BAB 4

IMPLEMENTASI DAN PENGUJIAN

4.1 Implementasi

Tahap implementasi merupakan tahap kelanjutan dari kegiatan perancangan sistem. Wujud dari hasil implementasi ini nantinya adalah sebuah sistem yang siap untuk diuji dan digunakan.

4.2 Perangkat Pendukung yang Digunakan

Dalam proses pembuatan aplikasi ini, tentunya membutuhkan perangkat keras (*Hardware*) dan perangkat lunak (*software*). Berikut merupakan penjelasan dari perangkat keras dan perangkat lunak yang digunakan.

4.2.1 Perangkat Keras (*Hardware*)

Perangkat keras yang digunakan selama pembangunan aplikasi ini memiliki spesifikasi sebagai berikut:

Tabel 4.1 Perangkat Keras yang Digunakan

Perangkat Keras	Keterangan
Processor	Intel Pentium 4 CPU 2.80GHz 2.10GHz
Memory	DDR2 2 GB
Harddisk	SATA 250 GB
Video Card	onBoard
Monitor	Resolusi 1366x768
Mouse	Optic 1 buah
Keyboard	1 buah

4.2.2 Perangkat Lunak (*Software*)

Perangkat lunak yang digunakan untuk digunakan dalam pembangunan game edukasi sejarah walisongo ini adalah sebagai berikut:

Tabel 4.2 Perangkat Lunak yang Digunakan

Perangkat Lunak	Keterangan
Windows 7 Ultimate	Sebagai sistem operasi
Greenfoot 2.2.0	Sebagai perangkat lunak pembangun
Microsoft Office 2007	Sebagai perangkat lunak penulisan laporan
StarUML	Sebagai perangkat lunak memodelkan analisi sistem

4.3 Implementasi Antar Muka

Berikut ini merupakan antarmuka *game* edukasi sejarah walisongo. Yang ditujukan untuk siswa-siswi kelas V SDN Karya Bhakti dan kelas VI MI Nidlomusibyan.

1. Tampilan Utama



Gambar 4.1 Tampilan Utama.

Layar ini berfungsi menampilkan Tampilan utama atau tampilan awal game dimana didalamnya tersidia empat pilihan, yaitu pilihan mulai, petunjuk, ucapan dan keluar.

2. Tampilan Petunjuk



Gambar 4.2 Tampilan Tempat Petunjuk

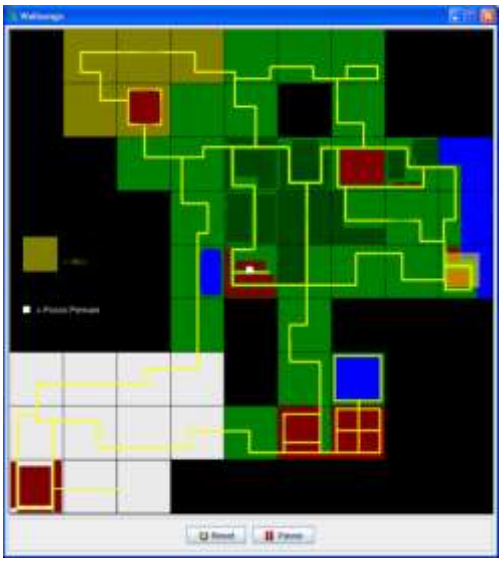
3. Tampilan Permainan



Gambar 4.3 Tampilan Tempat 1.

4. Tampilan Map

Layar ini befungsi menampilkan map permainan dan lokasi pemain di dalam permainan.



Gambar 4.4 Tampilan Map.

4.4 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.

Pada *game* edukasi sejarah walisongo digunakan pengujian *blackboxtesting* dan *whiteboxtesting*. Cara pengujian dilakukan melalui dua tahap, yaitu pengujian alpha (fungsional) dan pengujian beta.

4.4.1 Pengujian Alpha (Fungsional)

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

1. Rencana Pengujian

Rencana pengujian yang akan dilakukan pada *game* edukasi sejarah walisongo ini selengkapnya dapat dilihat pada Tabel 4.1

Tabel 4.3 Rencanan Pengujian Aplikasi

No	Komponen yang di Uji	Butir Uji	Jenis Uji
1.	Menu	Memilih Tombol Main	*Black Box*
		Tampil Tombol TampilPath	*BlackBox*
		Tampil Tombol HidePath	*BlackBox*
		Tampil Tombol KillAll	*BlackBox*
		Memilih Tombol TampilPath	*Black Box*
		Memilih Tombol HidePath	*BlackBox*
		Memilih Tombol KillAll	*Black Box*
2.	Kontrol Karakter	Bergerak ke atas	*Black Box*
		Bergerak ke bawah	*Black Box*
		Bergerak ke kiri	*Black Box*
		Bergerak ke kanan	*Black Box*
		Menembak	*Black Box*
		Mengisi ulang peluru	
		Menampilkan map dan melihat misi	*Black Box*
		Menampilkan foto yang telah terambil	*Black Box*

3.	Petunjuk	Pemain menggerakkan karakter	*BlackBox*
4.	Materi	Karakter Mendekati Objek	*BlackBox*
5.	Pencarian karakter oleh musuh (Algoritma A *Star*)	Musuh mencari node karakter degan jalur terpendek	*BlackBox* dan *WhiteBox*

2. Kasus dan Hasil Pengujian

Berikut ini adalah kasus untuk menguji perangkat lunak yang sudah dibangun menggunakan metode *Black Box* berdasarkan Tabel 4.1 Rencana pengujian yang telah dibuat sebelumnya.

Tabel 4.4 Kasus dan Hasil Uji

No	Kasus / Diuji	Skenario Uji	Hasil yang diharapkan	Hasil Pengujian
1.	Menu	Memilih Tombol Main	Masuk ke dunia *game*	[√] Berhasil [ ] Tidak Berhasil
		Tampil Tombol TampilPath	Menampilkan tombol ketika pemain menekan tombol *keyboard* "o"	[√] Berhasil [ ] Tidak Berhasil
		Tampil Tombol HidePath	Menampilkan tombol ketika pemain menekan tombol *keyboard* "o"	[√] Berhasil [ ] Tidak Berhasil
		Tampil Tombol KillAll	Menampilkan tombol ketika	[√] Berhasil

			pemain menekan tombol *keyboard* "o"	[ ] Tidak Berhasil
		Memilih Tombol Tampil Path	Menampilkan jalur terpendek yang akan dilalui musuh ke karakter pemain	[√] Berhasil [ ] Tidak Berhasil
		Memilih Tombol HidePath	Menghilangkan jalur yang akan dilalui musuh ke karakter pemain	[√] Berhasil [ ] Tidak Berhasil
		Memilih Tombol KillAll	Menghilangkan semua musuh pada tempat karakter berada.	[√] Berhasil [ ] Tidak Berhasil
2.	Kontrol karakter	Bergerak ke atas	ketika pemain menekan tombol *keyboard* "W" maka karakter bergerak ke atas	[√] Berhasil [ ] Tidak Berhasil
		Bergerak ke bawah	ketika pemain menekan tombol *keyboard* "S" maka karakter bergerak ke bawah	[√] Berhasil [ ] Tidak Berhasil
		Bergerak ke kiri	ketika pemain menekan tombol *keyboard* "A" maka karakter bergerak ke kiri	[√] Berhasil [ ] Tidak Berhasil
		Bergerak ke	ketika pemain	[√] Berhasil

		kanan	menekan tombol *keyboard* “D” maka karakter bergerak ke kanan	[ ] Tidak Berhasil
		Menembak	Ketika pemain menekan tombol *mouse*, karakter menembak	[√] Berhasil [ ] Tidak Berhasil
		Mengisi ulang peluru	Ketika pemain menekan tombol *keyboard* “R”, karakter mengisi ulang peluru dan menjadikan peluru ke maximal kembali yaitu 7	[√] Berhasil [ ] Tidak Berhasil
		Menampilkan map dan melihat misi	Ketika pemain menekan tombol *keyboard* “R”, karakter mengisi ulang peluru dan menjadikan peluru ke maximal kembali yaitu 7	[√] Berhasil [ ] Tidak Berhasil
		Menampilkan foto yang telah terambil	Ketika pemain telah mendapatkan foto dan menekan tombol *keyboard* “1,2,3,4,5,6,7,8, atau 9”, maka akan tampil foto pada	[√] Berhasil [ ] Tidak Berhasil

			tampilan pojok kanan game	
3.	Petunjuk	Menampilkan petunjuk permainan	Ketika pemain berada pada posisi awal permainan maka akan mucul petunjuk cara bermain dan ketika menggerakkan karakter akan menampilkan misi.	[√] Berhasil [ ] Tidak Berhasil
4.	Materi	Menampilkan materi sejarah walisongo	Ketika karakter mendekati objek, maka akan muncul materi sejarah walisongo	[√] Berhasil [ ] Tidak Berhasil
5.	Pencarian karakter oleh musuh	Musuh mencari karakter pemain	Ketika musuh mencari posisi karakter pemain, akan menggunakan jalur terpendek. untuk pengujian *whitebox* akan ditunjukan pada point ke 3	[√] Berhasil [ ] Tidak Berhasil

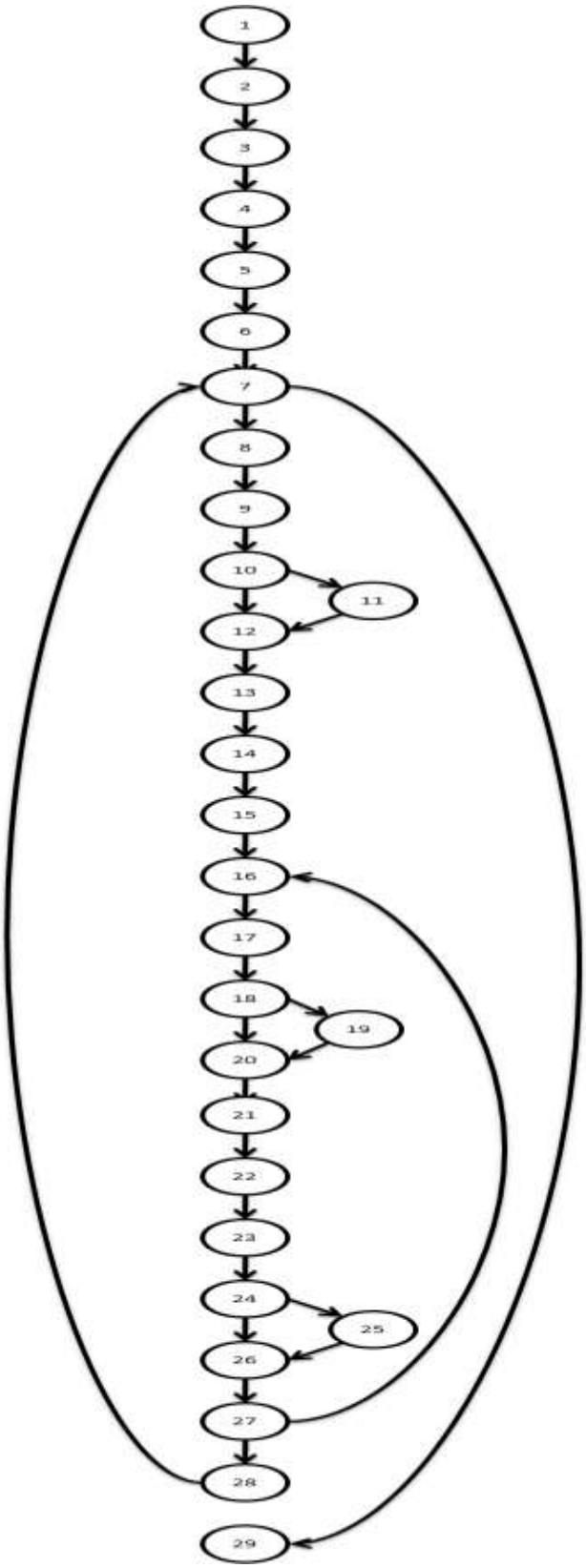
3. Pengujian Pencarian Karakter Pemain oleh musuh menggunakan algoritma A Star. Menguji apakah pencarian karakter pemain oleh musuh menggunakan algoritma A Star, yakni mencari jalur terpendek sudah berjalan sesuia dengan yang diharapkan. Berikut ini merupakan pengujian algoritma a star pada *game* edukasi sejarah walisongo dengan metode *whitebox*.

   a. algoritma yang akan diuji

   Berikut ini merupakan algoritma a star yang penerapannya menggunakan bahasa pemograman *java*.

```
1. ArrayList open = new ArrayList();
2. ArrayList closed = new ArrayList();
3. pathNodes = new PathNode[width][height];
4. PathNode current = new PathNode(s,heuristic(s,d));
5. open.add(current);
6. pathNodes[current.position.getX()][current.positio
   n.getY()] = current;
7. while(!open.isEmpty()){
8. int bestIndex = bestNode(open);
9. current = (PathNode)open.get(bestIndex);
10.     if(current.position.equals(d)){
11.      return returnPath(s,current);
12.     }
13.     open.remove(bestIndex);
14.     current.open = false;
```

```
15.     closed.add(current);
16.     for(int x = 0;x < 8;x++){
17.     Point nPoint = neighbour(current.position,
  x);
18.     if(pathNodes[nPoint.getX()][nPoint.getY()] !=
  null || !obstacleMapOpen(nPoint)){
19.     continue;
20.     }
21.     PathNode newNode = new
  PathNode(nPoint,current,heuristic(nPoint,d));
22.     open.add(newNode);
23.     pathNodes[nPoint.getX()][nPoint.getY()] =
  newNode;
24.     if(draw){
25.     imageMap.drawLine(current.position.getX() *
  pathResolution,current.position.getY() *
  pathResolution,nPoint.getX() *
  pathResolution,nPoint.getY() * pathResolution);
26.     }
27.     }
28.     }
29.     return null;
```



Gambar 4.5 ***Flow Graph Cascade Classifier***

Keterangan :

 = Menggambarkan Kondisi

 = Menggambarkan Aksi

a. *Cyclomatic Complexity* V(G)

E = 33 N = 29

V(G) = E – N + 2

V(G) = 33 – 29 + 2

V(G) = 6

Keterangan :

E = Jumlah aksi

N = Jumlah kondisi

b. *Independent Path*

Path 1 = 1-2-3-4-5-6-7-29

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

Path 3 = 1-2-3-4-5-6-7-8-9-10-11-12-13-14-15-16-17-18-19-20-21-22-23-24-25-26-27-16

Path 4 = 1-2-3-4-5-6-7-8-9-10-11-12-13-14-15-16-17-18-19-20-21-22-23-24-26-27-16

Path 5 = 1-2-3-4-5-6-7-8-9-10-11-12-13-14-15-16-17-18-20-21-22-23-24-26-27-16

Path 6 = 1-2-3-4-5-6-7-8-9-10-12-13-14-15-16-17-18-20-21-22- 23-24-26-27-16

a. *Graph Matrix Cascade Classifier*

Tabel 4.5 *Graph Matrix Cascade Classifier*

	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	E
1		1																												0
2			1																											0
3				1																										0
4					1																									0
5						1																								0
6							1																							0
7								1																					1	1
8									1																					0
9										1																				0
10											1	1																		1
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
25																										1				0
26																											1			0
27																1												1		1
28							1																							0
29																														0
SUM (E)+1																														6

Berdasarkan pengujian *whitebox* pada pencarian jalur terpendek karakter musuh ke karakter player yang menggunakan algoritma A star sudah berjalan dengan baik dan tidak ada pengulangan tak berhingga karena path pada *Flow Graph Cascade Classifier* sudah sesuai dengan *Independent Path* dan *Graph Matrix Cascade Classifier* yakni 6 (enam) *path.*

4.4.2 Pengujian *Beta*

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

4.4.2.1 Evaluasi dari Pengguna

Untuk mengetahui tanggapan dan penilaian pemain terhadap game ini. Telah disebarkan kuisioner kepada 20 responden dengan usia yang berdasarkan pada sasaran user. Berikut adalah pertanyaan dan hasil kuesioner yang telah dibagikan dengan menggunakan rumus :

$$\mathbf{Y = P/Q * 100\%}$$

Keterangan :

P = Banyaknya jawaban responden tiap soal

Q = Jumlah responden

Y = Nilai prosentase

Hasil pengujian *Beta* adalah sebagai berikut:

1. Menurut anda apakah *game* edukasi sejarah walisongo menarik?

Kategori Jawaban	Sangat menarik	Menarik	Kurang menarik	Tidak menarik
Frekuensi Jawaban	6	12	2	-
Persentase Nilai	30%	60%	10%	-

2. Apakah dengan adanya *game* ini anda lebih tertarik untuk mempelajari sejarah walisongo?

Kategori Jawaban	Sangat tertarik	Tertarik	Kurang menarik	Tidak menarik
Frekuensi Jawaban	8	9	3	-
Persentase Nilai	40%	45%	15%	-

3. Apakah dengan adanya *game* ini dapat membantu anda menambah pengetahuan mengenai sejarah walisongo?

Kategori Jawaban	Sangat terbantu	Terbantu	Kurang terbantu	Tidak Menarik
Frekuensi Jawaban	-	12	8	-
Persentase Nilai	-	60%	40%	-

4. Apakah materi mengenai sejarah walisongo yang disajikan pada *game* sudah sesuai dengan materi yang disampaikan di sekolah?

Kategori Jawaban	Sangat sesuai	Sesuai	Kurang sesuai	Tidak sesuai
Frekuensi Jawaban	15	5	-	-
Persentase Nilai	75%	25%	-	-

5. Apakah tampilan *game* edukasi sejarah walisongo menarik?

Kategori Jawaban	Sangat menarik	Menarik	Kurang menarik	Tidak menarik
Frekuensi Jawaban	-	5	10	5
Persentase Nilai	-	25%	60%	25%

6. Apakah petunjuk yang terdapat pada *game* edukasi sejarah walisongo membantu anda dalam memainkan game?

Kategori Jawaban	Sangat membantu	Membantu	Kurang membantu	Tidak membantu
Frekuensi Jawaban	1	9	10	-
Persentase Nilai	5%	45%	50%	-

7. Apakah anda kesulitan dalam menyelesaikan misi yang ada pada *game* ini?

Kategori Jawaban	Sangat Sulit	Sulit	Kurang sulit	Tidak sulit
Frekuensi Jawaban	7	3	8	2
Persentase Nilai	35%	15%	40%	10%

Berdasarkan jawaban hasil kuesioner yang diberikan kepada para responden, maka dapat diambil kesimpulan yaitu:

1. *Game* edukasi sejarah walisongo ini dapat membuat user tertarik sehingga mereka ingin mencoba memainkannya
2. *Game* edukasi sejarah walisongo ini dapat menarik perhatian user untuk mempelajari sejarah walisongo.
3. *Game* edukasi sejarah walisongo ini dapat membantu user dalam mempelajari sejarah walisongo.
4. Materi sejarah walisongo yang di sampaikan dalam *game* ini sesuai dengan materi yang diberikan di sekolah.
5. Tampilan pada *game* edukasi sejarah walisongo ini kurang disukai oleh user.
6. petunjuk yang diberikan untuk memainkan game ini cukup membantu user dalam memainkan *game*.
7. misi-misi yang diberikan sebagai tantangan permaianan pada game ini cukup memberikan tantangan pada user.