BAB 4

IMPLEMENTASI DAN PENGUJIAN

4.1 Implementasi

Proses implementasi merupakan tahapan dimana aplikasi yang sudah dibuat dan dirancang, diuji kelayakannya.

4.1.1 Implementasi Perangkat Keras

Perangkat keras yang digunakan untuk melakukan implementasi dan pengujian aplikasi yaitu sebagai berikut :

1. *Processor* dengan kecepatan 1.8 Ghz
2. Memori 256 MB
3. *Space* kosong di *harddisk* 11 MB
4. *VGA card*
5. Monitor
6. *Mouse*
7. *Keyboard*
8. *Speaker*

4.1.2 Implementasi Perangkat Lunak

Perangkat lunak yang digunakan untuk melakukan implementasi dan pengujian aplikasi yaitu sebagai berikut :

1. Sistem operasi *windows 8*
2. *Flash Player* 10
3. *GoGreen.exe*

4.1.3 Implementasi *Class*

Implementasi *class* merupakan implementasi dari perancangan *class diagram* yang terbentuk dari *use case diagram* ke dalam bentuk *file* fisik dengan ekstensi ".as".

Tabel 4.1 Implementasi Class

No	Nama *Class*	Nama *File* Fisik
1	*Main*	*Main*.as
2	*Hud*	*Hud*.as
3	*Node*	*Node*.as
4	*Astar*	*Astar*.as
5	Kuis	Kuis.as
6	Kepompong	Kepompong.as

7	Sasaran	Sasaran.as
8	Info	Info.as
9	*Level*	*Level*.as
10	*PathMv*	*PathMv*.as
11	Koin	Koin.as
12	Ulat	Ulat.as
13	TInfo	TInfo.as
14	*Sounds*	*Sounds*.as

4.1.4 Implementasi Antarmuka

Pada bagian ini akan ditampilkan implementasi antarmuka *menu* utama, *menu* pengaturan, *menu* bantuan, petunjuk *level* 1, permainan *level* 1, info *level* 1 ke-1, info *level* 1 ke-2, info *level* 1 ke-3, kuis *level* 1, petunjuk *level* 2, permainan *level* 2, info *level* 2 ke-1, info *level* 2 ke-2, info *level* 2 ke-3, jalur terpendek *level* 2, kuis *level* 2, petunjuk *level* 3, permainan *level* 3 *fase* 1, info *level* 3 *fase* 1 ke-1, info *level* 3 *fase* 1 ke-2, info *level* 3 *fase* 1 ke-3, kuis *level* 3 *fase* 1, permainan *level* 3 *fase* 2, info *level* 3 *fase* 2 ke-1, *level* 3 *fase* 2 ke-2, *level* 3 *fase* 2 ke-3, dan kuis *level* 3 *fase* 2.

4.1.4.1 Implementasi Antarmuka *Menu* Utama

Berikut adalah implementasi antarmuka *menu* utama dari permainan.



Gambar 4.1 Implementasi *Menu* Utama

4.1.4.2 Implementasi Antarmuka *Menu* Pengarutan

Berikut adalah implementasi antarmuka *menu* pengaturan permaian.



Gambar 4.2 Implementasi Antarmuka Pengaturan

4.1.4.3 Implementasi Antarmuka *Menu* Bantuan

Berikut adalah implementasi antarmuka *menu* bantuan permaian.



Gambar 4.3 Implementasi Antarmuka *Menu* Bantuan

4.1.4.4 Implementasi Antarmuka Petunjuk *Level 1*

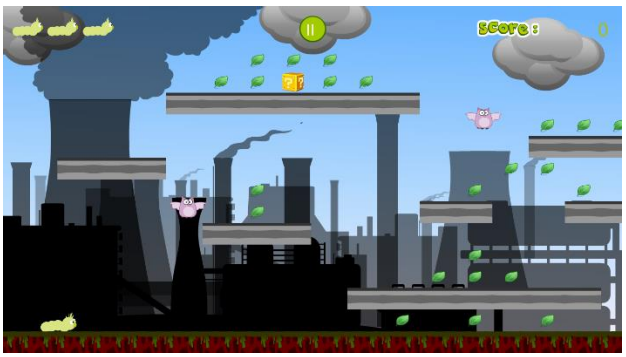
Berikut ini adalah implementasi antarmuka petunjuk *level* 1.



Gambar 4.4 Implementasi Antarmuka Petunjuk *Level* 1

4.1.4.5 Implementasi Antarmuka Permainan *Level* 1

Berikut ini adalah implementasi antarmuka permainan *level* 1.

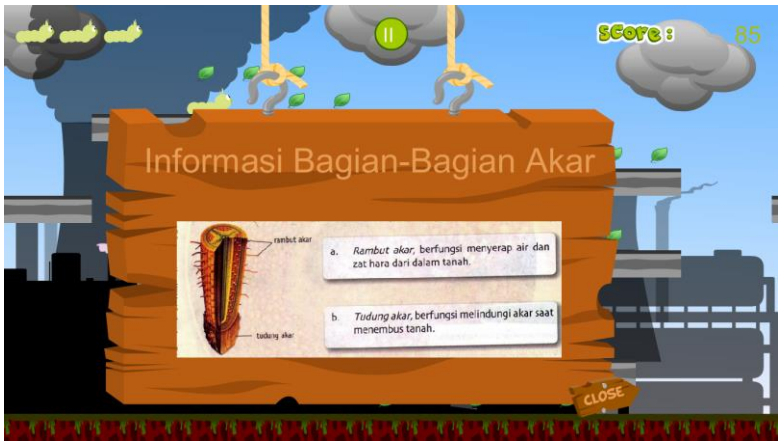


Gambar 4.5 Implementasi Antarmuka Permainan *Level* 1

4.1.4.6 Implementasi Antarmuka Informasi *Level* 1

Pada permainan *level* 1 terdapat 3 informasi. Berikut ini adalah masing-masing implementasi antarmuka informasi pada *level* 1.

1. Implementasi Antarmuka Informasi Bagian-Bagian Akar



Gambar 4.6 Implementasi Antarmuka Informasi Bagian-Bagian Akar

2. Implementasi Antarmuka Informasi Fungsi Akar



Gambar 4.7 Implementasi Antarmuka Informasi Fungsi Akar

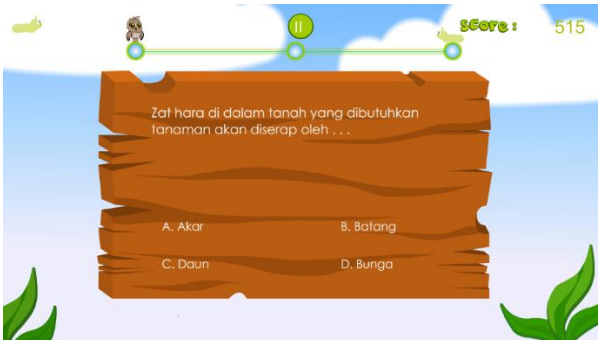
3. Implementasi Antarmuka Informasi Jenis Akar



Gambar 4.8 Implementasi Antarmuka Informasi Jenis Akar

4.1.4.7 Implementasi Antarmuka Kuis *Level* 1

Berikut ini adalah implementasi antarmuka kuis *level* 1.



Gambar 4.9 Implementasi Antarmuka Kuis *Level* 1

4.1.4.8 Implementasi Antarmuka Petunjuk *Level* 2

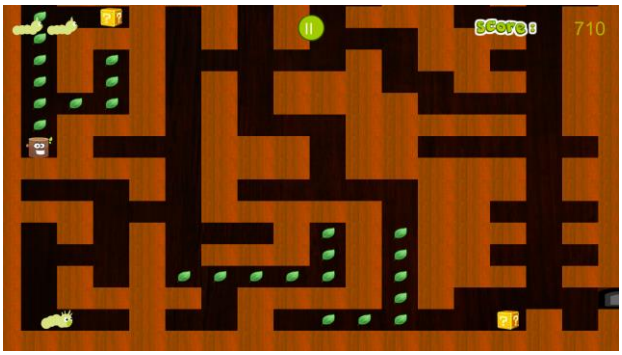
Berikut ini adalah implementasi antarmuka petunjuk *level* 2.



Gambar 4.10 Implementasi Antarmuka Petunjuk *Level* 2

4.1.4.9 Implementasi Antarmuka Permainan *Level* 2

Berikut ini adalah implementasi antarmuka permainan *level* 2.



Gambar 4.11 Implementasi Antarmuka Permaian *Level* 2

4.1.4.10 Implementasi Antarmuka Informasi *Level* 2

Pada permainan *level* 2 terdapat 3 informasi. Berikut ini adalah masing-masing implementasi antarmuka informasi pada *level* 2.

1. Implementasi Antarmuka Informasi Jenis Batang



Gambar 4.12 Implementasi Antarmuka Informasi Jenis Batang

2. Implementasi Antarmuka Informasi Jenis Batang Tidak Berkayu



Gambar 4.13 Implementasi Antarmuka Informasi Jenis Batang Tidak Berkayu

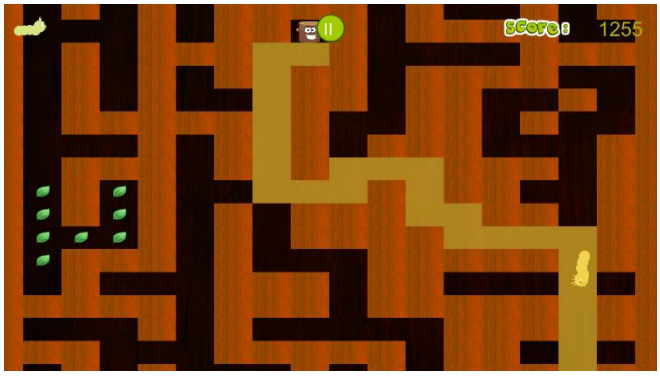
3. Implementasi Antarmuka Informasi Bagian Batang dan Fungsinya



Gambar 4.14 Implementasi Antarmuka Informasi Bagian Batang dan Fungsinya

4.1.4.11 Implementasi Antarmuka Jalur Terpendek *Level* 2

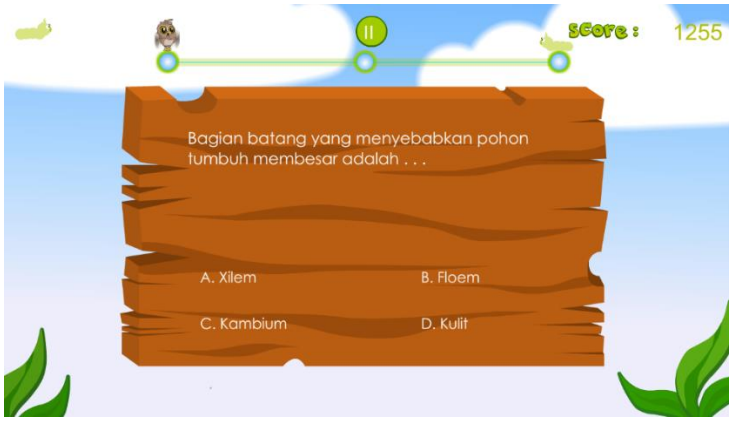
Berikut ini adalah implementasi antarmuka jalur terpendek pada *level* 2.



Gambar 4.15 Implementasi Antarmuka Jalur Terpendek Pada *Level* 2

4.1.4.12 Implementasi Antarmuka Kuis *Level* 2

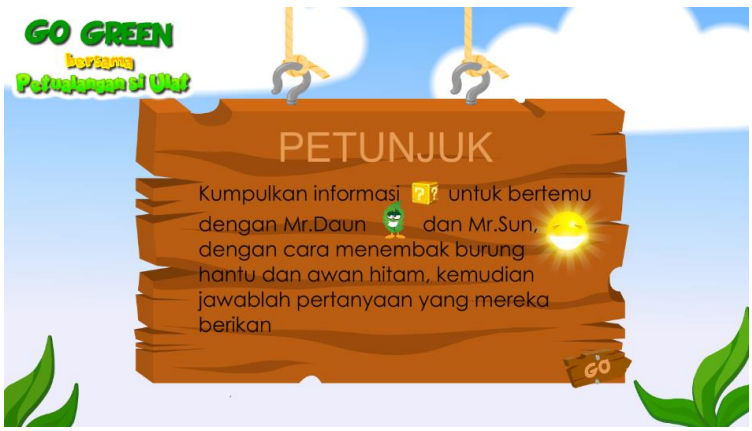
Berikut ini adalah implementasi antarmuka kuis *level* 2.



Gambar 4.16 Implementasi Antarmuka Kuis *Level* 2

4.1.4.13 Implementasi Antarmuka Petunjuk *Level* 3

Berikut ini adalah implementasi antarmuka petunjuk *level* 3.



Gambar 4.17 Implementasi Antarmuka Petunjuk *Level* 3

4.1.4.14 Implementasi Antarmuka Permainan *Level* 3 *Fase* 1

Berikut ini adalah implementasi antarmuka permainan *level* 3 *fase* 1.



Gambar 4.18 Implementasi Antarmuka Permainan *Level* 3 *Fase* 1

4.1.4.15 Implementasi Antarmuka Informasi *Level* 3 *Fase* 1

Pada permainan *level* 3 terdapat 3 informasi. Berikut ini adalah masing-masing implementasi antarmuka informasi pada *level* 3.

1. Implementasi Antarmuka Informasi Fotosintesis



Gambar 4.19 Implementasi Antarmuka Informasi Fotosintesis

2. Implementasi Antarmuka Informasi Bentuk Daun



Gambar 4.20 Implementasi Antarmuka Informasi Bentuk Daun

3. Implementasi Antarmuka Informasi Bagian Pada Daun



Gambar 4.21 Implementasi Antarmuka Informasi Bagian Pada Daun

4.1.4.16 Implementasi Antarmuka Kuis *Level* 3 *Fase* 1

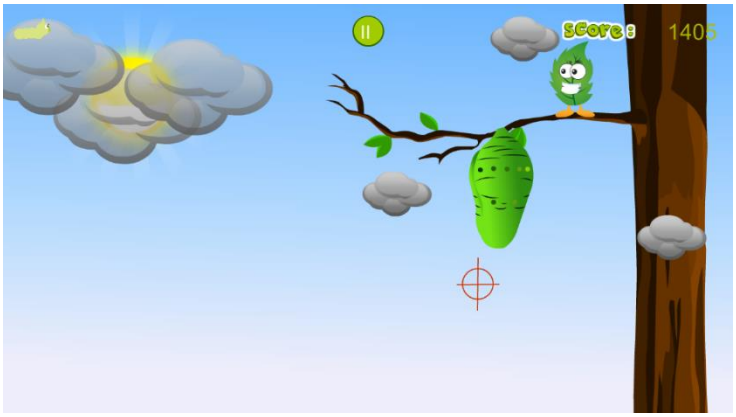
Berikut adalah implementasi antarmuka kuis *level* 3 *fase* 1.



Gambar 4.22 Implementasi Antarmuka Kuis *Level* 3 *Fase* 1

4.1.4.17 Implementasi Antarmuka Permainan *Level* 3 *Fase* 2

Berikut ini adalah implementasi antarmuka permainan *level* 3 *fase* 2.



Gambar 4.23 Implementasi Antarmuka Permainan *Level* 3 *Fase* 2

4.1.4.18 Implementasi Antarmuka Informasi *Level* 3 *Fase* 2

Pada permainan *level* 3 terdapat 3 informasi. Berikut ini adalah masing-masing implementasi antarmuka informasi pada *level* 3.

1. Implementasi Antarmuka Informasi Bagian dan Fungsi Bunga



Gambar 4.24 Implementasi Antarmuka Informasi Bagian dan Fungsi Bunga

2. Implementasi Antarmuka Informasi Bunga Sempurna dan Tidak Sempurna



Gambar 4.25 Implementasi Antarmuka Informasi Bunga Sempurna dan Tidak Sempurna

3. Implementasi Antarmuka Informasi Bagian dan Fungsi Buah



Gambar 4.26 Implementasi Antarmuka Informasi Bagian dan Fungsi Buah

4.1.4.19 Implementasi Antarmuka Kuis *Level* 3 *Fase* 2

Berikut adalah implementasi antarmuka kuis *level* 3 *fase* 2.



Gambar 4.27 Implementasi Antarmuka Kuis *Level* 3 *Fase* 2

4.1.5 Implementasi Pesan

Pada bagian ini akan ditampilkan implementasi pesan moral *level* 1, pesan moral *level* 2, pesan moral *level* 3.

4.1.5.1 Implementasi Antarmuka Pesan Moral *Level* 1

Berikut ini adalah implementasi antarmuka pesan moral *level* 1.



Gambar 4.28 Implementasi Antarmuka Pesan Moral *Level* 1

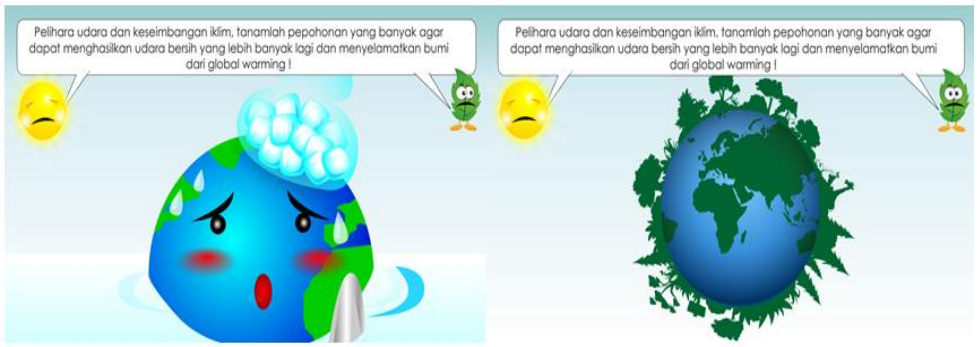
4.1.5.2 Implementasi Antarmuka Pesan Moral *Level* 2

Berikut ini adalah implementasi antarmuka pesan moral *level* 2.



Gambar 4.29 Implementasi Antarmuka Pesan Moral Level 2

4.1.5.3 Implementasi Antarmuka Pesan Moral *Level* 3



Gambar 4.30 Implementasi Antarmuka Pesan Moral *Level* 3

4.2 Pengujian Sistem

Pengujian bermaksud untuk mengetahui perangkat lunak yang dibuat telah memenuhi kriteria yang sesuai dengan tujuan perancangan perangkat lunak tersebut.

4.2.1 Rencana Pengujian

Pengujian akan dilakukan melalui dua tahap, yaitu pengujian *alpha* dan pengujian *beta*.

1. Pengujian *Alpha*

Metode yang digunakan dalam pengujian *alpha* adalah *metode black box* yang fokus pada persyaratan fungsional dari perangkat lunak yang dibangun.

2. Pengujian *Beta*

Metode yang digunakan dalam pengujian *beta* merupakan pengujian yang dilakukan secara objektif terhadap pengguna.

4.2.2 Pengujian *Alpha*

Pengujian dilakukan terhadap aplikasi untuk memastikan bahwa aplikasi berjalan dengan benar sesuai dengan kebutuhan dan tujuan yang diharapkan. Proses pengujian akan menggunakan *Black Box Testing*.

4.2.2.1 *Scenario* Pengujian

Scenario pengujian menjelaskan pengujian terhadap sistem yang ada pada *game go green* bersama petualangan si ulat. Pengujian dilakukan dengan melakukan tes setiap *use case* dan memungkinkan kesalahan yang terjadi pada setiap proses. Pengujian ini dilakukan secara *black box* yaitu pengujian dilakukan

dengan hanya memperhatikan masukan ke sistem dan keluaran sistem. Tabel berikut ini menjelaskan *scenario* pengujian yang akan dilakukan.

Tabel 4.2 *Scenario* Pengujian

Use case	**Butir Pengujian**	**Jenis Pengujian**
Memulai permainan	Mengklik tombol main	*Black Box*
	Menampilkan *loading*	*Black Box*
Membaca petunjuk	Mengklik tombol *go*	*Black Box*
	Menampilkan permainan	*Black Box*
Memainkan *level* 1	Menggerakan ulat ke kotak informasi *level* 1	*Black Box*
	Menambah *score level* 1 hasil dari mengenai daun	*Black Box*
	Mengenai *block* menampilkan ulat tidak bisa bergerak pada *level* 1	*Black Box*
	Mengurangi nyawa mengenai musuh *level* 1	*Black Box*
	Menambah *score level* 1 hasil dari mengambil kotak informasi *level* 1	*Black Box*
	Menampilkan informasi pada *level* 1	*Black Box*
	Mengklik tombol *close*	*Black Box*
	Menambah jumlah kotak informasi *level* 1	*Black Box*
	Menggerakan ulat ke tokoh pembantu *level* 1	*Black Box*
	Menampilkan kuis *level* 1	*Black Box*
Memainkan *level* 2	Menggerakan ulat ke kotak informasi *level* 2	*Black Box*
	Mengenai *block* menampilkan ulat tidak bisa bergerak pada *level* 2	*Black Box*
	Menambah *score level* 2 hasil dari mengenai daun	*Black Box*
	Menambah *score level* 2 hasil dari mengambil kotak informasi *level* 2	*Black Box*
	Menampilkan informasi pada *level* 2	*Black Box*
	Mengklik tombol *close*	*Black Box*
	Menambah jumlah kotak informasi *level* 2	*Black Box*
	Menggerakan ulat ke tokoh pembantu *level* 2	*Black Box*
	Menampilkan jalan keluar dari solusi A*	*Black Box*
	Menggerakan ulat ke pintu keluar	*Black Box*
	Menampilkan kuis *level* 2	*Black Box*
Memainkan *level* 3	Menembaki musuh	*Black Box*
	Menambah jumlah tertembak	*Black Box*
	Musuh mengenai kepompong maka nyawa berkurang	*Black Box*
	Musuh mati sama dengan 3 maka menampilkan informasi	*Black Box*
	Mengklik tombol *close*	*Black Box*
	Menambah *session*	*Black Box*
	Session lebih dari 3 maka menampilkan tanda tanya pada tokoh pembantu	*Black Box*
	Mengklik tanda tanya pada tokoh pembantu	*Black Box*
	Menampilkan kuis *level* 3	*Black Box*
	Nyawa sama dengan nol maka menampilkan permainan gagal	*Black Box*
Menjawab kuis	Menjawab kuis	*Black Box*
	Menambah *score* hasil dari menjawab benar	*Black Box*

	Jumlah pertanyaan sudah sama dengan 3 dengan salah tidak sama dengan 2 maka menampilkan pesan moral	*Black Box*
	Menjawab salah maka menambah 1 jumlah salah	*Black Box*
	Jumlah pertanyaan sudah sama dengan 3 dengan salah sama dengan 2 maka permainan gagal	*Black Box*
Melihat *menu* bantuan	Mengklik tombol bantuan pada *menu* utama	*Black Box*
	Menampilkan *menu* bantuan	*Black Box*
Mengatur suara	Mengklik tombol pengaturan	*Black Box*
	Menampilkan *menu* pengaturan	*Black Box*
	Mengklik ya pada matikan *backsound*	*Black Box*
	Mematikan *backsound* permaian	*Black Box*
	Mengklik ya pada *mute*	*Black Box*
	Mematikan semua *sound* pada permaian	*Black Box*

4.2.2.2 Pengujian *Black Box*

Tabel berikut adalah hasil dari pengujian *black box* yang dilakukan sesuai pengujian *scenario* pengujian yang telah dirancang sebelumnya.

Tabel 4.3 Hasil Pengujian *Black Box*

Memulai permainan		
Skenario	**Hasil yang diharapkan**	**Hasil pengujian**
Mengklik tombol main	Menampilkan *loading*	[√] berhasil [ ] tidak berhasil
Membaca petunjuk		
Skenario	**Hasil yang diharapkan**	**Hasil pengujian**
Mengklik tombol *go*	Menampilkan permainan	[√] berhasil [ ] tidak berhasil
Memainkan *level* 1		
Menggerakan ulat ke kotak informasi *level* 1	Menambah *score level* 1 hasil dari mengenai daun	[√] berhasil [ ] tidak berhasil
	Mengenai *block* menampilkan ulat tidak bisa bergerak pada *level* 1	[√] berhasil [ ] tidak berhasil
	Mengurangi nyawa *level* 1	[√] berhasil [ ] tidak berhasil
	Menambah *score level* 1 hasil dari mengambil kotak informasi pada *level* 1	[√] berhasil [ ] tidak berhasil
	Menampilkan informasi pada *level* 1	[√] berhasil [ ] tidak berhasil
Mengklik tombol *close*	Menambah jumlah kotak informasi pada *level* 1	[√] berhasil [ ] tidak berhasil
	Menggerakan ulat ke tokoh pembantu *level* 1	[√] berhasil [ ] tidak berhasil
	Menampilkan kuis pada *level* 1	[√] berhasil [ ] tidak berhasil
	Mengurangi nyawa mengenai musuh *level* 1	[√] berhasil [ ] tidak berhasil
	Menambah *score level* 1 hasil dari mengambil kotak informasi pada *level* 1	[√] berhasil [ ] tidak berhasil
Memainkan *level* 2		

Menggerakan ulat ke kotak informasi *level* 2	Mengenai *block* menampilkan ulat tidak bisa bergerak pada *level* 2	[√] berhasil [ ] tidak berhasil
	Menambah *score level* 2 hasil dari mengenai daun	[√] berhasil [ ] tidak berhasil
	Menambah *score level* 2 hasil dari mengambil kotak informasi pada *level* 2	[√] berhasil [ ] tidak berhasil
	Menampilkan informasi pada *level* 2	[√] berhasil [ ] tidak berhasil
Mengklik tombol *close*	Menambah jumlah kotak informasi pada *level* 2	[√] berhasil [ ] tidak berhasil
Menggerakan ulat ke tokoh pembantu *level* 2	Menampilkan jalan keluar dari solusi A*	[√] berhasil [ ] tidak berhasil
Menggerakan ulat ke pintu keluar	Menampilkan kuis pada *level* 2	[√] berhasil [ ] tidak berhasil
Memainkan *level* 3		
Menembaki musuh	Menambah jumlah musuk yang tertembak	[√] berhasil [ ] tidak berhasil
	Musuh mengenai kepompong maka nyawa berkurang	[√] berhasil [ ] tidak berhasil
	Musuh mati sama dengan 3 maka menampilkan informasi	[√] berhasil [ ] tidak berhasil
Mengklik tombol *close*	Menambah *session*	[√] berhasil [ ] tidak berhasil
	Session lebih dari 3 maka menampilkan tanda tanya pada tokoh pembantu	[√] berhasil [ ] tidak berhasil
Mengklik tanda tanya pada sun	Menampilkan kuis pada *level* 3	[√] berhasil [ ] tidak berhasil
	Nyawa sama dengan nol maka menampilkan permainan gagal	[√] berhasil [ ] tidak berhasil
Menjawab kuis		
Menjawab kuis	Menambah *score* hasil dari menjawab benar	[√] berhasil [ ] tidak berhasil
	Jumlah pertanyaan sudah sama dengan 3 dengan salah tidak sama dengan 2 maka menampilkan pesan moral	[√] berhasil [ ] tidak berhasil
	Menjawab salah maka menambah 1 jumlah salah	[√] berhasil [ ] tidak berhasil
	Jumlah pertanyaan sudah sama dengan 3 dengan salah sama dengan 2 maka permainan gagal	[√] berhasil [ ] tidak berhasil
Melihat *menu* bantuan		
Mengklik tombol bantuan pada *menu* utama	Menampilkan *menu* bantuan	[√] berhasil [ ] tidak berhasil
Mengatur suara		
Mengklik tombol pengaturan	Menampilkan *menu* pengaturan	[√] berhasil [ ] tidak berhasil
Mengklik ya pada matikan *backsound*	Mematikan *backsound* permaian	[√] berhasil [ ] tidak berhasil
Mengklik ya pada *mute*	Mematikan semua *sound* pada permaian	[√] berhasil [ ] tidak berhasil

4.2.2.3 Pengujinan *White Box*

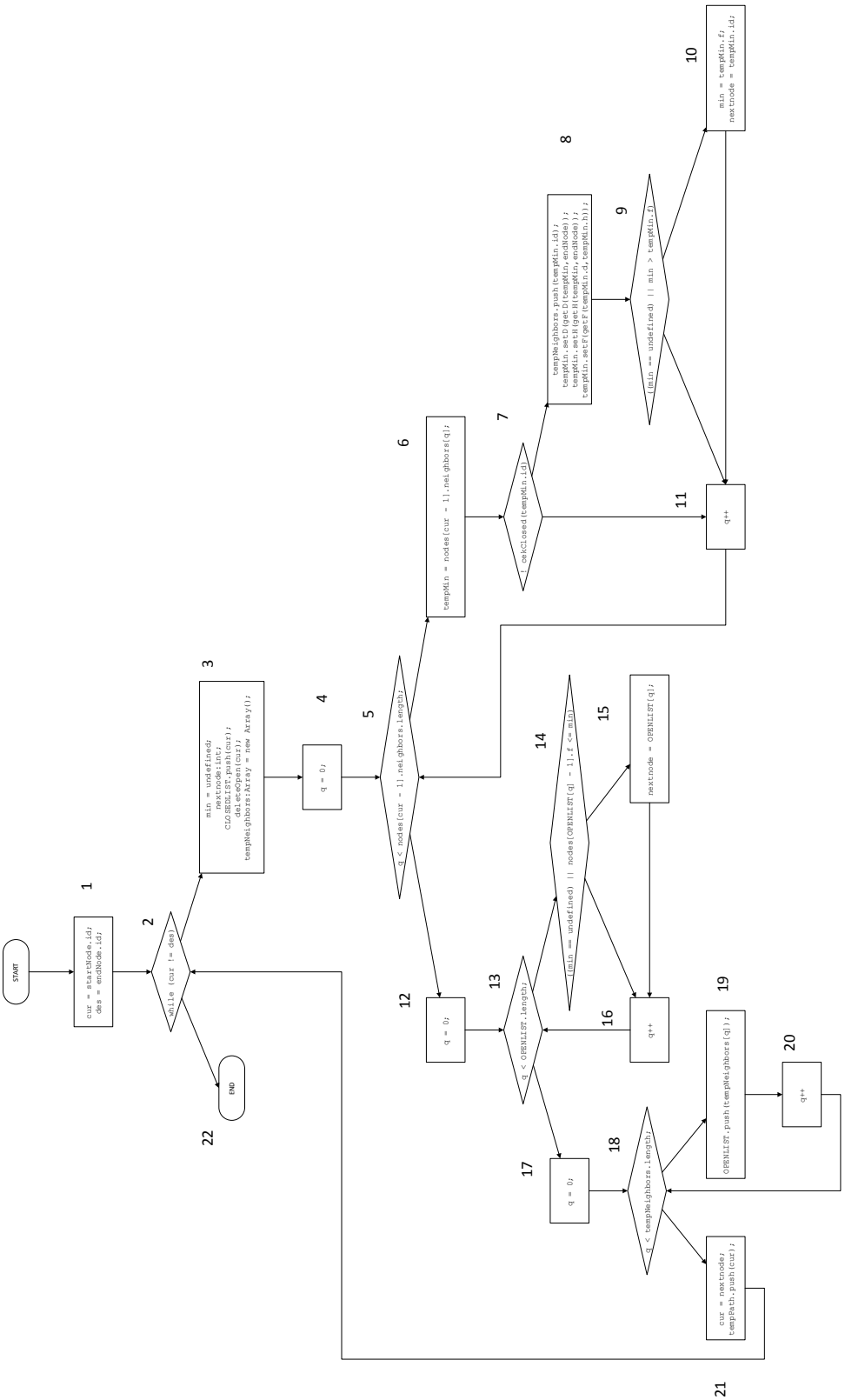
Pengujian *white box* merupakan pengujian yang didasarkan pada pengecekan terhadap detail perancangan, menggunakan struktur *control* dari desain aplikasi secara procedural untuk membagi pengujian ke dalam beberapa kasus pengujian untuk mengetahui apakah setiap *node* pada *independent path* dikerjakan. Serta untuk menguji alur logika, perulangan, dan validasi parameter masukannya. Berikut ini merupakan algoritma A* untuk *level* 2.

```
public function findPath(startNode:Node,endNode:Node):void{
    var cur:int;
    var des:int;
[1] cur = startNode.id;
[1] des = endNode.id;

[2] while (cur != des) {
[3]     var min = undefined;
[3]     var nextnode:int;
[3]     CLOSEDLIST.push(cur);
[3]     deleteOpen(cur);
[3]     var tempNeighbors:Array = new Array();
[4,5,11]for (var q = 0; q < nodes[cur - 1].neighbors.length; q++)
        {
[6]         var tempMin = nodes[cur - 1].neighbors[q];
[7]         if (! cekClosed(tempMin.id))
            {
[8]             tempNeighbors.push(tempMin.id);
[8]             tempMin.setD(getD(tempMin,endNode));
[8]             tempMin.setH(getH(tempMin,endNode));
[8]             tempMin.setF(getF(tempMin.d,tempMin.h));
[9]             if (((min == undefined) || min > tempMin.f))
                {
[10]                min = tempMin.f;
[10]                nextnode = tempMin.id;
                }
            }
        }
[12,13,16]for (q = 0; q < OPENLIST.length; q++)
        {
[14]        if (((min == undefined) || nodes[OPENLIST[q] - 1].f <= min))
            {
```

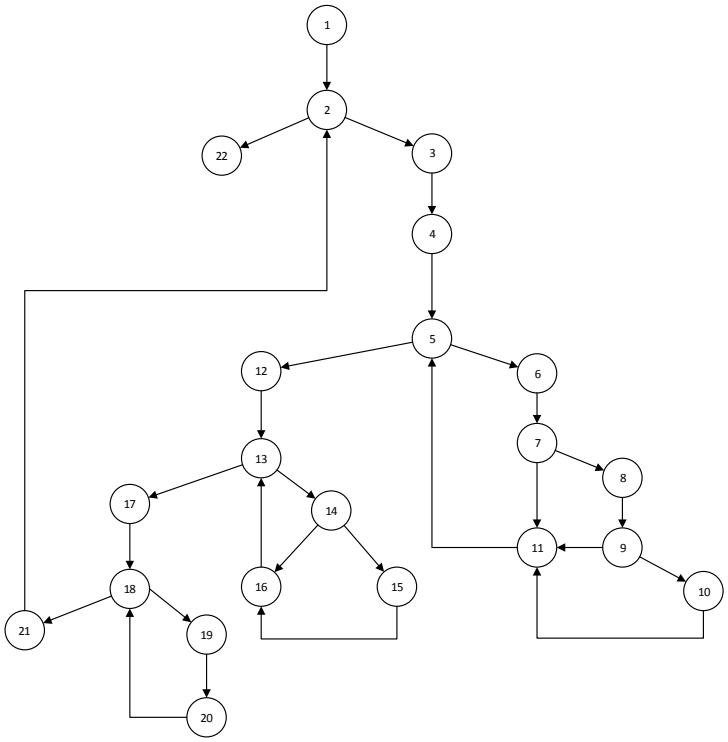
```
[15]              nextnode = OPENLIST[q];
             }
          }
[17,18,20]for (q = 0; q < tempNeighbors.length; q++)
          {
[19]         OPENLIST.push(tempNeighbors[q]);
          }
[21]      cur = nextnode;
[21]      tempPath.push(cur);
[22]}
```

Berdasarkan *pseudocode* tersebut maka *flow chart* algoritma A* adalah sebagai berikut :



Gambar 4.31 *flow chart* algoritma A*

Berdasarkan *flow chart* di atas maka *flow graph* algoritma A* adalah sebagai berikut :



Gambar 4.32 *Flowgraph* Algoritma A*

Dari gambar 4.31 dapat dihitung *cyclomatic complexity* atau V(G) sebagai berikut:

V(G) = E – N + 2

Keterangan :

 = menggambarkan kondisi (N)

 = menggambarkan aksi (E)

E = 28, N= 22

V(G) = 28 – 22 + 2

V(G) = 8

Jadi, *cyclomatic complexity* untuk gambar 4.31 adalah 8. Berdasarkan *cyclomatic complexity* tersebut, maka terdapat 8 *path* yang terdiri dari :

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

Path 2 : 1-2-3-4-5-6-7-8-9-11-5-12-13-14-15-16-13-17-18-19-20-18-21-2-22

Path 3 : 1-2-3-4-5-6-7-11-5-12-13-14-15-16-13-17-18-19-20-18-21-2-22

Path 4 : 1-2-3-4-5-12-13-14-15-16-13-17-18-19-20-18-21-2-22

Path 5 : 1-2-3-4-5-12-13-14-16-13-17-18-19-20-18-21-2-22

Path 6 : 1-2-3-4-5-12-13-17-18-19-20-18-21-2-22

Path 7 : 1-2-3-4-5-12-13-17-18-21-2-22

Path 8 : 1-2-22

Setelah *flowgraph* diberi nomor dan *cyclomatic complexity* telah dihitung selanjutnya dibuat tabel *graph matrics* seperti pada tabel berikut :

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	
1		1																					0
2			1																			1	1
3				1																			0
4					1																		0
5						1						1											1
6							1																0
7								1			1												1
8									1														0
9										1	1												1
10											1												0
11					1																		0
12													1										0
13														1			1						1
14															1	1							1
15																1							0
16													1										0
17																		1					0
18																			1		1		1
19																				1			0
20																		1					0
21		1																					0
22																							0
Jumlah																							7

V(G) = Jumlah *graph matrices* + 1

V(G) = 8

Setelah dilakukan pengujian hasil yang didapat dapat dilihat pada tabel berikut :

Tabel 4.4 Hasil Uji White Box Penerapan Algoritma A*

Path No	Node(n)	Hasil yang Diharapkan	Hasil Sesuai Uji Kasus	Keterangan
1	1-2-3-4-5-6-7-8-9-10-11-5-12-13-14-15-16-13-17-18-19-20-18-	Menghasilkan *rute* terpendek dari A*	Menghasilkan *rute* terpendek dari A*	[√] alur terlewati [ ] alur tidak terlewati

	21-2-22			
2	1-2-3-4-5-6-7-8-9-11-5-12-13-14-15-16-13-17-18-19-20-18-21-2-22	Memeriksa *node* tetangga dari *node* yang diperiksa	Memeriksa *node* tetangga dari *node* yang diperiksa	[√] alur terlewati [ ] alur tidak terlewati
3	1-2-3-4-5-6-7-11-5-12-13-14-15-16-13-17-18-19-20-18-21-2-22	Memeriksa apakah *node* tetangga dari *node* yang diperiksa masuk pada *closed*	Memeriksa apakah *node* tetangga dari *node* yang diperiksa masuk pada *closed*	[√] alur terlewati [ ] alur tidak terlewati
4	1-2-3-4-5-12-13-14-15-16-13-17-18-19-20-18-21-2-22	Memeriksa F(n) terkecil pada setiap *node* tetangga dari *node* yang diperiksa, sehingga menghasilkan dugaan sementara terhadap *node* yang terpilih dari *node* tetangga untuk dijadikan *node* yang selanjutnya akan menjadi *parent*	Memeriksa F(n) terkecil pada setiap *node* tetangga dari *node* yang diperiksa, sehingga menghasilkan dugaan sementara terhadap *node* yang terpilih dari *node* tetangga untuk dijadikan *node* yang selanjutnya akan menjadi *parent*	[√] alur terlewati [ ] alur tidak terlewati
5	1-2-3-4-5-12-13-14-16-13-17-18-19-20-18-21-2-22	Memeriksa *openlist*	Memeriksa *openlist*	[√] alur terlewati [ ] alur tidak terlewati
6	1-2-3-4-5-12-13-17-18-19-20-18-21-2-22	Memeriksa atau membandingan jarak terkecil dari *node* tetangga yang telah diperiksa dan menjadi nilai terkecil dengan *node* yang perada pada *openlist*	Memeriksa atau membandingan jarak terkecil dari *node* tetangga yang telah diperiksa dan menjadi nilai terkecil dengan *node* yang perada pada *openlist*	[√] alur terlewati [ ] alur tidak terlewati
7	1-2-3-4-5-12-13-17-18-21-2-22	Memasukan *node* yang tidak terpakai kedalam *openlist*	Memasukan *node* yang tidak terpakai kedalam *openlist*	[√] alur terlewati [ ] alur tidak terlewati
8	1-2-22	*Node* yang diperiksa sama dengan *node* tujuan	*Node* yang diperiksa sama dengan *node* tujuan	[√] alur terlewati [ ] alur tidak terlewati

Berdasarkan tabel diatas maka dapat disimpulkan hasil uji kasus setiap *node* pada *independent path A** berjalan sesuai hasil yang diharapkan yaitu semua alur sudah terlewati 1 kali.

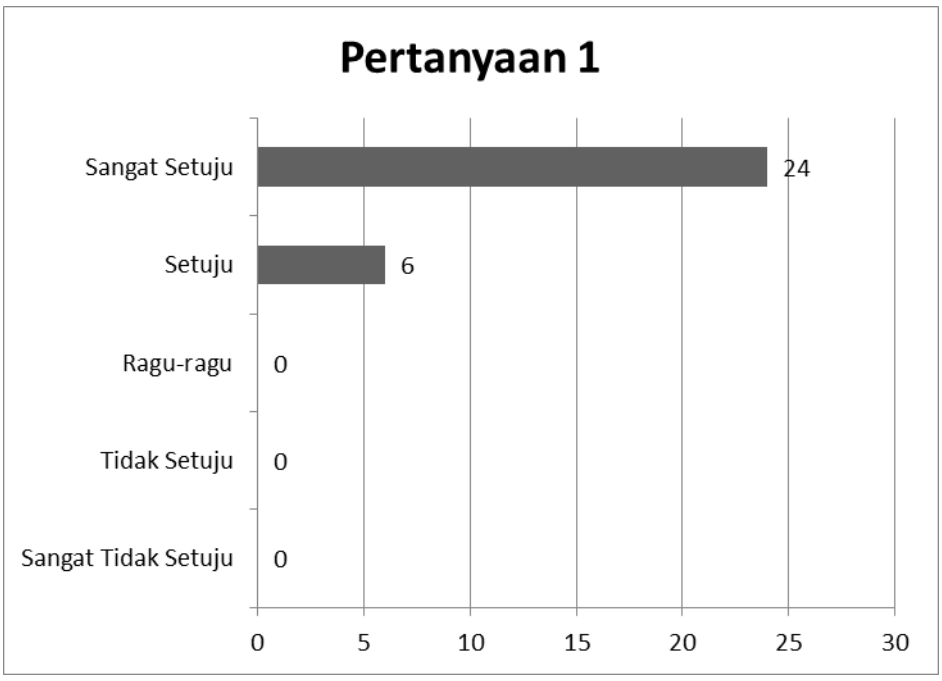
4.2.3 Pengujian *Beta*

Pengujian *beta* merupakan pengujian yang dilakukan secara *objectif*. Pengujian dilakukan terhadap 30 *responden* yang merupakan siswa berumur 9-10 tahun. *Responden* disesuaikan dengan analisis pengguna dimana kriteria secara umumnya adalah bisa menggunakan komputer, laki-laki atau perempuan, dapat membaca, dan tidak ada masalah penglihatan.

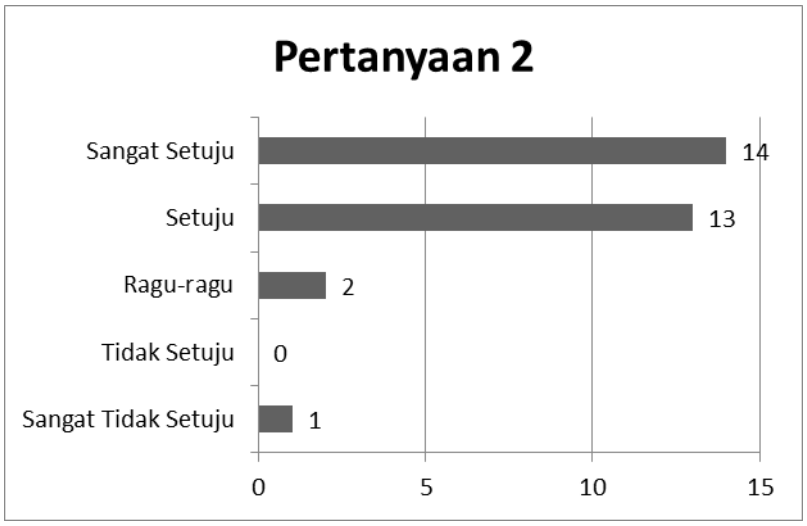
4.2.3.1 Kuesioner

Kuesioner dilakukan untuk mengetahui ketertarikan siswa terhadap *game* yang dibangun, meperkenalkan danpak global warming, bagian pohon beserta fungsinya. Pembuatan *Kuesioner* menggunakan *metode* skala *likert*, dimana masing-masing pertanyaan mempunyai pilihan sangat setuju, setuju, ragu-ragu, tidak setuju, dan sangat tidak setuju. Masing-masing pilihan tersebut diberi poin (5 sampai 1). Berikut ini adalah daftar pertanyaan *Kuesioner* yang diberikan, serta hasil yang didapat setelah nilainya dihitung.

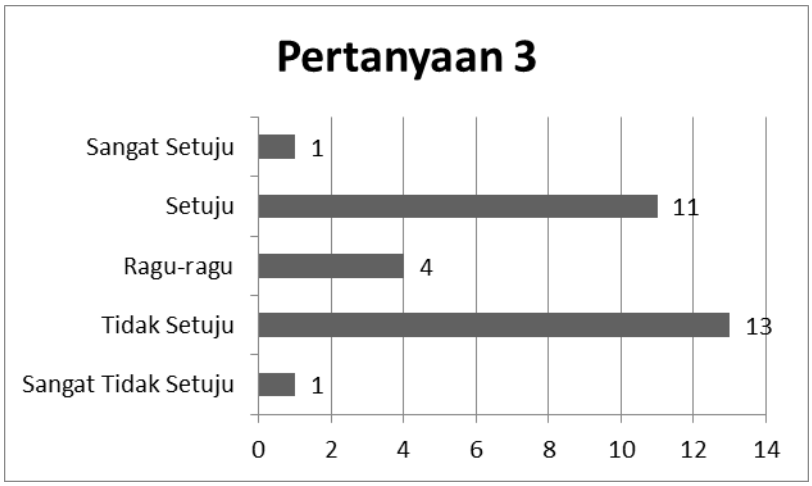
1. Menurut adik-adik, apakah tampilan *game* ini menarik ?



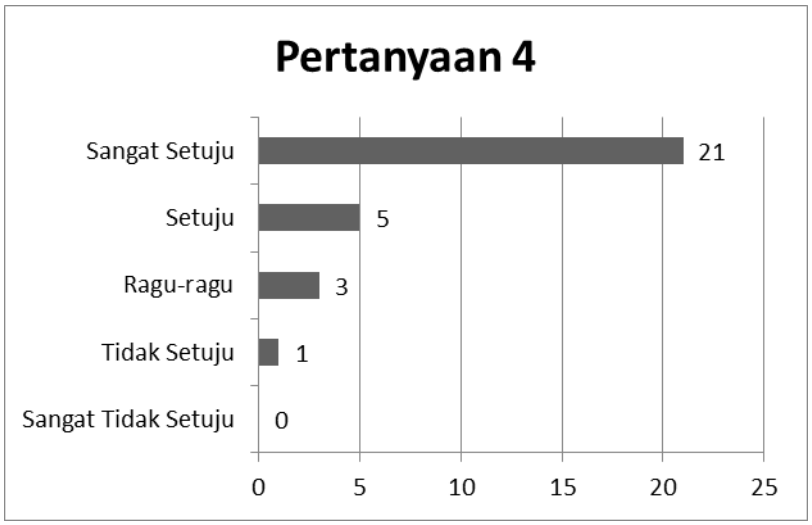
2. Menurut adik-adik, apakah *game* ini mudah digunakan ?



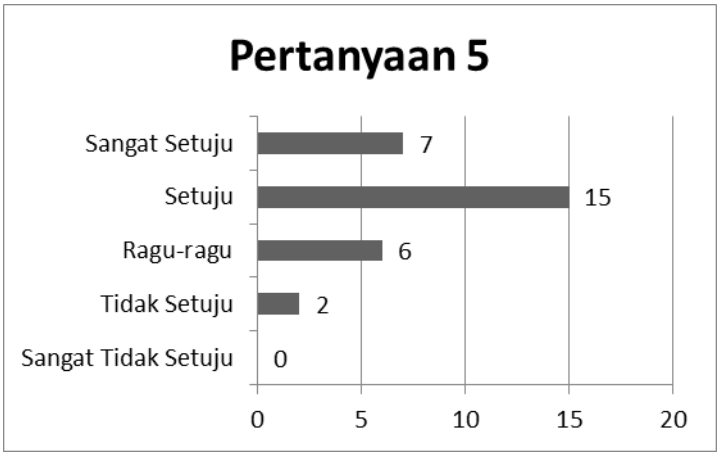
3. Menurut adik-adik, apakah musuh pada game ini sulit dikalahkan ?



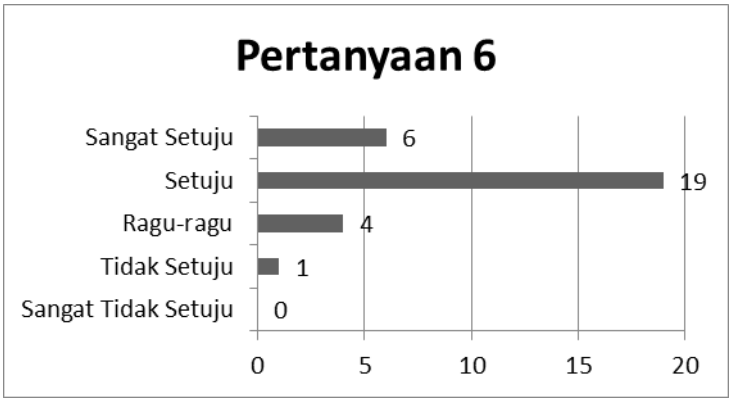
4. Apakah adik-adik tertarik untuk memainkan *game* ini lagi ?



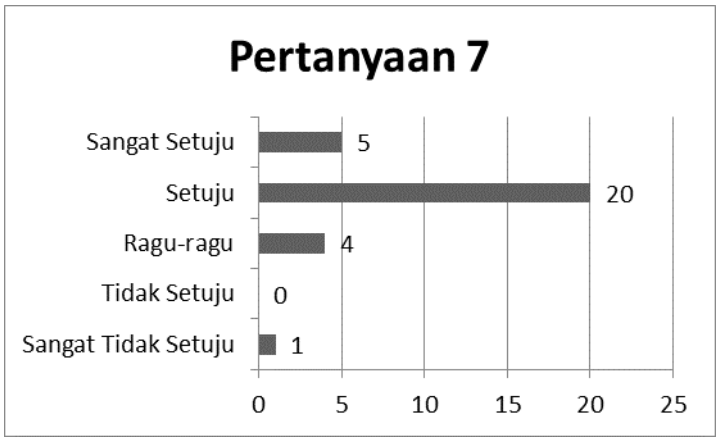
5. Apakah adik-adik mengenal bagian dan fungsi pada akar setelah memainkan *game* ini ?



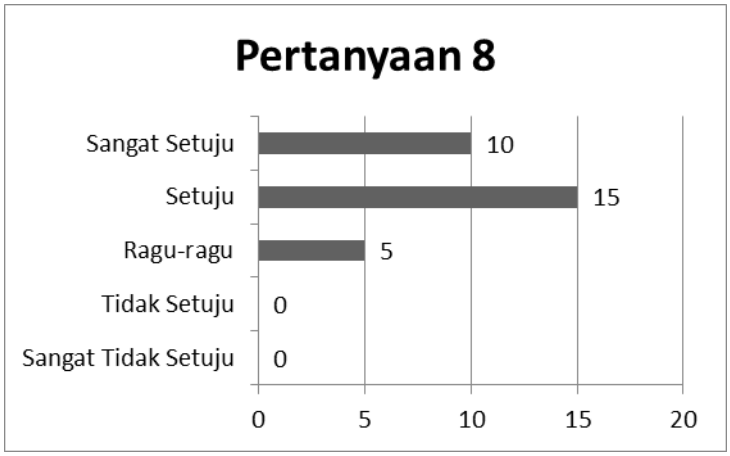
6. Apakah adik-adik mengenal bagian dan fungsi batang setelah memainkan *game* ini ?



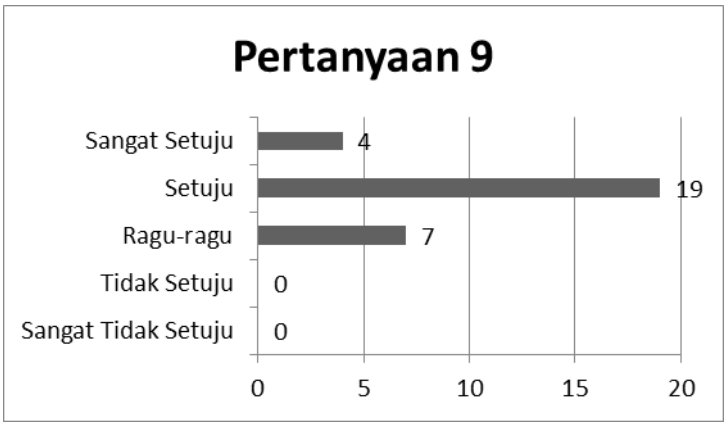
7. Apakah adik-adik mengenal bagian dan fungsi daun setelah memainkan *game* ini ?



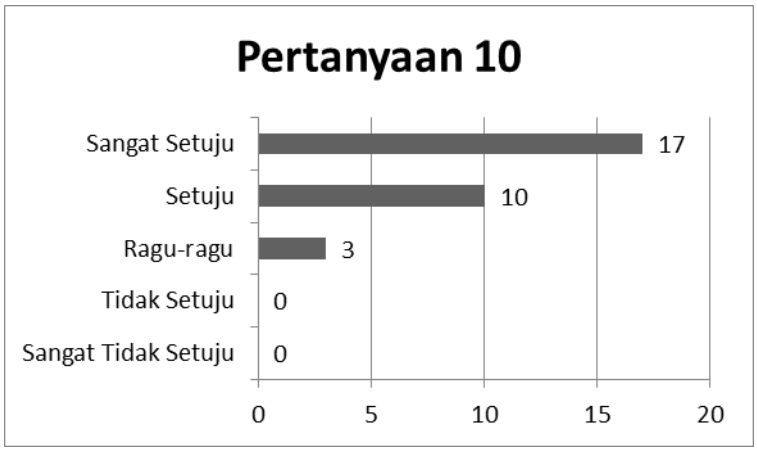
8. Apakah adik-adik mengenal fungsi fotosintesis setelah memainkan *game* ini ?



9. Apakah adik-adik mengenal dampak *global warming* setelah memainkan *game* ini ?



10. Apakah adik-adik mengenal gerakan *go green* menanam pohon penting untuk kelangsungan makhluk hidup setelah memainkan *game* ini ?



Dari hasil tersebut kemudian dikalikan sesuai poin yang diberikan pada masing-masing pilihan, yaitu :

1. Sangat Setuju : 5 poin
2. Setuju : 4 poin
3. Ragu-ragu : 3 poin
4. Kurang Setuju : 2 poin
5. Tidak Setuju : 1 poin

Setelah dikalikan, poin tersebut dijumlahkan dan kemudian dihitung persentasenya dengan membagi jumlah tadi oleh total kemungkinan nilai terbesar. Total kemungkinan nilai terbesar adalah kemungkinan poin tertinggi dikali jumlah responden.

Tabel 4.5 Hasil Perhitungan *Kuesioner*

	SS	S	RG	TS	STS	Jumlah	*Persentase*	Kesimpulan
Q1	120	24	0	0	0	144	96.00%	Sangat kuat
Q2	70	52	6	0	1	129	86.00%	Sangat kuat
Q3	5	44	12	26	1	88	58.67%	Cukup
Q4	105	20	9	2	0	136	90.67%	Sangat kuat
Q5	35	60	18	4	0	117	78.00%	Kuat
Q6	30	76	12	2	0	120	80.00%	Kuat
Q7	25	80	12	0	1	188	78.67%	Kuat
Q8	50	60	15	0	0	125	83.33%	Sangat kuat
Q9	20	76	21	0	0	117	78.00%	Kuat
Q10	85	40	9	0	0	134	89.33%	Sangat kuat

Kriteria kesimpulan hasil tersebut adalah sebagai berikut :

1. 0% - 20% : Sangat Lemah
2. 21% - 40% : Lemah
3. 41% - 60% : Cukup
4. 61% - 80% : Kuat
5. 81% - 100% : Sangat Kuat

Berdasarkan hasil yang didapat, maka kesimpulannya adalah sebagai berikut :

1. Tampilan *game* sangat menarik
2. *Game* sangat mudah digunakan
3. Musuh pada *game* ini cukup sulit dikalahkan
4. Responden sangat tertarik untuk memainkan *game* ini lagi

5. Responden mengenal bagian dan fungsi pada akar setelah memainkan *game* ini
6. Responden mengenal bagian dan fungsi pada batang setelah memainkan *game* ini
7. Responden mengenal bagian dan fungsi daun setelah memainkan *game* ini
8. Responden sangat mengenal fungsi fotosintesis setelah memainkan *game* ini
9. Responden mengenal dampak global warming setelah memainkan *game* ini
10. Responden sangat mengenal gerakan *go green* menanam pohon penting untuk kelangsungan makhluk hidup setelah memainkan *game* ini

4.2.3.2 Kesimpulan Pengujian *Beta*

Berdasarkan hasil pengujian *beta kuesioner* yang dibagikan pada 30 anak menyatakan 96% tampilan *game* menarik, 86% *game* mudah digunakan, 58,67% musuh pada *game* ini sulit dikalahkan, 90,67% responden tertarik untuk memainkan *game* ini lagi, 78% mengenal bagian dan fungsi pada akar setelah memainkan *game* ini, 80% mengenal bagian dan fungsi pada batang setelah memainkan *game* ini, 78,67% mengenal bagian dan fungsi daun setelah memainkan *game* ini, 83,33% mengenal fungsi fotosintesis setelah memainkan *game* ini, 78% mengenal dampak global warming setelah memainkan *game* ini dan 89,33% mengenal gerakan *go green* menanam pohon penting untuk kelangsungan makhluk hidup setelah memainkan *game* ini oleh karena itu dapat disimpulkan bahwa tampilan *game* sangat menarik, *game* sangat mudah digunakan, musuh pada *game* ini cukup sulit dikalahkan, responden tertarik untuk memainkannya lagi, dapat membuat responden mengenal bagian dan fungsi daun, mengenal fungsi fotosintesis, mengenal dampak *global warming*, sangat mengenal gerakan *go green* menanam pohon adalah penting untuk kelangsungan makhluk hidup sehingga *game* ini dapat dijadikan sarana untuk memperkenalkan dampak *global warming*, bagian pohon beserta fungsinya.