BAB IV
IMPLEMENTASI DAN PENGUJIAN

Pada bab ini akan dilakukan implementasi dan pengujian terhadap aplikasi permainan yang dibangun, yaitu *Game* Education "Ayo Belajar Puasa 2D". Hasil perancangan pada tahap sebelumnya kemudian diimplementasikan ke dalam bahasa pemrograman. Setelah tahap implementasi selesai, maka dilakukan tahap pengujian terhadap aplikasi *Game* Education "Ayo Belajar Puasa 2D".

IV.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.

IV.1.1 Implementasi Perangkat Keras

Perangkat keras yang digunakan untuk menjalankan *Game* Education "Ayo Belajar Puasa 2D" ini tidak harus yang berspesifikasi tinggi. Spesifikasi minimum yang dapat digunakan untuk menjalankan *Game* Education "Ayo Belajar Puasa 2D" adalah sebagai berikut :

1. Processor 2,0 GHz
2. RAM 1 GB
3. VGA 512 MB
4. Harddisk 80 GB
5. Monitor 17"
6. Keyboard

IV.1.2 Implementasi Perangkat Lunak

Adapun spesifikasi perangkat lunak yang digunakan untuk menjalankan menjalankan *Game* Education "Ayo Belajar Puasa 2D" adalah perangkat komputer yang telah terinstal Java.

IV.1.3 Implementasi Aplikasi

Ayo Belajar Puasa merupakan aplikasi permainan ber-*genre plafrom*. Pemain akan melakukan petualangan di sebuah lingkungan khidupan sehari-hari, Disetiap level, pemain akan berperan sebagai seorang anak laki-laki berumur lima tahun bernama Naofal, yang sedang berpuasa dengan nenahan lapar dan hawa nafsunya.

Permainan dikatakan *game over* apabila jumlah buku pemain berkurang menjadi 0, dan sebaliknya permainan dikatakan berhasil apabila pemain dapat menghindari bos dengan mempertahankan poin disetiap *level*-nya dengan baik.

Aplikasi permainan ini menggunakan *Greenfoot* dan teknik interaksi *Side Scrolling* yang dibangun dengan menggunakan bahasa pemrograman *Java.* Sehingga permainan ini akan berjalan apabila pemain telah menginstal *Java* pada perangkatnya.

IV.1.4 Implementasi Antar Muka

Adapun implementasi antarmuka pada aplikasi permainan Ayo Belajar Puasa adalah sebagai berikut :

1. Antarmuka Menu Utama

   Antarmuka menu utama merupakan tampilan menu utama pada saat pertama kali pemain membuka aplikasi permainan. Tampilan dari antarmuka menu utama dapat dilihat pada gambar 4.1.



Gambar IV.1.Tampilan Antarmuka Menu Utama

2. Antarmuka Intro Permainan

Antarmuka Intro permainan merupakan tampilan ketika pemain mengklik tombol mulai permainan sehingga langsung masuk ke *intro* permainan yang merupakan cerita tentang kehidupan Naofal yang sedang belajar berpuasa. Tampilan dari antarmuka intro permainan dapat dilihat pada gambar 4.2.

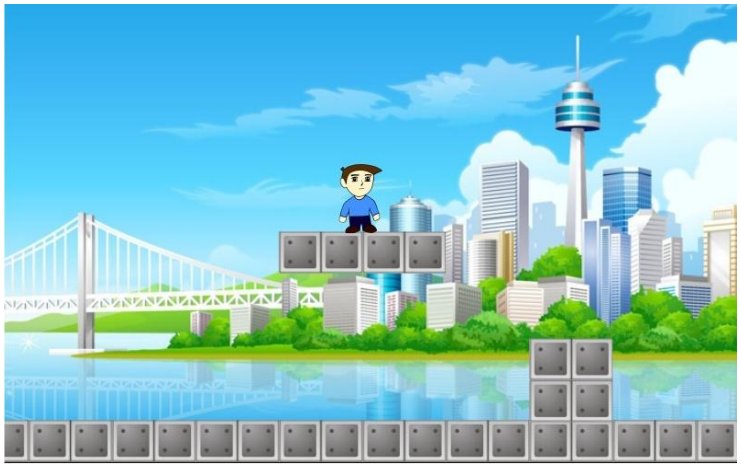


Gambar IV.2.Tampilan Antarmuka Intro Permainan

3. Antarmuka Mulai Permainan

Antarmuka mulai permainan merupakan tampilan ketika pemain selesai mengikuti intro permainan dan langsung masuk ke level 1.

Tampilan dari antarmuka mulai permainan level 1 dapat dilihat pada gambar 4.3.



Gambar IV.3.Tampilan Antarmuka Permainan level 1

IV.2 Pengujian

Dalam melakukan pengujian terhadap *Game* Edukasi Ayo Belajar Pussa 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.

IV.2.1 Pengujian Alpha

Pada tahap ini, pengujian dilakukan untuk memastikan *Game* Edukasi Ayo Belajar Puasa telah berjalan dengan benar dan sesuai dengan kebutuhan yang diinginkan. Pengujian *alpha* menitik beratkan kepada persyaratan fungsional perangkat lunak.

IV.2.1.1 Skenario Pengujian Aplikasi

Skenario pengujian menjelaskan pengujian terhadap sistem yang ada pada *Game* Edukasi Ayo Belajar Puasa.Skenario pengujian dari aplikasi *Game* Edukasi Ayo Belajar Puasa dapat dilihat pada tabel 4.1.

Tabel 4.1. Skenario Pengujian Aplikasi Permainan Ayo Belajar Puasa

No	Komponen yang di uji	Skenario Pengujian	Jenis Pengujian
1	Menu Utama	Klik Menu Pembuat	*Black Box*
		Klik Menu Bantuan	*Black Box*
		Klik Menu Bermain	*Black Box*
		Klik Menu Keluar	*Black Box*
2	Menu Mulai Permainan	Klik Menu Mulai Permainan	*Black Box*
		Permainan *Level* 1	*Black Box*
3	Menu Petunjuk	Klik Menu Petunjuk	*Black Box*
		Klik Tombol Kembali	*Black Box*
4	Menu Pembuat	Klik Menu Pembuat	*Black Box*
		Klik Tombol Kembali	*Black Box*
5	Menu Keluar	Klik Menu Keluar	*Black Box*
6	*Collision Detection*	Deteksi Tabrakan Entitias Buku Dengan Entitas Musuh	*White Box*
		Deteksi Tabrakan Entitas Pemain Dengan Entitas Musuh	*White Box*
7	Algoritma pencarian (A star)	Algoritma pencarian Pada tiap *Level*	*White Box*

IV.2.1.2 Kasus Dan 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

A. Pengujian Menu Mulai Permainan

Pengujian mulai permainan dengan kasus dan hasil uji, kemudian diamati dan menghasilkan kesimpulan dapat dilihat pada tabel 4.2.

Tabel 4.2. Pengujian Mulai Permainan

Kasus dan Hasil Uji (Data Benar)			
Masukan	Keluaran yang diharapkan	Pengamatan	Kesimpulan
Klik mulai permainan	Dapat menampilkan intro permainan dan berlanjut ke map *level* 1	Menampilkan intro permainan dan berlanjut ke map *level* 1. Sesuai yang diharapkan.	[√] Diterima [ ] Ditolak
Menyelesaikan misi *level* 1	Dapat menampilkan map *level* 2.	Menampilkan map *level* 2. Sesuai yang diharapkan.	[√] Diterima [ ] Ditolak
Menyelesaikan misi *level* 2	Dapat menampilkan map *level 3*	Menampilkan map *level* 3. Sesuai yang diharapkan.	[√] Diterima [ ] Ditolak
Menyelesaikan misi *level* 3	Dapat menampilkan keterangan menang	Menampilkan keterangan menang. Sesuai yang diharapkan.	[√] Diterima [ ] Ditolak
Game Over	Dapat menampilkan keterangan *Game* Over	Menampilkan keterangan *Game* Over. Sesuai yang diharapkan.	[√] Diterima [ ] Ditolak

B. Pengujian Menu Petunjuk

Pengujian petunjuk dengan kasus dan hasil uji, kemudian diamati dan menghasilkan kesimpulan dapat dilihat pada tabel 4.3.

Tabel 4.3. Pengujian Tampil Petunjuk

Kasus dan Hasil Uji (Data benar)			
Masukan	Keluaran yang diharapkan	Pengamatan	Kesimpulan
Menekan tombol Petunjuk	Dapat menampilkan informasi petunjuk permainan	Menampilkan informasi petunjuk permainan. Sesuai yang diharapkan.	[√] Diterima [ ] Ditolak
Menekan tombol kembali	Menampilkan menu utama	Dapat menampilkan menu utama. Sesuai yang diharapkan.	[√] Diterima [ ] Ditolak

C. Pengujian Menu Pembuat.

Pengujian menu pembuat dengan kasus dan hasil uji, kemudian diamati dan menghasilkan kesimpulan dapat dilihat pada tabel 4.4.

Tabel 4.4. Pengujian Tampil Pembuat

Kasus dan Hasil Uji(Data benar)			
Masukan	Keluaran yang diharapkan	Pengamatan	Kesimpulan
Menekan tombol Pembuat	Dapat menampilkan informasi pembuat aplikasi permainan.	Menampilkan informasi pembuat aplikasi permainan. Sesuai yang diharapkan	[√] Diterima [ ] Ditolak
Menekan tombol kembali	Dapat menampilkan menu utama	Menampilkan menu utama. Sesuai yang diharapkan	[√] Diterima [ ] Ditolak

D. Pengujian Menu Keluar

Pengujian menu Keluar dengan kasus dan hasil uji, kemudian diamati dan menghasilkan kesimpulan dapat dilihat pada tabel 4.5.

Tabel 4.5. Pengujian Keluar Permainan

Kasus dan Hasil Uji(Data benar)			
Masukan	Keluaran yang diharapkan	Pengamatan	Kesimpulan
Menekan tombol Keluar	Dapat keluar dari permaian.	Menampilkan keluar dari permaian. Sesuai yang diharapkan	[√] Diterima [ ] Ditolak

IV.2.1.3 Kasus dan Hasil Pengujian White Box

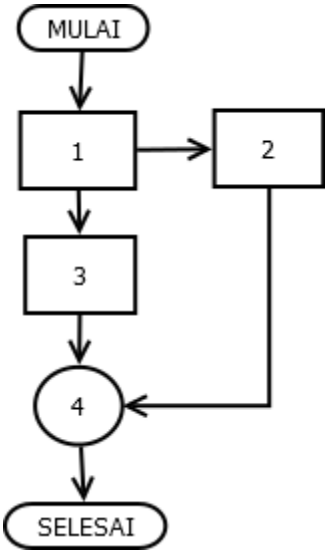
Di bawah ini merupakan kasus *collision detection* dan algoritma pencarian untuk menguji perangkat lunak yang sudah dibangun dengan menggunakan metode *white box*, berdasarkan skenario pengujian aplikasi yang sudah dijabarkan.

A. Pengujian *Collision Detection*

Pengujian *collision detection* merupakan proses dimana akan diuji deteksi tubrukan antara buku dengan musuh dan juga antara musuh dengan pemain pada *Game* edukasi Ayo Belajar Puasa.

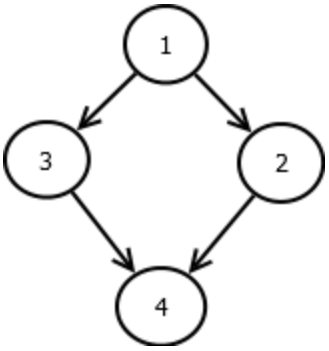
a. *Collision Detection* Buku dengan Musuh

Adapun *Flowchart* pengujian pada *collision detection* buku dengan musuh dapat dilihat pada gambar 4.6.



Gambar IV.4.Flowchart Pengujian Collision Detection Buku dengan Musuh

Flowgraph yang merupakan hasil dari perubahan *flowchart* dapat dilihat pada gambar 4.5.



Gambar IV.5.Flow Graph Collision Detection Buku dengan Musuh

Keterangan : = Menggambarkan kondisi

= Menggambarkan aksi

Tahap Pengujian *Collision Detection* buku dengan Musuh

1. *Cyclomatic Complexity* V(G)

   Edge = 4, Node = 4

   V(G) = Edge – Node + 2

   V(G) = 4 – 4 + 2

   V(G) = 0 + 2

   V(G) = 2

2. *Independent Path*

   Path 1 = 1, 2, 4

   Path 2 = 1, 3, 4

3. *Predicate Node*

   V(G) = Jumlah node yang memiliki lebih dari 1 jalur + 1

   V(G) = 1 + 1

   V(G) = 2

4.Graph Matriks

Graph Matriks pengujian pada *collision detection* buku dengan musuh dapat dilihat pada tabel 4.6.

Tabel 4.6. Graph Matriks Collision Detection Peluru dengan Musuh

	1	2	3	4	
1		1	1		1
2				1	0
3				1	0
4					0
Hasil					1

V(G) = Jumlah *Graph Matriks* + 1

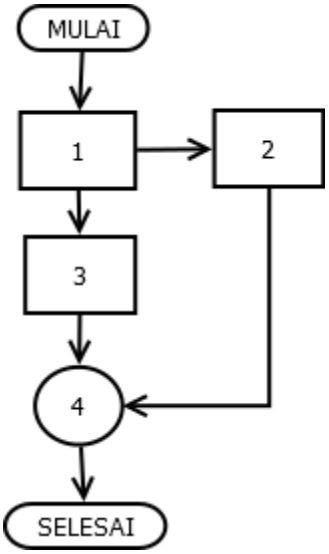
V(G) = 1 + 1

V(G) = 2

Berdasarkan hasil pengujian dihasilkan nilai *Cyclomatic Complexity* yang sama yaitu 2, maka dapat disimpulkan bahwa pada proses *collision detection* buku dengan musuh berjalan dengan baik, karena setiap pengujian menghasilkan nilai yang sama.

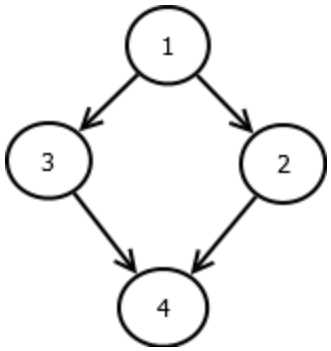
b. *Collision Detection* Musuh dengan Pemain

Adapun *Flowchart* pengujian pada *collision detection* musuh dengan brankas dapat dilihat pada gambar 4.6.



Gambar IV.6.*Flowchart* Pengujian *Collision Detection* Musuh dengan Brankas

Flowgraph yang merupakan hasil dari perubahan *flowchart* dapat dilihat pada gambar 4.7.



Gambar IV.7.Flowgraph Collision Detection Musuh dengan Pemain

Keterangan : = Menggambarkan kondisi

= Menggambarkan aksi

Tahap Pengujian *Collision Detection* buku dengan Musuh

1. *Cyclomatic Complexity* V(G)

   Edge = 4, Node = 4

   V(G) = Edge – Node + 2

   V(G) = 4 – 4 + 2

   V(G) = 0 + 2

   V(G) = 2

2. *Independent Path*

   Path 1 = 1, 2, 4

   Path 2 = 1, 3, 4

3. *Predicate Node*

   V(G) = Jumlah node yang memiliki lebih dari 1 jalur + 1

   V(G) = 1 + 1

   V(G) = 2

4. *Graph Matriks*

   Graph Matriks pengujian pada *collision detection* pemain dengan musuh dapat dilihat pada tabel 4.7.

Tabel 4.7. Graph Matriks Collision Detection Musuh dengan Pemain

	1	2	3	4	
1		1	1		1
2				1	0
3				1	0
4					0
Hasil					1

V(G) = Jumlah *Graph Matriks* + 1

V(G) = 1 + 1

V(G) = 2

Berdasarkan hasil pengujian dihasilkan nilai *Cyclomatic Complexity* yang sama yaitu 2, maka dapat disimpulkan bahwa pada proses *collision detection* pemain dengan musuh berjalan dengan baik, karena setiap pengujian menghasilkan nilai yang sama.

B. Pengujian Algoritma pencarian

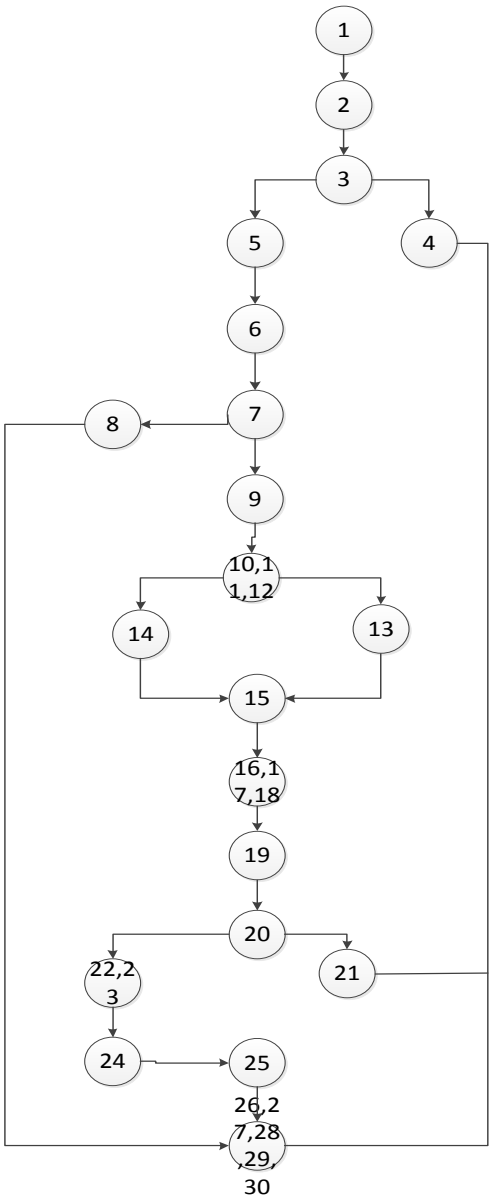
Pengujian Algoritma pencarian merupakan proses dimana akan diuji terhadap musuh dalam melakukan pencarian rute terpendek yang terdapat di

setiap akhir *level* pada *Game* Edukasi Ayo Belajar Puasa.

Berikuta dalah *sourcecode* metode *path finding* dengan menggunakan algoritma Pencarian :

```
1: var Open = S
   var Closed = array;
2: while sukses or open==null{
3: if (open==null){
4: gagal;
5: { else
6: BestNode==Simpul di open=min F;
   pindah BestNode dari open ke closed;
7: if (BestNode==goal)
8: sukses;
9: } else
10. bangkitkan suksesor BestNode
11. for(suksesor=0,suksesor<suksesor.length)
    g(suksesor)=g(BestNode)+actual cost(BestNode)
12. suksesor);
    Check suksesor;
13. if (suksesor==open)
14. old=simpul di open;
    BestNode=old;
15. if (g(old)=better)
16. ubah parent old ke BestNode;
    Ubah g dan f pada old;
17. }
18. }
19. if suksesor==closed)
20. old=simpul di closed;
21. if(g(old)=better)
22. ubah parent old ke BestNode
    ubah nilai g dan f
23. While(suksesor!=open)
24. if (suksesor==open)
25. continue;
26. else {
27. masukan suksesor ke open;
    Tambahkan suksesor=BestNode
    f = g + h
28. }
29. }
30. }
```

Berdasarkan *sourcecode* di atas maka *flow graph* algoritma pencarian adalah dapat dilihat pada gambar 4.13:



Gambar IV. 8 *Flow Graph* Algoritma Pencarian

Keterangan:

= Menggambarkan jumlah edges.

= Menggambarkan jumlah node/simpul.

Dari Gambar 4.13 dapat dihitung *cyclomatic complexity* sebagai berikut :

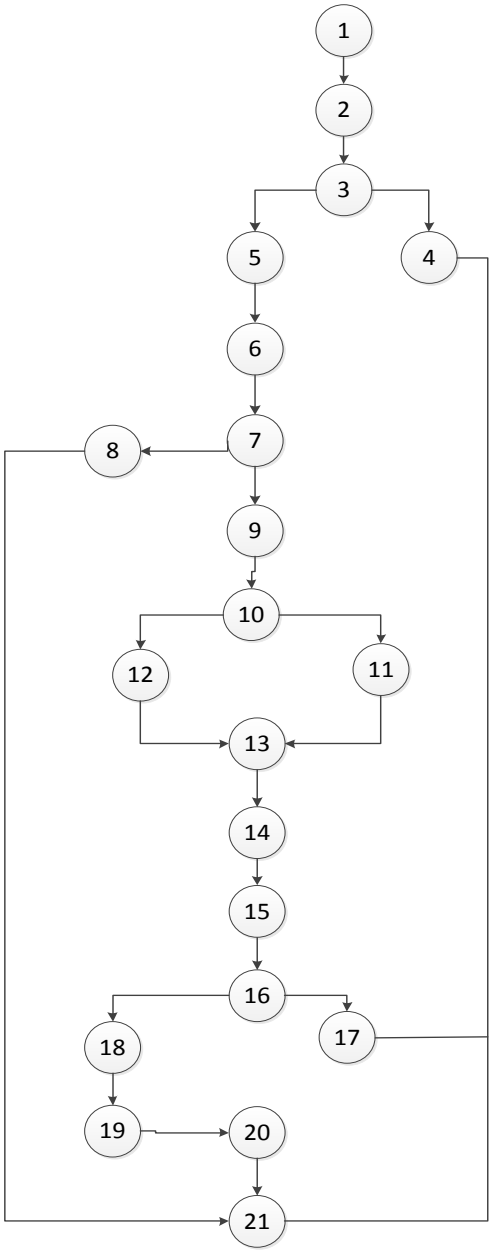
1) V(G) = E – N + 2
2) V(G) = 24 – 21 + 2
3) V(G) = 3 + 2 = 5

Keterangan :E = jumlah edges/panah, N = jumlah node/simpul

Jadi, *cyclomatic complexity* untuk gambar 4.13 adalah 5. Berdasarkan *cyclomaticcomplexity* tersebut, maka terdapat 5 *path* yang terdiri dari :

1) *Path 2* = 1,2,3,5,6,7,9,10,11,12,15,16,17,18,19,20,22,23,25,26,27,28,29,30
2) *Path 2* = 1,2,3,4,26,27,28,29,30
3) *Path 3* = 1,2,3,5,6,7,8,26,27,28,29,30
4) *Path4*3 = 1,2,3,5,6,7,9,10,11,12,13,14,15,16,17,18,19,20,21,26,27,28,29,30
5) *Path 5* = 1,2,3,5,6,7,9,10,11,12,13,14,15,16,17,18,19,20,22,23,24,25,26,27,28,29,30

Penomoran ulang *flow graph* atau penyederhanaan berdasarkan kondisi atau simpul sebagai berikut.



Gambar IV.9 Penomoran ulang *Flow Graph*

Graph Matriks Algoritma Pencarian

Tabel IV.1 *Graph Matriks* Algoritma Pencarian

N	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	N(E)-1
1		1																				1-1=0
2			1																			1-1=0
3				1	1																	2-1=1
4																					1	1-1=0
5						1																1-1=0
6							1															1-1=0
7								1	1													2-1=1
8																					1	1-1=0
9										1												2-1=1
10											1	1										1-1=0
11													1									1-1=0
12													1					1				2-1=1
13														1								1-1=0
14															1							1-1=0
15																1						1-1=0
16																	1	1				2-1=1
17																					1	1-1=0
18																			1			1-1=0
19																				1		
20																					1	
21																						
Sum (E)+1																						4+1=5

Keterangan:

1. Baris dan kolom merepresentasikan simpul.
1. Nilai 1 merepresentasikan adanya keterhubungan antar simpul. Hasil Uji Algoritma Pencarian.

Tabel 4.9 Hasil Uji Algoritma Pencarian

No	Kasus uji	Pengamatan	Yang diharapkan	Keterangan

1	1: var Open = S
	var Closed = array;
	2: while sukses or open==null{
	3: if (open==null){
	5: { else
	6: BestNode==Simpul di open=min F;
	pindah BestNode dari open ke closed;
	7: if (BestNode==goal)
	8: sukses;
	9: } else
	10. bangkitkan suksesor BestNode
	11. for(suksesor=0,suksesor<suksesor.length)
	g(suksesor)=g(BestNode)+actual cost(BestNode)
	12. suksesor);
	Check suksesor;
	15. if (g(old)=better)
	16. ubah parent old ke BestNode;
		membuka open list tidak kosong, dan menemukan tujuan kemudian berhenti	Mencari node tujuan kemudian berhenti	[✓]dikerjakan
				[]tidak dikerjakan

No	Kasus uji	Pengamatan	Yang diharapkan	Keterangan

	Ubah g dan f pada old;
	17. }
	18. }
	19. if suksesor==closed)
	20. old=simpul di closed;
	21.
	if(g(old)=better)
	23.
	While(suksesor!=open)
	25. continue;
	26. else {
	27. masukan suksesor ke open;
	Tambahkan suksesor=BestNode
	f = g + h
	28. }
	29. }
	30. }			

2	1: var Open = S
	var Closed = array;
	2: while sukses or open==null{
	3: if (open==null){
	4: gagal;
	26. else {
	27. masukan suksesor ke open;
	Tambahkan suksesor=BestNode
	f = g + h
	28. }
	29. }
	30. }
		Menentukan node tujuan dengan nilai f terkecil.	Node n' ditemukanndari openList dengan f(n) terkecil	[✓]dikerjakan
				[] tidakdikerjakan

No	Kasus uji	Pengamatan	Yang diharapkan	Keterangan

3	1: var Open = S
	var Closed = array;
	2: while sukses or open==null{
	3: if (open==null){
	5: { else
	6: BestNode==Simpul di open=min F;
	pindah BestNode dari open ke closed;
	7: if (BestNode==goal)
	8: sukses;
	26. else {
	27. masukan suksesor ke open;
	Tambahkan suksesor=BestNode
	f = g + h
	28. }
	29. }
	30. }
		Menentukan *best node*	Menentukan
			best node
				[✓]dikerjakan
				[]
				Tidak dikerjakan

4	1: var Open = S
	var Closed = array;
	2: while sukses or open==null{
	3: if (open==null){
	5: { else
	6: BestNode==Simpul di open=min F;
	pindah BestNode dari open ke closed;
	7: if (BestNode==goal)
	9: } else
	10. bangkitkan suksesor BestNode
	11.
		Membangkitkan best node.
		.
			Membangkitkan best node	[✓]dikerjakan
				[]
				Tidak dikerjakan

No	Kasus uji	Pengamatan	Yang diharapkan	Keterangan

	for(suksesor=0,suksesor<suksesor.length)
	
	g(suksesor)=g(BestNode)+actual cost(BestNode)
	12. suksesor);
	Check suksesor;
	13. if (suksesor==open)
	14. old=simpul di open;
	BestNode=old;
	15. if (g(old)=better)
	16. ubah parent old ke BestNode;
	Ubah g dan f pada old;
	17. }
	18. }
	19. if suksesor==closed)
	20. old=simpul di closed;
	21. if(g(old)=better)
	22. ubah parent old ke BestNode
	ubah nilai g dan f
	23. While(suksesor!=open)
	24. if (suksesor==open)
	25. continue;
	26. else {
	27. masukan suksesor ke open;
	Tambahkan suksesor=BestNode			

No	Kasus uji	Pengamatan	Yang diharapkan	Keterangan
	f = g + h 28. } 29. } 30. }			

5	1: var Open = S
	var Closed = array;
	2: while sukses or open==null{
	3: if (open==null){
	5: { else
	6: BestNode==Simpul di open=min F;
	pindah BestNode dari open ke closed;
	7: if (BestNode==goal)
	9: } else
	10. bangkitkan suksesor BestNode
	11. for(suksesor=0,suksesor<suksesor.length)
	
	g(suksesor)=g(BestNode)+actual cost(BestNode)
	12. suksesor);
	Check suksesor;
	13. if (suksesor==open)
	14. old=simpul di open;
	BestNode=old;
	15. if (g(old)=better)
	16. ubah parent old ke BestNode;
	Ubah g dan f pada old;
	17. }
		Mengecek hasil nilai yang di uji check best node dari node tujuan' ke node semula, Tiap node Hanya memiliki satu best node	Menghasilkan best node	[✓]dikerjakan
				[]
				
				Tidak dikerjakan

No	Kasus uji	Pengamatan	Yang diharapkan	Keterangan
	18. } 19. if suksesor==closed) 20. old=simpul di closed; 22. ubah parent old ke BestNode ubah nilai g dan f 23. While(suksesor!=open) 24. if (suksesor==open) 25. continue; 26. else { 27. masukan suksesor ke open; Tambahkan suksesor=BestNode f = g + h 28. } 29. } 30. }			

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

IV.2.1.4 Kesimpulan 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.

IV.2.2 Pengujian Beta

Pengujian beta merupakan pengujian yang dilakukan secara objektif, dimana pengujian langsung melibatkan para pengguna aplikasi permainan secara langsung dengan menggunakan kuisioner mengenai kepuasan para pengguna terhadap *game* edukasi yang dibangun. Pengujian dilakukan terhadap 30 responden dengan usia 6 tahun sampai 12 tahun dan setiap responden diberikan 5 pernyataan untuk kemudian dilakukan penilaian dengan memilih 5 pilihan yang telah disediakan untuk setiap pernyataannya.

IV.2.2.1. Kuesioner

Data hasil kuisioner yang telah diberikan dapat dicari presentasinya dari masing-masing jawaban dengan menggunakan rumus sebagai berikut :

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

Keterangan:

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 kuisioner terhadap 30 anak yang berusia 6- 12 tahun adalah sebagai berikut :

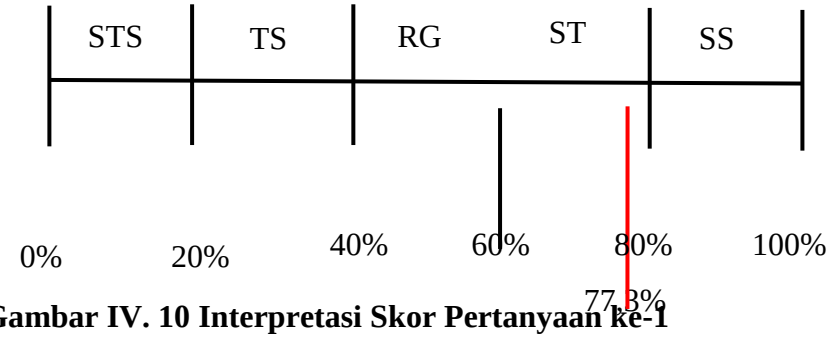
1. Apakah Adik-adik senang memainkan *Game* “Ayo Belajar Puasa” 2D ini?
   Hasil kuisioner untuk pertanyaan ke-1 dapat dilihat pada tabel 4.9.

Tabel IV. 10 Hasil Kuisioner Pertanyaan Pertama

Pernyataan	Jawaban	Skor	Responden	S
1	Sangat Setuju	5	11	55
	Setuju	4	11	44
	Ragu-Ragu	3	3	9
	Tidak Setuju	2	3	6
	Sangat Tidak Setuju	1	2	2
JUMLAH			**30**	**116**

$$P = \frac{116}{150} x100\% = 77{,}3\%$$

Berdasarkan hasil presentase nilai di atas, maka dapat disimpulkan bahwa tingkat persetujuan *game* edukasi "Ayo Belajar Puasa" menyenangkan dimainkan oleh anak-anak adalah sebesar adalah 77,3% dari yang diharapkan 100%, atau dapat dikategorikan sebagai setuju, untuk lebih jelasnya dapat dilihat pada gambar 4.10



Gambar IV. 10 Interpretasi Skor Pertanyaan ke-1

2. Apakah Adik-adik lebih suka belajar tentang puasa di sekolah atau lewat *Game* Education seperti "Ayo Belajar Puasa" 2D ini?

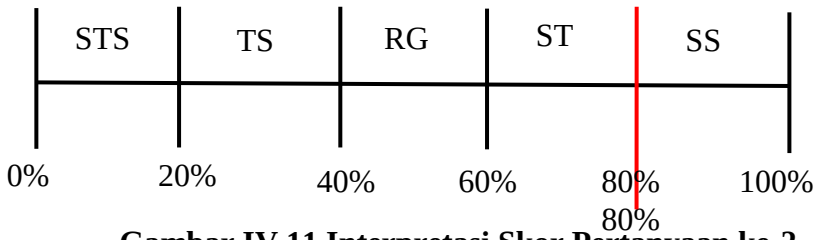
   Hasil kuisioner untuk pertanyaan ke-2 dapat dilihat pada tabel 4.10.

Tabel 4.8. Hasil kuisioner Pertanyaan ke-2

Pernyataan	Jawaban	Skor	Responden	S
2	Sangat Setuju	5	10	50
	Setuju	4	14	56
	Ragu-Ragu	3	3	9
	Tidak Setuju	2	2	4
	Sangat Tidak Setuju	1	1	1
JUMLAH			**30**	**120**

$$P = \frac{120}{150} x100\% = 80\%$$

Berdasarkan hasil presentase nilai di atas, maka dapat disimpulkan bahwa tingkat persetujuan belajar lewat *game* edukasi "Ayo Belajar Puasa" ini lebih disukai daripada belajar tentang berpuasa di Sekolah adalah sebesar adalah 80% dari yang diharapkan 100%, atau dapat dikategorikan sebagai sangat setuju, untuk lebih jelasnya dapat dilihat pada gambar 4.11.



Gambar IV.11 Interpretasi Skor Pertanyaan ke-2

3. Apakah setelah adik-adik memainkan *Game* "Ayo Belajar Puasa" 2D ini jadi lebih mengetahui tentang makna puasa itu sendiri?

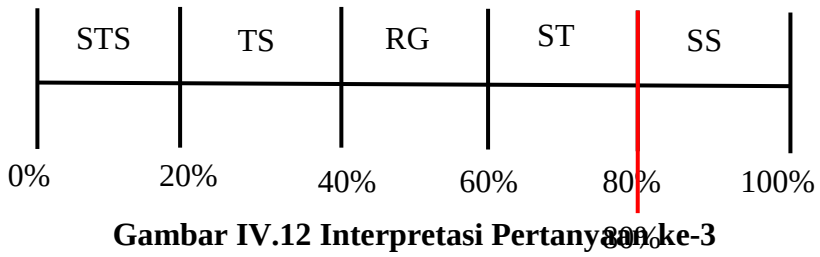
   Hasil kuisioner untuk pernyataan ke-3 dapat dilihat pada tabel 4.11.

Tabel 4.9. Hasil kuisioner pertanyaan ke-3

Pernyataan	Jawaban	Skor	Responden	S
3	Sangat Setuju	5	11	55
	Setuju	4	13	52
	Ragu-Ragu	3	2	6
	Tidak Setuju	2	3	6
	Sangat Tidak Setuju	1	1	1
JUMLAH			**30**	**120**

$$P = \frac{120}{150} x100\% = 80\%$$

Berdasarkan hasil presentase nilai di atas, maka dapat disimpulkan bahwa tingkat persetujuan bahwa Anak-anak tersebut jadi lebih mengerti tentang makna puasa yang sebenarnya adalah sebesar adalah 80% dari yang diharapkan 100%, atau dapat dikategorikan sebagai sangat setuju, untuk lebih jelasnya dapat dilihat pada gambar 4.12.



Gambar IV.12 Interpretasi Pertanyaan ke-3

4. Apakah setelah Adik-adik memainkan *Game* "Ayo Belajar Puasa" 2D jadi lebih mengetahui apa yang sebenarnya harus dilakukan saat sedang berpuasa selain menahan lapar dan dahaga?

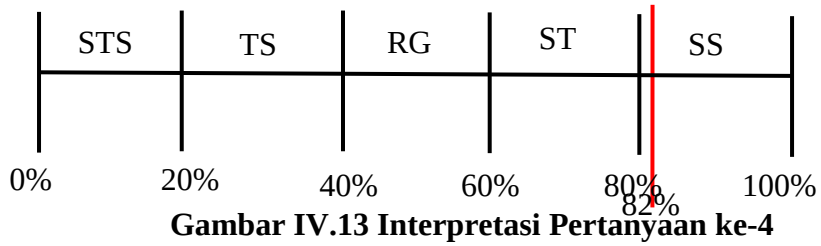
   Hasil kuisioner untuk pertanyaan ke-4 dapat dilihat pada tabel 4.12.

Tabel 4.10. Hasil kuisioner pertanyaan ke-4

Pernyataan	Jawaban	Skor	Responden	S
4	Sangat Setuju	5	13	65
	Setuju	4	11	44
	Ragu-Ragu	3	3	9
	Tidak Setuju	2	2	4
	Sangat Tidak Setuju	1	1	1
JUMLAH			**30**	**123**

$$P = \frac{123}{150} x100\% = 82\%$$

Berdasarkan hasil presentase nilai di atas, maka dapat disimpulkan bahwa tingkat persetujuan bahwa Anak-anak jadi lebih mengetahui apa yang harus dilakukan apabila sedang berpuasa selain menahan lapar dan dahaga adalah sebesar 82% dari yang diharapkan 100%, atau dapat dikategorikan sebagai setuju, untuk lebih jelasnya dapat dilihat pada gambar 4.13.



Gambar IV.13 Interpretasi Pertanyaan ke-4

5. Apakah Adik-Adik suka dengan Karakter, Latar tempat dan Cerita yang ada di dalam *Game* "Ayo Belajar Puasa" 2D ini?

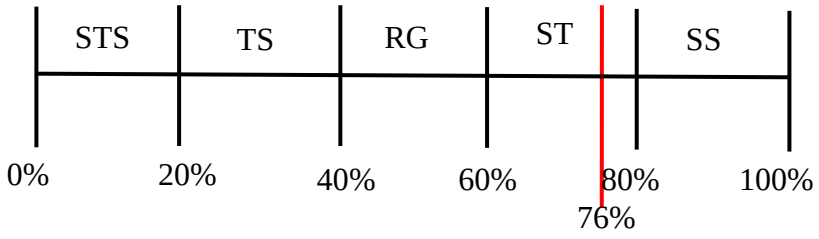
   Hasil kuisioner untuk pernyataan ke-5 dapat dilihat pada tabel 4.14.

Tabel 4.11. Hasil kuisioner ketertarikan pada aplikasi permainan

Pernyataan	Jawaban	Skor	Responden	S
5	Sangat Setuju	5	9	45
	Setuju	4	14	56
	Ragu-Ragu	3	2	6
	Tidak Setuju	2	2	4
	Sangat Tidak Setuju	1	3	3
JUMLAH			**30**	**114**

$$P = \frac{114}{150} x100\% = 76\%$$

Berdasarkan hasil presentase nilai di atas, maka dapat disimpulkan bahwa tingkat persetujuan bahwa Anak-anak menyukai Gambar-gambar, tempat dan Karakter yang ada dalam *Game* edukasi Ayo Belajar Puasa ini adalah sebesar 76% dari yang diharapkan 100%, atau dapat dikategorikan sebagai setuju, untuk lebih jelasnya dapat dilihat pada gambar 4.14.



Gambar IV.5 Interpretasi Pertanyaan ke-5

Berdasarkan jawaban terhadap kuesioner, maka dapat diambil kesimpulan yaitu:

- Kelebihan dari *Game* “Ayo Belajar Puasa” yaitu:
  1. *Game* ini dapat digunakan sebagai media pembelajaran tentang berpuasa dan dapat memberikan kesan menyenangkan.
  2. *Game* ini dapat dengan mudah dimengerti oleh pengguna untuk memainkannya dan dapat dapat menyajikan animasi yang menarik.
  3. *Game* ini dapat memberikan pemahaman bagi anak-anak tentang makna puasa yang sebenarnya tidak hanya menahan lapar dan dahaga.

- Kekurangan dari *Game* “Ayo Belajar Puasa” yaitu:
  Untuk anak-anak usia 6-12 tahun, penyajian materi yang disampaikan dan pesan-pesan sulit difahami.