BAB 4

IMPLEMENTASI DAN PENGUJIAN SISTEM

4.1 Implementasi Sistem

Pada bab ini merupakan tahap implementasi dari aplikasi permainan *Fun Math For Smart*. Berdasarkan hasil analisis pada bab sebelumnya kedalam bahasa pemograman yang dapat dimengerti oleh komputer. Setelah tahap implementasi, maka akan dilakukan tahap pengujian terhadap aplikasi permainan *fun math for smart*.

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 pengujian aplikasi ini tidak harus yang berspesifikasi tinggi. Spesifikasi minimum yang dapat digunakan untuk menjalankan aplikasi permainan Fun math for smart.

Spesifikasi perangkat keras:

1. Prosesor : 2.0 GHz
2. Memory : 1024MB RAM
3. VGA : 756MB
4. Hardisk : 320GB
5. Monitor
6. Mouse dan Keyboard.

4.2.2 Perangkat Lunak (Software)

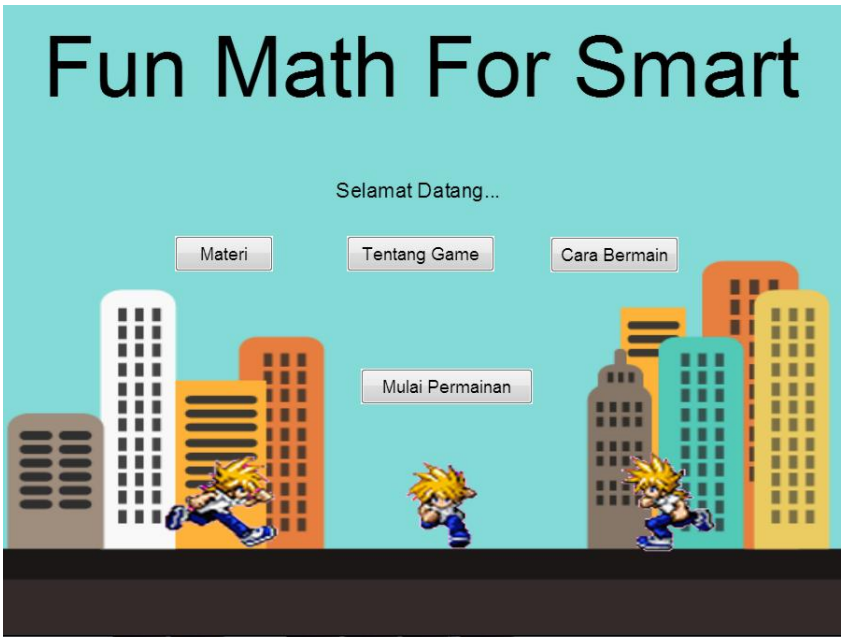
Perangkat lunak yang digunakan selama pembangunan aplikasi ini memiliki spesifikasi sebagai berikut :

a. Sistem Operasi Windows 7
b. Construct 2 R152
c. Web Browser (Google Chrome, Mozilla firefox)

4.2.3 Implementasi Antarmuka

1. Tampilan Menu

Layar ini menampilkan menu utama dalam permainan *fun math for smart*. Terdapat empat buah tombol, tombol materi untuk menampilkan materi, tombol tentang game untuk menampilkan tentang game, tombol cara bermain untuk menampilkan cara bermain dalam dgame, dan tombol mulai bermain untuk menampilkan intro.



Gambar 4.1 Tampilan Menu Game

2. Tampilan Cara Bermain

Layar ini menampilkan cara bermain. Dalam tampilan ini terdapat beberapa tombol, tombol menu utama untuk kembali ke menu utama untuk tombol stage 1, 2, 3, 4 untuk menampilkan cara bermain di setiap stage.



Gambar 4.2 Tampilan Cara Bermain

3. Tampilan Tentang

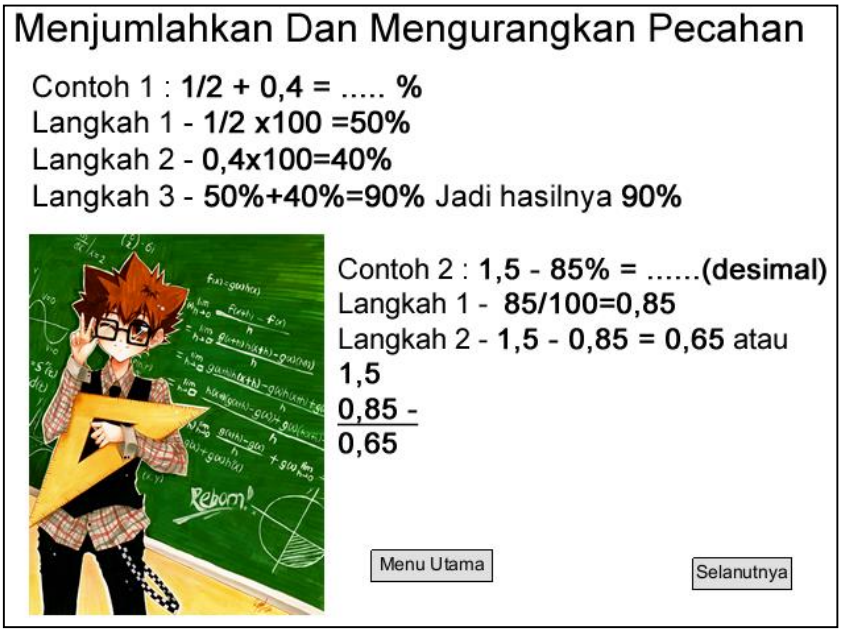
Layar ini menampilkan tentang game, terdapat tombol menu utama untuk kembali ke menu utama.



Gambar 4.3 Tampilan Tentang Game

4. Tampilan Materi

Layar ini menampilkan materi matematika. Dalam tampilan ini jika menekan tombol selanjutnya akan menampilkan materi selanjutnya.



Gambar 4.4 Tampilan Materi

5. Tampilan Intro

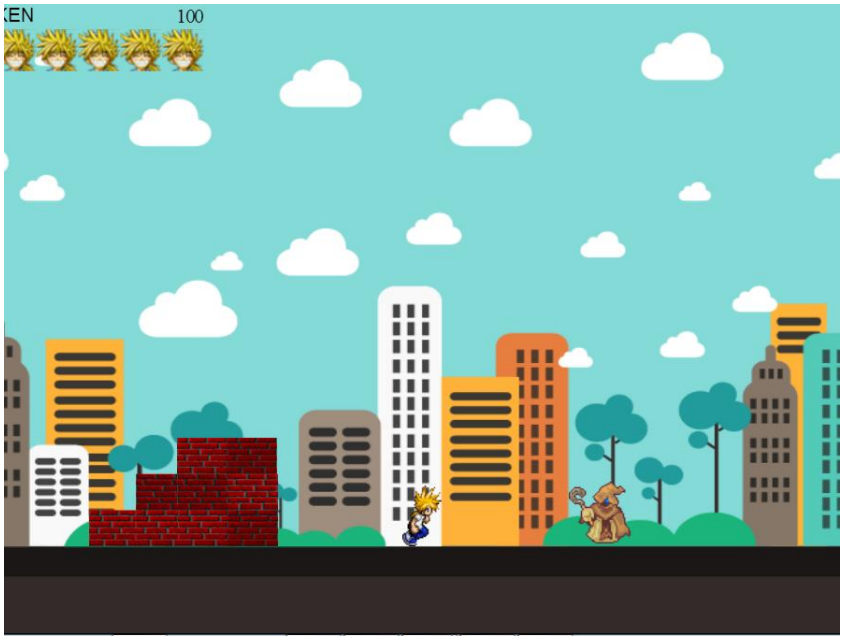
Layar ini berfungsi menampilkan intro pada permainan, apabila pemain memilih tombol mulai pada aplikasi permainan *Fun Math For Smart*. Pada layar intro terdapat text yang akan menampilkan alur cerita pada aplikasi permainan *Fun Math For Smart*.



Gambar 4.4 Tampilan Intro Game

6. Tampilan Stage 1

Layar ini berfungsi menampilkan stage 1, dalam stage ini terdapat NPC yang jika beradu dengan hero maka akan muncul pertanyaan jika menjawab dengan benar maka NPC akan hilang dan akan mendapatkan poin, jika salah maka darah akan berkurang.



Gambar 4.7 Tampilan Stage 1

7. Tampilan Stage 2

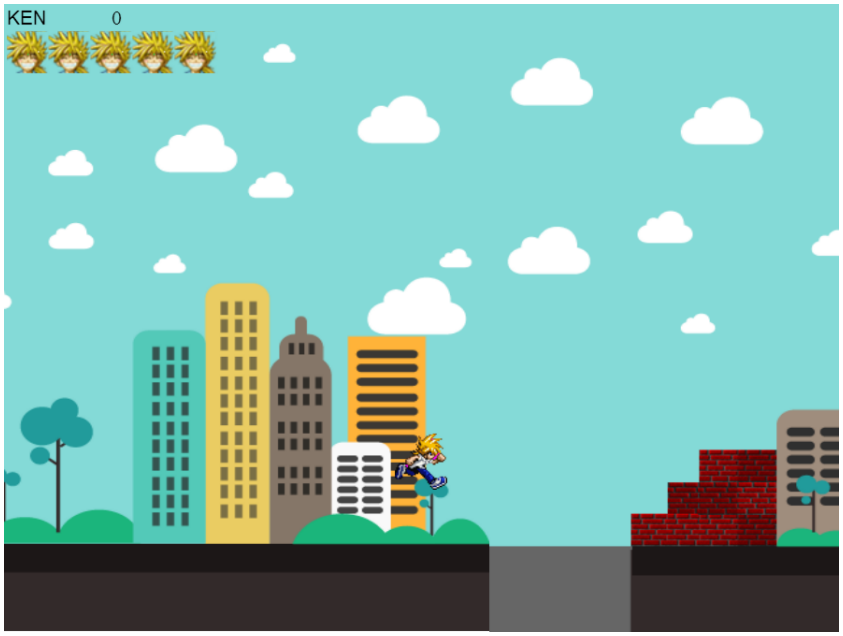
Layar ini berfungsi menampilkan stage 2, dalam stage ini pemain harus menembak 3 musuh dan mencari kotak pertanyaan. Setelah menjawab pertanyaan akan muncul kunci untuk membuka pintu.



Gambar 4.6 Tampilan Stage 1

8. Tampilan Stage 3

Layar ini berfungsi menampilkan stage 1, dalam stage ini pemain akan berlari menghadapi rintangan dan akan muncul pertanyaan, jika bisa menjawab maka akan terbebas dari rintangan tersebut dan jika salah menjawab maka pemain akan terkena rintangan.



Gambar 4.8 Tampilan Stage 3

9. Tampilan Stage 4

Layar ini berfungsi menampilkan stage 4, dalam stage ini pemain akan menghadapi raja. Disini pemain harus mengalahkan raja dengan menjawab pertanyaan, jika dalam 10 detik tidak bisa menjawab maka raja akan menjawab pertanyaan tersebut.



Gambar 4.9 Tampilan Stage 4

10. Tampilan Gameover

Layar ini berfungsi menampilkan tampilan permainan berakhir. Pada layar tercantum skor pada akhir permainan.



Gambar 4.10 Tampilan Gameover

4.3 Pengujian Sistem

Pengujian sistem merupakan hal terpenting yang dilakukan untuk menemukan kekurangan atau kesalahan pada perangkat lunak yang diuji. Dalam melakukan pengujian terhadap aplikasi permainan permainan *Fun Math For Smart* dilakukan dengan dua cara, 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 beta.

4.3.1 Pengujian Alpha (Fungsional)

Pengujian dilakukan untuk memastikan aplikasi permainan *Fun Math For Smart* telah berjalan dengan benar dan sesuai dengan kebutuhan yang diinginkan. Pengujian alpha berfokus kepada persyaratan fungsional perangkat lunak.

4.3.1.1 Rencana pengujian

Rencana pengujian yang akan dilakukan pada aplikasi ini selengkapnya terlihat pada tabel 4.I.

Tabel 4.1 Rencana Pengujian Aplikasi

No.	Komponen yang diuji	Detail Pengujian	Jenis Uji
1.	Menu	Memilih Tombol Mulai Permainan	Black Box
		Memilih Tombol Cara Bermain	Black Box
		Memilih Tombol Tentang	Black Box
		Memilih Tombol Materi	Black Box
2.	Tentang	Menampilkan Halaman Tentang	Black Box
		Memilih Tombol Menu Utama	Black Box
3.	Cara Bermain	Menampilkan Halaman Cara Bermain	Black Box

No.	Komponen yang diuji	Detail Pengujian	Jenis Uji
		Memilih Tombol Menu Utama	Black Box
		Memilih Tombol Stage 1	Black Box
		Memilih Tombol Stage 2	Black Box
		Memilih Tombol Stage 3	Black Box
		Memilih Tombol Stage 4	Black Box
4.	Materi	Menampilkan Halaman Materi	Black Box
		Memilih Tombol Menu Utama	Black Box
		Memilih Tombol Selanjutnya	Black Box
5.	Intro	Menampilkan Halaman Intro	Black Box
		Memilih Tombol Skip	Black Box
		Menampilkan karakter hero	Black Box
		Menampilkan karakter NPC	Black Box
		Menampilkan text	Black Box
		Mengaktifkan Musik	Black Box
6.	Arena Stage 1	Menampilkan arena stage 1	Black Box
		Menampilkan Health	Black Box
		Menampilkan Score	Black Box
		Menghitung Score	Black Box
		Menampilkan Karakter Hero	Black Box

No.	Komponen yang diuji	Detail Pengujian	Jenis Uji
		Menampilkan Karakter NPC	Black Box
		Menampilkan Item	Black Box
		Menampilkan Pertanyaan	Black Box
		Menekan tombol spasi untuk melompat	Black Box
		Menekan tombol kiri untuk bergerak ke kiri	Black Box
		Menekan tombol kanan untuk Bergerak ke kanan	Black Box
7.	Arena Stage 2	Menampilkan arena stage 2	Black Box
		Menampilkan Health	Black Box
		Menampilkan Score	Black Box
		Menghitung Score	Black Box
		Menampilkan Karakter Hero	Black Box
		Menampilkan Karakter NPC	Black Box
		Menampilkan Item	Black Box
		Menampilkan Pertanyaan	Black Box
		Menekan tombol spasi untuk menembak	Black Box
		Menekan tombol kiri untuk bergerak ke kiri	Black Box
		Menekan tombol kanan untuk Bergerak ke kanan	Black Box
		Menekan tombol atas untuk bergerak ke atas	Black Box
		Menekan tombol bawah untuk bergerak ke bawah	Black Box

No.	Komponen yang diuji	Detail Pengujian	Jenis Uji
8.	Area Stage 3	Menampilkan arena stage 3	Black Box
		Menampilkan Health	Black Box
		Menampilkan Score	Black Box
		Menghitung Score	Black Box
		Menampilkan Karakter Hero	Black Box
		Menampilkan Rintangan	Black Box
		Menampilkan Item	Black Box
		Menampilkan Pertanyaan	Black Box
		Menekan tombol spasi untuk melompat	Black Box
9.	Arena Stage 4	Menampilkan arena stage 4	Black Box
		Menampilkan Health	Black Box
		Menampilkan Score	Black Box
		Menghitung Score	Black Box
		Menampilkan Karakter Hero	Black Box
		Menampilkan Karakter Raja	Black Box
		Menampilkan Pertanyaan	Black Box
10.	Game Over	Menampilkan Halaman Game Over	Black Box
		Menampilkan Score Terakhir	Black Box
		Memilih Tombol Main Lagi	Black Box

No.	Komponen yang diuji	Detail Pengujian	Jenis Uji
11.	Algoritma A*	Pathfinding Pada Stage 2	White Box

4.3.1.2 Kasus Hasil Pengujian

Berdasarkan hasil uji perangkat lunak yang telah disusun, maka dapat dilakukan hasil uji kemudian diamati dan menghasilkan kesimpulan dapat dilihat pada tabel 4.2

Tabel 4.2 Kasus dan Hasil Uji

Kasus dan hasil uji (data normal)				
.No	**Kasus /diuji**	**Skenario Uji**	**Hasil Yang diharapkan**	**Hasil Pengijian**
1.	Menu	Memilih Tombol Mulai Permainan	Menampilkan Halaman Intro	[√] diterima [ ] ditolak
		Memilih Tombol Cara Bermain	Menampilkan Halaman Cara Bermain	[√] diterima [ ] ditolak
		Memilih Tombol Tentang	Menampilkan Halaman Tentang	[√] diterima [ ] ditolak
		Memilih Tombol Materi	Menampilkan Halaman Materi	[√] diterima [ ] ditolak
2.	Tentang	Menampilkan Halaman Tentang	Halaman Tentang Tampil	[√] diterima [ ] ditolak
		Memilih Tombol Menu Utama	Menampilkan Halaman Menu	[√] diterima [ ] ditolak
3.	Cara Bermain	Menampilkan Halaman Cara Bermain	Halaman Cara Bermain Tampil	[√] diterima [ ] ditolak
		Memilih Tombol	Menampilkan	[√] diterima

Kasus dan hasil uji (data normal)				
.No	**Kasus /diuji**	**Skenario Uji**	**Hasil Yang diharapkan**	**Hasil Pengijian**
		Menu Utama	Halaman Menu Utama	[ ] ditolak
		Memilih Tombol Stage 1	Menampilkan Cara Bermain Stage 1	[√] diterima [ ] ditolak
		Memilih Tombol Stage 2	Menampilkan Cara Bermain Stage 2	[√] diterima [ ] ditolak
		Memilih Tombol Stage 3	Menampilkan Cara Bermain Stage 3	[√] diterima [ ] ditolak
		Memilih Tombol Stage 4	Menampilkan Cara Bermain Stage 4	[√] diterima [ ] ditolak
4.	Materi	Menampilkan Halaman Materi	Halaman Materi Tampil	[√] diterima [ ] ditolak
		Memilih Tombol Menu Utama	Menampilkan Halaman Menu Utama	[√] diterima [ ] ditolak
		Memilih Tombol Selanjutnya	Menampilkan Halaman Materi Selanjutnya	[√] diterima [ ] ditolak
5.	Intro	Menampilkan Halaman Intro	Halaman Intro Tampil	[√] diterima [ ] ditolak
		Memilih Tombol Skip	Menampilkan Arena Stage 1	[√] diterima [ ] ditolak
		Menampilkan karakter hero	Karakter Hero Tampil	[√] diterima [ ] ditolak
		Menampilkan karakter NPC	Karakter NPC Tampil	[√] diterima [ ] ditolak
		Menampilkan text	Text Tampil	[√] diterima

Kasus dan hasil uji (data normal)				
.No	**Kasus /diuji**	**Skenario Uji**	**Hasil Yang diharapkan**	**Hasil Pengijian**
				[ ] ditolak
		Mengaktifkan Musik	Musik Aktif	[√] diterima [ ] ditolak
6.	Arena Stage 1	Menampilkan arena stage 1	Arena Stage 1 Tampil	[√] diterima [ ] ditolak
		Menampilkan Health	Healt Tampil	[√] diterima [ ] ditolak
		Menampilkan Score	Score Tampil	[√] diterima [ ] ditolak
		Menghitung Score	Score Bertambah	[√] diterima [ ] ditolak
		Menampilkan Karakter Hero	Karakter Hero Tampil	[√] diterima [ ] ditolak
		Menampilkan Karakter NPC	Karakter NPC Tampil	[√] diterima [ ] ditolak
		Menampilkan Item	Item Tampil	[√] diterima [ ] ditolak
		Menampilkan Pertanyaan	Pertanyaan Tampil	[√] diterima [ ] ditolak
		Menekan tombol spasi untuk melompat	Karakter Lompat	[√] diterima [ ] ditolak
		Menekan tombol kiri untuk bergerak ke kiri	Karakter Bergerak Ke Kiri	[√] diterima [ ] ditolak
		Menekan tombol kanan untuk Bergerak	Karakter Bergerak Ke Kanan	[√] diterima [ ] ditolak

Kasus dan hasil uji (data normal)				
.No	**Kasus /diuji**	**Skenario Uji**	**Hasil Yang diharapkan**	**Hasil Pengijian**
		ke kanan		
7.	Arena Stage 2	Menampilkan arena stage 2	Arena Stage 2 Tampil	[√] diterima [ ] ditolak
		Menampilkan Health	Healt Tampil	[√] diterima [ ] ditolak
		Menampilkan Score	Score Tampil	[√] diterima [ ] ditolak
		Menghitung Score	Score Bertambah	[√] diterima [ ] ditolak
		Menampilkan Karakter Hero	Karakter Hero Tampil	[√] diterima [ ] ditolak
		Menampilkan Karakter NPC	Karakter NPC Tampil	[√] diterima [ ] ditolak
		Menampilkan Item	Item Tampil	[√] diterima [ ] ditolak
		Menampilkan Pertanyaan	Pertanyaan Tampil	[√] diterima [ ] ditolak
		Menekan tombol spasi untuk menembak	Karakter Menembak	[√] diterima [ ] ditolak
		Menekan tombol kiri untuk bergerak ke kiri	Karakter Bergerak Ke Kiri	[√] diterima [ ] ditolak
		Menekan tombol kanan untuk Bergerak ke kanan	Karakter Bergerak Ke Kanan	[√] diterima [ ] ditolak
		Menekan tombol atas	Karakter Bergerak Ke	[√] diterima

Kasus dan hasil uji (data normal)				
.No	**Kasus /diuji**	**Skenario Uji**	**Hasil Yang diharapkan**	**Hasil Pengijian**
		untuk bergerak ke atas	Atas	[ ] ditolak
		Menekan tombol bawah untuk bergerak ke bawah	Karakter Bergerak Ke Bawah	[√] diterima [ ] ditolak
8.	Area Stage 3	Menampilkan arena stage 3	Arena Stage 2 Tampil	[√] diterima [ ] ditolak
		Menampilkan Health	Healt Tampil	[√] diterima [ ] ditolak
		Menampilkan Score	Score Tampil	[√] diterima [ ] ditolak
		Menghitung Score	Score Bertambah	[√] diterima [ ] ditolak
		Menampilkan Karakter Hero	Karakter Hero Tampil	[√] diterima [ ] ditolak
		Menampilkan Rintangan	Rintangan Tampil	[√] diterima [ ] ditolak
		Menampilkan Item	Item Tampil	[√] diterima [ ] ditolak
		Menampilkan Pertanyaan	Pertanyaan Tampil	[√] diterima [ ] ditolak
		Menekan tombol spasi untuk melompat	Karakter Melompat	[√] diterima [ ] ditolak
9.	Arena Stage 4	Menampilkan arena stage 4	Arena Stage 2 Tampil	[√] diterima [ ] ditolak
		Menampilkan Health	Healt Tampil	[√] diterima [ ] ditolak

Kasus dan hasil uji (data normal)				
.No	**Kasus /diuji**	**Skenario Uji**	**Hasil Yang diharapkan**	**Hasil Pengijian**
		Menampilkan Score	Score Tampil	[√] diterima [ ] ditolak
		Menghitung Score	Score Bertambah	[√] diterima [ ] ditolak
		Menampilkan Karakter Hero	Karakter Hero Tampil	[√] diterima [ ] ditolak
		Menampilkan Karakter Raja	Karakter Raja Tampil	[√] diterima [ ] ditolak
		Menampilkan Pertanyaan	Pertanyaan Tampil	[√] diterima [ ] ditolak
10.	Game Over	Menampilkan Halaman Game Over	Halaman Game Over Tampil	[√] diterima [ ] ditolak
		Menampilkan Score Terakhir	Score Terakhir Tampil	[√] diterima [ ] ditolak
		Memilih Tombol Main Lagi	Menampilkan Halaman Stage	[√] diterima [ ] ditolak

4.3.1.3 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.

4.3.1.4 Hasil Pengujian White Box

Di bawah ini merupakan kasus algoritma A* (A star) untuk menguji perangkat lunak yang sudah dibangun dengan menggunakan metode white box, berdasarkan skenario pengujian aplikasi yang sudah dijabarkan.

Pengujian proses algoritma A* merupakan proses pengujian dari pengimplementasian algoritma A* menggunakan pengujian whitebox, dan menguji apakah algoritma A* (A Star) berjalan dengan seharusnya pada musuh.

a. Pseudo Algoritma A*

```
1  create the open list of nodes, initially containing
2  only our starting node
3     create the closed list of nodes, initially empty
4     while (we have not reached our goal) {
5          consider the best node in the open list (the
   node with the lowest f value)
6          if (this node is the goal) {
7               then we're done
8          }
           else {
9               move the current node to the closed list
   and consider all of its neighbors
10              for (each neighbor) {
11                   if (this neighbor is in the closed
   list and our current g value is lower) {
12                        update the neighbor with the
13 new, lower, g value
14                        change the neighbor's parent to
15 our current node
16                   }
17                   else if (this neighbor is in the
   open list and our current g value is lower) {
18
```

```
19                      update the neighbor with the
new, lower, g value
20
                        change the neighbor's parent to
21
our current node
22                  }
23
                    else this neighbor is not in either
24
the open or closed list {
25                      add the neighbor to the open
list and set its g value
26
                    }

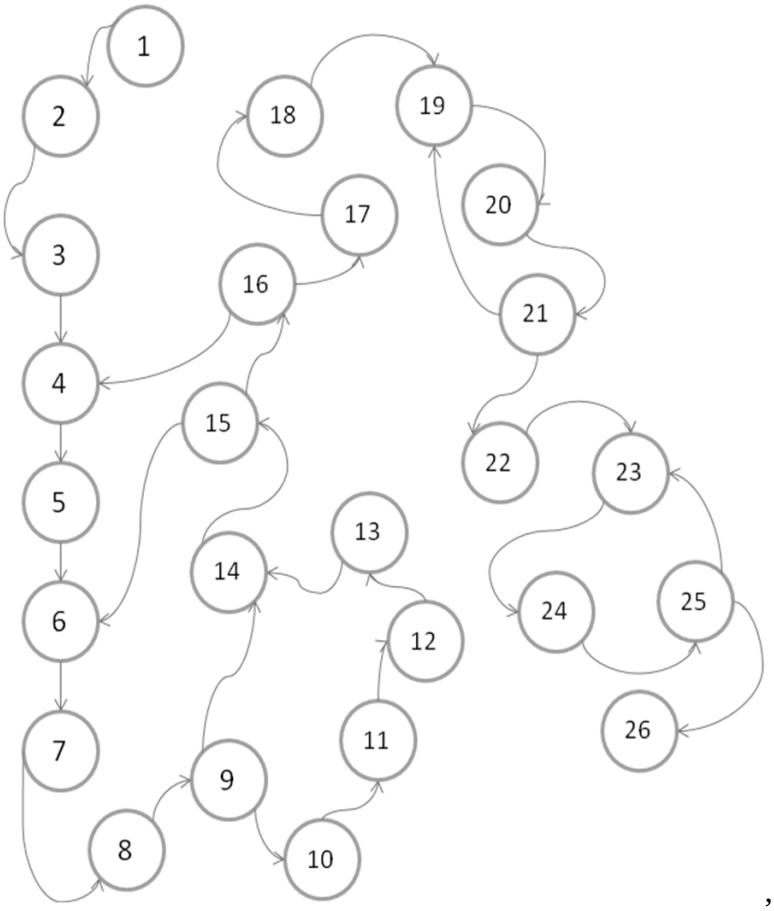
                }

            }

        }
```

Gambar 4.11 Pseudo Algoritma A*

b. Flow Graph



,

Gambar 4.12 Flow Graph A *

Keterangan :

= Menggambarkan kondisi

= Menggambarkan aksi

1. Cyclomatic Complexity (V(G)

   V(G) = E – N + 2

   V(G) = 30 – 26 + 2

   V(G) = 4 + 2

V(G) = 6

Di mana :

E = jumlah edge pada grafik alir

N = jumlah node pada grafik alir

2. Independent path

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

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

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

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

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

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

3. Graph Matrices

Berikut ini adalah graph matrices dari algoritma A*yang dapat dilihat pada tabel 4.4.

Tabel 4.3 Graph Matrices

	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	Hasil
1		1																									0
2			1																								0
3				1																							0
4					1																						0
5						1																					0
6							1																				0
7								1																			0
8									1																		0
9										1	1																1
10														1													0
11												1															0
12													1														0
13														1													0
14															1												0
15						1										1											1
16				1													1										1
17																		1									0
18																			1								0
19																				1							0
20																					1						0
21																			1			1					1
22																							1				0
23																								1			0
24																									1		0
25																							1			1	1
26																											Sum(E)+1=6 5+1=6

Berikut ini adalah hasil dari pengujian whitebox penelusuran lgoritma A* pada objek musuh pada aplikasi permainan Fun Math For Smart. Dapat dilihat pada tabel 4.5.

Tabel 4.4 Hasil Pengujian White Box

No path	Node (n)	Hasil yang diharapkan	Hasil tes	kesimpulan
1	1-2-3-4-5-6-7-8-9-10-14-15-16-17-18-19-20-21-22-23-24-25-26	Menampilkan rute ke player	Menampilkan rute ke player	Path dikerjakan
2	1-2-3-4-5-6-7-8-9-11-12-13-14-15-16-17-18-19-20-21-22-23-24-25-26	Mengecek node dalam OPEN list	Node dalam OPEN list terdefinisi	Path dikerjakan

No path	Node (n)	Hasil yang diharapkan	Hasil tes	kesimpulan
3	1-2-3-4-5-6-7-8-9-10-14-15-6-7-8-9-10-14-15-16-17-18-19-20-21-22-23-24-25-26	Mengecek node bukan berada dalam OPEN list	Node dalam OPEN list terdefinisi	Path dikerjakan
4	1-2-3-4-5-6-7-8-9-10-14-15-16-4-5-6-7-8-9-11-12-13-14-15-16-17-18-19-20-21-22-23-24-25-26	Membandingkan Nilai F antara nodeawal dengan node tetangga	Node tetangga menjadi current node	Path dikerjakan
5	1-2-3-4-5-6-7-8-9-10-14-15-16-17-18-19-20-21-19-20-21-22-23-24-25-26	Mencari nilai F setiap node terkecil dalam antrian	Nilai node yang mempunyai F terkecil terdefinisi	Path dikerjakan
6	1-2-3-4-5-6-7-8-9-10-14-15-16-17-18-19-20-21-22-23-24-25-23-24-25-26	Mengecek node terakhir sama dengan node awal	Node akhir = node awa	Path dikerjakan

Berdasarkan hasil pengamatan dan pengujian path maka dapat disimpulkan beberapa poin sebagai berikut :

1. Seluruh path dilalui dan dieksekusi

2. Setiap kondisi setidaknya diproses satu kali

3. Tidak ada perulangan tak hingga

4.3.2 Pengujian *Beta*

Pengujian *beta* merupakan pengujian yang dilakukan secara objektif dimana diuji secara langsung ke lapangan, dengan menggunakan kuesioner mengenai tanggapan *user* terhadap *game* yang telah dibangun. Adapun metode penilaian pengujian yang digunakan yaitu metode kuantitatif berdasarkan data sampel dari *user*.

Untuk mengetahui tanggapan dan penilaian pemain terhadap *game* ini. Telah disebarkan kuesioner kepada responden berdasarkan target *user* dan dilakukan perhitungan jawaban terbanyak.

Adapun berikut pertanyaan kuisioner yang dibagikan kepada *user* adalah sebagai berikut :

1. Apakah permainan *Fun Math For Smart* mudah untuk dimainkan?
2. Apakah permainan *Fun Math For Smart* bisa dijadikan sebagai media pembelajaran yang interaktif ?
3. Apakah permainan *Fun Math For Smart* mampu menarik perhatian serta minat belajar ?
4. Apakah pertanyaan pada permainan *Fun Math For Smart* mudah untuk dijawab ?
5. Apakah permainan *Fun Math For Smart* menarik untuk dimainkan ?
6. Apakah setelah memainkan permainan *Fun Math For Smart* akan memainkannya lagi ?

Tabel 4.5 Skor Jawaban Kuesioner

Jawaban	Skor	Skor Maksimum
Sangat Setuju	5	100
Setuju	4	80
Ragu - ragu	3	60
Kurang Setuju	2	40
Tidak Setuju	1	10

Tabel-tabel diatas adalah jawaban dari setiap pertanyaan kuesioner yang akan dibagikan kepada *user*. Interpretasi skor perhitungan = total skor / skor maksimum x 100% . Dengan berpatokan pada tabel 4.6 berikut ini:

Tabel 4.6 Kriteria Interpretasi Skor

Rentang Kriteria	Kesimpulan
81- 100%	Sangat Bagus
61 - 80%	Bagus
41 - 60%	Cukup Bagus
11 - 40%	Kurang Bagus
0 - 10%	Sangat Tidak Bagus

4.3.2.1 Hasil Kuesioner

Berikut ini adalah hasil perhitungan dari masing-masing jawaban yang sudah dihitung nilainya dengan menggunakan perhitungan rumus-rumus diatas:

1. Apakah permainan *Fun Math For Smart* mudah untuk dimainkan?

Tabel 4.7 Hasil Jawaban Kuesioner Pertanyaan Pertama

Pernyataan	Jawaban	Skor	Responden	Hasil Skor
1	Sangat Setuju	5	9	45
	Setuju	4	20	80
	Ragu-Ragu	3	1	3
	Tidak Setuju	2	0	0
	Sangat Tidak Setuju	1	0	0
JUMLAH			**30**	**128**

Interpretasi skor = 128 / 150 x 100 = 85,3%

Berdasarkan hasil presentase, dapat di simpulkan bahwa dari hasil pertanyaan pertama responden menjawab sangat bagus.

2. Apakah permainan *Fun Math For Smart* bisa dijadikan sebagai media pembelajaran yang interaktif ?

Tabel 4.8 Hasil Jawaban Kuesioner Pertanyaan Kedua

Pernyataan	Jawaban	Skor	Responden	Hasil Skor
2	Sangat Setuju	5	7	35
	Setuju	4	18	72
	Ragu-Ragu	3	5	15
	Tidak Setuju	2	0	0
	Sangat Tidak Setuju	1	0	0
JUMLAH			**30**	**122**

Interpretasi skor = 122 / 1150 x 100 = 81,3%

Berdasarkan hasil presentase, dapat di simpulkan bahwa dari hasil pertanyaan kedua responden menjawab sangat bagus.

3. Apakah permainan *Fun Math For Smart* mampu menarik perhatian serta minat belajar ?

Tabel 4.9 Hasil Jawaban Kuesioner Pertanyaan Ketiga

Pernyataan	Jawaban	Skor	Responden	Hasil Skor
3	Sangat Setuju	5	3	15
	Setuju	4	21	84
	Ragu-Ragu	3	6	18
	Tidak Setuju	2	0	0
	Sangat Tidak Setuju	1	0	0
JUMLAH			**30**	**117**

Interpretasi skor = 117 / 150 x 100 = 78%

Berdasarkan hasil presentase, dapat di simpulkan bahwa dari hasil pertanyaan ketiga responden menjawab bagus.

4. Apakah pertanyaan pada permainan *Fun Math For Smart* mudah untuk dijawab ?

Tabel 4.10 Hasil Jawaban Kuesioner Pertanyaan Keempat

Pernyataan	Jawaban	Skor	Responden	Hasil Skor
4	Sangat Setuju	5	4	20
	Setuju	4	16	64
	Ragu-Ragu	3	7	21
	Tidak Setuju	2	2	4
	Sangat Tidak Setuju	1	1	1
JUMLAH			**30**	**110**

Interpretasi skor = 110 / 150 x 100 = 73,3%

Berdasarkan hasil presentase, dapat di simpulkan bahwa dari hasil pertanyaan keempat responden menjawab bagus.

5. Apakah permainan *Fun Math For Smart* menarik untuk dimainkan ?

Tabel 4.11 Hasil Jawaban Kuesioner Pertanyaan Kelima

Pernyataan	Jawaban	Skor	Responden	Hasil Skor
5	Sangat Setuju	5	7	35
	Setuju	4	21	84
	Ragu-Ragu	3	2	6
	Tidak Setuju	2	0	0
	Sangat Tidak Setuju	1	0	0
JUMLAH			**30**	**125**

Interpretasi skor = 125 / 150 x 100 = 83,5%

Berdasarkan hasil presentase, dapat di simpulkan bahwa dari hasil pertanyaan kelima responden menjawab sangat bagus.

6. Apakah setelah memainkan permainan *Fun Math For Smart* akan memainkannya lagi ?

Tabel 4.12 Hasil Jawaban Kuesioner Pertanyaan Keenam

Pernyataan	Jawaban	Skor	Responden	Hasil Skor
6	Sangat Setuju	5	3	15
	Setuju	4	23	92
	Ragu-Ragu	3	4	12
	Tidak Setuju	2	0	0
	Sangat Tidak Setuju	1	0	0
JUMLAH			**30**	**119**

Interpretasi skor = 119 / 150 x 100 = 79,3%

Berdasarkan hasil presentase, dapat di simpulkan bahwa dari hasil pertanyaan keenam responden menjawab bagus.

4.3.2.2 Hasil Pengujian *Beta*

Dari pengujian *beta* yang telah dilakukan yaitu dengan pengujian pilihan jawaban dari kuesioner yang telah dibagikan di lapangan, dapat disimpulan bahwa

program aplikasi yang telah dibuat dinilai dapat membantu pengguna dalam belajar. Selain itu juga dapat menambah wawasan pengguna terhadap mata pelajaran Matematika.