BAB IV
IMPLEMENTASI DAN PENGUJIAN

4.1 Implementasi

Proses implementasi adalah tahapan dimana aplikasi yang sudah dibangun dan di rancang lalu di uji kelayakannya. Pemograman pada *game* ini menggunakan *Adobe Flash Professional CS3*, Selanjutnya dioperasikan sebagaimana mestinya sesuai fungsi dan kelayakannya untuk digunakan.

4.2 Perangkat Pendukung yang Digunakan

Dalam proses pembangunan 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 implementasi dan pengujian aplikasi ini memiliki spesifikasi sebagai berikut:

1. *CPU* : *AMD Phennom II X4 955 Processor* 3.20 Ghz
2. *Memory* : 2 GB
3. *VGA* : *NVIDIA GeForce GT 630* 2 GB
4. *Keyboard* : 1 buah
5. *Speaker* : Ya

4.2.2 Perangkat Lunak (*Software*)

Perangkat lunak yang digunakan untuk implementasi dan pengujian *Game* Edukasi *Save Family From Malaria* menggunakan *Adobe Flash Player 11*.

4.3 Implementasi Aplikasi

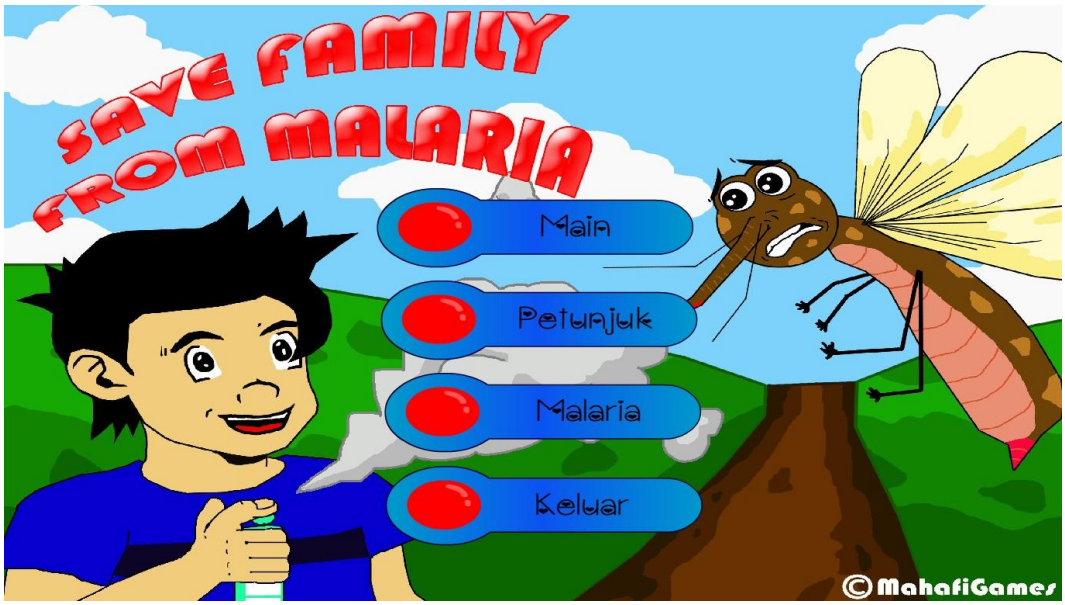
File yang terdapat pada *Game* Edukasi *Save Family From Malari*a ini, dibungkus menjadi satu dalam bentuk *Game* Edukasi *Save Family From Malari*a.zip, berikut langkah-langkah untuk menjalankan tersebut:

1. *Extract Game* Edukasi *Save Family From Malari*a.zip
2. Akan ada *folder* bentuk *Game* Edukasi *Save Family From Malari*a

3. *Double* Klik pada Malaria.exe atau Malaria.swf yang ada di *folder Game* Edukasi *Save Family From Malari*a
4. Akan tampil *window Game* Edukasi *Save Family From Malari*a, dan hal tersebut menandakan bahwa *game* sukses dijalankan.

4.4 Implementasi Antar Muka

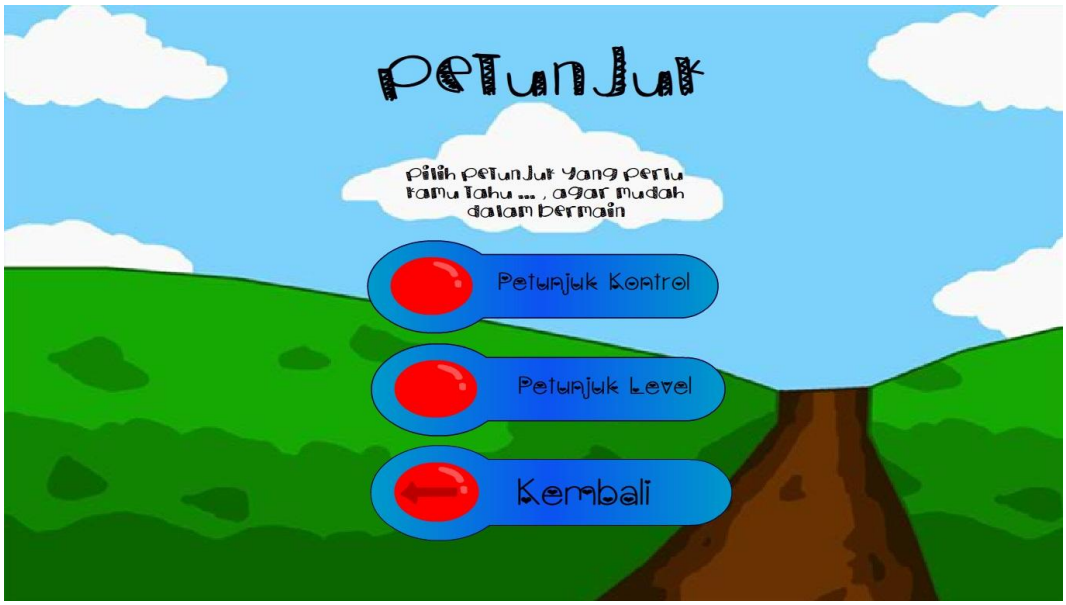
4.4.1 Tampilan Utama



Gambar 4.1 Tampilan Utama

Layar ini berfungsi untuk menampilkan tampilan utama atau tampilan awal game dimana didalamnya tersedia lima pilihan, yaitu pilihan Main, Petunjuk, Malaria, dan Keluar.

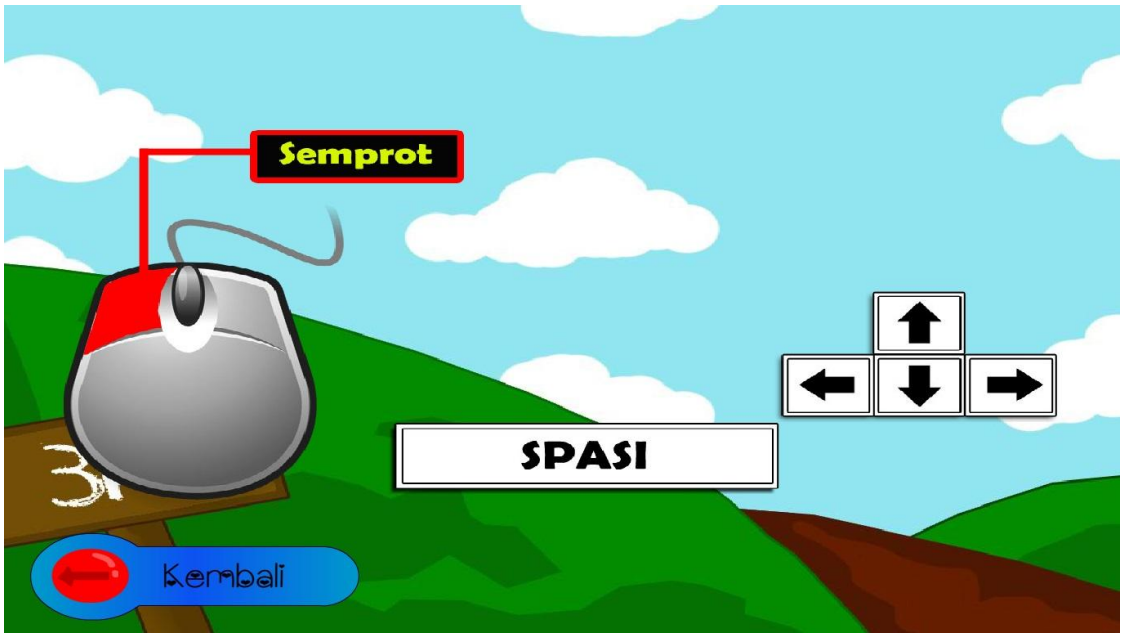
4.4.2 Tampilan Petunjuk



Gambar 4.2 Tampilan Petunjuk

Pada menu petunjuk terdapat pilihan Petunjuk Kontrol, Petunjuk Level, dan Kembali untuk kembali ke menu utama.

4.4.3 Tampilan Petunjuk Kontrol



Gambar 4.3 Tampilan Petunjuk Kontrol

Tampilan Petunjuk Kontrol menampilkan petunjuk kontrol dari permainan agar memudahkan pengguna dalam memainkan *game*.

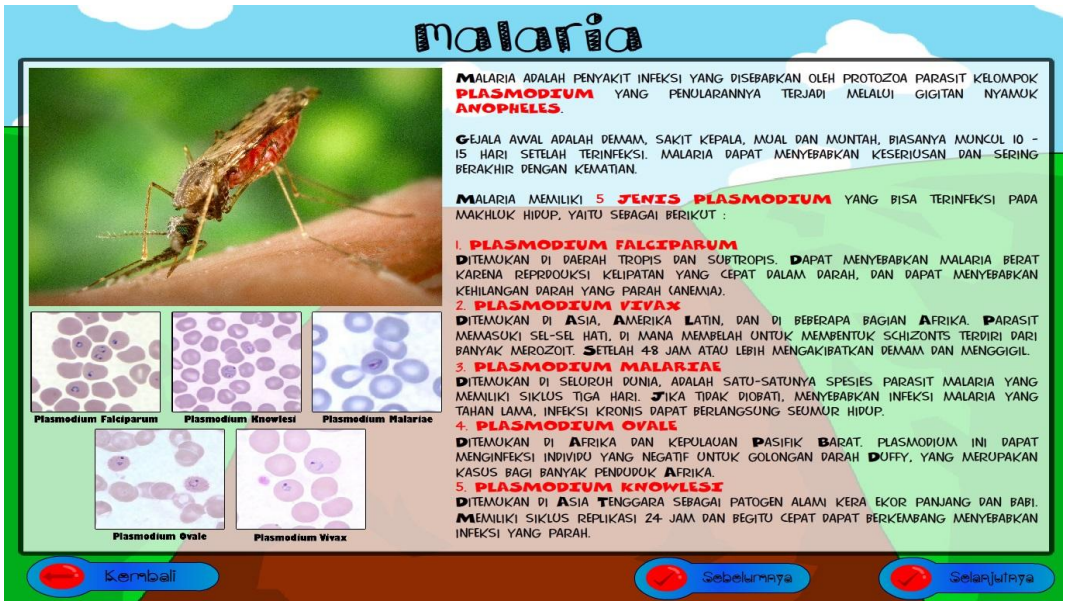
4.4.4 Tampilan Petunjuk Level



Gambar 4 4 Tampilan Petunjuk Level

Tampilan Petunjuk Level menampilkan petunjuk pada setiap level dari permainan agar memudahkan pengguna dalam mengetahui apa saja yang perlu dilakukan di setiap level permainan.

4.4.5 Tampilan Malaria



Gambar 4.5 Tampilan Malaria

Pada menu Malaria pemain dapat menggetahui info tentang malaria, nyamuk anopheles, dampak dari malaria dan pencegahannya.

4.4.6 Tampilan Cerita dan Info Materi



Gambar 4.6 Tampilan Permainan Cerita dan Info Materi

Pada menu malaria pemain dapat menggetahui info tentang malaria, nyamuk *anopheles*, dampak dari malaria dan pencegahannya.

4.4.7 Tampilan Permainan



Gambar 4.7 Tampilan Permainan

Layar menampilkan permainan utama dimana pemain harus menembakan atau menyemprotkan obat nyamuk ke nyamuk-nyamuk *anopheles* untuk mendapatkan *score*.

4.4.8 Tampilan Permainan Sesuai Materi



Gambar 4.8 Tampilan Permainan Sesuai Materi

Layar menampilkan permainan yang sesuai dengan materi pada level yang sedang dilalui pemain. Materi berupa pencegahan malaria seperti memasang kelambu, menguras bak mandi, menutup tempat air, mengubur wadah, dan menyemprotkan DDT.

4.4.9 Tampilan Setelah Menang



Gambar 4. 9 Tampilan Setelah Menang

Layar ini akan menampilkan notifikasi kemenangan bila pemain berhasil meyemprot beberapa nyamuk, menyelesaikan permainan sesuai materi dan menuju pintu hutan.

4.5 Pengujian

Pengujian adalah hal yang perlu dilakukan untuk mengetahui 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.

4.5.1 Pengujian Alpha

Pengujian alpha dilakukan pada sisi pengembang yang dianggap pemakai dan merekam semua kesalahan dan masalah pemakaian. Pengujian alpha dilakukan pada sebuah lingkungan yang terkendali.

Pendekatan perancangan kasus uji menggunakan dua metode, yaitu *white box* dan *Black Box Testing*. *White box* adalah metode perancangan kasus uji yang

menggunakan struktur kontrol program. *Black box* adalah metode pengujian yang berfokus pada persyaratan fungsional perangkat lunak.

4.5.1.1 Rencana Pengujian

Rencana pengujian yang akan dilakukan pada aplikasi ini selengkapnya terlihat pada Tabel 4.1

Tabel 4.1 Rencana Pengujian Aplikasi

No	Komponen yang di uji	Skenario Pengujian	Jenis Pengujian
1.	Menu Utama	Memilih menu main	*Black box*
		Memilih menu petunjuk	*Black box*
		Memilih menu malaria	*Black box*
		Memilih menu keluar	*Black box*
2.	Menu Main	Memilih menu main	*Black box*
		Menekan tombol spasi	*Black box*
		Mengurangi darah karakter	*Black box*
		Menambah score	*Black box*
3.	Menu Petunjuk	Memilih menu petunjuk	*Black box*
		Memilih menu petunjuk kontrol	*Black box*
		Memilih menu petunjuk level	*Black box*
		Memilih tombol kembali	*Black box*
4.	Menu Malaria	Memilih menu malaria	*Black box*
		Memilih tombol kembali	*Black box*
5.	Menggerakan karakter	Menekan tombol atas	*Black box*
		Menekan tombol bawah	*Black box*
		Menekan tombol kiri	*Black box*
		Menekan tombol kanan	*Black box*
6.	Menembak	Menekan klik kiri	*Black box*
7.	Memasang kelambu	Drag and drop kelambu	*Black box*
		Menambah bonus score	*Black box*
8.	Menguras bak mandi	Menekan tutup bak mandi	*Black box*
		Drag sikat	*Black box*
		Menekan keran	*Black box*
		Menambah bonus score	*Black box*
9.	Menutup tempat air	Drag and drop tutup	*Black box*
		Menambah bonus score	*Black box*
10.	Mengubur wadah	Drag and drop wadah	*Black box*
		Menekan scope	*Black box*
		Menambah bonus score	*Black box*

11.	Melawan ratu	Mengurangi darah ratu	*Black box*
		Algoritma pencarian A*	*White box*
		Menambah bonus score	*Black box*

4.5.1.2 Pengujian White Box

Pengujian yang akan digunakan dalam menguji *algoritma A** adalah dengan menggunakan metode pengujian white box. Pengujian white box merupakan pengujian yang didasarkan pada pengecekan terhadap detail perancangan dan menggunakan struktur kontrol dari desain aplikasi secara procedural untuk membagi pengujian ke dalam beberapa kasus pengujian.

Penentuan kasus uji disesuaikan dengan struktur sistem, pengetahuan mengenai aplikasi digunakan untuk mengidentifikasikan kasus uji tambahan. Tujuan penggunaan white box adalah untuk menguji semua statement program. Penggunaan metode pengujian white box dilakukan untuk:

1. Memberikan jaminan bahwa semua independent paths suatu modul dikerjakan minimal satu kali.
2. Semua keputusan logis untuk kondisi true atau false pada aplikasi telah dikerjakan.

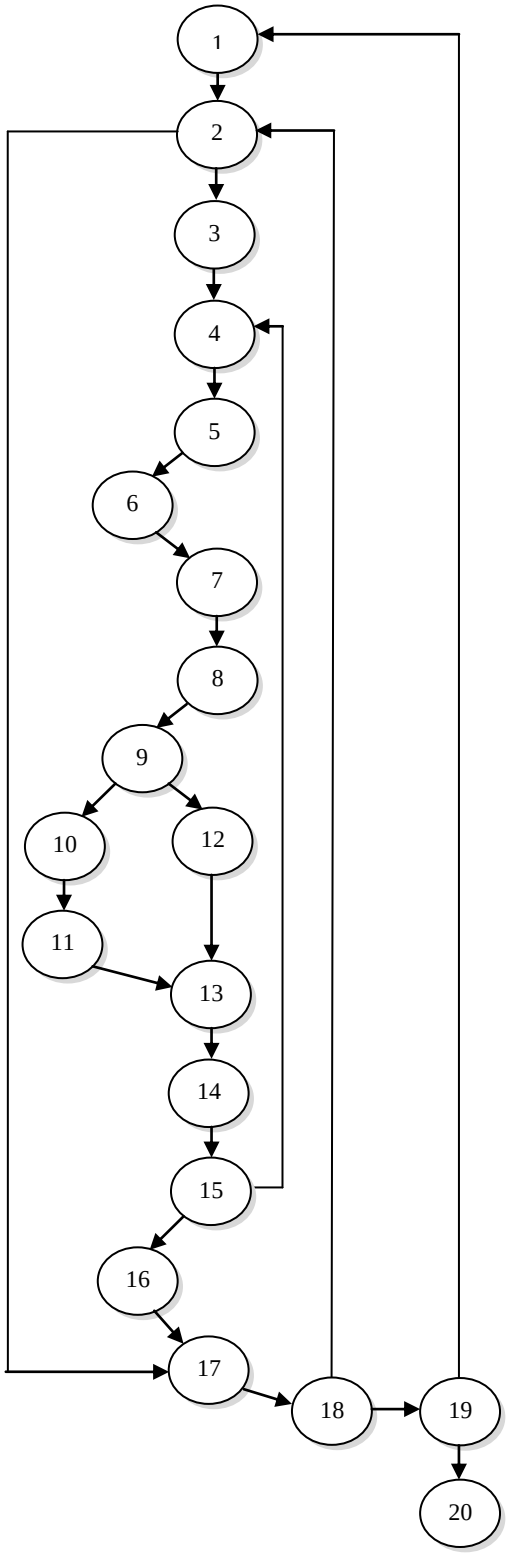
Apabila semua independen, paths telah dikerjakan, maka aplikasi yang bersangkutan telah benar. Tetapi jika terdapat satu saja node pada suatu Independen paths yang tidak dikerjakan, maka masih terdapat kesalahan pada aplikasi tersebut.

Berikut ini algoritma deskriptif algoritma pencarian A*:

```
1 : inisialisasi list
     OPEN = kosong
     CLOSED = kosong
     masukkan node awal ke list OPEN
         currentNode = firstNode
      menentukan perhitungan A* (A star)
         currentNode.g = 0
         currentNode.h = Pathfinder.euclidianHeuristic(currentNode,
         destinationNode)
         currentNode.f = currentNode.g + currentNode.h
2 :  while currentNode tidak sama dengan destinationNode
3 :      connectedNodes = connectedNodeFunction( currentNode )
4 :      for i = 1 to jumlah connectedNodes
             testNode = connectedNodes[i]
5 :          if testNode = currentNode & testNode.traversable = false
6 :                      g = currentNode.g + travelCost
                         h = Pathfinder.euclidianHeuristic( testNode,
                         destinationNode)
                         f = g + h
7 :          end if
8 :           if Pathfinder.isOpen(testNode, OPEN) dan
             Pathfinder.isClosed( testNode, CLOSED)
9 :           if testNode.f > f
10 :                     testNode.f = f
                         testNode.g = g
                         testNode.h = h
                         testNode.parentNode = currentNode
                         OPEN. parentNode (testNode)
11 :         end if
12 :         else
                         testNode.f = f
                         testNode.g = g
                         testNode.h = h
                         testNode.parentNode = currentNode
                         OPEN.push(testNode)
13:          end if
14:          end for
15:          if OPEN.length = 0
                         return null
16:          end if
17:          OPEN.sortOn('f', Array.NUMERIC)
                         currentNode = OPEN.shift()
18:  end while
19:  return Pathfinder.buildPath(destinationNode, firstNode)
20: end
```

Gambar 4.10 Deskriptif Algoritma A*

a. ***Flow Graph***



Gambar 4.11 *Flow Graph* Algoritma A*

Keterangan :

 = Menggambarkan Kondisi

 = Menggambarkan Aksi

b. Cyclomatic Complexity (VG)

V(G) = E – N + 2

V(G) = 24 – 20 +2

V(G) = 4+2 = 6

Keterangan :

E = Jumlah aksi

N = Jumlah Kondisi

P = Jumlah bagian dari *graph* yang tidak terkoneksi

c. Independent Path

Path 1 = 1 – 2 – 17 – 18 – 19 – 20

Path 2 = 1 – 2 – 17 – 18 – 2 – 17 – 18 – 19 – 20

Path 3 = 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 9 – 10 – 11 – 13 – 14 – 15 – 16 – 17 – 18 – 19 – 20

Path 4 = 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 9 – 12 – 13 – 14 – 15 – 16 – 17 – 18 – 19 – 20

Path 5 = 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 9 – 12 – 13 – 14 – 4 – 5 – 6 – 7 – 8 – 9 – 12 – 13 – 14 – 15 – 16 – 17 – 18 – 19 – 20

Path 6 = 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 9 – 12 – 13 – 14 – 15 – 16 – 17 – 18 – 2 – 17 – 18 – 19 – 20

d. Graph Matrix

Tabel 4.2 *Graph Matriks* Algoritma A*

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1		1																		
2			1														1			
3				1																
4					1															
5						1														
6							1													
7								1												
8									1											
9												1								
10									1		1									
11													1							
12													1							
13														1						
14				1											1					
15																1				
16																	1			
17																		1		
18		1																	1	
19	1																			1
20																				

Keterangan :

1. Baris dan kolom merepresentasikan simpul
2. Nilai 1 merepresentasikan adanya keterhubungan antar simpul

e. Hasil Uji

Tabel 4.3 Hasil Uji Algoritma A*

Kasus dan Hasil Uji		
No. *path*	**Kondisi**	**Kesimpulan**
1	Mencocokan node awal sama dengan node tujuan	*Path* dikerjakan
2	Mencari nilai F paling kecil dengan nilai heuristik menggunakan diagonal manhattan	*Path dikerjakan*
3	Mencari nilai F paling kecil dengan nilai heuristik menggunakan manhattan distance	*Path dikerjakan*
4	Mencari nilai F paling kecil apabila neighbor CLOSED	*Path dikerjakan*
5	Mencari nilai F paling kecil apabila neighbor OPEN atau neighboor kurang dari neighbor g	*Path dikerjakan*
6	Koneksi terhadap neighbor tidak ada, maka tidak dilakukan pencarian nilai F	*Path dikerjakan*

f. Kesimpulan

Dari hasil pengujian algoritma A* dari entitas NPC yang dilakukan, didapat kesimpulan bahwa seluruh *node* yang ada pada setiap *independent path* telah dikerjakan, serta tidak ada pengulangan tak berhingga.

4.5.1.3 Pengujian *Black Box*

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 4.2 Kasus dan Hasil Uji *Black Box*

No.	Kasus yang akan di uji	Skenario dan Hasil Uji		
		Skenario Uji	Hasil yang Diharapkan	Kesimpulan
1.	Menu Utama	Memilih Menu Main	Sistem menampilkan stage permainan	[√] Berhasil [ ] Tidak Berhasil
		Memilih Menu Petunjuk	Sistem menampilkan halaman menu petunjuk	[√] Berhasil [ ] Tidak Berhasil
		Memilih Menu Malaria	Sistem menampilkan halaman menu tentang malaria	[√] Berhasil [ ] Tidak Berhasil
		Memilih Menu Keluar	Keluar dari aplikasi	[√] Berhasil [ ] Tidak Berhasil
2.	Menu Main	Memilih Menu Main	Sistem menampilkan stage permainan	[√] Berhasil [ ] Tidak Berhasil
		Menekan tombol spasi	Sistem menampilkan tombol pause	[√] Berhasil [ ] Tidak Berhasil

		Mengurangi darah karakter	Sistem mengurangi darah karakter	[√] Berhasil [ ] Tidak Berhasil
		Menambah score	Sistem menambah score	[√] Berhasil [ ] Tidak Berhasil
3.	Menu Petunjuk	Memilih Menu Petunjuk	Sistem menampilkan halaman petunjuk	[√] Berhasil [ ] Tidak Berhasil
		Memilih Menu Petunjuk Kontrol	Sistem menampilkan halaman petunjuk kontrol	[√] Berhasil [ ] Tidak Berhasil
		Memilih Menu Petunjuk Level	Sistem menampilkan halaman petunjuk level	[√] Berhasil [ ] Tidak Berhasil
		Memilih tombol kembali	Sistem menampilkan menu utama	[√] Berhasil [ ] Tidak Berhasil
4.	Menu Malaria	Memilih Menu Malaria	Sistem menampilkan halaman tentang malaria	[√] Berhasil [ ] Tidak Berhasil
		Memilih tombol kembali	Sistem menampilkan menu utama	[√] Berhasil [ ] Tidak Berhasil
5.	Menggerakkan Karakter	Menekan tombol atas	Sistem menggerakkan karakter loncat	[√] Berhasil [ ] Tidak Berhasil
		Menekan tombol bawah	Sistem menggerakkan karakter masuk pintu hutan	[√] Berhasil [ ] Tidak Berhasil

		Menekan tombol kiri	Sistem menggerakkan karakter ke kiri	[√] Berhasil [ ] Tidak Berhasil
		Menekan tombol kanan	Sistem menggerakkan karakter ke kanan layar	[√] Berhasil [ ] Tidak Berhasil
6.	Menembak	Menekan klik kiri	Sistem mengeluarkan peluru dari karakter	[√] Berhasil [ ] Tidak Berhasil
7.	Memasang kelambu	Drag and drop kelambu	Sistem mencocokan kelambu dengan kasur	[√] Berhasil [ ] Tidak Berhasil
		Menambah bonus score	Sistem menambahkan bonus score	[√] Berhasil [ ] Tidak Berhasil
8.	Menguras bak mandi	Menekan tutup bak mandi	Sistem membuka tutup bak mandi	[√] Berhasil [ ] Tidak Berhasil
		Drag sikat	Sistem membersihkan kotoran dengan sikat	[√] Berhasil [ ] Tidak Berhasil
		Menekan keran	Sistem menutup keran bak mandi	[√] Berhasil [ ] Tidak Berhasil
		Menambah bonus score	Sistem menambahkan bonus score	[√] Berhasil [ ] Tidak Berhasil
9.	Menutup tempat air	Drag and drop tutup	Sistem mencocokan tutup dengan tempat air	[√] Berhasil [ ] Tidak Berhasil
		Menambah bonus score	Sistem menambahkan bonus score	[√] Berhasil [ ] Tidak Berhasil

10.	Mengubur wadah	Drag and drop wadah	Sistem memasukan wadah ke lubang	[√] Berhasil [ ] Tidak Berhasil
		Menekan scope	Sistem mengubur lubang	[√] Berhasil [ ] Tidak Berhasil
		Menambah bonus score	Sistem menambahkan bonus score	[√] Berhasil [ ] Tidak Berhasil
11.	Melawan ratu	Mengurangi darah ratu	Sistem mengurahi darah ratu saat terkena semprotan	[√] Berhasil [ ] Tidak Berhasil
		Algoritma pencarian A*	Dapat dilihat pada sub bab 4.5.1.2 Kasus dan Hasil Pengujian (*White box Testing*)	
		Menambah bonus score	Sistem menambahkan bonus score	[√] Berhasil [ ] Tidak Berhasil

4.5.1.4 Kesimpulan Pengujian Alpha

Dari hasil pengujian yang dilakukan, dapat disimpulkan bahwa sistem sudah berjalan sesuai dengan yang diharapkan. Secara fungsional sistem sudah dapat menghasilkan keluaran yang diharapkan.

4.5.2 Pengujian Beta

4.5.2.1 Evaluasi dari Pengguna

Untuk mengetahui tanggapan dan penilaian pemain terhadap game ini. Telah disebarkan kuesioner kepada 30 responden dengan usia yang berdasarkan pada sasaran pengguna. Berdasarkan data hasil kuesioner tersebut, dapat dicari presentase masing-masing jawaban dengan menggunakan rumus skala likert sebagai berikut :

$$P = \frac{S}{skorideal} \times 100\%$$

Keterangan :

P = Nilai presentase yang dicari

S = Jumlah frekuensi dikalikan dengan nilai yang ditetapkan jawaban

skorideal = Nilai tertinggi dikalikan dengan jumlah sampel

100% = Bilangan tetap

Hasil pengujian *Beta* adalah sebagai berikut:

1. Kuesioner Terhadap Siswa – Siswi Sekolah Dasar

Hasil Kuesioner:

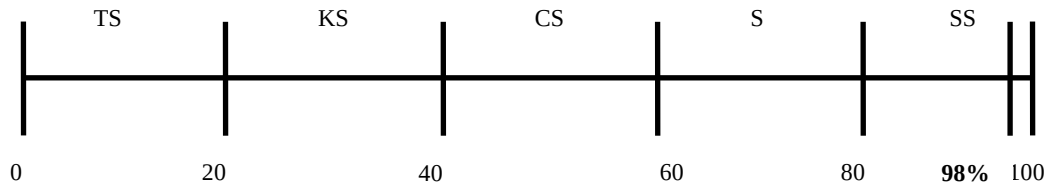
1. Apakah game edukasi Save Family From Malaria dapat memberikan pengetahuan tentang pencegahan malaria bagi anda?

Tabel 4.5 Tabel Perhitungan Quisoner Pertanyaan 1

Pertanyaan	**No**	**Keterangan**	**Skor**	**Frekuensi**	**S**
1	1	Sangat Setuju	5	27	135
	2	Setuju	4	3	12
	3	Cukup Setuju	3	0	0
	4	Kurang Setuju	2	0	0
	5	Tidak Setuju	1	0	0
Jumlah				**30**	**147**

$$p = \frac{147}{150} X\ 100\% = 98\%$$

Berdasarkan hasil presentase diatas maka dapat disimpulkan bahwa penilaian *game* dapat memberikan pengetahuan pencegahan malaria adalah 98% dari yang diharapkan 100% atau bisa dikategorikan sebagai sangat setuju. Berikut kriteria interpretasi skor :



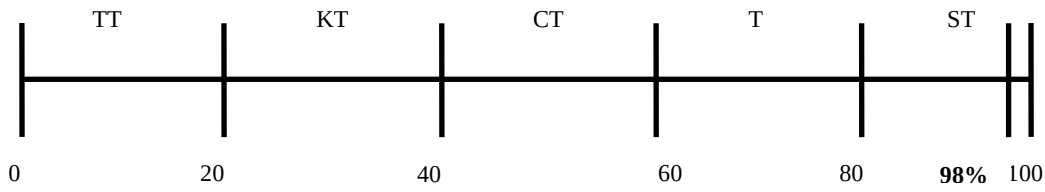
2. Apakah dengan adanya game ini anda lebih tertarik untuk mengetahui pencegahan malaria?

Tabel 4.6 Tabel Perhitungan Quisoner Pertanyaan 2

Pertanyaan	**No**	**Keterangan**	**Skor**	**Frekuensi**	**S**
2	1	Sangat Tertarik	5	28	140
	2	Tertarik	4	1	4
	3	Cukup Tertarik	3	1	3
	4	Kurang Tertarik	2	0	0
	5	Tidak Tertarik	1	0	0
Jumlah				**30**	**147**

$$p = \frac{147}{150} X\ 100\% = 98\%$$

Berdasarkan hasil persentase diatas maka dapat disimpulkan bahwa pengguna sangat tertarik untuk mengetahui pencegahan malaria setelah memainkan *game* ini adalah 98% dari yang diharapkan 100% atau bisa dikategorikan sebagai sangat tertarik. Berikut kriteria interpretasi skor :



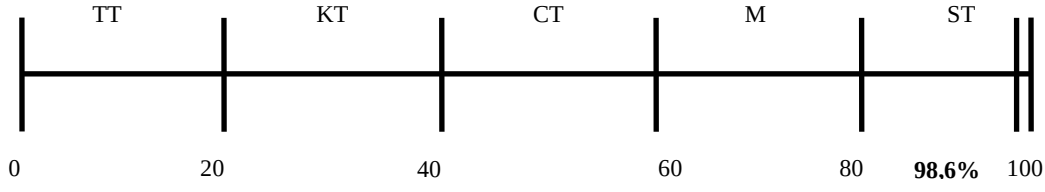
3. Apakah dengan adanya game ini anda merasa terhibur dalam mempelajari pencegahan malaria?

Tabel 4.7 Tabel Perhitungan Quisoner Pertanyaan 3

Pertanyaan	**No**	**Keterangan**	**Skor**	**Frekuensi**	**S**
3	1	Sangat Terhibur	5	28	140
	2	Menghibur	4	2	8
	3	Cukup Terhibur	3	0	0
	4	Kurang Terhibur	2	0	0
	5	Tidak Terhibur	1	0	0
Jumlah				**30**	**148**

$$p = \frac{148}{150} X\ 100\% = 98.6\%$$

Berdasarkan hasil persentase diatas maka dapat disimpulkan, bahwa penilaian terhadap tampilan dan animasi *game* ini dapat menghibur pengguna untuk menambah pengetahuan pencegahan malaria adalah 98,6% dari yang diharapkan 100% atau bisa dikategorikan sebagai sangat terhibut. Berikut interpretasi skor :



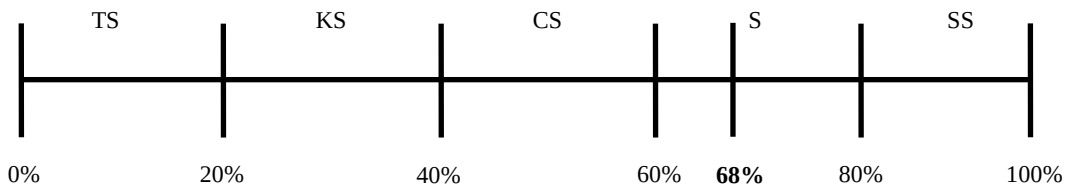
4. Apakah dengan penyajian yang interaktif dan edukatif anda mendapatkan pemahaman yang lebih jelas tentang pencegahan malaria?

Tabel 4.8 Tabel Perhitungan Quisoner Pertanyaan 4

Pertanyaan	**No**	**Keterangan**	**Skor**	**Frekuensi**	**S**
4	1	Sangat Setuju	5	28	140
	2	Setuju	4	2	8
	3	Cukup Setuju	3	0	0
	4	Kurang Setuju	2	0	0
	5	Tidak Setuju	1	0	0
Jumlah				**30**	**148**

$$p = \frac{148}{150} X\ 100\% = 98.6\%$$

Berdasarkan hasil persentase diatas maka dapat disimpulkan bahwa melalui penyajian yang interaktif dan edukatif pada *game* ini dapat memberikan pemaham yang lebih jelas tentang pencegahan malaria adalah 98,6% dari yang diharapkan 100% atau bisa dikategorikan sebagai sangat setuju. Berikut adalah interpretasi skor :



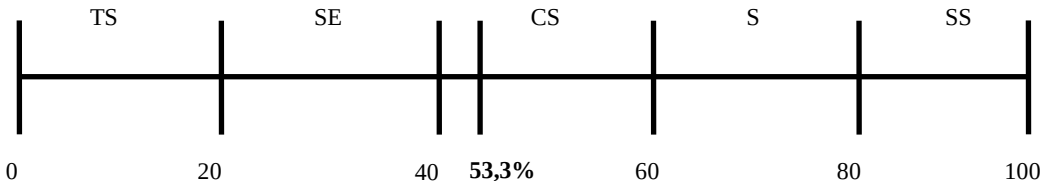
5. Apakah anda kesulitan dalam memainkan game pencegahan malaria pada game edukasi Save Family From Malaria?

Tabel 4.8 Tabel Perhitungan Quisoner Pertanyaan 5

Pertanyaan	No	Keterangan	Skor	Frekuensi	S
5	1	Sangat Sulit	5	7	35
	2	Sulit	4	0	0
	3	Cukup Sulit	3	8	24
	4	Sedang	2	6	12
	5	Tidak Sulit	1	9	9
Jumlah				**30**	**128**

$$p = \frac{80}{150} X\ 100\% = 53{,}3\%$$

Berdasarkan hasil persentase diatas maka dapat disimpulkan, bahwa penilaian terhadap tingkat kesulitan dari permainan di dalam *game* ini adalah 53,3% dari yang diharapkan 100% atau bisa dikategorikan sebagai cukup sulit. Berikut interpretasi skor :



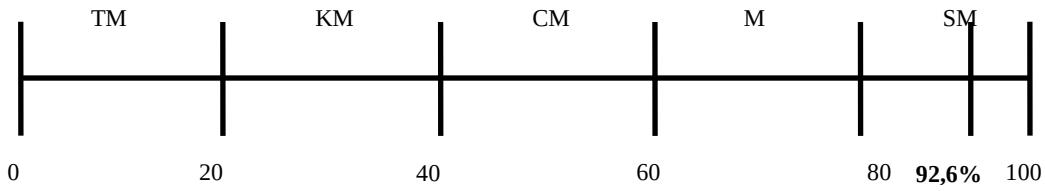
6. Apakah petunjuk yang terdapat pada game edukasi Save Family From Malaria membantu anda dalam memainkan game ini?

Tabel 4.9 Tabel Perhitungan Quisoner Pertanyaan 6

Pertanyaan	No	Keterangan	Skor	Frekuensi	S
6	1	Sangat Membantu	5	24	120
	2	Membantu	4	3	12
	3	Cukup Membantu	3	2	6
	4	Kurang Membantu	2	0	0
	5	Tidak Membantu	1	1	1
Jumlah				**30**	**139**

$$p = \frac{139}{150} X\ 100\% = 92{,}6\%$$

Berdasarkan hasil persentase diatas maka dapat disimpulkan, bahwa penilaian terhadap petunjuk dalam *game* ini telah membantu Anda dalam memainkan *game* adalah 92,6% dari yang diharapkan 100% atau bisa dikategorikan sebagai sangat membantu. Berikut interpretasi skor :



4.5.2.2 Kesimpulan Pengujian Beta

Berdasarkan hasil persentase yang didapatkan dari hasil pengisian kuesioner yang diberikan kepada siswa SD, maka diperoleh kesimpulan bahwa aplikasi yang di bangun telah memenuhi tujuan yang ingin dicapai yaitu memberikan pengetahuan terhadap masyarakat khususnya anak-anak tentang bahaya dan cara pencegahan penyakit malaria, anak-anak dapat tertarik sehingga paham bagaimana pencegahan malaria yang baik, dan memberikan hiburan yang

edukatif dengan alur cerita dan pesan-pesan tentang malaria yang interaktif dengan adanya animasi. Berikut adalah hasil presentase dari setiap pertanyaan, yaitu :

1. Sebanyak 98% responden menyatakan sangat setuju game edukasi Save Family From Malaria dapat memberikan pengetahuan tentang pencegahan malaria.
2. Sebanyak 98% responden menyatakan menjadi sangat tertarik tertarik untuk mengetahui pencegahan malaria setelah memainkan game ini.
3. Sebanyak 98,6% responden menyatakan sangat terhibur dalam mempelajari pencegahan malaria dengan game ini.
4. Sebanyak 98,6% menyatakan sangat setuju dengan penyajian yang interaktif dan edukatif dapat memberikan pemahaman yang lebih jelas tentang pencegahan malaria
5. Sebanyak 53,3% menyatakan cukup culit dalam memainkan game pencegahan malaria pada game edukasi Save Family From Malaria.
6. Sebanyak 92,6% menyatakan petunjuk yang terdapat pada game edukasi Save Family From Malaria sangat membantu dalam memainkan game ini