BAB 4
IMPLEMENTASI DAN PENGUJIAN

4.1. 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.

4.1.1. Implementasi Perangkat Keras

Perangkat keras yang digunakan untuk implementasi *game* edukasi yang dibuat tertera pada tabel 4.1.

Tabel 4.1. Spesifikasi Implementasi Perangkat Keras

No.	Perangkat Keras	Spesifikasi
1	Processor	Processor 2.6 Ghz
2	Monitor	Monitor Standar
3	VGA	VGA On-Board 64 MB
4	Memori	128 MB
5	*Space Hardisk*	80 GB
8	*Mouse* dan *Key Board*	Standar

4.1.2. Implementasi Perangkat Lunak

Perangkat lunak yang digunakan untuk implementasi *game* edukasi yang dibuat tertera pada tabel 4.2.

Tabel 4.2. Spesifikasi Implementasi Perangkat Lunak

No.	Perangkat Lunak	Keterangan
1	Sistem Operasi	Windows XP
2	*Plugin*	Adobe Flash Player

4.1.3. Implementasi Aplikasi

Game Edukasi *Smart Labyrinth* merupakan aplikasi *game* edukasi ber-*genre maze game* (labirin). *Game* ini memiliki konten edukasi matematika dasar berupa mengenal bilangan romawi, sudut, satuan waktu, satuan berat, satuan panjang, bilangan bulat, pecahan bilangan, dan perpangkatan. Dalam *game* ini terdapat 3 *level* dan pada setiap *level* memiliki 6 *stage* yang harus diselesaikan. Pada setiap *stage* terdapat musuh yang akan selalu mengejar pemain saat akan mengambil konten edukasi.

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.

Aplikasi *game* ini berbasis Desktop yang dibangun dengan Adobe Flash menggunakan bahasa pemrograman *ActionScript 2.0*. Sehingga aplikasi *game* ini hanya dapat dijalankan apabila pemain telah memiliki *plugin flash player* pada perangkat komputernya.

4.1.4. Implementasi Antarmuka

Implementasi antarmuka dilakukan dengan setiap tampilan program yang dibangun. Antarmuka yang disajikan berikut merupakan antarmuka yang mewakili keseluruhan program.

1. Implementasi Antarmuka Menu Utama

Gambar 4.1 menunjukan antarmuka menu utama dimana halaman tersebut menampilkan tombol main, tombol nilai tertinggi, tombol bantuan, dan tombol keluar.



Gambar 4.1. Antarmuka Menu Utama

2. Implementasi Tampilan Pemilihan Level

Gambar 4.2 merupakan implementasi tampilan pemilihan level merupakan tampilan pilihan tingkat kesulitan permainan.



Gambar 4.2. Antarmuka Pemilihan Level

3. Implementasi Tampilan Nilai Tertinggi

Gambar 4.3 merupakan implementasi tampilan nilai tertinggi menampilkan nilai tertinggi dan nilai terakhir dari permainan.



Gambar 4.3. Antarmuka Tampilan Nilai Tertinggi

4. Implementasi Tampilan Bantuan

Gambar 4.4 merupakan implementasi tampilan bantuan menampilkan petunjuk atau cara bermain *game*.



Gambar 4.4. Antarmuka Tampilan Bantuan

5. Implementasi *Game* Tingkat Mudah

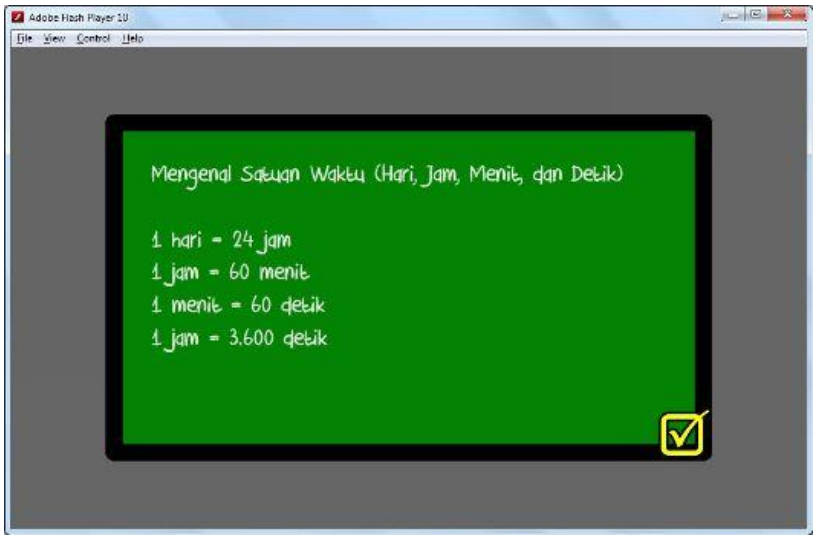
Gambar 4.5 merupakan implementasi *game* tingkat mudah menampilkan tampilan *game* dengan tingkat kesulitan mudah.



Gambar 4.5. Antarmuka *Game* Tingkat Mudah

6. Implementasi Tampilan Materi Edukasi

Gambar 4.6 merupakan implementasi tampilan materi edukasi menunjukkan materi edukasi, tampilan ini akan muncul ketika *player* mengambil *item* berupa lembaran kertas dalam *game*.



Gambar 4.6. Antarmuka Tampilan Materi Edukasi

7. Implementasi Tampilan Bermain *Mini Game Drag & Drop*

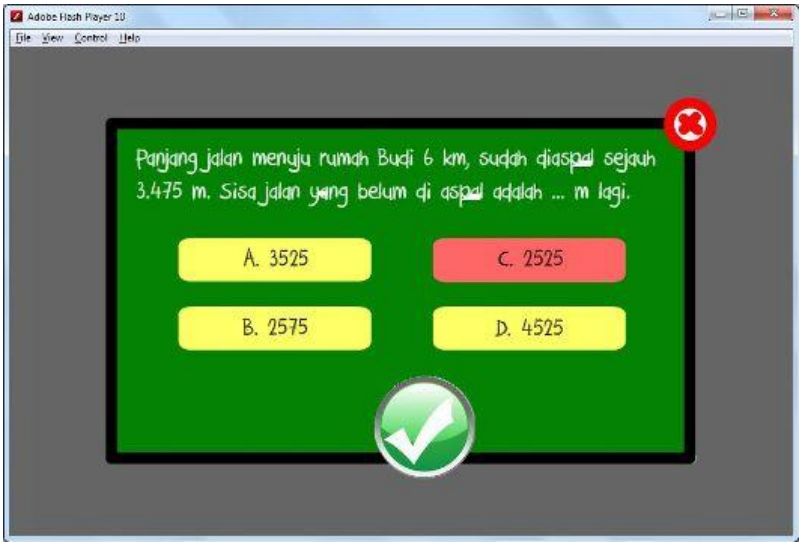
Gambar 4.7 merupakan implementasi tampilan bermain *mini game drag & drop* merupakan tampilan yang akan muncul saat *player* mengambil *item* pensil dalam *game*.



Gambar 4.7. Antarmuka Tampilan Bermain *Mini Game Drag & Drop*

8. Implementasi Tampilan Bermain Kuis

Gambar 4.8 merupakan implementasi tampilan bermain kuis merupakan tampilan yang menampilkan soal-soal latihan berupa pilihan ganda. Tampilan ini akan muncul saat *player* mengambil *item* berupa pensil dalam *game*.



Gambar 4.8. Antarmuka Tampilan Bermain Kuis

9. Implementasi Tampilan Tamat *Game* Tingkat Mudah

Gambar 4.9 merupakan implementasi tampilan tamat *game* tingkat mudah merupakan tampilan saat *player* berhasil menyelesaikan *game* tingkat mudah.



Gambar 4.9. Antarmuka Tampilan Tamat *Game* Tingkat Mudah

4.2. 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.

4.2.1. 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.

4.2.1.1. Skenario Pengujian Aplikasi

Skenario pengujian menjelaskan pengujian terhadap sistem yang ada pada aplikasi *game* edukasi *smart labyrinth*. Skenario pengujian yang akan dilakukan pada aplikasi ini selengkapnya dapat dilihat pada table 4.3.

Tabel 4.3. Skenario Pengujian Alpha

No	Komponen Yang Diuji	Skenario Pengujian	Jenis Pengujian
1	Menu Utama	Memilih Tombol Main	*Black box*
		Memilih Tombol Nilai Tertinggi	*Black box*
		Memilih Tombol Bantuan	*Black box*
		Memilih Tombol Keluar	*Black box*
2	Menu Main	Memilih Tombol Mudah	*Black box*
		Memilih Tombol Sedang	*Black box*
		Memilih Tombol Sulit	*Black box*
		Memilih Tombol Kembali	*Black box*
3	Menu Nilai Tertinggi	Memilih Tombol Kembali	*Black box*
4	Menu Bantuan	Memilih Tombol Kembali	*Black box*
		Memilih Tombol *Next*	*Black box*
		Memilih Tombol *Previous*	*Black box*
5	Keluar Aplikasi *Game*	Memilih Tombol YA	*Black box*
		Memilih Tombol TIDAK	*Black box*
6	Menu Permainan	Memilih Tombol Menu Utama	*Black box*
		Memilih Tombol Tanda Ceklis	*Black box*
		Memilih Tombol Jawaban Kuis	*Black box*
7	Gerak Karakter *Player*	Bergerak ke atas	*Black box*
		Bergerak ke kiri	*Black box*
		Bergerak ke bawah	*Black box*

No	Komponen Yang Diuji	Skenario Pengujian	Jenis Pengujian
		Bergerak ke kanan	*Black box*
8	Gerak Musuh	Penelusuran Algoritma A*	*White box*

4.2.1.2. 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 Menu Utama

Pengujian menu utama dengan kasus dan hasil pengujian yang menghasilkan kesimpulan dapat dilihat pada tabel 4.4.

Tabel 4.4. Pengujian Menu Utama

Kasus dan hasil pengujian			
Masukan	**Yang diharapkan**	**Pengamatan**	**kesimpulan**
Tombol main	Menampilkan pilihan *level* yang tersedia pada *game*	Tombol dapat berfungsi sesuai dengan yang diharapkan	Diterima
Tombol nilai tertinggi	Menampilkan informasi nilai tertinggi	Tombol dapat berfungsi sesuai dengan yang diharapkan	Diterima
Tombol bantuan	Menampilkan informasi bantuan	Tombol dapat berfungsi sesuai dengan yang diharapkan	Diterima
Tombol keluar	Keluar dari *game*	Tombol dapat berfungsi sesuai dengan yang diharapkan	Diterima

2. Pengujian Menu Main

Pengujian Menu Main dengan kasus dan hasil pengujian yang menghasilkan kesimpulan dapat dilihat pada tabel 4.5.

Tabel 4.5. Pengujian Menu Main

Kasus dan hasil pengujian			
Masukan	**Yang diharapkan**	**Pengamatan**	**kesimpulan**
Tombol mudah	Menampilkan permainan dengan *Level* mudah	Tombol dapat berfungsi sesuai dengan yang diharapkan	Diterima
Tombol sedang	Menampilkan permainan dengan *Level* sedang	Tombol dapat berfungsi sesuai dengan yang diharapkan	Diterima
Tombol sulit	Menampilkan permainan dengan *Level* sulit	Tombol dapat berfungsi sesuai dengan yang diharapkan	Diterima

3. Pengujian Nilai Tertinggi

Pengujian nilai tertinggi dengan kasus dan hasil pengujian yang menghasilkan kesimpulan dapat dilihat pada tabel 4.6.

Tabel 4.6. Pengujian Nilai Tertinggi

Kasus dan hasil pengujian			
Masukan	**Yang diharapkan**	**Pengamatan**	**kesimpulan**
Tombol kembali	Menampilkan kembali menu utama	Tombol dapat berfungsi sesuai dengan yang diharapkan	Diterima

4. Pengujian Bantuan

Pengujian bantuan dengan kasus dan hasil pengujian yang menghasilkan kesimpulan dapat dilihat pada tabel 4.7.

Tabel 4.7. Pengujian Bantuan

Kasus dan hasil pengujian			
Masukan	**Yang diharapkan**	**Pengamatan**	**kesimpulan**
Tombol kembali	Menampilkan kembali menu utama	Tombol dapat berfungsi sesuai dengan yang diharapkan	Diterima
Tombol *next*	Menampilkan bantuan selanjutnya	Tombol dapat berfungsi sesuai dengan yang diharapkan	Diterima
Tombol *previous*	Menampilkan bantuan sebelumnya	Tombol dapat berfungsi sesuai dengan yang diharapkan	Diterima

5. Pengujian Keluar Aplikasi *Game*

Pengujian keluar aplikasi *game* dengan kasus dan hasil pengujian yang menghasilkan kesimpulan dapat dilihat pada tabel 4.8.

Table 4.8. Pengujian Keluar Aplikasi *Game*

Kasus dan hasil pengujian			
Masukan	**Yang diharapkan**	**Pengamatan**	**kesimpulan**
Tombol YA	Keluar dari *game*	Tombol dapat berfungsi sesuai dengan yang diharapkan	Diterima
Tombol TIDAK	Menampilkan kembali menu utama	Tombol dapat berfungsi sesuai dengan yang diharapkan	Diterima

6. Pengujian Menu Permainan

Pengujian menu permainan dengan kasus dan hasil pengujian yang menghasilkan kesimpulan dapat dilihat pada tabel 4.9.

Tabel 4.9. Pengujian Menu Permainan

Kasus dan hasil pengujian			
Masukan	**Yang diharapkan**	**Pengamatan**	**kesimpulan**
Tombol Menu Utama	Keluar permainan dan kembali ke menu utama	Tombol dapat berfungsi sesuai dengan yang diharapkan	Diterima
Tombol Tanda Ceklis	Menutup materi edukasi dan melanjutkan permainan	Tombol dapat berfungsi sesuai dengan yang diharapkan	Diterima
Tombol Jawaban Kuis	Menampilkan informasi benar atau salah	Tombol dapat berfungsi sesuai dengan yang diharapkan	Diterima

7. Pengujian Gerak Karakter

Pengujian Gerak karakter *player* dengan kasus dan hasil pengujian yang menghasilkan kesimpulan dapat dilihat pada tabel 4.10.

Tabel 4.10. Pengujian Gerak Karakter

Kasus dan hasil pengujian			
Masukan	**Yang diharapkan**	**Pengamatan**	**kesimpulan**
Atas	Karakter *player* bergerak ke atas	Karakter *player* bergerak ke atas sesuai dengan yang diharapkan	Diterima
Kiri	Karakter *player* bergereak ke kiri	Karakter *player* bergerak ke kiri sesuai dengan yang diharapkan	Diterima
Bawah	Karakter *player* bergerek ke bawah	Karakter *player* bergerak ke bawah sesuai dengan yang diharapkan	Diterima
Kanan	Karakter *player* bergerak ke kanan	Karakter *player* bergerak ke kanan sesuai dengan yang diharapkan	Diterima

4.2.1.3. Kesimpulan Hasil Pengujian Alpha

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

4.2.1.4. Kasus dan Hasil Pengujian (White Box)

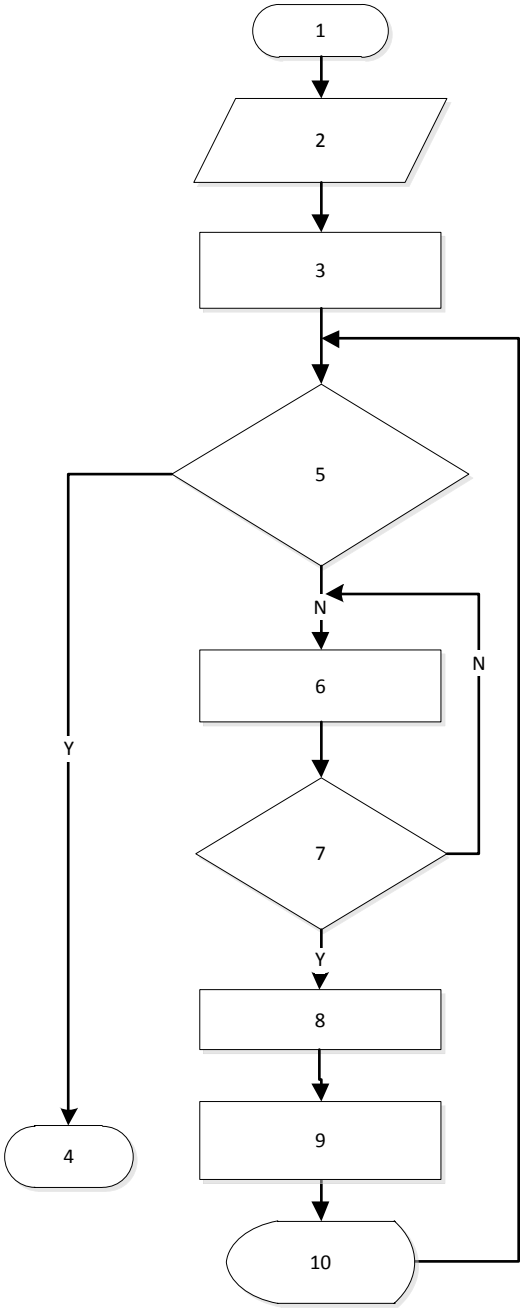
Berikut ini adalah kasus menguji perangkat lunak yang telah dibangun menggunakan metode *white-box* berdasarkan tabel 4.3. Pengujian *white-box* menggunakan *flow graph* yang digunakan untuk menggambarkan alur dari algoritma dan *grap matrix* yang digunakan untuk menggenerasi *flow graph*. Berikut pengujian *white-box* dapat dilihat pada tabel 4.11.

Tabel 4.11. Pengujian *White-box*

No	*Coding*
1	path.name = "node_" + starty + "_" + startx;
2	path.belum_dicek = [];
3	path[path.name] = {x:startx, y:starty, visited:true, parentx:null, parenty:null, cost:cost};
4	path.belum_dicek[path.belum_dicek.length] = path[path.name];
5	while (path.belum_dicek.length > 0){
6	var N = path.belum_dicek.shift();
7	if (N.x == targetx and N.y == targety){
8	buatjalur(N);
9	path.done = true;
10	break;}
11	else{
12	G = G + 10;
13	addNode(N,N.x + 1,N.y,targetx,targety,G,H,F);
14	addNode(N,N.x,N.y + 1,targetx,targety,G,H,F);

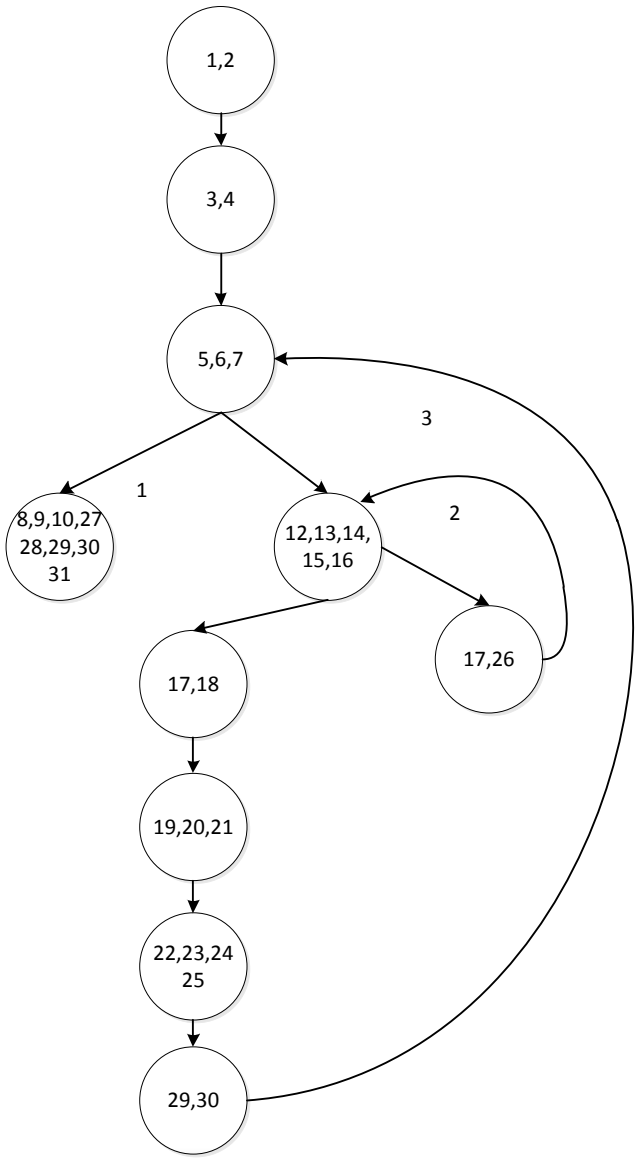
15	addNode(N,N.x - 1,N.y,targetx,targety,G,H,F);
16	addNode(N,N.x,N.y - 1,targetx,targety,G,H,F);}}
17	function addNode(musuh1, x, y, targetx, targety, G, H,F){
18	if (game["lt_" + y + "_" + x].jalan){
19	var cost = (Math.abs(x - targetx) + Math.abs(y - targety)) * 10;
20	var F = cost + G;
21	path[path.name] = {x:x, y:y, visited:false, parentx:musuh1.x, parenty:musuh1.y, cost:cost};
22	for (var i = 0; i < path.belum_dicek.length; i++){
23	if (cost < path.belum_dicek[i].cost){
24	path.belum_dicek.splice(i,0,path[path.name]);
25	break;}}
26	else {}
27	function buatjalur(musuh1){
28	while (musuh1.parentx != null){
29	game.path[game.path.length] = musuh1.x;
30	game.path[game.path.length] = musuh1.y;}
31	musuh.moving = true;}

a. *Flowchart* Pencarian Jalur



Gambar 4.10. *Flowchart* Pencarian Jalur

b. *Flow Graph* Penelusuran Algoritma A*



Gambar 4.11. *Flow Graph* Algoritma A*

Keterangan :

= Menggambarkan kondisi

= Menggambarkan aksi

Dari gambar 4.11 dapat dihitung *cyclomatic complexity* sebagai berikut:

Region = 3

V(G) = 11 *edge* – 10 *node* + 2 = 3

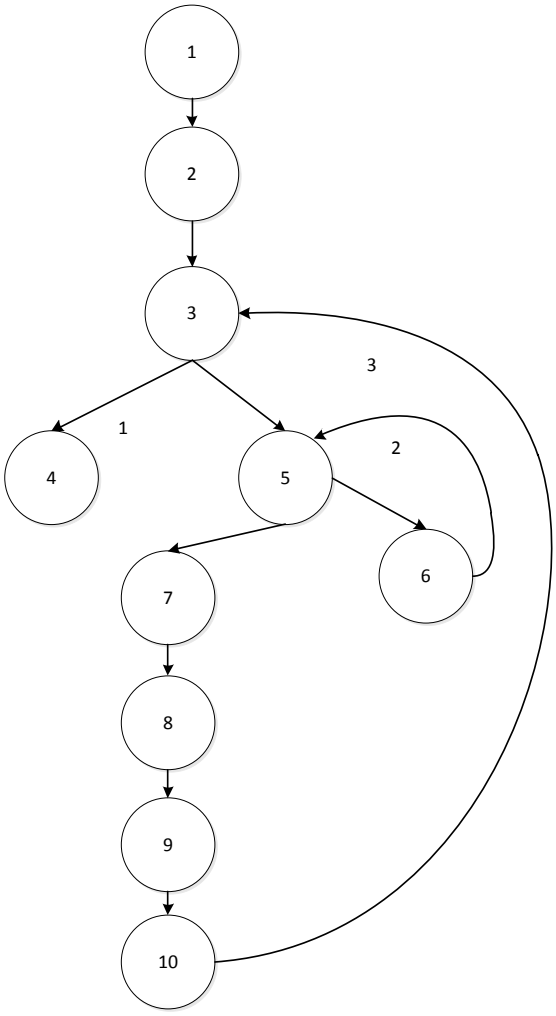
Jadi, *cyclomatic complexity* untuk gambar 4.11 adalah 3. Berdasarkan *cyclomatic complexity* tersebut, maka terdapat 3 *path* yang terdiri dari:

Path 1 : 1-2-3-4-5-6-7-8-9-10-27-28-29-30-31

Path 2 : 1-2-3-4-5-6-7-12-13-14-15-16-17-26-12-13-14-15-16-17-18-19-20-21-22-23-24-25-29-30-5-6-7-8-9-10-27-28-29-30-31

Path 3 : 1-2-3-5-6-7-12-13-14-15-16-17-18-19-20-21-22-23-24-25-26-27-28-29-30-5-6-7-8-9-10-27-28-29-30-31

Maka gambar 4.11 disederhanakan berdasarkan kondisi atau simpul seperti yang ditunjukkan pada gambar 4.12 sebagai berikut:



Gambar 4.12. Penyederhanaan *Flow Graph* Algoritma A*

c. *Graph Matriks* penelusuran Algoritma A*

Tabel 4.12. *Graph Matrix* Algoritma A*

	1	2	3	4	5	6	7	8	9	10	E
1		1									0
2			1								0
3				1	1						1
4											0
5						1	1				1
6					1						0
7								1			0
8									1		0
9										1	0
10			1								0
										SUM(E)+1	3

Merujuk pada tabel 3.4 pada bab 3 untuk pengujian nomor *path* pada butir uji, maka dibuatlah tabel kesimpulan algoritma A* seperti yang ditunjukkan pada tabel 4.13 sebagai berikut:

Tabel 4.13. Kesimpulan Algoritma A*

No Path	Butir Uji	Hasil yang diharapkan	Hasil sesuai kasus uji	Kesimpulan
1	Path.name = posisi target (5,2)	Membuat jalur pencarian	Membuat jalur pencarian	Sesuai dengan yang diharapkan
2	Path.name dengan kondisi penghalang ((3,1),(3,2),(3,3))	Musuh menemukan penghalang dan tidak dapat dilalui	Musuh menemukan penghalang dan tidak dapat dilalui	Sesuai dengan yang diharapkan
3	Path.name mendapatkan nilai terendah (2,2)	Musuh dapat menemukan posisi dengan nilai terendah	Musuh dapat menemukan posisi dengan nilai terendah	Sesuai dengan yang diharapkan

4.2.2. 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.

4.2.2.1. Kuesioner

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

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

dimana,

P = Nilai presentase yang dicari

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

skorideal = 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* edukasi *Smart Labyrinth* ini menarik untuk dimainkan?

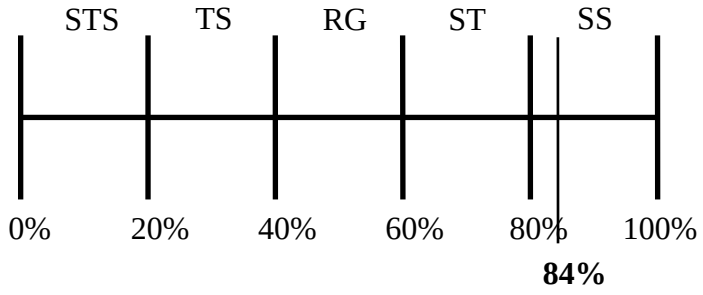
   Hasil kuesioner untuk pertanyaan ke-1 dapat dilihat pada tabel 4.14.

Tabel 4.14. Hasil Kuesioner Pertanyaan 1

Pertanyaan	Jawaban	Skor	Responden	S
1	Sangat Setuju	5	8	40
	Setuju	4	20	80
	Ragu-Ragu	3	2	6
	Tidak Setuju	2	0	0
	Sangat Tidak Setuju	1	0	0
JUMLAH			**30**	**126**

$$P = \frac{126}{150}\ x\ 100\ \% = \ 84\ \%$$

Dapat disimpulkan bahwa penilaian terhadap menarik tidaknya pada aplikasi *Game* Edukasi *Smart Labyrinth* adalah 84% dari yang diharapkan 100%, atau dapat dikategorikan sebagai sangat setuju.



Gambar 4.13. Pengukuran Garis Kontinum Pertanyaan 1

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

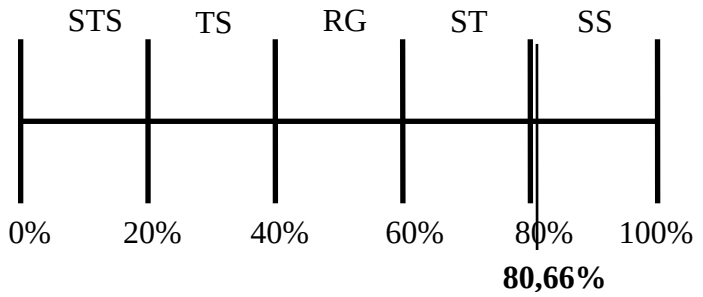
Hasil Kuesioner untuk pertanyaan ke-2 dapat dilihat pada tabel 4.15.

Tabel 4.15. Hasil Kuesioner Pertanyaan 2

Pertanyaan	Jawaban	Skor	Responden	S
2	Sangat Setuju	5	6	30
	Setuju	4	19	76
	Ragu-Ragu	3	5	15
	Tidak Setuju	2	0	0
	Sangat Tidak Setuju	1	0	0
JUMLAH			**30**	**121**

$$P = \frac{121}{150}\ x\ 100\ \% = \ 80{,}66\ \%$$

Dapat disimpulkan bahwa penilaian terhadap tampilan secara umum dari aplikasi *Game* Edukasi *Smart Labyrinth* adalah 80,66% dari yang diharapkan 100%, atau dapat dikategorikan sebagai setuju.



Gambar 4.14. Pengukuran Garis Kontinum Pertanyaan 2

3. Apakah *sound* yang dimiliki *game* sudah bagus?

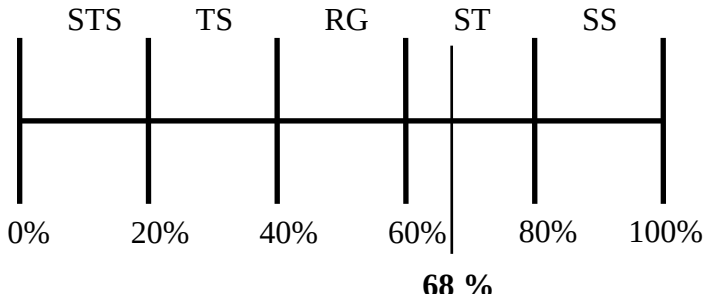
   Hasil kuesioner untuk pertanyaan ke-3 dapat dilihat pada tabel 4.16.

Tabel 4.16. Hasil Kuesioner Pertanyaan 3

Pertanyaan	Jawaban	Skor	Responden	S
3	Sangat Setuju	5	3	15
	Setuju	4	8	32
	Ragu-Ragu	3	17	51
	Tidak Setuju	2	2	4
	Sangat Tidak Setuju	1	0	0
JUMLAH			**30**	**102**

$$\frac{102}{150}\ x\ 100\ \%\ =\ 68\ \%$$

Dapat disimpulkan bahwa penilaian terhadap *sound* yang dimiliki pada aplikasi *Game* Edukasi *Smart Labyrinth* adalah 68% dari yang diharapkan 100%, atau dapat dikategorikan sebagai setuju.



Gambar 4.15. Pengukuran Garis Kontinum Pertanyaan 3

4. Apakah penerapan konsep labirin dan menempatkan musuh dalam *game* sudah sesuai?

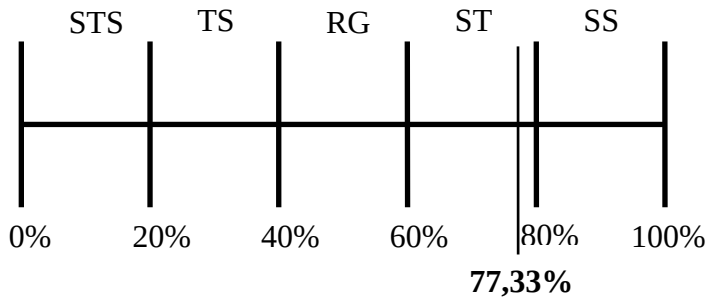
   Hasil kuesioner untuk pertanyaan ke-4 dapat dilihat pada tabel 4.17.

Tabel 4.17. Hasil Kuesioner Pertanyaan 4

Pertanyaan	Jawaban	Skor	Responden	S
4	Sangat Setuju	5	8	40
	Setuju	4	13	52
	Ragu-Ragu	3	6	18
	Tidak Setuju	2	3	6
	Sangat Tidak Setuju	1	0	0
JUMLAH			**30**	**116**

$$P = \frac{116}{150} x\ 100\ \% = \ 77{,}33\ \%$$

Dapat disimpulkan bahwa penilaian terhadap penerapan konsep labirin dan menempatkan musuh pada aplikasi *Game* Edukasi *Smart Labyrinth* sudah sesuai adalah 77,33% dari yang diharapkan 100%, atau dapat dikategorikan sebagai setuju.



Gambar 4.16. Pengukuran Garis Kontinum Pertanyaan 4

5. Apakah penyajian konten edukasi matematika yang terdapat dalam *game* sudah baik?

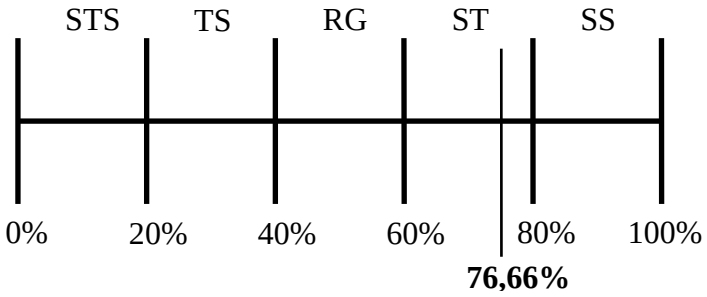
   Hasil kuesioner untuk pertanyaan ke-5 dapat dilihat pada tabel 4.18.

Tabel 4.18. Hasil Kuesioner Pertanyaan 5

Pertanyaan	Jawaban	Skor	Responden	S
5	Sangat Setuju	5	10	50
	Setuju	4	14	56
	Ragu-Ragu	3	3	9
	Tidak Setuju	2	3	6
	Sangat Tidak Setuju	1	0	0
JUMLAH			**30**	**115**

$$P = \frac{115}{150} x\ 100\ \% = \ 76{,}66\ \%$$

Dapat disimpulkan bahwa penilaian terhadap penyajian konten edukasi matematika pada aplikasi *Game* Edukasi *Smart Labyrinth* sudah baik adalah 76,66% dari yang diharapkan 100%, atau dapat dikategorikan sebagai setuju.



Gambar 4.17. Pengukuran Garis Kontinum Pertanyaan 5

6. Apakah konten edukasi matematika yang dimiliki *game* membantu meningkatkan minat kamu dalam mempelajari matematika?

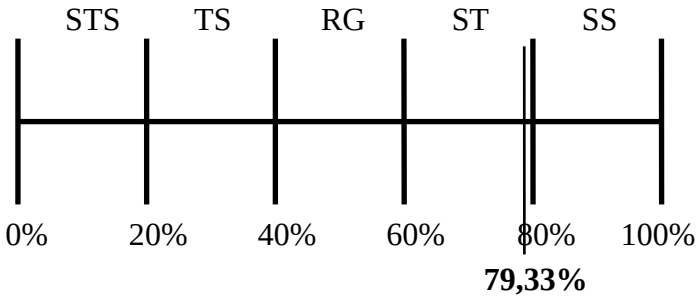
   Hasil kuesioner untuk pertanyaan ke-6 dapat dilihat pada tabel 4.19.

Tabel 4.19. Hasil Kuesioner Pertanyaan 6

Pertanyaan	Jawaban	Skor	Responden	S
6	Sangat Setuju	5	5	25
	Setuju	4	19	76
	Ragu-Ragu	3	6	18
	Tidak Setuju	2	0	0
	Sangat Tidak Setuju	1	0	0
JUMLAH			**30**	**119**

$$P = \frac{119}{150}\ x\ 100\ \% = \ 79{,}33\ \%$$

Dapat disimpulkan bahwa penilaian terhadap konten edukasi matematika pada aplikasi *Game* Edukasi *Smart Labyrinth* dapat membantu meningkatkan minat anak-anak dalam memepelajari matematika adalah 79,33% dari yang diharapkan 100%, atau dapat dikategorikan sebagai setuju.



Gambar 4.18. Pengukuran Garis Kontinum Pertanyaan 6