BAB 4
IMPLEMENTASI DAN PENGUJIAN

4.1 Implementasi

Tahap implementasi merupakan tahap penciptaan perangkat lunak kedalam bahasa pemograman yang dimengerti oleh komputer, serta merupakan tahap kelanjutan dari kegiatan perancangan sistem.

4.1.1 Implementasi Perangkat Keras

Perangkat keras yang akan digunakan untuk menjalankan aplikasi *game* cerita rakyat fable adalah perangkat mobile yang sudah diuji coba terlebih dahulu. Adapun spesifikasi perangkat keras yang digunakan saat implementasi dan pengujian aplikasi *game* cerita rakyat fable dapat dilihat pada tabel 4.1.

Tabel 4.1 Spesifikasi Perangkat Keras

Spesifikasi Perangkat Keras	
Processor	CPU Dual – Core 1 GHz Cortex-A5GPU
RAM	768 Mb
Memory	4 Gb
VGA	GPU PowerVR SGX540
Display	Display 480 x 800 Pixels, 4.3 Inches (Multi Touch)

4.1.2 Implementasi Perangkat Lunak

Spesifikasi perangkat lunak yang dibutuhkan untuk menjalankan aplikasi *game* cerita rakyat fabel yaitu sistem operasi android 4.0 Ice Cream Sandwich dan Adobe Air untuk dapat menjalankan game ini di android.

4.1.3 Implementasi Aplikasi

Aplikasi *game* cerita rakyat fabel merupakan sebuah *game* ber-genre *adventure* yang terdiri dari beberapa *stage* yang dimana pemain harus menyelesaikan tantangan di setiap *stage*nya untuk dapat melanjutkan ke *stage* berikutnya.

4.1.4 Implementasi Antarmuka

Adapun implementasi antarmuka pada aplikasi *game* Cerita rakyat fabel adalah sebagai berikut :

4.1.4.1 Antarmuka Menu Utama

Antarmuka menu utama merupakan tampilan awal pada saat aplikasi dijalankan. Terdapat 2 menu untuk menu utama seperti pada gambar 4.1.



Gambar 4.1 Menu Utama

4.1.4.2 Antarmuka Menu Option

Antarmuka option adalah antramuka pada saat pemain menyentuh tombol option di menu utama. Dalam menu option terdapat 2 submenu, yaitu sub menu musik dan sub menu developer, sub menu musik berfungsi untuk mematikan dan menyalakan musik, sedangkan sub menu developer berfungsi untuk menampilkan pembuat game.



Gambar 4.2 Menu Option

4.1.4.3 Implementasi Antarmuka Play

Antarmuka *play* merupakan tampilan yang akan ditampilkan ketika pengguna mengklik tombol *play* pada tampilan menu utama, pengguna akan menuju ke intro dari permainan yang menceritakan certa awal dari kancil dan buaya sebelum permainan dimulai. berikut adalah tampilan intro permainan yang dapat dilihat pada gambar 4.3.



Gambar 4.3 Intro Pertama

4.1.4.4 Implementasi Petunjuk Permainan *Stage* 1

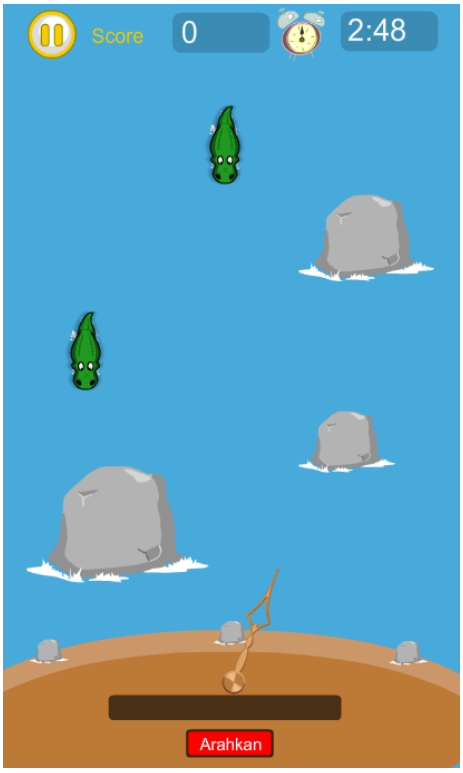
Setiap akan memulai *stage* akan ada informasi petunjuk bagaimana cara memainkan permainan di *stage* tersebut. Petunjuk permainan dalam *stage* 1 dapat ditunjukan dalam gambar 4.4 berikut.



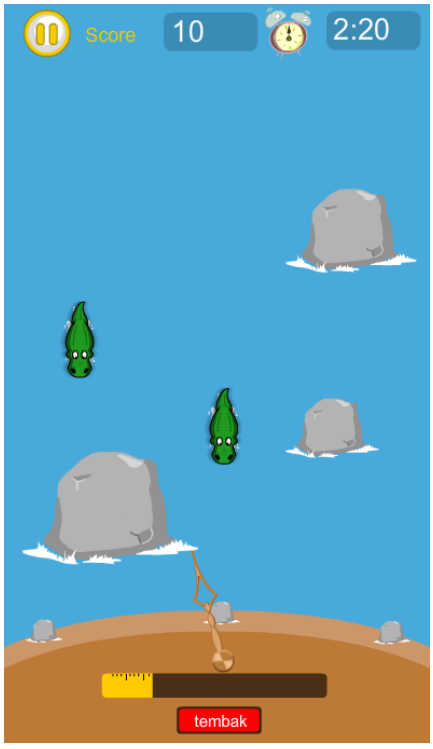
Gambar 4.4 Petunjuk Permainan *Stage* 1

4.1.4.5 Implementasi Permainan *Stage* 1

Stage 1 akan menceritakan kancil sedang melempari buaya dengan batu. Dalam pengaplikasiannya di *game*, akan ada busur yang menunjukan arah lemparan batu dan aka nada bar untuk menentukan jauh atau dekatnya lemparan batu. *Stage* 1 dapat dilihat pada gambar 4.5.



Gambar 4.5 Pengarahan Batu *Stage* 1



Gambar 4.6 Penentuan Jarak Lemparan *Stage* 1

4.1.4.6 Implementasi Petunjuk Permainan *Stage* 2

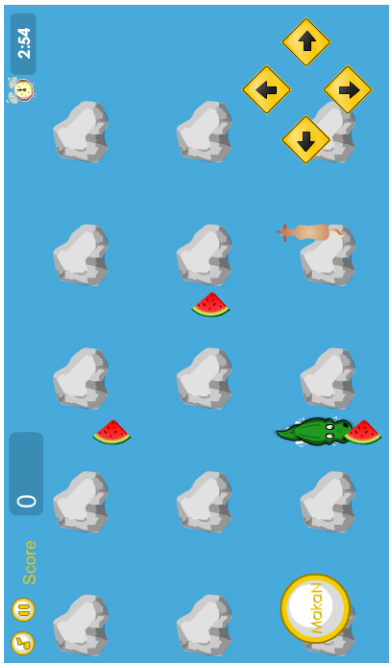
Petunjuk permainan dalam *stage* 2 dapat ditunjukan dalam gambar 4.7 berikut.



Gambar 4.7 Petunjuk Permainan *Stage* 2

4.1.4.7 Implementasi Permainan *Stage* 2

Stage 2 menceritakan bahwa kancil sedang berada di sungai di atas batu, pada saat tersebut buah-buahan jatuh ke sungai dari hutan sebrang sungai. Maka tugas kancil di *stage* 2 ini adalah memakan buah – buahan yang jatuh dan mengumpulkan point untuk menuju ke *stage* 3. *Stage* 2 dapat dilihat pada gambar 4.8 berikut.



Gambar 4.8 Permainan *Stage* 2

4.1.4.8 Implementasi Petunjuk Permainan *Stage* 3

Petunjuk permainan dalam *stage* 2 dapat ditunjukan dalam gambar 4.9 berikut.



Gambar 4.9 Petunjuk Permainan *Stage* 3

4.1.4.9 Implementasi Permainan *Stage* 3

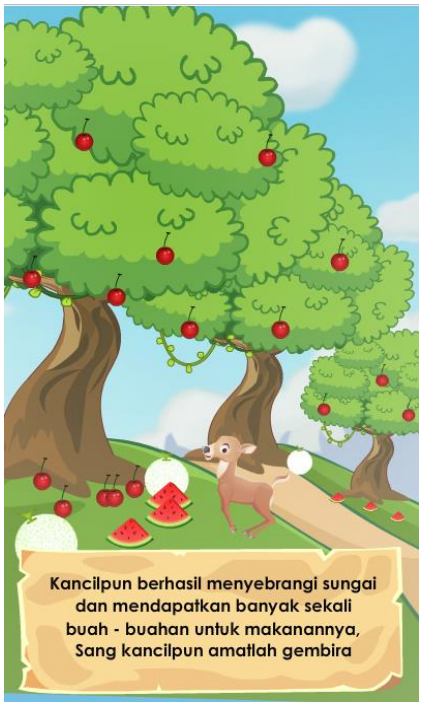
Dalam *stage* ini kancil harus melewati batu untuk sampai ke sebrang sungai. *Stage* 3 dapat dilihat pada gambar 4.10 berikut.



Gambar 4.10 Permainan *Stage* 3

4.1.4.10 Implementasi Memenangkan Permainan

Setelah kancil menyelesaikan semua tantangan dan dapat menyebrangi sungai, maka kancil sampai ke hutan sebrang sungai dan memakan banyak buah – buahan. Intro menang dapat dilihat pada gambar 4.11 berikut.



Gambar 4.11 Intro Memenangkan Permainan

4.2 Pengujian

Pengujian dilakukan betujuan untuk menemukan kesalahan kekurangan pada perangkat lunak yang diuji. Pengujian tersebut dimaksudkan untuk mengetahui perangkat lunak yang dibuat sudah memenuhi kriteria yang sesuai dengan tujuan perancangan perangkat lunak.

Pengujian dilakukan dengan dua macam teknik yaitu pengujian *white box* dan Pengujian *black box*. Adapun pengujian *black box* lebih berfokus pada menemukan kesalahan program, sedangkan pengujian *white box* berfokus pada struktur internal (*source code*) program, dimana setiap baris kode yang akan di-*compile* dicek satu persatu untuk mengetahui apakah masih terjadi *error*. Cara pengujian dilakukan melalui dua tahap, yaitu tahap pengujian *alpha* dan tahap pengujian *beta*

4.2.1 Pengujian Alpha

Pada tahap alpha, pengujian dilakukan untuk memastikan aplikasi *game* cerita rakyat fabel dapat berjalan sesuai dengan kebutuhan dan tujuan yang diharapkan. Pendekatan perancangan kasus uji menggunakan dua macam teknik yaitu *white box* dan *black box*.

4.2.1.1 Skenario Pengujian Aplikasi

Skenario pengujian menjelaskan pengujian terhadap sistem yang ada pada aplikasi *game* cerita rakyat fabel. Berikut ini merupakan skenario pengujian dari aplikasi *game* dapat dilihat pada tabel 4.2.

Tabel 4.2 Skenario Pengujian

No	Komponen yang diuji	Butir Uji	Jenis Pengujian
1	Menu Utama	Sentuh Menu Play	*Black Box*
		Sentuh Menu Option	*Black Box*
2	Menu Play	Menampilkan intro permainan	*Black Box*
3	Menu Option	Sentuh On pada musik	*Black Box*
		Sentuh Off pada musik	*Black Box*
4	Menu Petunjuk Permainan *Stage* 1	Sentuh Main	*Black Box*
5	Menu Petunjuk Permainan *Stage* 2	Sentuh Main	*Black Box*
6	Menu Petunjuk Permainan *Stage* 3	Sentuh Main	*Black Box*
7	Kontrol Busur	Sentuh Arahkan	
		Sentuh Tembak	
8	Menu Pause	Tampil Pause Menu	*Black Box*
		Sentuh On atau Off pada musik	*Black Box*
		Klik Tombol Main Menu	*Black Box*
		Klik Tombol Back	*Black Box*
9	Kontrol Karakter Kancil	Karakter Bergerak Kedepan	*Black Box*
		Karakter Bergerak Kekanan	*Black Box*
		Karakter Bergerak Kekiri	*Black Box*

		Karakter Bergerak Kebelakang	*Black Box*
		Karakter Memakan Buah	*Black Box*
10	Algoritma SMA*	Implementasi Algoritma SMA* Pada Musuh Untuk Melakukan Pencarian Terhadap Buah	*White Box*

4.2.1.2 Kasus Hasil Pengujian (*Black Box*)

Kasus dan hasil uji perangkat lunak yang sudah dibangun dengan menggunakan teknik *black box* berdasarkan pengamatan yang dilakukan dan menghasilkan kesimpulan dari pengujian tersebut adalah sebagai berikut:

1. **Pengujian Menu Utama**

Pengujian menu utama merupakan proses dimana pengguna menyentuh menu-menu yang ada pada tampilan menu utama. Adapun hasil dari proses pengujian dapat dilihat pada tabel 4.3.

Table 4.3 Pengujian Menu Utama

Kasus dan Hasil Uji			
Masukan	**Keluaran yang diharapkan**	**Pengamatan**	**Kesimpulan**
Klik tombol Play	Sistem melakukan proses untuk menampilkan *intro* permainan	Sistem menampilkan *introl* permainan	[√] Berhasil [ ] Tidak Berhasil
Klik tombol Option	Sistem melakukan proses untuk menampilkan halaman option	Sistem menampilkan halaman option	[√] Berhasil [ ] Tidak Berhasil

2. **Pengujian Menu Play**

Pengujian menu mulai merupakan proses setelah pengguna mengklik tombol *play* pada menu utama yang selanjutkan akan menampilkan *intro* permainan. Adapun hasil dari proses pengujian dapat dilihat pada tabel 4.4.

Table 4.4 Pengujian Menu Play

Kasus dan Hasil Uji			
Masukan	**Keluaran yang diharapkan**	**Pengamatan**	**Kesimpulan**
Klik Tombol Play	Sistem melakukan proses untuk menampilkan halaman *intro* permainan	Sistem menampilkan halaman *intro* permainan	[√] Berhasil [ ] Tidak Berhasil

3. Pengujian Menu Option

Pengujian menu option merupakan proses dimana pengguna dapat melihat dan menkofigurasi pengaturan yang tersedia pada *game*. Adapun hasil dari proses pengujian dapat dilihat pada tabel 4.5.

Table 4.5 Pengujian Menu Option

Kasus dan Hasil Uji			
Masukan	**Keluaran yang diharapkan**	**Pengamatan**	**Kesimpulan**
Klik tombol Option	Sistem melakukan proses untuk menampilkan menu option permainan	Sistem menampilkan menu option permainan	[√] Berhasil [ ] Tidak Berhasil
Klik tombol Off pada musik	Sistem melakukan proses untuk mematikan musik	Sistem mematikan musik	[√] Berhasil [ ] Tidak Berhasil
Klik tombol On pada musik	Sistem melakukan proses untuk menyalakan musik	Sistem menyalakan musik	[√] Berhasil [ ] Tidak Berhasil
Klik Tombol	Sistem melakukan	Sistem	[√] Berhasil

back	proses untuk kembali menampilkan menu utama	menampilkan halaman menu utama	[ ] Tidak Berhasil

4. Pengujian Menu Petunjuk *Stage* 1

Pengujian menu petunjuk *stage* 1 merupakan proses setelah sistem menampilkan *intro* permainan. Adapun hasil dari proses pengujian dapat dilihat pada tabel 4.6.

Table 4.6 Pengujian Petunjuk *Stage* 1

Kasus dan Hasil Uji			
Masukan	**Keluaran yang diharapkan**	**Pengamatan**	**Kesimpulan**
Klik Tombol Main	Sistem melakukan proses untuk menampilkan halaman *stage* 1 permainan	Sistem menampilkan halaman *stage* 1 permainan	[√] Berhasil [ ] Tidak Berhasil

5. Pengujian Menu Petunjuk *Stage* 2

Pengujian menu petunjuk *stage* 2 merupakan proses setelah sistem menampilkan keterangan menang di permainan *stage* 1. Adapun hasil dari proses pengujian dapat dilihat pada tabel 4.7.

Table 4.7 Pengujian Petunjuk *Stage* 2

Kasus dan Hasil Uji			
Masukan	**Keluaran yang diharapkan**	**Pengamatan**	**Kesimpulan**
Klik Tombol Main	Sistem melakukan proses untuk menampilkan halaman *stage* 2 permainan	Sistem menampilkan halaman *stage* 2 permainan	[√] Berhasil [ ] Tidak Berhasil

6. Pengujian Menu Petunjuk *Stage* 3

Pengujian menu petunjuk *stage* 3 merupakan proses setelah sistem menampilkan keterangan menang di permainan *stage* 2. Adapun hasil dari proses pengujian dapat dilihat pada tabel 4.8.

Table 4.8 Pengujian Petunjuk *Stage* 3

Kasus dan Hasil Uji			
Masukan	**Keluaran yang diharapkan**	**Pengamatan**	**Kesimpulan**
Klik Tombol Main	Sistem melakukan proses untuk menampilkan halaman *stage* 3 permainan	Sistem menampilkan halaman *stage* 3 permainan	[√] Berhasil [ ] Tidak Berhasil

7. Pengujian Menu *Pause*

Pengujian *menu pause* merupakan proses dimana pengguna dapat menghentikan sementara jalannya permainan sewaktu-waktu atau ingin keluar ditengah-tengah jalannya permainan. Adapun hasil dari proses pengujian dapat dilihat pada tabel 4.9.

Table 4.9 Pengujian Menu *Pause*

Kasus dan Hasil Uji			
Masukan	**Keluaran yang diharapkan**	**Pengamatan**	**Kesimpulan**
Sentuh Off pada musik	Sistem melakukan proses untuk mematikan musik	Sistem mematikan musik	[√] Berhasil [ ] Tidak Berhasil
Sentuh On pada musik	Sistem melakukan proses untuk menyalakan musik	Sistem menyalakan musik	[√] Berhasil [ ] Tidak Berhasil
Sentuh back	Sistem melakukan	Sistem	[√] Berhasil

	proses untuk kembali menampilkan permainan	menampilkan halaman permainan	[ ] Tidak Berhasil
Sentuh Main menu	Sistem melakukan proses untuk kembali ke menu utama	Sistem menampilkan menu utama	[√] Berhasil [ ] Tidak Berhasil

8. Pengujian Kotrol Busur

Pengujian kontrol busur merupakan proses dimana pemain dapat mengarahkan busur untuk melempar buaya dengan kekuatan yang telah ditentukan oleh pemain. Adapun hasil dari proses pengujian dapat dilihat pada tabel 4.10.

Table 4.10 Pengujian Kontrol Busur

Kasus dan Hasil Uji			
Masukan	**Keluaran yang diharapkan**	**Pengamatan**	**Kesimpulan**
Sentuh arahakan	Sistem melakukan proses untuk menghentikan pergerakan busur	Sistem menghentikan pergerakan busur	[√] Berhasil [ ] Tidak Berhasil
Sentuh tembak	Sistem melakukan proses untuk mengukur jauh atau dekatnya lemparan menggunakan bar sebagai indikator.	Sistem melempar batu sesuai dengan indikator lemparan.	[√] Berhasil [ ] Tidak Berhasil

9. Pengujian Kontrol Karakter Kancil

Pengujian kontrol karakter kancil merupakan proses yang dilakukan oleh pemain untuk menggerakkan karakter kancil pada permainan. Adapun hasil dari proses pengujian dapat dilihat pada tabel 4.11.

Tabel 4.11 Pengujian Kontrol Karakter Kancil

Kasus dan Hasil Uji			
Masukan	**Keluaran yang diharapkan**	**Pengamatan**	**Kesimpulan**
Sentuh panah atas	Karakter akan bergerak atas	Karakter kancil bergerak atas	[√] Berhasil [ ] Tidak Berhasil
Sentuh panah bawah	Karakter kancil akan bergerak bawah	Karakter kancil bergerak bawah	[√] Berhasil [ ] Tidak Berhasil
Sentuh panah kanan	Karakter kancil akan bergerak kekanan	Karakter kancil bergerak kanan	[√] Berhasil [ ] Tidak Berhasil
Sentuh panah kiri	Karakter kancil akan bergerak kekiri	Karakter kancil bergerak kekiri	[√] Berhasil [ ] Tidak Berhasil
Sentuh tombol makan	Karakter kancil akan mengambil buah yang berada didekatnya	Karakter kancil mengambil buah yang berada didekatnya	[√] Berhasil [ ] Tidak Berhasil

4.2.2 Kasus Hasil Pengujian (*White Box Texting*)

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

Pengujian Algoritma SMA* merupakan proses di mana akan diuji terhadap musuh dalam melakukan pencarian rute terpendek yang terdapat di dalam permainan cerita rakyat fabel.

1. Prosedru Pencarian Jalan

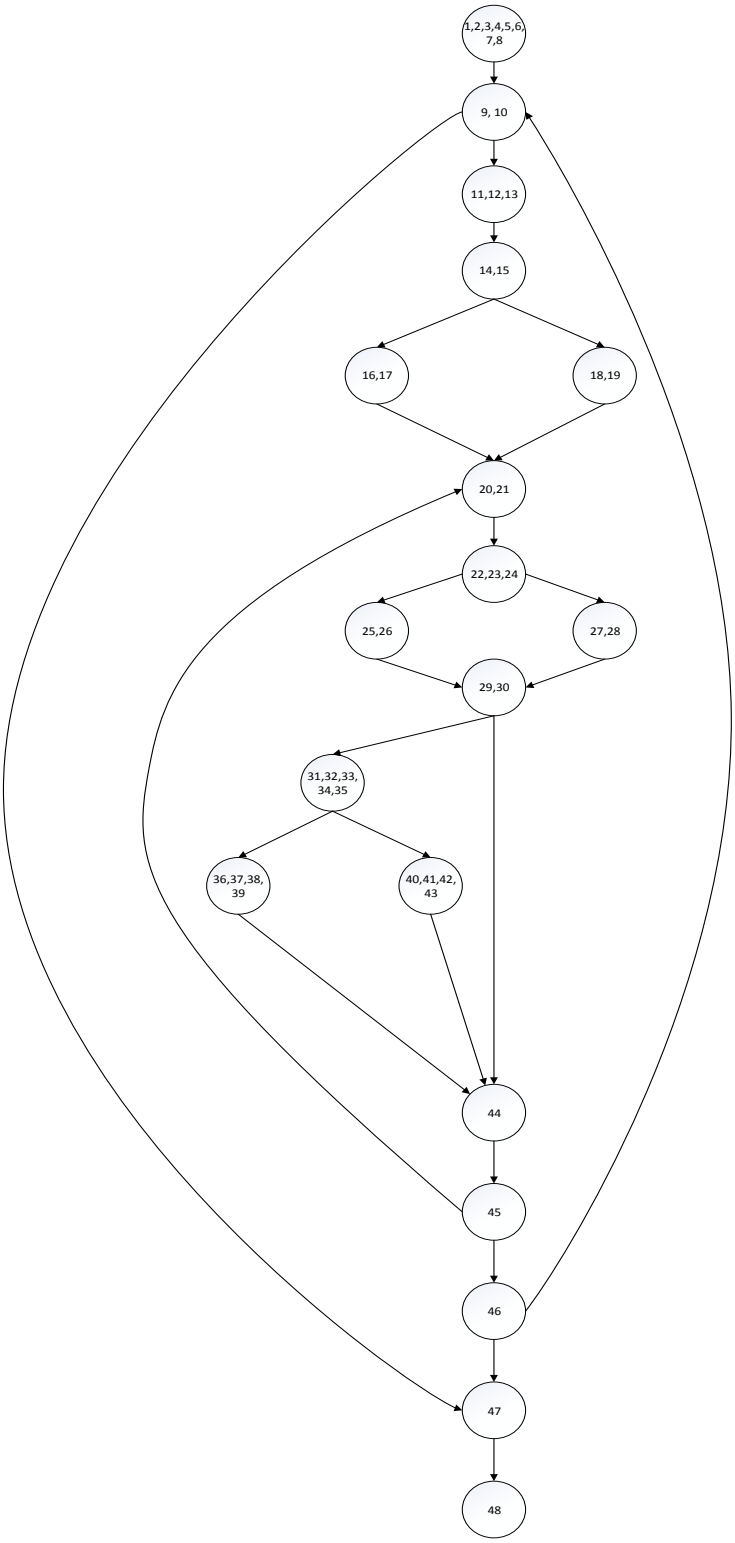
Berikut adalah *sourcecode* algortima yang digunakan dalam prosedur pencarian jalan di *game* cerita rakyat fabel yang diterapkan pada buaya dengan menggunakan bahasa *actionscript* 3.

```
function pathFinder()
{
var startNode:Nodes = varHolder[cur];
startNode.g = 0;
startNode.f = 0;
push(startNode);
startNode.opened = true;
var tahap = 0;
while (opened.length > 0)
{
var gh,gg;
var node = pop();
node.closes = true;
if (node.id == des)
{
return backtrace(des);
}
getTetangga(node.id);
var ttgs = node.tetangga;
for (var d = 0; d < ttgs.length; d++)
{
var ttg = ttgs[d];
if (ttg.closes)
{
continue;
}
var cost:Number = (ttg.xs - node.xs === 0 || ttg.ys - node.ys === 0 ? 1:Math.SQRT2);
var ng:Number = node.g + cost;
```

```
29  if (! ttg.opened || ng < ttg.g)
30  {
31  ttg.g = ng;
32  ttg.h = ttg.h || heuristic(Math.abs(ttg.xs - endN.xs),Math.abs(ttg.ys - endN.ys));
33  ttg.f = ttg.g + ttg.h;
34  ttg.par = node;
35  if (! ttg.opened)
36  {
37  push(ttg);
38  ttg.opened = true;
39  }
40  else
41  {
42  listUpdate(ttg);
43  }
44  }
45  }
46  }
47  return [];
48  }
```

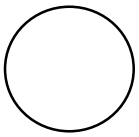
Gambar 4.12 ***Source Code Pathfind***

a. Flow Graph



Gambar 4.13 Flow Graph Prosedur Pencarian Jalan

Keterangan :

 = Menggambarkan Kondisi

 = Menggambarkan Aksi

b. *Cyclomatic Complexcity* V(G)

E = 24, N = 19

V(G) = E – N + 2

V(G) = 24 – 19 + 2

V(G) = 7

Keterangan =

E = Jumlah Aksi

N = Jumlah Kondisi

c. *Independent Path*

Path 1 = 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 9 – 47 – 48

Path 2 = 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 9 – 10 – 11 – 12 – 13 – 14 – 15 – 16 – 17 – 20 – 21 – 22 – 23 – 24 – 25 – 26 – 29 – 30 – 31 – 32 – 33 – 34 – 35 – 36 – 37 – 38 – 39 – 44 – 45

Path 3 = 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 9 – 10 – 11 – 12 – 13 – 14 – 15 – 18 – 19 – 20 – 21 – 22 – 23 – 24 – 25 – 26 – 29 – 30 – 31 – 32 – 33 – 34 – 35 – 36 – 37 – 38 – 39 – 44 – 45

Path 4 = 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 9 – 10 – 11 – 12 – 13 – 14 – 15 – 16 – 17 – 20 – 21 – 22 – 23 – 24 – 27 – 28 – 29 – 30 – 31 – 32 – 33 – 34 – 35 – 36 – 37 – 38 – 39 – 44 – 45

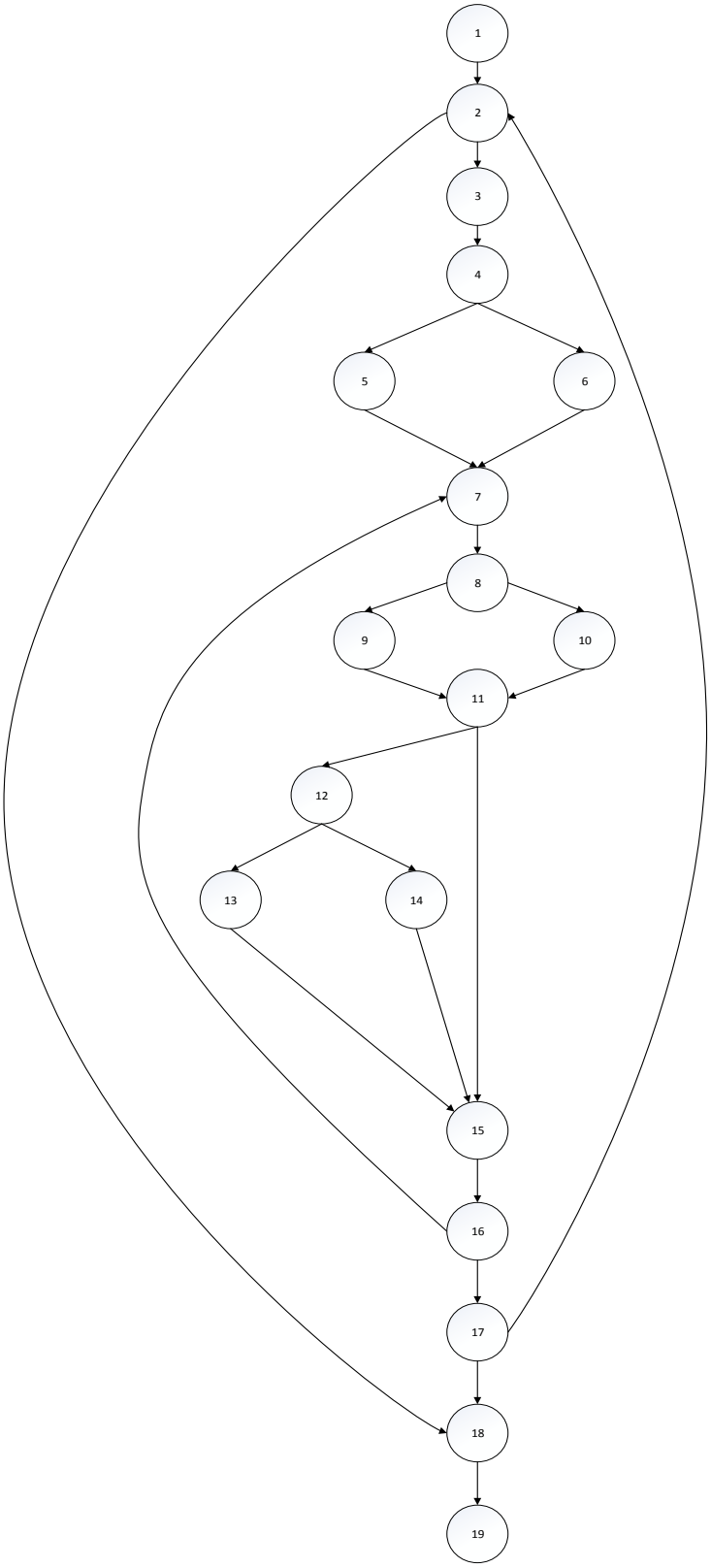
Path 5 = 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 9 – 10 – 11 – 12 – 13 – 14 – 15 – 18 – 19 – 20 – 21 – 22 – 23 – 24 – 27 – 28 – 29 – 30 – 31 – 32 – 33 – 34 – 35 – 36 – 37 – 38 – 39 – 44 – 45

Path 6 = 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 9 – 10 – 11 – 12 – 13 – 14 – 15 – 16 – 17 – 20 – 21 – 22 – 23 – 24 – 27 – 28 – 29 – 30 – 31 – 32 – 33 – 34 – 35 – 40 – 41 – 42 – 43 – 44 – 45

Path 7 = 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 9 – 10 – 11 – 12 – 13 – 14 – 15 – 18 – 19 – 20 – 21 – 22 – 23 – 24 – 27 – 28 – 29 – 30 – 31 – 32 – 33 – 34 – 35 – 40 – 41 – 42 – 43 – 44 – 45

d. *Graph Matriks* Algoritma

Penomoran ulang *flowgraph* atau penyederhanaan berdasarkan kondisi atau simpul sebagai berikut:



Gambar 4.14 Penyederhanaan Flow Graph Algoritma Pencarian Jalan

Tabel 4.12 *Graph Matriks* Algoritma

N	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	N(E) -1
1		1																		1-1=0
2			1															1		2-1=1
3				1																1-1=0
4					1	1														2-1=1
5							1													1-1=0
6							1													1-1=0
7								1												1-1=0
8									1	1										2-1=1
9											1									1-1=0
10											1									1-1=0
11												1								1-1=0
12													1	1						2-1=1
13															1					1-1=0
14															1					1-1=0
15																1				1-1=0
16							1										1			2-1=1
17		1																1		2-1=1
18																			1	1-1=0
19																				1-1=0
																			SUM(E)+1	6+1=7

Keterangan :

1. Baris dan kolom mempresentasikan simpul.
2. Nilai 1 mempresentasikan keterhubungan antara simpul.

e. Hasil Uji Algoritma SMA*

Tabel 4.13 Hasil Uji Algoritma SMA*

No	Kasus uji	Yang diharapkan	Pengamatan	Keterangan
1	- `var startNode:Nodes = varHolder[cur];` - `startNode.g = 0;` - `startNode.f = 0;` - `push(startNode);` - `startNode.opened = true;` - `var tahap = 0;` - `while (opened.length > 0)`	Node awal di mulai dan memasukkan node g dan f ke *open*. Dan dilakukan pengecekan nilai *openlist*.	Nilai node tidak lebih dari 0	[√] Alur terlewati [ ] Alur tidak terlewati

No	Kasus uji	Yang diharapkan	Pengamatan	Keterangan
	- return [];			
2	- while (opened.length > 0) - var gh,gg; - var node = pop(); - node.closes = true; - if (node.id == des) - return backtrace(des);	Ketika nilai di node *open* > 0 masukkan node ke *close*	Node masuk ke *close list*	[√] Alur terlewati [ ] Alur tidak terlewati
3	- while (opened.length > 0) - var gh,gg; - var node = pop(); - node.closes = true; - getTetangga(node.id); - var ttgs = node.tetangga; - for (var d = 0; d < ttgs.length; d++) - var ttg = ttgs[d]; - if (ttg.closes) - continue;	Prosedur pembangkitan suksesor dari *initial state*.	Pembangkitan suksesor berhasil dilakukan.	[√] Alur terlewati [ ] Alur tidak terlewati
4	- while (opened.length > 0) - var gh,gg; - var node = pop(); - node.closes = true; - getTetangga(node.id); - var ttgs = node.tetangga; - for (var d = 0; d < ttgs.length; d++) - var ttg = ttgs[d]; - var cost:Number = (ttg.xs - node.xs === 0 || ttg.ys - node.ys === 0 ? 1:Math.SQRT2);	Proses pembangkitan nilai dari setiap suksesor	Pembangkitan nilai berhasil dilakukan.	[√] Alur terlewati [ ] Alur tidak terlewati

No	Kasus uji	Yang diharapkan	Pengamatan	Keterangan
	- var ng:Number = node.g + cost;			
5	- if (! ttg.opened || ng < ttg.g) - ttg.g = ng; - ttg.h = ttg.h || heuristic(Math.abs(ttg.xs - endN.xs),Math.abs(ttg.ys - endN.ys)); - ttg.f = ttg.g + ttg.h; - ttg.par = node;	Proses perhitungan nilai G dan H dan percarian nilai F	Proses perhitungan berhasil dilakukan.	[√] Alur terlewati [ ] Alur tidak terlewati
6	- if (! ttg.opened || ng < ttg.g) - ttg.g = ng; - ttg.h = ttg.h || heuristic(Math.abs(ttg.xs - endN.xs),Math.abs(ttg.ys - endN.ys)); - ttg.f = ttg.g + ttg.h; - ttg.par = node; - if (! ttg.opened) - push(ttg); - ttg.opened = true;	Proses memasukkan suksesor ke *open list*	Pemasukan suksesor ke *open list* berhasil dilakukan	[√] Alur terlewati [ ] Alur tidak terlewati
7	- if (! ttg.opened || ng < ttg.g) - ttg.g = ng; - ttg.h = ttg.h || heuristic(Math.abs(ttg.xs - endN.xs),Math.abs(ttg.ys - endN.ys)); - ttg.f = ttg.g + ttg.h; - ttg.par = node; - else - listUpdate(ttg);	Proses update nilai dari perhitungan G dan H kepada F	Perhitungan berhasil dilakukan.	[√] Alur terlewati [ ] Alur tidak terlewati

f. Kesimpulan

Dari hasil pengujian algoritma SMA* yang dilakukan, maka didapat kesimpulan bahwa seluruh *node* yang ada pada setiap *independent path* telah dikerjakan, serta tidak terjadi pengulangan tak terhingga.

4.2.3 Pengujian Beta

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

4.2.3.1 Kuisioner

Kuesioner disebarkan kepada 30 orang responden berdasarkan tujuan pengguna yaitu sekitar 7 sampai 12 tahun, dari hasil kuesioner tersebut akan dilakukan perhitungan agar dapat diambil kesimpulan terhadap penilaian penerapan sistem yang telah dibangun. Kuesioner terdiri dari 8 pernyataan dengan menggunakan sistem penskoran dengan skala pengukuran yang digunakan adalah skala likert, Berikut skor untuk jawaban kuesioner.

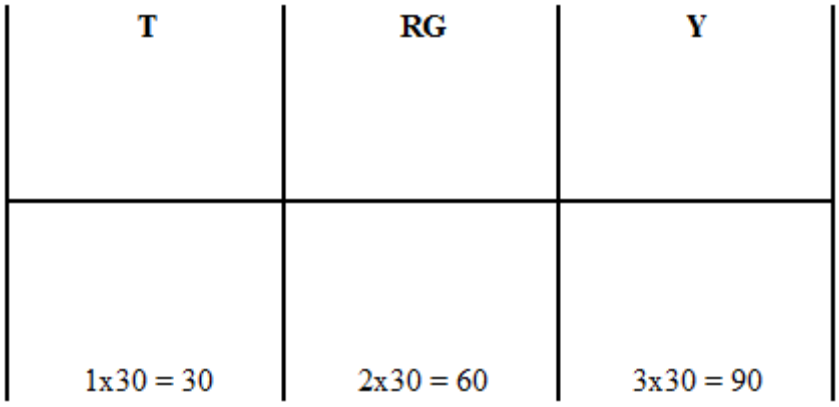
Tabel 4.14 Skor Jawaban Pada Kuisioner

Skala Jawaban	Keterangan	Skor
Y	Ya	3
RG	Ragu-ragu	2
T	Tidak	1

Data yang diperoleh dari pemberian kuesioner kepada responden dapat dianalisis dengan menghitung rata-rata jawaban berdasarkan *scoring* setiap jawaban dari responden. Analisis yang dapat dilakukan yaitu sebagai berikut.

a. Jumlah skor ideal (kriterium) untuk seluruh item = 3 × jumlah responden
b. Rata-rata skor dalam persentase = (jumlah skor : jumlah skor ideal) × 100%
c. Jumlah skor ideal pada penelitian ini yaitu : Jumlah skor ideal = 3×30 = 90

Secara kontinum dapat digambarkan seperti berikut:



Gambar 4.15 Penilaian Kuisioner

Adapun pertanyaan yang diajukan dalam kuesioner untuk pengguna aplikasi ini adalah sebagai berikut :

1. Apakah adik mengetahui cerita kancil dan buaya ?
2. Apakah dalam game ini diceritakan cerita kancil dan buaya ?
3. Apakah sama cerita kancil dan buaya di game dan cerita kancil dan buaya yang adik dengar / baca ?
4. Apakah adik menyukai game ini ?
5. Apakah game ini menyenangkan ?
6. Apakah adik kesulitan memainkan game ini ?
7. Apakah adik ingin memainkan lagi game ini ?
8. Apakah adik tertarik untuk membaca cerita kancil dan buaya atau cerita yang lainnya?

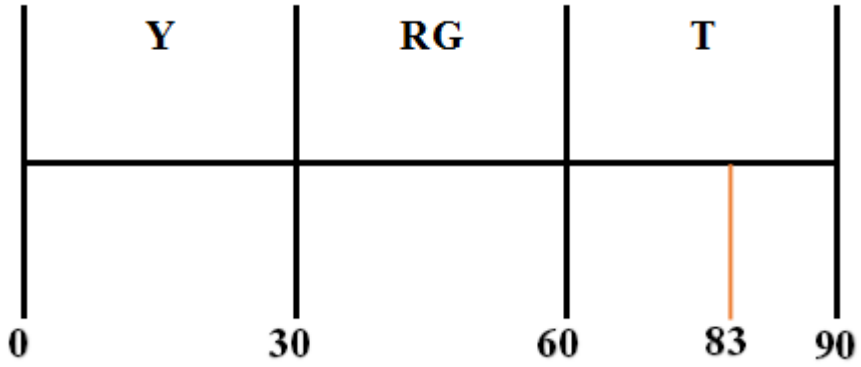
Berikut adalah hasil kuisioner masing-masing jawaban yang sudah dihitung dengan menggunakan persamaan skala kontinum. Kuisioner ini ditujukan kepada 30 responden.

1. Apakah adik mengetahui cerita kancil dan buaya ?

Tabel 4.15 Hasil Pengujian Kuisioner No.1

Pertanyaan	Jawaban	Skor	Responden	S
	Ya	3	24	72
	Ragu-ragu	2	5	10
	Tidak	1	1	1
Jumlah			**30**	**83**

Berdasarkan hasil skor nilai di atas, maka dapat disimpulkan bahwa penilaian terhadap *game* ini adalah 83 dari yang diharapkan 90, atau dapat disimpulkan bahwa responden mengetahui cerita kancil dan buaya.



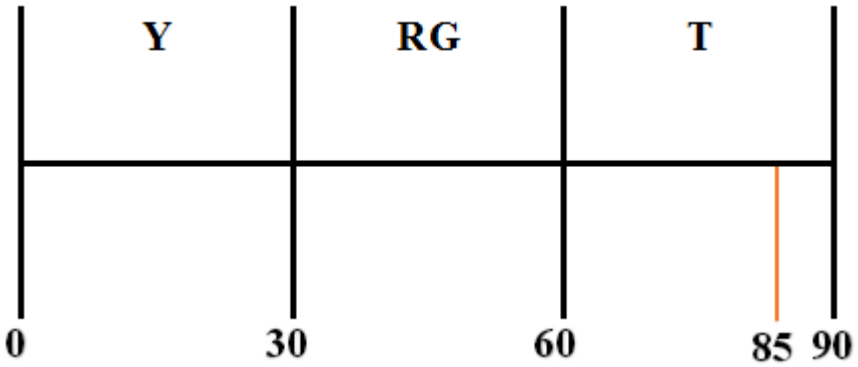
Gambar 4.16 *Interprestasi* Skor Pertanyaan No. 1

2. Apakah dalam game ini diceritakan cerita kancil dan buaya ?

Tabel 4.16 Hasil Pengujian Kuisioner No. 2

Pertanyaan	Jawaban	Skor	Responden	S
	Ya	3	26	78
	Ragu-ragu	2	3	6
	Tidak	1	1	1
Jumlah			**30**	**85**

Berdasarkan hasil skor nilai di atas, maka dapat disimpulkan bahwa penilaian terhadap *game* ini adalah 85 dari yang diharapkan 90, atau dapat disimpulkan bahwa didalam *game* diceritakan cerita kancil dan buaya.



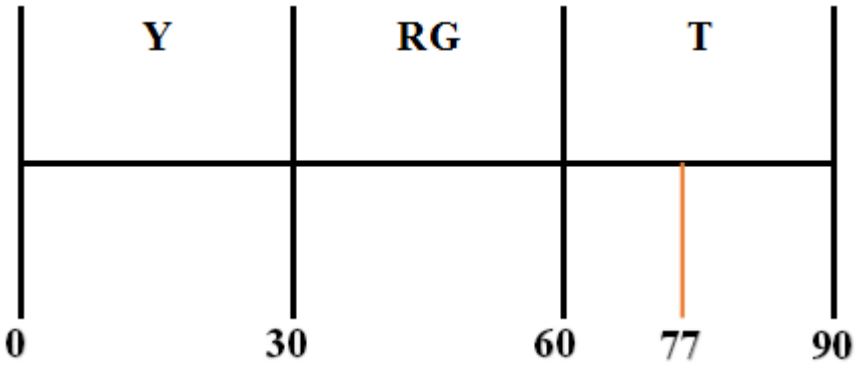
Gambar 4.17 *Interprestasi* Skor Pertanyaan No 2

3. Apakah sama cerita kancil dan buaya di game dan cerita kancil dan buaya yang adik dengar / baca ?

Tabel 4.17 Hasil Pengujian Kuisioner No. 3

Pertanyaan	Jawaban	Skor	Responden	S
	Ya	3	20	60
	Ragu-ragu	2	7	14
	Tidak	1	3	3
Jumlah			**30**	**77**

Berdasarkan hasil skor nilai di atas, maka dapat disimpulkan bahwa penilaian terhadap *game* ini adalah 77 dari yang diharapkan 90, atau dapat disimpulkan bahwa cerita dalam *game* sama dengan cerita sebenarnya.



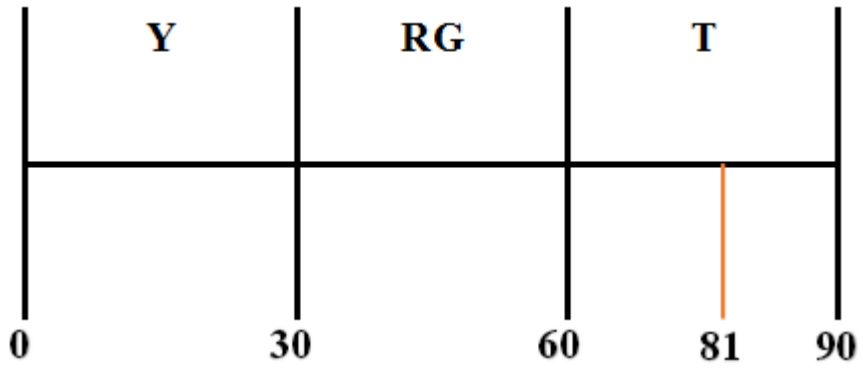
Gambar 4.18 *Interprestasi* Skor Pertanyaan No 3

4. Apakah adik menyukai game ini ?

Tabel 4.18 Hasil Pengujian Kuisioner No. 4

Pertanyaan	Jawaban	Skor	Responden	S
	Ya	3	23	69
	Ragu-ragu	2	5	10
	Tidak	1	2	2
Jumlah			**30**	**81**

Berdasarkan hasil skor nilai di atas, maka dapat disimpulkan bahwa penilaian terhadap *game* ini adalah 81 dari yang diharapkan 90, atau dapat disimpulkan bahwa responden menyukai *game* ini.



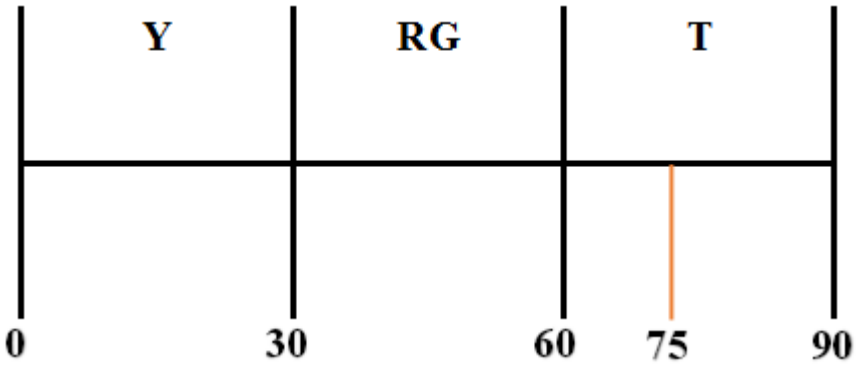
Gambar 4.19 *Interprestasi* Skor Pertanyaan No 4

5. Apakah game ini menyenangkan ?

Tabel 4.19 Hasil Pengujian Kuisioner No. 5

Pertanyaan	Jawaban	Skor	Responden	S
	Ya	3	19	72
	Ragu-ragu	2	7	14
	Tidak	1	4	4
Jumlah			**30**	**75**

Berdasarkan hasil skor nilai di atas, maka dapat disimpulkan bahwa penilaian terhadap *game* ini adalah 75 dari yang diharapkan 90, atau dapat disimpulkan bahwa *game* ini menyenangkan bagi responden.



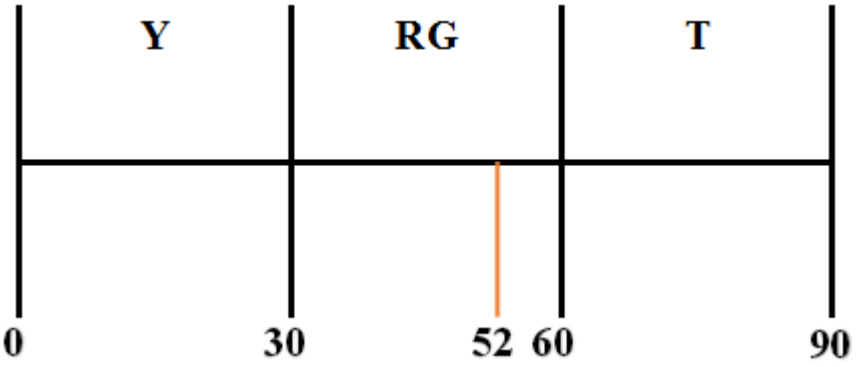
Gambar 4.20 *Interprestasi* Skor Pertanyaan No 5

6. Apakah adik kesulitan memainkan game ini ?

Tabel 4.20 Hasil Pengujian Kuisioner No. 6

Pertanyaan	Jawaban	Skor	Responden	S
	Ya	3	9	27
	Ragu-ragu	2	4	8
	Tidak	1	17	17
Jumlah			**30**	**52**

Berdasarkan hasil skor nilai di atas, maka dapat disimpulkan bahwa penilaian terhadap *game* ini adalah 52 dari yang diharapkan 90, atau dapat disimpulkan bahwa responden tidak kesulitan dalam memainkan *game* ini.



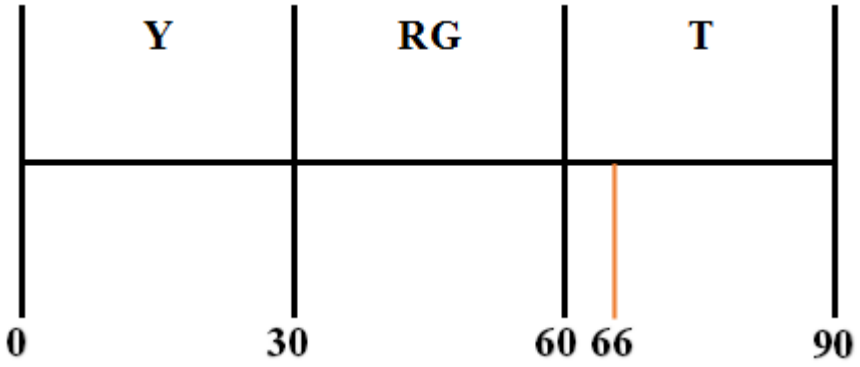
Gambar 4.21 *Interprestasi* Skor Pertanyaan No 6

7. Apakah adik ingin memainkan lagi game ini ?

Tabel 4.21 Hasil Pengujian Kuisioner No. 7

Pertanyaan	Jawaban	Skor	Responden	S
	Ya	3	13	39
	Ragu-ragu	2	10	20
	Tidak	1	7	7
Jumlah			**30**	**66**

Berdasarkan hasil skor nilai di atas, maka dapat disimpulkan bahwa penilaian terhadap *game* ini adalah 66 dari yang diharapkan 90, atau dapat disimpulkan bahwa responden ragu – ragu untuk memainkan lagi *game* ini.



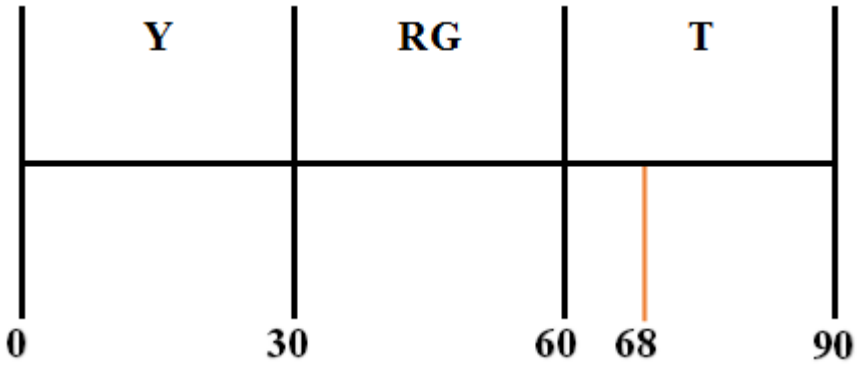
Gambar 4.22 *Interprestasi* Skor Pertanyaan No 7

8. Apakah adik tertarik untuk membaca cerita kancil dan buaya atau cerita yang lainnya?

Tabel 4.22 Hasil Pengujian Kuisioner No. 8

Pertanyaan	Jawaban	Skor	Responden	S
	Ya	3	16	48
	Ragu-ragu	2	6	12
	Tidak	1	8	8
Jumlah			**30**	**68**

Berdasarkan hasil skor nilai di atas, maka dapat disimpulkan bahwa penilaian terhadap *game* ini adalah 68 dari yang diharapkan 90, atau dapat disimpulkan bahwa responden tidak terlalu tertarik untuk membaca cerita kancil dan buaya atau cerita fabel lainnya.



Gambar 4.23 *Interprestasi* Skor Pertanyaan No 8