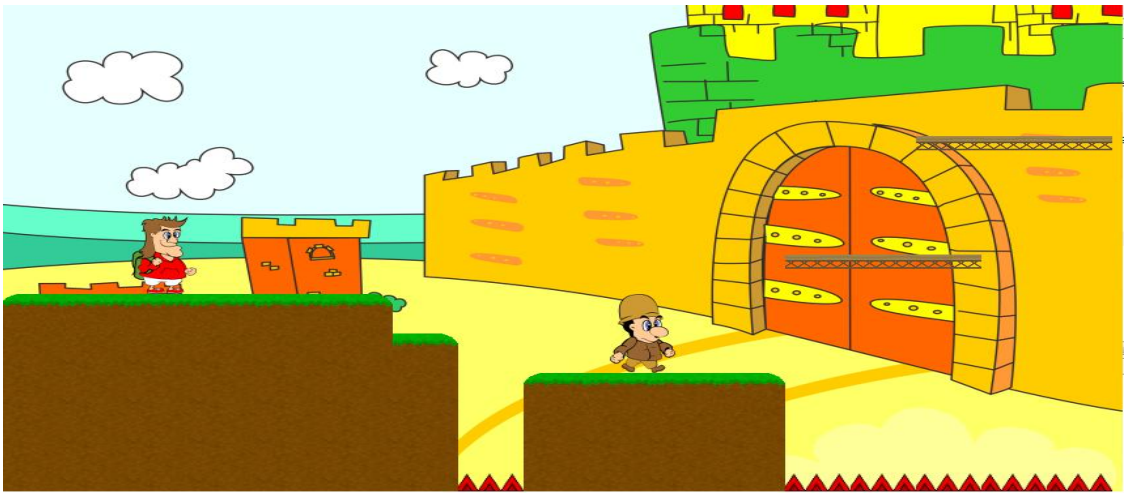
- **D. Tampil level 2**



Gambar 4.4 Tampilan level 2

Layar ini berfungsi untuk menampilkan level 2.

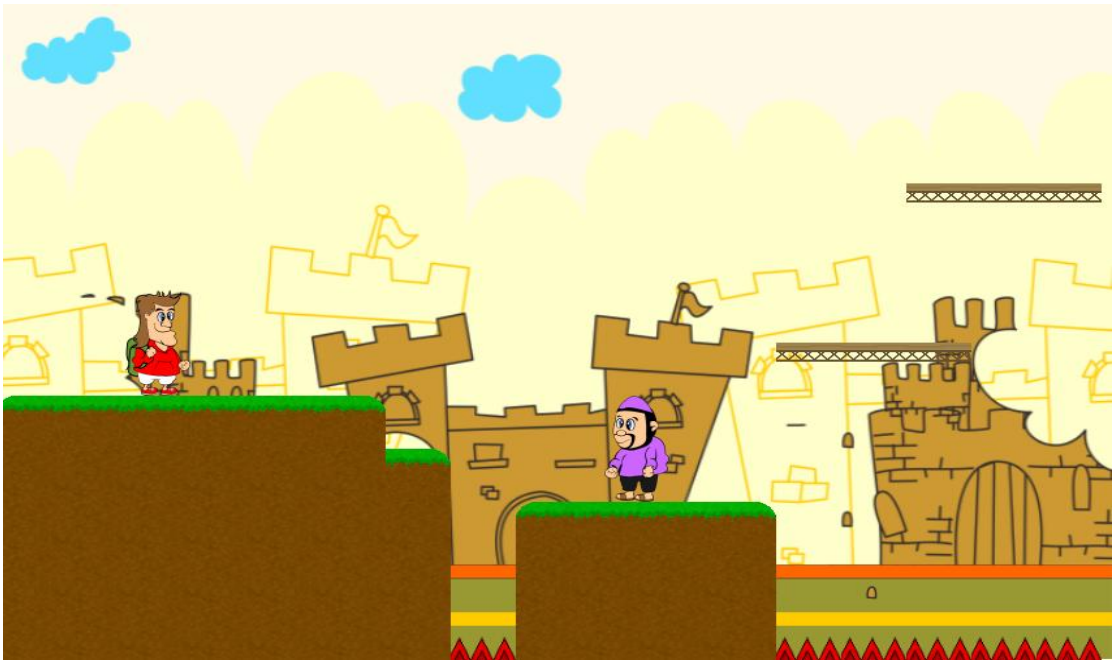
- **E. Tampil level 3.**



Gambar 4.5 Tampilan level 3

Layar ini berfungsi untuk menampilkan level 3.

- **F. Tampil Bonus Track**



Gambar 4.6 Tampilan Bonus Track

Layar ini berfungsi untuk menampilkan bonus track.

· **G. Tampil bantuan.**



Gambar 4.7 Tampilan bantuan

Layar ini berfungsi untuk menampilkan bantuan, dimana Pemain dapat melihat tombol apa saja yang digunakan saat bermain game.

H. Tampil keluar.



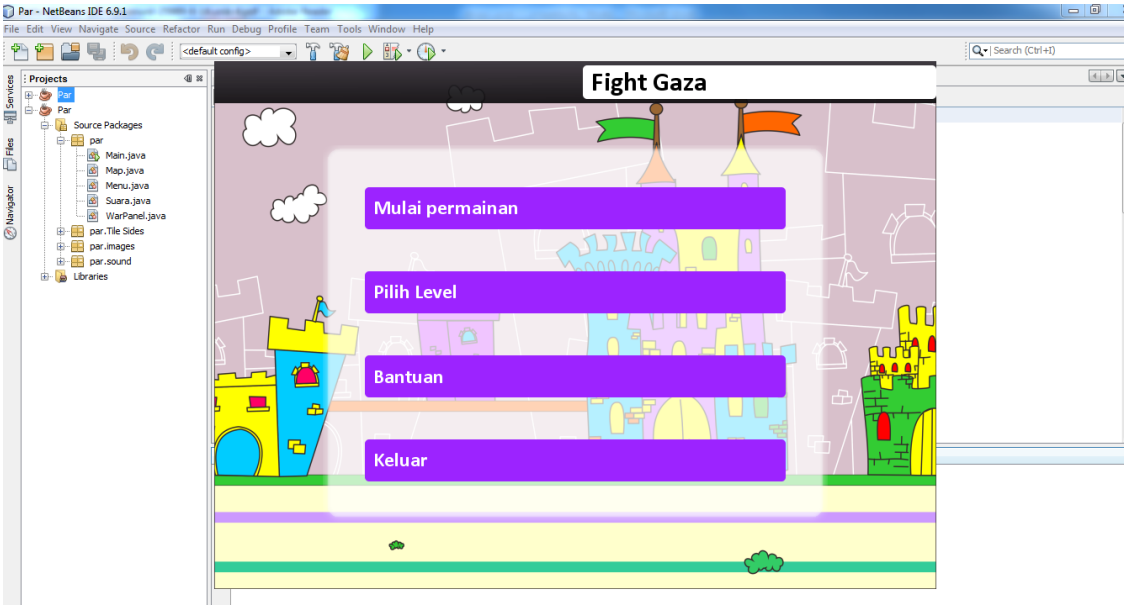
Gambar 4.8 Tampilan keluar

Layar ini berfungsi untuk menampilkan pesan keluar, dimana Pemain dapat memilih untuk keluar permainan atau tidak, tekan tombol ya untuk keluar permainan tekan tombol tidak untuk kembali ke menu utama game.

4.2.2 Tampilan Pengembang Game

Layar menampilkan NetBeans sebagai editor yang digunakan untuk membuat game 2D Indonesia for Gaza.

Tampilan Pengembangan Game



Gambar 4.9 Tampilan pengembang game

4.3 Pengujian Sistem

Pengujian sistem merupakan hal terpenting yang dilakukan untuk menemukan kekurangan atau kesalahan pada perangkat lunak yang diuji. Pengujian bermaksud untuk mengetahui perangkat lunak yang dibuat sudah memenuhi kriteria yang sesuai dengan tujuan perancangan perangkat lunak tersebut.

4.3.1 Pengujian Alpha (Fungsional)

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

A. Rencana Pengujian

Rencana pengujian yang akan dilakukan pada aplikasi ini selengkapnya terlihat pada tabel di bawah.

Tabel 4.3 Rencana Pengujian Aplikasi

Kelas uji	Detail pengujian	Jenis pengujian
Mulai Permainan	Memulai permainan	Black box
Pilih level	Menampilkan tampilan pilih level	Black box
Bantuan	Menampilkan tampilan bantuan	Black box
Keluar	Menampilkan tampilan keluar permainan	Black box
Tombol - tombol	Testing tombol-tombol yang digunakan dalam aplikasi yang dibangun	Black box

B. Kasus dan hasil pengujian

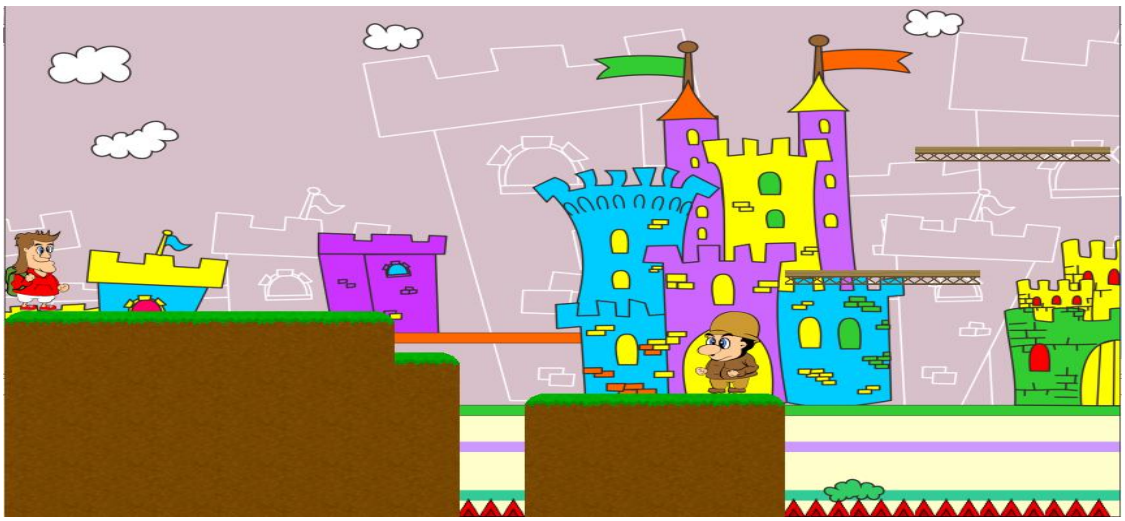
Berdasarkan hasil pengujian dan yang telah disusun, maka dapat dilakukan pengujian sebagai berikut. Pengujian yang dilakukan meliputi pengujian mulai permainan.

1. Pengujian Tombol mulai permainan

Pengujian ini dilakukan untuk memulai Aplikasi game yang telah dibuat.



Gambar 4.10 Tekan tombol mulai permainan



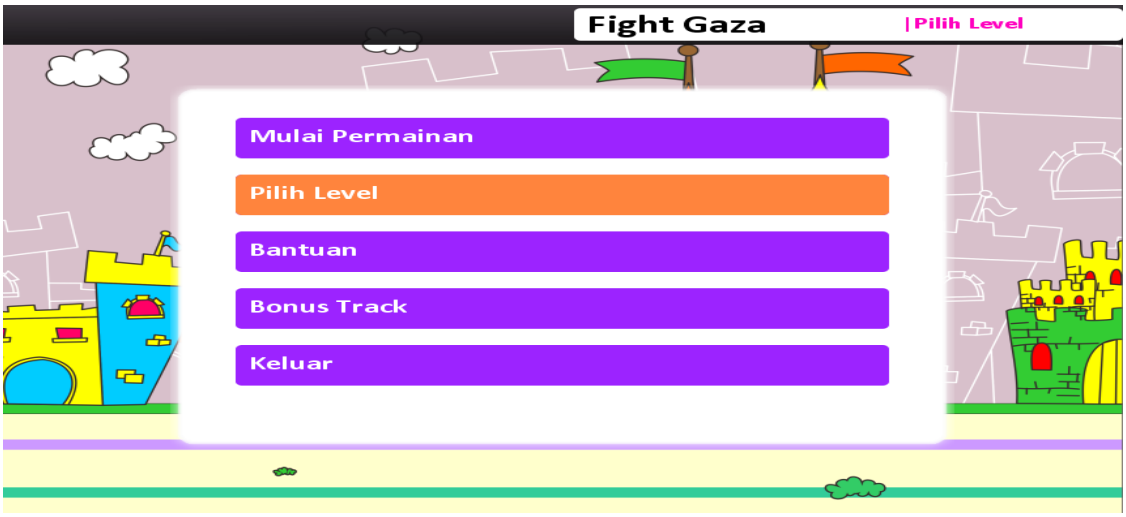
Gambar 4.11 Tampilan mulai permainan

Tabel 4.4 Pengujian Tombol mulai permainan

Kasus dan hasil uji (data normal)			
Data masukan	Yang diharapkan	Pengamatan	Kesimpulan
Pilih menu "Mulai permainan"	Memulai permainan, mucul intro permainan, muncul layar permainan	Permainan dimulai	[√] Tampil [ ] Tidak

2. Pengujian Tombol pilih level

Pengujian ini dilakukan untuk memulai Aplikasi game yang telah dibuat, melihat level mana saja yang sudah diselesaikan, bermain game sesuai level yang dipilih.



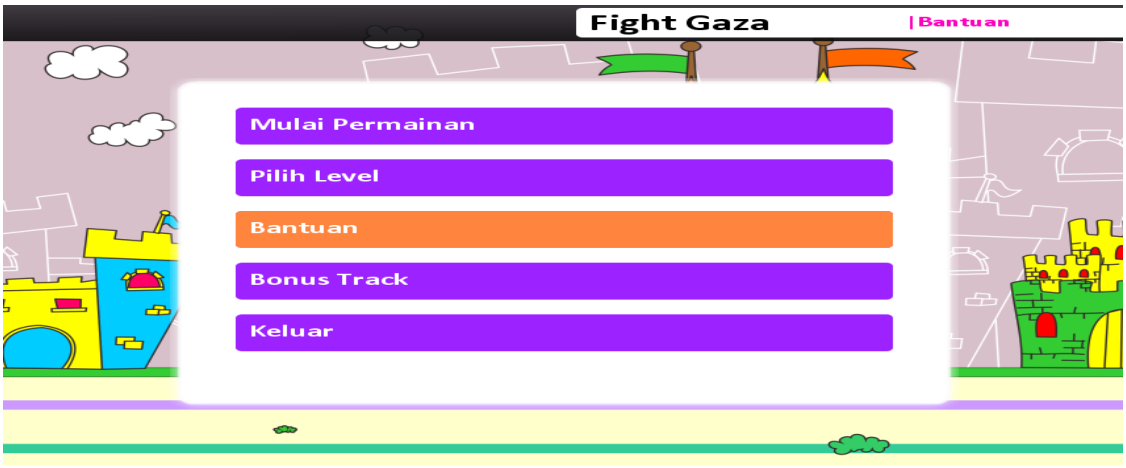
Gambar 4.12 Tekan tombol pilih level

Tabel 4.5 Pengujian Tombol pilih level

Kasus dan hasil uji (data normal)			
Data masukan	Yang diharapkan	Pengamatan	Kesimpulan
Pilih menu "pilih level"	Memulai permainan, untuk kemudian memilih level, menampilkan intro	Permainan dimulai	[□] Tampil [□ □] Tidak

3. Pengujian Tombol Bantuan

Pengujian ini dilakukan untuk menampilkan informasi dan petunjuk yang ada di dalam permainan.



Gambar 4.13 Klik tombol bantuan



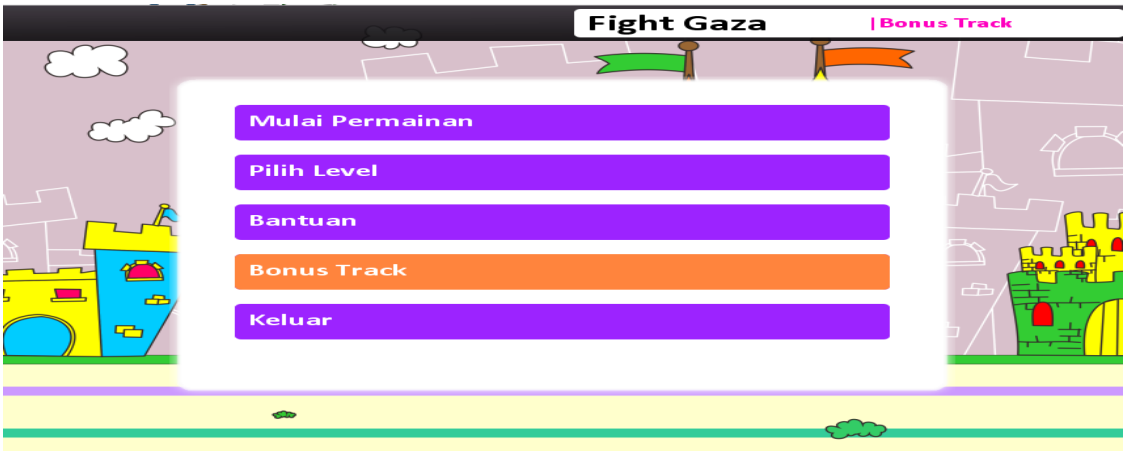
Gambar 4.14 Tampilan bantuan permainan

Tabel 4.6 Pengujian Tombol bantuan

Kasus dan hasil uji (data normal)			
Data masukan	Yang diharapkan	Pengamatan	Kesimpulan
Pilih menu “bantuan”	Melihat informasi untuk memainkan permainan/aplikasi	Petunjuk permainan	[□] Tampil [□ □] Tidak

4. Pengujian Tombol Bonus Track

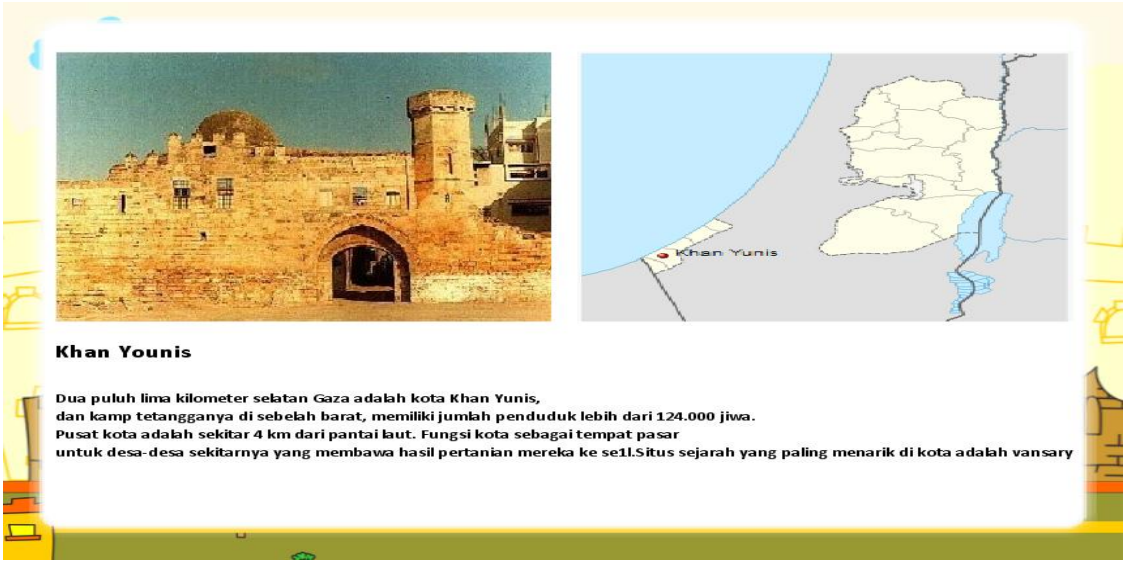
Pengujian ini dilakukan untuk menampilkan Layar bonus track dan informasi tentang gaza yang ada di dalam permainan.



Gambar 4.15 Klik tombol Bonus track



Gambar 4.16 Tampilan Bonus track



Gambar 4.17 Tampilan informasi di Bonus track

Tabel 4.7 Pengujian Tombol bonus track

Kasus dan hasil uji (data normal)			
Data masukan	Yang diharapkan	Pengamatan	Kesimpulan
Pilih menu “Bonus Track”	Melihat informasi yang ada dalam permainan/aplikasi	Informasi permainan	[□] Tampil [□ □] Tidak

5. Pengujian Tombol-tombol yang Digunakan pada Aplikasi

Pengujian ini dilakukan untuk menampilkan informasi dan petunjuk yang ada di dalam permainan.

Tabel 4.8 Pengujian Tombol-tombol yang digunakan

Kasus dan hasil uji (data normal)			
Data masukan	Yang diharapkan	Pengamatan	Kesimpulan
Tombol A	Garuda bergerak kearah kiri	Pergerakan garuda	[□] Bergerak [□ □] Tidak
Tombol D	Garuda bergerak kearah kanan	Pergerakan garuda	[□] Bergerak [□ □] Tidak
Tombol W	Garuda bergerak kearah atas	Pergerakan garuda	[□] Bergerak [□ □] Tidak

Tombol Spasi	Garuda bergerak meloncat	Pergerakan garuda	[□] Bergerak [□ □] Tidak
Tombol Spasi dan A	Garuda bergerak meloncat kearah kiri	Pergerakan garuda	[□] Bergerak [□ □] Tidak
Tombol Spasi dan D	Garuda bergerak meloncat ke arah kanan	Pergerakan garuda	[□] Bergerak [□ □] Tidak

4.3.2 Kasus dan Hasil Pengujian (*White Box Testing)*

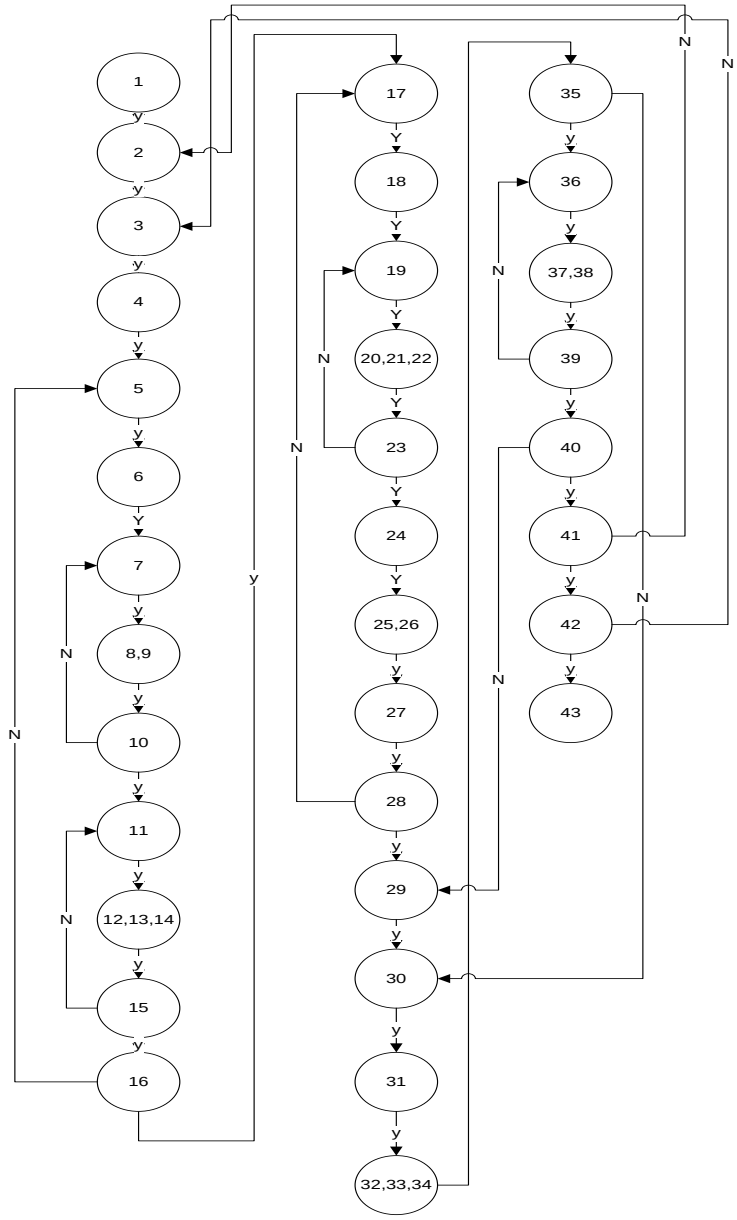
Pengujian *White Box* Algoritma A*Berikut ini merupakan algoritma A* untuk setiap *level* yang penerapannya menggunakan bahasa pemrograman Java:

```
1.  public void musuhMove(double timePassed){
2.  for (int i = 0; i < musuhX.length; i++){
3.  if (!(musuhX[i] == 0 && musuhY[i] == 0)){
4.  musuhAnim[i] += (timePassed * 1000);
5.  if (musuhAnim[i] > 200){
6.  musuhAnim[i] -= 200;
7.  if (musuhDir[i] < 4){
8.  musuhDir[i]++;
9.  musuhDir[i] %= 4;
10. }
11. else if (musuhDir[i] < 8){
12. musuhDir[i]++;
13. musuhDir[i] %= 4;
14. musuhDir[i] += 4;
15. }
16. }
17. if (musuhDir[i] < 4){
18. musuhX[i] -= 2.0 * timePassed;
19. if (musuhX[i] < 0){
20. musuhX[i] = 0;
21. musuhDir[i] = 4;
22. musuhX[i] += 2.0 * timePassed;
23. }
24. else if (solid[map[(int) ((musuhY[i] << 7) + (musuhX[i] -
    0.4))]] || solid[map[(int) ((musuhY[i] << 7) + (musuhX[i] +
```

```
    0.4))]] || !solid[map[(int) (((musuhY[i] + 1) << 7) +
    musuhX[i])]] || map[(int) (((musuhY[i] + 1) << 7) +
    musuhX[i])] == 6){
25. musuhDir[i] = 4;
26. musuhX[i] += 2.0 * timePassed;
27. }
28. }
29. else if (musuhDir[i] < 8){
30. musuhX[i] += 2.0 * timePassed;
31. if (musuhX[i] > mapMaxX){
32. musuhX[i] = mapMaxX;
33. musuhDir[i] = 4;
34. musuhX[i] -= 2.0 * timePassed;
35. }
36. else if (solid[map[(int) ((musuhY[i] << 7) + (musuhX[i] -
    0.4))]] || solid[map[(int) ((musuhY[i] << 7) + (musuhX[i] +
    0.4))]] || !solid[map[(int) (((musuhY[i] + 1) << 7) +
    musuhX[i])]] || map[(int) (((musuhY[i] + 1) << 7) +
    musuhX[i])] == 6){
37. musuhDir[i] = 0;
38. musuhX[i] -= 2.0 * timePassed;
39. }
40. }
41. }
42. }
43. }
```

A. *Flow Graph* Algoritma A*

Berdasarkan *pseudocode* di atas maka *flow graph* algoritma A* adalah



Gambar 4.18 *Flow Graph* Algoritma A*

Keterangan:

= Menggambarkan kondisi

= Menggambarkan aksi

B. *Cyclomatic Complexity* V(G)

E = 34, N = 44

V(G) = E – N + 2

V(G) = 34 – 44 + 2

V(G) = 10 + 2 = 12

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

Path 2 = 1-2-3-4-5-6-7-8-9-10-7-8-9-10-11-12-13-14-15-16-17-18-19-20-21-22-23-24-25-26-27-28-29-30-31-32-33-34-35-36-37-38-39-40-41-42-43

Path 3 = 1-2-3-4-5-6-7-8-9-10-11-12-13-14-15-11-12-13-14-15-16-17-18-19-20-21-22-23-24-25-26-27-28-29-30-31-32-33-34-35-36-37-38-39-40-41-42-43

Path 4 = 1-2-3-4-5-6-7-8-9-10-11-12-13-14-15-16-5-6-7-8-9-10-11-12-13-14-15-16-17-18-19-20-21-22-23-24-25-26-27-28-29-30-31-32-33-34-35-36-37-38-39-40-41-42-43

Path 5 = 1-2-3-4-5-6-7-8-9-10-11-12-13-14-15-16-17-18-19-20-21-22-23-19-20-21-22-23-24-25-26-27-28-29-30-31-32-33-34-35-36-37-38-39-40-41-42-43

Path 6 = 1-2-3-4-5-6-7-8-9-10-11-12-13-14-15-16-17-18-19-20-21-22-23-24-25-26-27-24-25-26-27-28-29-30-31-32-33-34-35-36-37-38-29-30-31-32-33-34-35-36-37-38-39-40-41

Path 7 = 1-2-3-4-5-6-7-8-9-10-11-12-13-14-15-16-17-18-19-20-21-22-23-24-25-26-27-28-17-18-19-20-21-22-23-24-25-26-27-28-29-30-31-32-33-34-35-36-37-38-39-40-41-42-43

Path 8 = 1-2-3-4-5-6-7-8-9-10-11-12-13-14-15-16-17-18-19-20-21-22-23-24-25-26-27-28-29-30-31-32-33-34-35-31-32-33-34-35-36-37-38-39-40-41-42-43

Path 9 = 1-2-3-4-5-6-7-8-9-10-11-12-13-14-15-16-17-18-19-20-21-22-23-24-25-26-27-28-29-30-31-32-33-34-35-36-37-38-39-36-37-38-39-40-41-42-43

Path 10 = 1-2-3-4-5-6-7-8-9-10-11-12-13-14-15-16-17-18-19-20-21-22-23-24-25-26-27-28-29-30-31-32-33-34-35-36-37-38-39-40-29-30-31-32-33-34-35-36-37-38-39-40-41-42-43

Path 11 = 1-2-3-4-5-6-7-8-9-10-11-12-13-14-15-16-17-18-19-20-21-22-23-24-25-26-27-28-29-30-31-32-33-34-35-36-37-38-39-40-41-2-3-4-5-6-7-8-9-10-11-12-13-14-15-16-17-18-19-20-21-22-23-24-25-26-27-28-29-30-31-32-33-34-35-36-37-38-39-40-41-42-43

Path 12 = 1-2-3-4-5-6-7-8-9-10-11-12-13-14-15-16-17-18-19-20-21-22-23-24-25-26-27-28-29-30-31-32-33-34-35-36-37-38-39-40-41-42-3-4-5-6-7-8-9-10-11-12-13-14-15-16-17-18-19-20-21-22-23-24-25-26-27-28-29-30-31-32-33-34-35-36-37-38-39-40-41-42-43

D. *Graph Matriks* Algoritma A*

Tabel 4.9 Graph Matrik Algoritma A*

N	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	N(E)-1
1		1																																											1-1 = 0
2			1																																										1-1 = 0
3				1																																									1-1 = 0
4					1																																								1-1 = 0
5						1																																							1-1 = 0
6							1																																						1-1 = 0
7								1																																					1-1 = 0
8									1																																				1-1 = 0
9										1																																			1-1 = 0
10							1				1																																		2-1 = 1
11												1																																	1-1= 0
12													1																																1-1 = 0
13														1																															1-1 = 0
14															1																														1-1 = 0
15										1						1																													2-1 = 1
16				1													1																												2-1 = 1
17																		1																												1-1 = 0
18																			1																											1-1 = 0
19																				1																										1-1 = 0
20																					1																									1-1 = 0
21																						1																								1-1 = 0
22																							1																							1-1 = 0
23																			1					1																						2-1 = 1
24																									1																					1-1 = 0
25																										1																				1-1 = 0
26																											1																			1-1 = 0
27																								1				1																		2-1 = 1
28																	1												1																	2-1 = 1
29																														1																1-1 = 0
30																															1															1-1 = 0
31																																1														1-1 = 0
32																																	1													1-1 = 0
33																																		1												1-1 = 0
34																																			1											1-1 = 0
35																															1					1										2-1 = 1
36																																					1									1-1 = 0
37																																						1								1-1 = 0
38																																							1							1-1 = 0
39																																				1				1						2-1 = 1
40																													1												1					2-1 = 1
41			1																																							1				2-1 = 1
42		1																																										1		2-1 = 1
43																																												1		1-1 = 0
																												SUM(E)+1																		11+1 = 12

Keterangan:

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

E. Hasil Uji Algoritma A*

Tabel 4.10 Hasil Uji Algoritma A*

Kasus dan Hasil Uji				
No. *Path*	**Butir uji algoritma *A****	***Node* (n), input kondisi**	**Pengamatan**	**Kesimpulan**
1	Rute = null, rute tidak ada	1(*void*) 2(*void*) 3(*void*) 4(*void*) 5(*void*) 6(*void*) 7(*void)* 8(*void*) 9(*void*) 10(*void*) 11(*void*) 12(*void*) 13(*void*) 14(*void*) 15(*void*) 16(*void*) 17(*void*) 18(*void*) 19(*void*) 20(*void*) 21(*void*)	Rute tidak ditemukan	*Path* dikerjakan
2	Current Node = Goal. Rute ditemukan	1,(*void*) 2(*void*) 3(*void*) 4(*void*)	Rute ditemukan	*Path* dikerjakan

Kasus dan Hasil Uji				
No. *Path*	**Butir uji algoritma *A****	***Node* (n), input kondisi**	**Pengamatan**	**Kesimpulan**
		6(void)		
3	Node tidak dapat dilalui, pengujian collision atau unwalkable node	1(*void*) 2(*void*) 3(*void*) 4(*void*) 5(*void*) 6(*void*) 5(*void*) 6(*void*) 7(*void*) 8(*void*) 9(*void*) 10(*void*) 11(*void*) 12(*void*) 13(*void*) 14(*void*) 15(*void*) 16(*void*) 17(*void*) 18(*void*) 19(*void*) 20(*void*) 21(*void*)	Node menjadi unwalkable	*Path* dikerjakan
4	Masukkan semua *neighbor node* ke dalam open dan	1(*void*) 2(*void*) 3(*void*)	*Current node* dan nilai F,G,H masuk ke dalam	*Path* dikerjakan

Kasus dan Hasil Uji				
No. *Path*	**Butir uji algoritma *A****	***Node* (n), input kondisi**	**Pengamatan**	**Kesimpulan**
	hitung nilai F,G,H. Set *current node*	4(*void*) 5(*void*) 6(*void*) 7(*void*) 4(*void*) 5(*void*) 6(*void*) 7(*void)* 8(*void*) 9(*void*) 10(*void*) 11(*void*) 12(*void*) 13(*void*) 14(*void*) 15(*void*) 16(*void*) 17(*void*) 18(*void*) 19(*void*) 20(*void*) 21(*void*)	open	
5	Nilai G yang didapat lebih kecil dari nilai G sebelumnya. Kalkulasi ulang nilai F,G,H	1(*void*) 2(*void*) 3(*void*) 4(*void*) 5(*void*)	Nilai F,G,H, dikalkulasi ulang	*Path* dikerjakan

Kasus dan Hasil Uji				
No. *Path*	**Butir uji algoritma *A****	***Node* (n), input kondisi**	**Pengamatan**	**Kesimpulan**
		6(*void*) 7(*void)* 8(*void*) 9(*void*) 4(*void*) 5(*void*) 6(*void*) 7(*void)* 8(*void*) 9(*void*) 10(*void*) 11(*void*) 12(*void*) 13(*void*) 14(*void*) 15(*void*) 16(*void*) 17(*void*) 18(*void*) 19(*void*) 20(*void*) 21(*void*)		
6	Open = null, tidak ada *node* di dalam open	1(*void*) 2(*void*) 3(*void*) 4(*void*) 5(*void*)	Tidak ada *node* di dalam open	*Path* dikerjakan

Kasus dan Hasil Uji				
No. *Path*	**Butir uji algoritma *A****	***Node* (n), input kondisi**	**Pengamatan**	**Kesimpulan**
		6(*void*) 7(*void)* 8(*void*) 9(*void*) 10(*void*) 3(*void*) 4(*void*) 5(*void*) 6(*void*) 7(*void)* 8(*void*) 9(*void*) 10(*void*) 11(*void*) 12(*void*) 13(*void*) 14(*void*) 15(*void*) 16(*void*) 17(*void*) 18(*void*) 19(*void*) 20(*void*) 21(*void*)		
7	Open = null, tidak ada *node* di dalam open	1(*void*) 2(*void*) 3(*void*)		

Kasus dan Hasil Uji				
No. *Path*	**Butir uji algoritma *A****	***Node* (n), input kondisi**	**Pengamatan**	**Kesimpulan**
		4(*void*) 5(*void*) 6(*void*) 7(*void)* 8(*void*) 9(*void*) 10(*void*) 3(*void*) 4(*void*) 5(*void*) 6(*void*) 7(*void)* 8(*void*) 9(*void*) 10(*void*) 11(*void*) 12(*void*) 13(*void*) 14(*void*) 15(*void*) 16(*void*) 17(*void*) 18(*void*) 19(*void*) 20(*void*) 21(*void*)		

Kasus dan Hasil Uji				
No. *Path*	**Butir uji algoritma *A****	***Node* (n), input kondisi**	**Pengamatan**	**Kesimpulan**
8	Rute = null, rute tidak ada	1(*void*) 2(*void*) 3(*void*) 4(*void*) 5(*void*) 6(*void*) 7(*void)* 8(*void*) 9(*void*) 10(*void*) 11(*void*) 12(*void*) 13(*void*) 14(*void*) 15(*void*) 16(*void*) 17(*void*) 18(*void*) 19(*void*) 20(*void*) 21(*void*)	Rute tidak ditemukan	*Path* dikerjakan
9	Current Node = Goal. Rute ditemukan	1,(*void*) 2(*void*) 3(*void*) 4(*void*) 6(void)	Rute ditemukan	*Path* dikerjakan
10	Node tidak dapat	1(*void*)	Node menjadi	*Path*

Kasus dan Hasil Uji				
No. *Path*	**Butir uji algoritma *A****	***Node* (n), input kondisi**	**Pengamatan**	**Kesimpulan**
	dilalui, pengujian collision atau unwalkable node	2(*void*) 3(*void*) 4(*void*) 5(*void*) 6(*void*) 5(*void*) 6(*void*) 7(*void*) 8(*void*) 9(*void*) 10(*void*) 11(*void*) 12(*void*) 13(*void*) 14(*void*) 15(*void*) 16(*void*) 17(*void*) 18(*void*) 19(*void*) 20(*void*) 21(*void*)	unwalkable	dikerjakan
11	Masukkan semua *neighbor node* ke dalam open dan hitung nilai F,G,H. Set *current node*	1(*void*) 2(*void*) 3(*void*) 4(*void*) 5(*void*)	*Current node* dan nilai F,G,H masuk ke dalam open	*Path* dikerjakan

Kasus dan Hasil Uji				
No. *Path*	**Butir uji algoritma *A****	***Node* (n), input kondisi**	**Pengamatan**	**Kesimpulan**
		6(*void*) 7(*void*) 4(*void*) 5(*void*) 6(*void*) 7(*void)* 8(*void*) 9(*void*) 10(*void*) 11(*void*) 12(*void*) 13(*void*) 14(*void*) 15(*void*) 16(*void*) 17(*void*) 18(*void*) 19(*void*) 20(*void*) 21(*void*)		
12	Open = null, tidak ada *node* di dalam open	1(*void*) 2(*void*) 3(*void*) 4(*void*) 5(*void*) 6(*void*) 7(*void)*		

Kasus dan Hasil Uji				
No. *Path*	**Butir uji algoritma *A****	***Node* (n), input kondisi**	**Pengamatan**	**Kesimpulan**
		8(*void*) 9(*void*) 10(*void*) 3(*void*) 4(*void*) 5(*void*) 6(*void*) 7(*void)* 8(*void*) 9(*void*) 10(*void*) 11(*void*) 12(*void*) 13(*void*) 14(*void*) 15(*void*) 16(*void*) 17(*void*) 18(*void*) 19(*void*) 20(*void*) 21(*void*)		

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.

4.3.3 Pengujian Betha

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.

A. Kuesioner

Kuesioner disebarkan menggunakan teknik *Simple Random Sampling* dimana anggota *sample* dari populasi dilakukan secara acak tanpa memperhatikan strata yang ada dalam populasi. Kuesioner disebarkan kepada 30 orang dari umur 13 tahun sampai 15 tahun, dari hasil kuesioner tersebut akan dilakukan perhitungan agar dapat diambil kesimpulan terhadap penilaian penerapan sistem yang telah dibangun. Kuesioner terdiri dari 7 pernyataan dengan menggunakan Sistem penskoran dengan skala pengukuran yang digunakan adalah skala likert, Berikut skor untuk jawaban kuesioner

Tabel 4.11 Skor Jawaban Pada Kuesioner

Skala Jawaban	Keterangan	Skor
SS	Sangat Setuju	5
ST	Setuju	4
KS	Kurang setuju	3
TS	Tidak Setuju	2
STS	Sangat Tidak Setuju	1

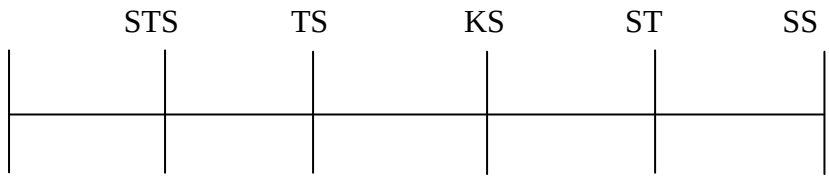
Data yang diperoleh dari pemberian kuesioner kepada responden dapat dianalisis dengan menghitung rata-rata jawaban berdasarkan scoring setiap jawaban dari responden. Analisis yang dapat dilakukan yaitu sebagai berikut.

Jumlah skor ideal (kriterium) untuk seluruh item = 5 × jumlah responden

Rata-rata skor dalam persentase = (jumlah skor : jumlah skor ideal) × 100%

Jumlah skor ideal pada penelitian ini yaitu; Jumlah skor ideal = 5×30 = 150

Secara kontinum dapat digambarkan seperti berikut:



Skor 1 x 30 = 30 2 x 30 = 60 3 x 30 = 90 4 x 30 = 120 5 x 30 = 150

Berikut ini adalah daftar pernyataan yang akan dibagikan pada responden daftar pernyataan :

1) Apakah game 2D Indonesia for Gaza ini menarik bagi anda?
2) Apakah dengan bermain game ini anda mengetahui apa itu Gaza, sejarah dan keindahan kota tersebut?
3) Apakah anda setuju materi atau informasi tentang Gaza mudah dimengerti?
4) Apakah Tampilan menu utama game Indonesia for Gaza mudah digunakan?
5) Apakah anda setuju game ini merupakan salah satu media pembelajaran yang interaktif?
6) Apakah karakter-karakter dalam permainan ini menarik?
7) Apakah anda terpacu untuk memainkan game ini kembali?

B. Evaluasi Dari Pengguna

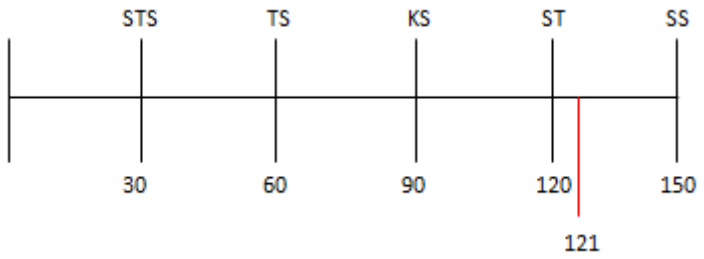
Berikut ini untuk mengetahui tanggapan dan penilaian pemain terhadap game Indonesia for Gaza. Telah disebarkan kuesioner kepada 30 responden dengan usia yang berdasarkan pada sasaran user yaitu dari usia 12 sampai 15 tahun. Berikut adalah hasil kuesioner yang telah dibagikan:

1. Aplikasi *game* 2D *Indonesia for Gaza* sudah menarik

Hasil kuesioner pernyataan nomor 1 sebagai berikut :

Pernyataan	Keterangan	Skor	Frekuensi	S
1.	Sangat setuju	5	6	30
	Setuju	4	19	76
	Kurang setuju	3	5	15
	Tidak setuju	2	-	-
	Sangat tidak setuju	1	-	-
Jumlah			**30**	**121**

Garis kontinum



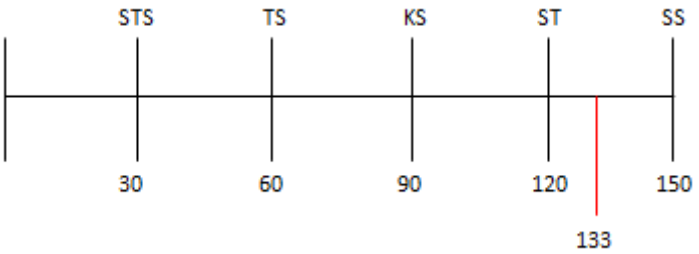
Berdasarkan hasil skor nilai di atas, maka dapat disimpulkan bahwa penilaian terhadap pertanyaan tentang setuju tidakkah anda bahwa game ini menarik, skor yang didapat adalah 121 dari yang diharapkan 150 atau dapat dikategorikan sebagai setuju. Maka dapat disimpulkan bahwa game ini menarik.

2. *Game Indonesia for Gaza* memberikan informasi tentang kota Gaza.

Hasil kuesioner pernyataan nomor 2 sebagai berikut :

Pernyataan	Keterangan	Skor	Frekuensi	S
2.	Sangat setuju	5	10	50
	Setuju	4	13	62
	Kurang setuju	3	7	21
	Tidak setuju	2	-	-
	Sangat tidak setuju	1	-	-
Jumlah			**30**	**133**

Garis kontinum



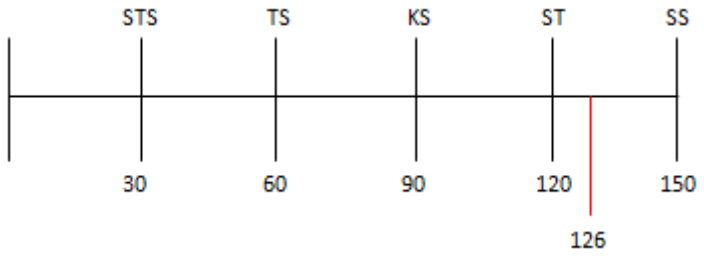
Berdasarkan hasil skor nilai di atas, maka dapat disimpulkan bahwa penilaian terhadap pertanyaan tentang setuju tidakkah bahwa dengan game ini anda mengetahui apa itu Gaza, skor yang didapat adalah 133 dari yang diharapkan 150 atau dapat dikategorikan sebagai setuju. Maka dapat disimpulkan bahwa game ini memberikan informasi tentang Gaza.

3. Materi tentang Gaza mudah dimengerti.

Hasil kuesioner pernyataan nomor 3 sebagai berikut :

Pernyataan	Keterangan	Skor	Frekuensi	S
3.	Sangat setuju	5	8	40
	Setuju	4	20	80
	Kurang setuju	3	2	6
	Tidak setuju	2	-	-
	Sangat tidak setuju	1	-	-
Jumlah			**30**	**126**

Garis kontinum



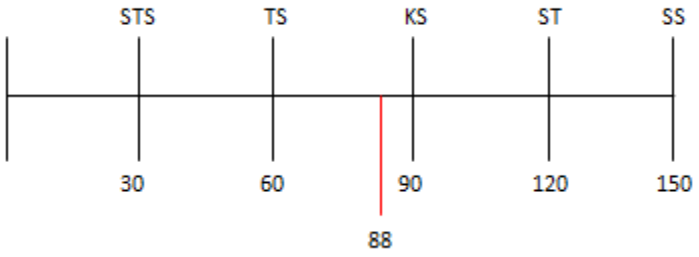
Berdasarkan hasil skor nilai di atas, maka dapat disimpulkan bahwa penilaian terhadap pertanyaan tentang setuju tidakkah anda bahwa informasi di game ini mudah dimengerti, skor yang didapat adalah 126 dari yang diharapkan 150 atau dapat dikategorikan sebagai setuju. Maka dapat disimpulkan bahwa Materi tentang Gaza di game ini mudah dimengerti.

4. Tampilan menu utama game *Indonesia for Gaza* mudah digunakan

Hasil kuesioner pernyataan nomor 4 sebagai berikut :

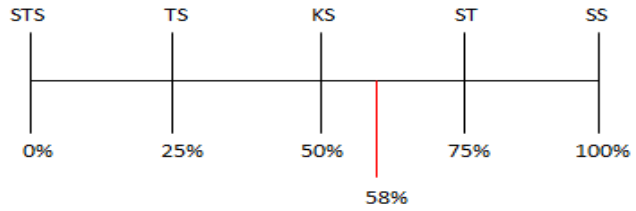
Pernyataan	Keterangan	Skor	Frekuensi	S
4.	Sangat setuju	5	2	10
	Setuju	4	10	40
	Kurang setuju	3	9	27
	Tidak setuju	2	2	4
	Sangat tidak setuju	1	7	7
Jumlah			**30**	**88**

Garis kontinum



Berdasarkan hasil skor nilai di atas, maka dapat disimpulkan bahwa penilaian terhadap pertanyaan tentang setuju tidakkah anda bahwa tampilan menu utama game ini mudah digunakan, skor yang didapat adalah 88 dari yang diharapkan 150 atau dapat dikategorikan sebagai tidak setuju karena jumlah skor ini terletak diantara tidak setuju dan kurang setuju namun lebih cenderung kurang setuju 88 mendekati 90 dapat dimodelkan berdasarkan hasil persentasi $\frac{88}{150}\ x\ 100\% = 58{,}67\%$ yang berarti kurang setuju. Maka dapat disimpulkan tampilan menu utama *game Indonesia for Gaza* belum cukup baik dan belum mudah digunakan.

Garis range skor rata-rata

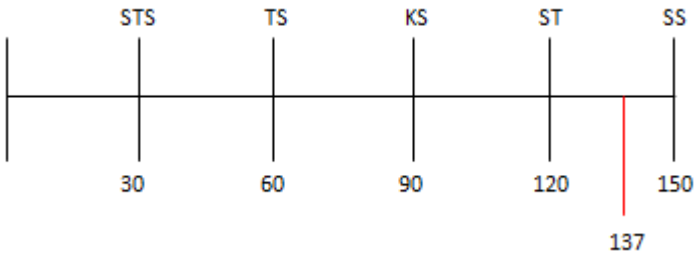


5. *Game Indonesia for Gaza* menjadi *game* pembelajaran interaktif.

Hasil kuesioner pernyataan nomor 5 sebagai berikut :

Pernyataan	Keterangan	Skor	Frekuensi	S
5.	Sangat setuju	5	20	100
	Setuju	4	7	28
	Kurang setuju	3	3	9
	Tidak setuju	2	-	-
	Sangat tidak setuju	1	-	-
Jumlah			**30**	**137**

Garis kontinum



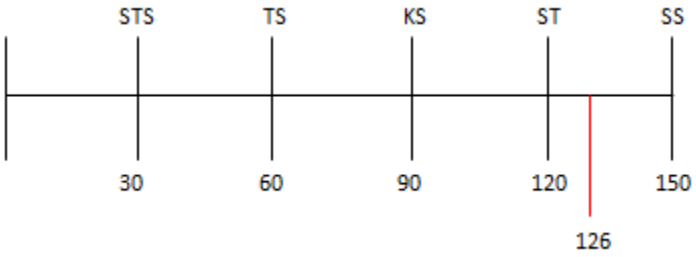
Berdasarkan hasil skor nilai di atas, maka dapat disimpulkan bahwa penilaian terhadap pertanyaan tentang setuju tidakkah anda bahwa game ini menjadi pembelajaran yang interaktif, skor yang didapat adalah 137 dari yang diharapkan 150, atau dapat dikategorikan sebagai setuju. Maka dapat disimpulkan bahwa g*ame Indonesia for Gaza* dapat menjadi *game* pembelajaran interaktif.

6. Karakter dalam *game Indonesia for Gaza* menarik.

Hasil kuesioner pernyataan nomor 6 sebagai berikut :

Pernyataan	Keterangan	Skor	Frekuensi	S
6.	Sangat setuju	5	9	45
	Setuju	4	18	72
	Kurang setuju	3	3	9
	Tidak setuju	2	-	-
	Sangat tidak setuju	1	-	-
Jumlah			**30**	**126**

Garis kontinum



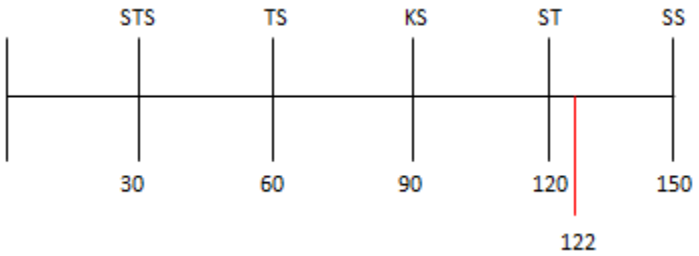
Berdasarkan hasil skor nilai di atas, maka dapat disimpulkan bahwa penilaian terhadap pertanyaan tentang setuju tidakkah anda bahwa game ini memiliki karakter yang menarik didalamnya, skor yang didapat adalah 126 dari yang diharapkan 150, atau dapat dikategorikan sebagai setuju. Maka dapat disimpulkan bahwa penilaian terhadap karakter dalam game ini adalah menarik.

7. Game ini membuat pemainnya untuk bermain kembali

Hasil kuesioner pernyataan nomor 7 sebagai berikut :

Pernyataan	Keterangan	Skor	Frekuensi	S
7.	Sangat setuju	5	7	35
	Setuju	4	18	72
	Kurang setuju	3	5	15
	Tidak setuju	2	-	-
	Sangat tidak setuju	1	-	-
Jumlah			**30**	**122**

Garis kontinum



Berdasarkan hasil skor nilai di atas, maka dapat disimpulkan bahwa penilaian terhadap pertanyaan tentang setuju tidakkah anda bahwa game ini membuat pemainnya untuk bermain kembali, skor yang diperoleh sebesar 122 dari yang diharapkan 150, atau dapat dikategorikan setuju. Maka dapat bahwa game ini membuat pemainnya ingin bermain *game* ini kembali.

Berdasarkan hasil kuesioner, para responden memberikan saran yang dapat disimpulkan sebagai berikut:

1. Penambahan karakter dan pengurangan karakter.
2. Perbaikan tingkat kesulitan di game.
3. Perbaikan Map pada game.

4.3.4 Kelebihan

Kelebihan dari game 2D Indonesia for Gaza yaitu :

1. Pembangunan aplikasi game 2D Indonesia for Gaza sangat diperlukan sebagai media untuk memperkenalkaan gaza melalui game, khususnya sejarah dan keindahanya dulu
2. Sistem single player dan karakter-karakter yang sudah dibangun dapat dikatakan menarik.

4.3.5 Kekurangan

Kekurangan dari game 2D Indonesia for Gaza yaitu :

1. Sisi sejarah dan keindahan gaza belum terangkat secara maksimal.
2. Game lumayan sulit dimainkan.
3. Tidak memiliki sistem multiplayer.
4. fitur pilih karakter kurang efektif
5. Perjalanan di permainan masih singkat tidak terlalu jauh