BAB 4

IMPLEMENTASI DAN PENGUJIAN

Pada bab ini akan dilakukan implementasi dan pengujian terhadap aplikasi yang dibangun. Tahapan ini dilakukan setelah analisis dan perancangan selesai dilakukan dan selanjutnya akan diimplementasikan kedalam bahasa pemograman. Setelah implementasi maka dilakukan pengujian terhadap aplikasi. Aplikasi yang telah dibangun akan diimplementasikan untuk mengetahui apakah sistem tersebut dapat berjalan sesuai dengan tujuannya atau tidak.

4.1 Implementasi

Tujuan implementasi adalah untuk menerapkan perancangan yang telah dilakukan terhadap sistem sehingga *user* dapat memberi masukan demi berkembangnya sistem yang telah dibangun sebagai simulasi dari permainan *shopping time*. Perangkat keras dan perangkat lunak yang digunakan untuk melakukan implementasi dan pengujian aplikasi yaitu sebagai berikut :

1. Perangkat keras

Perangkat keras yang digunakan untuk pengujian aplikasi dijelaskan pada tabel 4.1.

Tabel 4.1. Perangkat keras yang digunakan

Komputer	Spesifikasi perangkat keras
Prosesor	Intel dual-core 1.73 Ghz
Layar	Resolusi 1280 x 800
Monitor	LCD 14 inc
Memori	Memori 1 GB
Harddisk	Harddisk 120 GB
Keyboard dan mouse	(OnBoard)

2. Perangkat lunak

Perangkat lunak yang digunakan untuk menguji permainan ini menggunakan Browser atau Flash Player dengan meng-klik file shoppingtime.exe.

4.1.1 Implementasi Aplikasi

- **Tampilan halaman utama**

Perancangan aplikasi tampilan halaman utamapermainan *shopping time* dapat dilihat sebagai berikut:



Gambar 4.1 Tampilan halaman utama *shopping time*

Layar ini merupakan tampilan awal pada permainan *shopping time*, dimana terdapat lima tombol menu yaitutombol Mulai, Petunjuk, Pengaturan, Kredit dan Keluar.

- Tombol Mulai untuk masuk ke halaman permainan.
- Tombol Petunjuk untuk masuk ke halaman petunjuk.
- Tombol Pengaturan untuk masuk ke halaman pengaturan.
- Tombol Kredit untuk masuk ke halaman kredit.
- Tombol Keluar untuk keluar dari aplikasi.

Adapun gambar untuk setiap menu dapat di lihat sebagai berikut:



Gambar 4.2 Tampilan awal permainan *shopping time*



Gambar 4.3 Tampilan halaman petunjuk permainan *shopping time*



Gambar 4.4 Tampilan halaman pengaturan permainan *shopping time*



Gambar 4.5 Tampilan halaman kredit permainan *shopping time*

- **Tampilan isi toko**



Gambar 4.6 Tampilan Halaman Isi Toko

Layar ini berfungsi untuk menampilkan isi toko. Di masing-masing toko terdapat 4 jenis barang yang berbeda yang bisa dibeli.

- **Tampilan proses pembelian barang level 1**



Gambar 4.7 Tampilan proses pembelian barang level 1

Layar ini berfungsi untuk menampilkan proses pembelian barang pada level 1 dimana proses pembelian berhasil jika jawaban benar.

- **Tampilan proses pembelian barang level 2 dan level 3**



Gambar 4.8 Tampilan proses pembelian barang level 2 dan level 3

Layar ini berfungsi untuk menampilkan proses pembelian barang pada level 2 dan level 3 dimana proses pembelian berhasil jika jawaban benar.

- **Tampilan halaman pindah level 1 ke level 2**



Gambar 4.9 Tampilan halaman pidah level 1 dan level 2

Layar ini berfungsi untuk menampilkan halaman perpindahan dari level 1 ke level 2.

- **Tampilan halaman pindah level 1 ke level 2**



Gambar 4.10 Tampilan halaman pidah level 2 dan level 3

Layar ini berfungsi untuk menampilkan halaman perpindahan dari level 2 ke level 3.

4.1.2 Implementasi Contoh Kasus Jalur A*

Salah satu contoh implementasi simulasi yang akan digunakan untuk menemukan jalur yaitu dengan menentukan salah satu titik tujuan, misalnya titik tujuan toko mobil-mobilan. Setelah menentukan titik tujuan maka pencarian *rute* akan dilakukan dengan algoritma A* yang menghasilkan berapa panjang jarak yang ditempuh yang diasumsikan dengan berapa banyak kotak atau *node* yang dilewati. Implementasi jalur dengan algoritma A* dapat terlihat pada gambar 4.11 dengan hasil jalan ditemukan simpul yang dilewati 6 dan simpul yang diperiksa 23.



Gambar 4.11 Implementasi *Rute* Algoritma A* *search*

4.2 Pengujian

Pengujian pada sistem yang baru dibangun dilakukan melalui dua tahap yaitu pengujian *alpha* dan pengujian *betha*.

4.2.1 Pengujian *Alpha*

Metode yang digunakan dalam pengujian *alpha* adalah metode *whitebox* dan *blackbox*.

- **Rencana pengujian**

Berdasarkan rencana pengujian, maka dapat dilakukan pengujian alpha dengan *blackbox* pada aplikasi permainan *shopping time* yang dijelaskan pada tabel 4.2.

Tabel 4.2 Rencana Pengujian Permainan

Kelas Uji	Detail Pengujian	Jenis Uji
Algoritma A*	Testing implementasi algoritma A*	*White box*
Mulai	Memulai permainan	*Black box*
Petunjuk	Menampilkan Halaman Petunjuk	*Black box*
Pengaturan	Menampilkan Halaman Petunjuk	*Black box*
Kredit	Menampilkan Halaman Petunjuk	*Black box*
Pembelian barang	Testing halaman pembelian barang	*Black box*

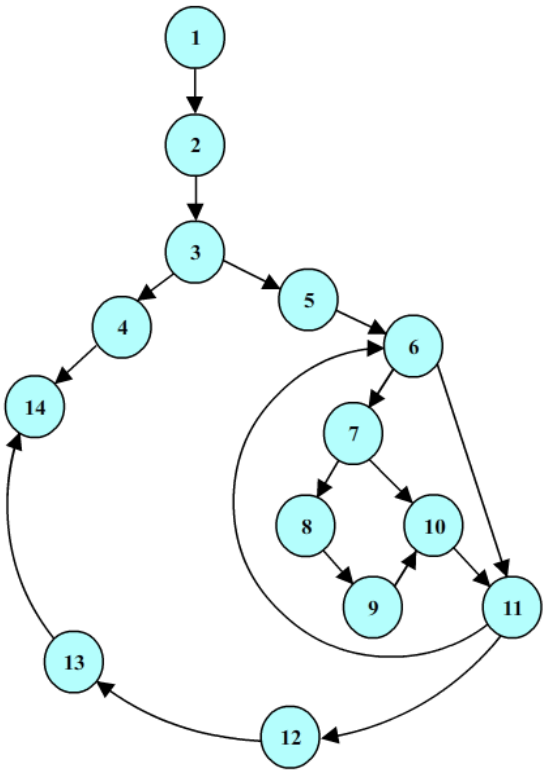
4.2.1.1 Pengujian *White-Box*

Pengujian *white-box* atau *white-box testing* merupakan metode perancangan *test case* yang menggunakan struktur kontrol dari perancangan prosedural dalam mendapatkan *test case*. Adapun metode yang digunakan dalam pengujian *white-box* ini adalah metode Basis Path. Metode Basis Path mengijinkan pendesain kasus uji untuk membuat perkiraan lojik yang kompleks dari desain prosedural dan menggunakan perkiraan ini untuk mendefinisikan aliran eksekusi .

Berikut ini *pseudocode* algoritma A*

```
1.  masukkan nilaiAwal ke openList
2.     while (openList tidak kosong){
          currenNode = temukan nilai f yang paling kecil di
          openList
3.        if currenNode final kembali ke path yang berhasil
4.          masukkan currenNode ke closeList dan hapus dari
            openList
            exit
5.        else
6.          foreach nodeTetangga dari currentNode
7.            if nodeTetangga tidak berada di openList
                tambahkan nodeTetangga ke openList
                simpan g, h, dan f lalu simpan currenParent
8.              if nodeTetangga ada di openlist tapi current_g
                lebih baik dari previouse_g
                  simpan g dan f, lalu simpan currenParent
9.              endif
10.           endif
11.         endforeach
12.       endif
13.    endwhile
14. end
```

Berdasarkan *pseudocode* diatas maka diagram alirnya sebagai berikut:



Gambar 4.12 Diagram Alir Algoritma A*

Dari diagram alir diatas, dapat dihitung *cyclomatic complexity*, yakni:

1. Flowgraph mempunyai 6 region
2. V(G) = 18 edge – 14 node + 2 = 6
3. V(G) = 5 predicate node + 1 = 6

Dari hasil perhitungan cyclomatic complexity terdapat 6 independent path yaitu:

Path 1 = 1 – 2 – 3 – 4 – 14

Path 2 = 1 – 2 – 3 – 5 – 6 – 7 – 8 – 9 – 10 – 11 – 12 – 13 – 14

Path 3 = 1 – 2 – 3 – 5 – 6 – 7 – 10 – 11 – 12 – 13 – 14

Path 4 = 1 – 2 – 3 – 5 – 6 – 7 – 8 – 9 – 10 – 11 – 6 –7 –8 –9 –10 –11 –12 –13 – 14

Path 5 = 1 – 2 – 3 – 5 – 6 – 7 – 10 – 11 – 6 – 7 – 10 – 11 – 12 – 13 – 14

Path 6 = 1 – 2 – 3 – 5 – 6 – 11 – 12 – 13 – 14

Tabel 4.3 Tabel kondisi *path* algoritma A*

No	Path	Kondisi	Hasil Pengujian
1	*Path 1*	*Path* ini dijalankan dalam kondisi node yang dituju langsung ditemukan.	[√] Berhasil [ ] Tidak Berhasil
2	*Path 2*	*Path* ini dijalankan dalam kondisi *node* yang dituju telah ditemukan dengan memeriksa minimal 1 *current node* dan *node* tetangga tidak berada di *open list* dan nilai g sekarang lebih baik dari nilai g sebelumnya.	[√] Berhasil [ ] Tidak Berhasil
3	*Path 3*	*Path* ini dijalankan dalam kondisi *node* yang dituju telah ditemukan dengan memeriksa minimal 1 *current node* dan *node* tetangga tidak berada di *open list* dan tidak ada *node* pembanding untuk menentukan nilai g terbaik.	[√] Berhasil [ ] Tidak Berhasil

4	*Path 4*	*Path* ini dijalankan dalam kondisi *node* yang dituju telah ditemukan dengan memeriksa minimal 2 *current node* dan *node* tetangga tidak berada di *open list* dan nilai g sekarang lebih baik dari nilai g sebelumnya.	[√] Berhasil [ ] Tidak Berhasil
5	*Path 5*	*Path* ini dijalankan dalam kondisi *node* yang dituju telah ditemukan dengan memeriksa minimal 2 *current node* dan *node* tetangga tidak berada di *open list* dan tidak ada *node* pembanding untuk menentukan nilai g terbaik.	[√] Berhasil [ ] Tidak Berhasil
6	*Path 6*	*Path* ini dijalankan dalam kondisi *node* yang dituju telah ditemukan dengan memeriksa minimal 1 *current node* dan node tetangga berada di *open list*.	[√] Berhasil [ ] Tidak Berhasil

Berdasarkan pengujian yang dilakukan, maka bisa diambil kesimpulan bahwa masing-masing *independent path* yang dihasilkan bisa dijalankan minimal satu kali.

Selain itu, untuk melakukan uji coba basis path di atas juga digunakan graph matrik. Graph matrik merupakan matrik empat persegi yang mempunyai ukuran yang sama dengan jumlah node pada flowgraph. Adapun graph matrik pada flowgraph A* di atas adalah sebagai berikut:

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	N(E)-1
1		**1**													**1-1=0**
2			**1**												**1-1=1**
3				**1**	**1**										**2-1=1**
4														**1**	**1-1=0**
5						**1**									**1-1=0**
6							**1**				**1**				**2-1=1**
7								**1**		**1**					**2-1=1**
8									**1**						**1-1=0**
9										**1**					**1-1=0**
10											**1**				**1-1=0**
11						**1**						**1**			**2-1=1**
12													**1**		**1-1=0**
13														**1**	**1-1=0**
14															**0**
												SUM(E)+1			**5+1=6**

Gambar 4.13 Graph Matrik A*

4.2.1.2 Pengujian *Blackbox*

Metode *blackbox* adalah metode yang fokus pada persyaratan fungsional dari perangkat lunak yang dibangun.

- **Kasus dan hasil pengujian**

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

1. Pengujian tombol mulai

Pengujian ini dilakukan untuk memulai Aplikasi permainan *shopping time* yang telah dibuat.

Tabel 4.4 Pengujian Tombol Mulai

Kasus dan Hasil Uji (Data Normal)			
Data yang dimasukkan	**Realisasi yang diharapkan**	**Pengamatan**	**Kesimpulan**
Klik tombol 	Memulai permainan	Permainan dapat dimulai	[v] Diterima [ ] Ditolak

2. Pengujian tombol petunjuk

Pengujian ini dilakukan untuk menampilkan halaman petunjuk.

Tabel 4.5 Pengujian Tombol Petunjuk

Kasus dan Hasil Uji (Data Normal)			
Data yang dimasukkan	**Realisasi yang diharapkan**	**Pengamatan**	**Kesimpulan**
Klik tombol 	Untuk menampilkan cara bermain *shopping time*	Tombol dapat berfungsi sesuai yang diharapkan	[v] Diterima [ ] Ditolak

3. Pengujian tombol pengaturan

Pengujian ini dilakukan untuk menampilkan halaman pengaturan.

Tabel 4.6 Pengujian Tombol Pengaturan

Kasus dan Hasil Uji (Data Normal)			
Data yang dimasukkan	**Realisasi yang diharapkan**	**Pengamatan**	**Kesimpulan**
Klik tombol	Untuk menampilkan	Tombol dapat	[v] Diterima

	Pengaturan suara permainan *shopping time*	berfungsi sesuai yang diharapkan	[ ] Ditolak

4. Pengujian tombol kredit

   Pengujian ini dilakukan untuk menampilkan halaman kredit.

Tabel 4.7 Pengujian Tombol Kredit

Kasus dan Hasil Uji (Data Normal)			
Data yang dimasukkan	**Realisasi yang diharapkan**	**Pengamatan**	**Kesimpulan**
Klik tombol 	Untuk menampilkan pembuat permainan *shopping time*	Tombol dapat berfungsi sesuai yang diharapkan	[v] Diterima [ ] Ditolak

5. Pengujian permainan

   Pengujian ini dilakukan untuk melakukan permainan.

Tabel 4.8 Pengujian Permainan

Kasus dan Hasil Uji (Data Normal)			
Data yang dimasukkan	**Realisasi yang diharapkan**	**Pengamatan**	**Kesimpulan**
Klik toko	Untuk masuk ke halaman pembelian	Tombol dapat berfungsi sesuai yang diharapkan	[v] Diterima [ ] Ditolak
Isi textbox 	Untuk menjawab pengurangan uang jajan dan harga barang yang dibeli	Textbox dapat berfungsi sesuai yang diharapkan	[v] Diterima [ ] Ditolak
Klik 	Untuk menjawab warna barang yang dibeli	Tombol dapat berfungsi sesuai yang diharapkan	[v] Diterima [ ] Ditolak

Klik tombol 	Untuk mebeli barang	Tombol dapat berfungsi sesuai yang diharapkan	[v] Diterima [ ] Ditolak
Klik tombol 	Untuk kembali ke halaman awal permainan	Tombol dapat berfungsi sesuai yang diharapkan	[v] Diterima [ ] Ditolak
Klik tombol 	Untuk kembali ke halaman awal permainan	Tombol dapat berfungsi sesuai yang diharapkan	[v] Diterima [ ] Ditolak
Uang habis	Saat uang yang diberikan bisa dibelanjakan semua	Muncul halaman lanjut ke level berikutnya	[v] Diterima [ ] Ditolak
Waktu habis	Saat waktu yang diberikan habis dan uang yang ada masih tersisa	Muncul hamanan *game over*	[v] Diterima [ ] Ditolak
Karakter	Saat posisi titik tujuan di-klik, karakter bergerak ke titik tujuan	Karakter bergerak ke titik tujuan	[v] Diterima [ ] Ditolak
Pencuri	Saat muncul bisa mencuri barang-barang yang ada di toko	Barang-barang yang dicuri tidak bisa dibeli	[v] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Tidak Normal)			
Data yang dimasukkan	**Realisasi yang diharapkan**	**Pengamatan**	**Kesimpulan**
Isi textbox Salah	Saat tombol beli di-klik muncul pesan “jawaban salah, coba lagi”	Hasil sesuai yang Diharapkan	[v] Diterima [ ] Ditolak
Isi radiobox Salah	Saat tombol beli di-klik muncul pesan “jawaban salah, coba lagi”	Hasil sesuai yang Diharapkan	[v] Diterima [ ] Ditolak

4.2.2 Pengujian Beta

Pengujian beta merupakan pengujian yang dilakukan secara objektif dimana pengujian dilakukan secara langsung kepada responden dengan membuat kuesioner mengenai kepuasan pengguna dengan kandungan *point* syarat.

Berdasarkan hasil pengujian beta, dicari persentase masing-masing jawaban dengan menggunakan rumus:

Y = P / Q * 100%

Keterangan:

P = jumlah jawaban responden tiap soal

Q = jumlah responden

Y = nilai persentase

Berikut ini adalah hasil persentase masing-masing jawaban yang sudah dihitung nilainya dengan menggunakan rumus di atas. Kuesioner ini telah diujikan kepada 10 orang anak dengan rentang usia 6-13, dimana anak usia 6-8 tahun memerlukan pendamping untuk memainkan permainan *shooping time* ini. Hasil pengujian beta adalah sebagai berikut:

1. Apakah kamu setuju tampilan permainan *shopping time* ini menarik?

Tabel 4.9 Hasil pengujian beta pertanyaan No 1

Kategori Jawaban	**Frekuensi Jawaban**	**Jumlah Populasi *Sample***	**Jumlah Persentase**
Sangat Setuju	10	30	33.33%
Setuju	20	30	66.67%
Ragu-ragu	0	30	0%
Tidak Setuju	0	30	0%
Sangat Tidak Setuju	0	30	0%

2. Apakah kamu setuju karakter yang ada di permainan *shopping time* bagus dan cocok untuk permainan ini?

Tabel 4.10 Hasil pengujian beta pertanyaan No 2

Kategori Jawaban	**Frekuensi Jawaban**	**Jumlah Populasi *Sample***	**Jumlah Persentase**
Sangat Setuju	15	30	50%
Setuju	11	30	36.7%
Ragu-ragu	4	30	13,3%
Tidak Setuju	0	30	0%
Sangat Tidak Setuju	0	30	0%

3. Apakah kamu setuju permainan shopping time mudah dimainkan?

Tabel 4.11. Hasil pengujian beta pertanyaan No 3

Kategori Jawaban	**Frekuensi Jawaban**	**Jumlah Populasi *Sample***	**Jumlah Persentase**
Sangat Setuju	10	30	33,3%
Setuju	14	30	46,7%
Ragu-ragu	3	30	10%
Tidak Setuju	3	30	10%
Sangat Tidak Setuju	0	30	0%

4. Apakah kamu setuju pertanyaan yang ada di permainan shopping time mudah dijawab?

Tabel 4.22 Hasil pengujian beta pertanyaan No 4

Kategori Jawaban	**Frekuensi Jawaban**	**Jumlah Populasi *Sample***	**Jumlah Persentase**
Sangat Setuju	9	30	30%
Setuju	18	30	60%
Ragu-ragu	1	30	3,3%
Tidak Setuju	2	30	6,7%
Sangat Tidak Setuju	0	30	0%

5. Apakah kamu setuju dengan bermain permainan shopping time kamu bisa belajar berhitung?

Tabel 4.13 Hasil pengujian beta pertanyaan No 5

Kategori Jawaban	**Frekuensi Jawaban**	**Jumlah Populasi *Sample***	**Jumlah Persentase**
Sangat Setuju	14	30	46,7%
Setuju	16	30	53,3%
Ragu-ragu	0	30	0%
Tidak Setuju	0	30	0%
Sangat Tidak Setuju	0	30	0%

6. Apakah kamu setuju dengan bermain permainan shopping time kamu bisa belajar mengenali warna?

Tabel 4.14 Hasil pengujian beta pertanyaan No 6

Kategori Jawaban	**Frekuensi Jawaban**	**Jumlah Populasi *Sample***	**Jumlah Persentase**
Sangat Setuju	9	30	20%
Setuju	19	30	63,3%
Ragu-ragu	2	30	6,7%
Tidak Setuju	0	30	0%
Sangat Tidak Setuju	0	30	0%

7. Apakah kamu setuju kalau permainan ini masih menyenangkan jika dimainkan kembali?

Tabel 4.15 Hasil pengujian beta pertanyaan No 7

Kategori Jawaban	**Frekuensi Jawaban**	**Jumlah Populasi *Sample***	**Jumlah Persentase**
Sangat Setuju	11	30	36,7%
Setuju	18	30	60%
Ragu-ragu	1	30	3,3%

Tidak Setuju	0	30	0%
Sangat Tidak Setuju	0	30	0%

8. Apakah kamu setuju permainan *shopping time* sulit dimainkan?

Tabel 4.16 Hasil pengujian beta pertanyaan No 8

Kategori Jawaban	**Frekuensi Jawaban**	**Jumlah Populasi *Sample***	**Jumlah Persentase**
Sangat Setuju	0	30	0%
Setuju	3	30	10%
Ragu-ragu	8	30	26,7%
Tidak Setuju	16	30	53,3%
Sangat Tidak Setuju	3	30	10%

9. Apakah kamu setuju dengan bermaian permainan *shopping time* kamu lebih termotivasi untuk belajar?

Tabel 4.17 Hasil pengujian beta pertanyaan No 9

Kategori Jawaban	**Frekuensi Jawaban**	**Jumlah Populasi *Sample***	**Jumlah Persentase**
Sangat Setuju	11	30	36,7%
Setuju	15	30	50%
Ragu-ragu	2	30	6,7%
Tidak Setuju	2	30	6,6%
Sangat Tidak Setuju	0	30	0%

10. Apakah kamu setuju musik yang ada di permainan ini menarik?

Tabel 4.18 Hasil pengujian beta pertanyaan No 10

Kategori Jawaban	**Frekuensi Jawaban**	**Jumlah Populasi *Sample***	**Jumlah Persentase**
Sangat Setuju	10	30	33,3%
Setuju	19	30	63,4%
Ragu-ragu	1	30	3,3%

Tidak Setuju	0	30	0%
Sangat Tidak Setuju	0	30	0%

Berdasarkan jawaban terhadap kuesioner, maka dapat diambil kesimpulan yaitu :

- Kelebihan dari permainan *shopping time* yaitu :
  1. Aplikasi permainan ini dapat digunakan sebagai pemicu semangat anak untuk belajar.
  2. Pengguna dapat dengan mudah mengerti dalam memainkan aplikasi permainan *shopping time*.
  3. Gambar yang terdapat didalam permainan ini disenangi oleh pengguna.
  4. Karakter yang dipakai cocok untuk aplikasi permianan ini.
- Kekurangan dari permainan *shopping time* yaitu:
  1. Untuk anak 6-8 tahun, waktu yang diberikan terlalu cepat dan pertanyaan yang ada cukup sulit.
  2. Karakter yang ada cuma satu.