BAB 4
IMPLEMENTASI DAN PENGUJIAN

4.1 Implementasi

Setelah melakukan tahap analisis selanjutnya melakukan Implementasi dan Pengujian. Tahap implementasi merupakan tahap penciptaan perangkat lunak dan juga tahap kelanjutan dari kegiatan perancangan aplikasi, tahap ini merupakan tahap dimana aplikasi siap untuk dioperasikan. Aplikasi ini diimplementasikan dengan nama *Game* Budak Pahatu Lalis dengan *file extension*-nya *.exe.*

4.1.1 Implementasi Perangkat Keras

Perangkat keras yang digunakan untuk implementasi *game* budak pahatu lalis adalah :

1. Processor Intel Core i5 M 460 2.6 GHz
2. RAM 2 GB
3. VGA 1GB
4. Harddisk 500 GB
5. Monitor
6. Mouse, Keyboard, and Speaker

4.1.2 Implementasi Perangkat Lunak

Perangkat lunak yang digunakan untuk implementasi *game* budak pahatu lalis adalah :

1. Sistem Operasi Windows 7 Home Premium
2. Graphic Card
3. DirectX 9

4.1.3 Implementasi Aplikasi

Game budak pahatu lalis merupakan *game* edukasi yang diangkat dari cerita rakyat yang berasal dari Jawa Barat "Kisah Budak Pahatu Lalis" *game* ini dibuat untuk mengangkat kembali cerita rakyat berjudul kisah budak pahatu lalis agar cerita ini diketahui kembali oleh anak-anak

Game ini dibangun dengan *storyline* berdasarkan cerita yang terdapat pada kisah budak pahatu lalis dengan *genre game* yaitu *adventure*. Game ini terdiri dari 3 *stage* dimana *stage* 1 dengan tantangan mencari buah-buahan seperti yang terdapat pada cerita dimana kedua anak yang menjadi tokoh cerita sering mencari buah-buahan disekitar rumah mereka, *stage* 2 pemain yang memainkan tokoh sang kakak dalam cerita dituntut untuk melindungi adiknya didalam *game* scenario cerita ini diilustrasikan dengan datangnya penjahat yang mencegat mereka dalam perjalanan pulang menuju ke rumah, dan *stage* 3 pemain dituntut untuk mencari ular yang memakan sang adik kemudian melawannya lalu membebaskan kembali sang adik yang telah ditelan oleh ular tersebut.

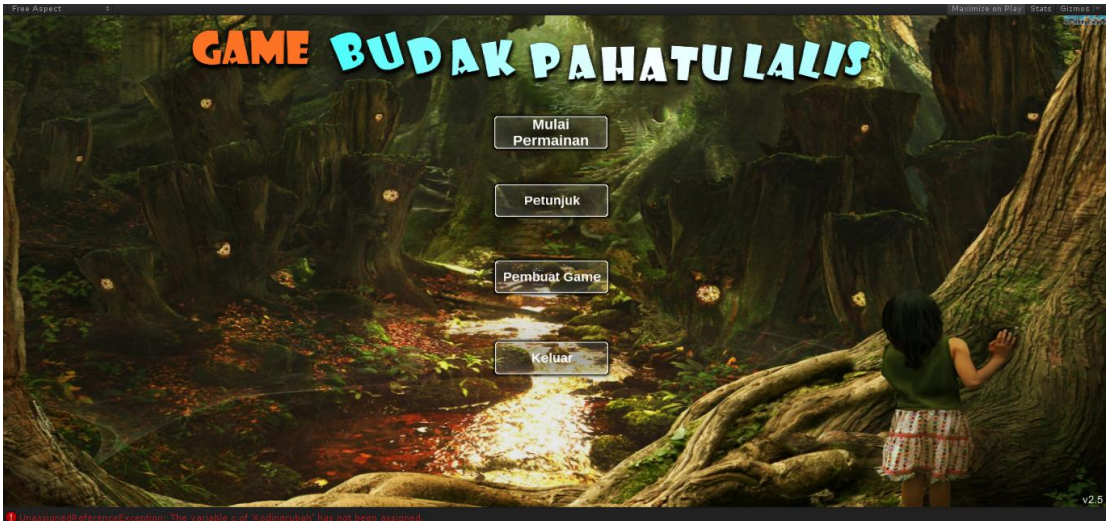
Unsur edukasi yang terdapat pada *game* ini adalah edukasi mengenai pesan moral dan sifat-sifat positif dalam kehidupan yang dimasukan pada masing-masing *stage* didalam *game*. Seperti *stage* 1 dimana sang kakak dituntut untuk mencari buah-buahan yaitu pesan moral agar sang kakak bertanggung jawab dalam memenuhi kebutuhan sang adik, *stage* 2 sang kakak melawan semua musuh yang datang untuk mengganggu mereka berdua yaitu pesan moral agar anak berani dan kuat dalam menghadapi semua cobaan yang datang bukan malah lari dan meninggalkan sang adik, dan *stage* 3 yang juga klimaks pada cerita dimana sang adik dimakan oleh ular disinilah pesan moral paling tinggi yaitu sifat untuk sepenuhnya bertanggung jawab terhadap adik tercintanya, sang kakak dituntut untuk mencari ular tersebut kemudian melawannya bukan hanya tinggal diam dan menyerah.

4.1.4 Implementasi Antarmuka

Implementasi antarmuka dibuat untuk melihat tampilan pada setiap *scene* dari program yang dibangun. Berikut ini merupakan implementasi antarmuka yang ada pada *game* budak pahatu lalis

1. **Antarmuka Menu Utama**

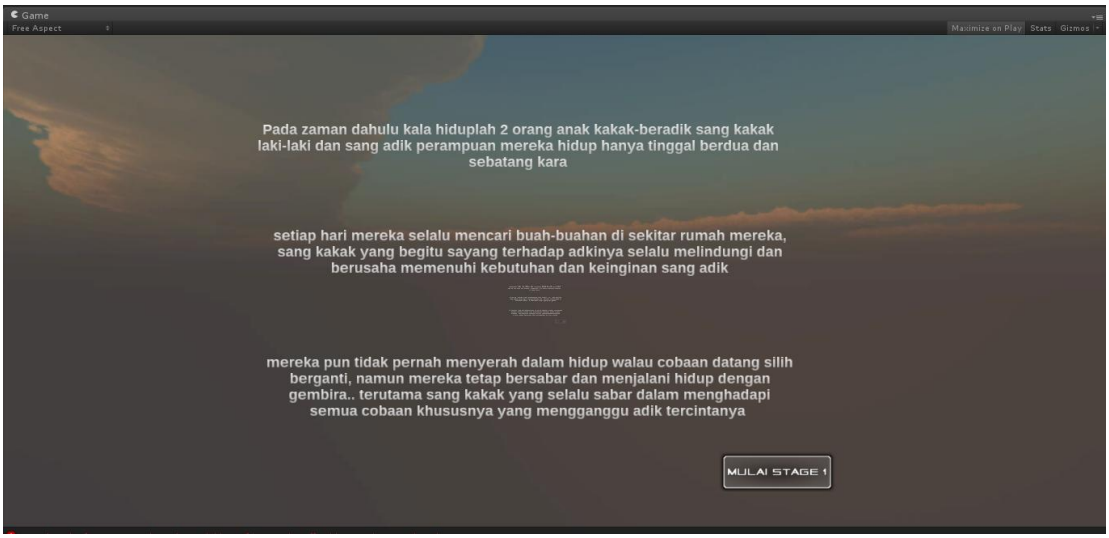
   Menu Utama merupakan *scene* awal yang terdapat pada *game* budak pahatu lalis didalam menu utama terdapat 4 menu yaitu. Mulai Permainan untuk mulai bermain, Petunjuk untuk melihat petunjuk *game*, Pembuat Game untuk melihat *profile* pembuat *game*, serta menu Keluar untuk keluar dari aplikasi gambar dari menu utama terdapat pada gambar 4.1



Gambar 4.1 Antarmuka Menu Utama

2. **Antarmuka Menu Intro**

   Menu Intro merupakan *scene* yang menampilkan ulasan cerita rakyat kisah budak pahatu lalis secara singkat, menu intro merupakan menu awal sebelum pemain masuk ke *scene game* dan mulai memainkan *game* Menu Intro dibuat agar pemain memiliki gambaran sekilas tentang apa yang akan dimainkan gambar Menu Intro terdapat pada gambar 4.2



Gambar 4.2 Antarmuka Menu Intro

3. Antarmuka *Scene Game Stage* 1

Scene game stage 1 merupakan *scene* dimana pemain memainkan *game stage* 1, disini pemain dituntut untuk mencari buah sampai point makanan bernilai 200 sebelum nyawa adik habis



Gambar 4.3 *Scene game stage* 1

4. Antarmuka *Scene Game Stage* 2

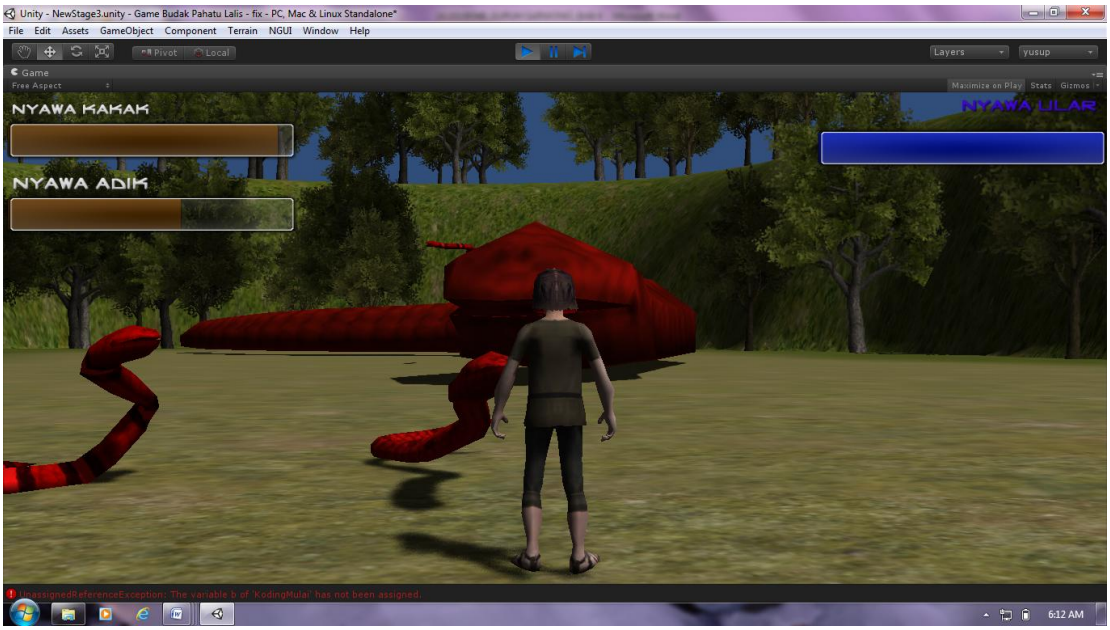
Scene game stage 2 merupakan *scene* dimana pemain memainkan *game stage* 2, disini pemain dituntut untuk melawan semua musuh yang datang menghadang perjalanan menuju ke rumah pemain dituntut untuk bisa melawan semua musuh, satu musuh bernilai 10 berusahalah sampai poin musuh bernilai 200 namun nyawa sang kakak jangan sampai habis



Gambar 4.4 *Scene game stage* 2

5. Antarmuka *Scene Game Stage* 3

Scene *game stage* 3 merupakan *scene* dimana pemain memainkan *game stage* 3, disini pemain dituntu untuk mencari ular yang telah memakan sang adik kemudian melawannya agar sang adik kembali selamat sang kakak dituntut untuk menemukan dan mengalahkan sebelum nyawa sang adik habis



Gambar 4.5 *Scene game stage* 3

6. Antarmuka Menu Petunjuk

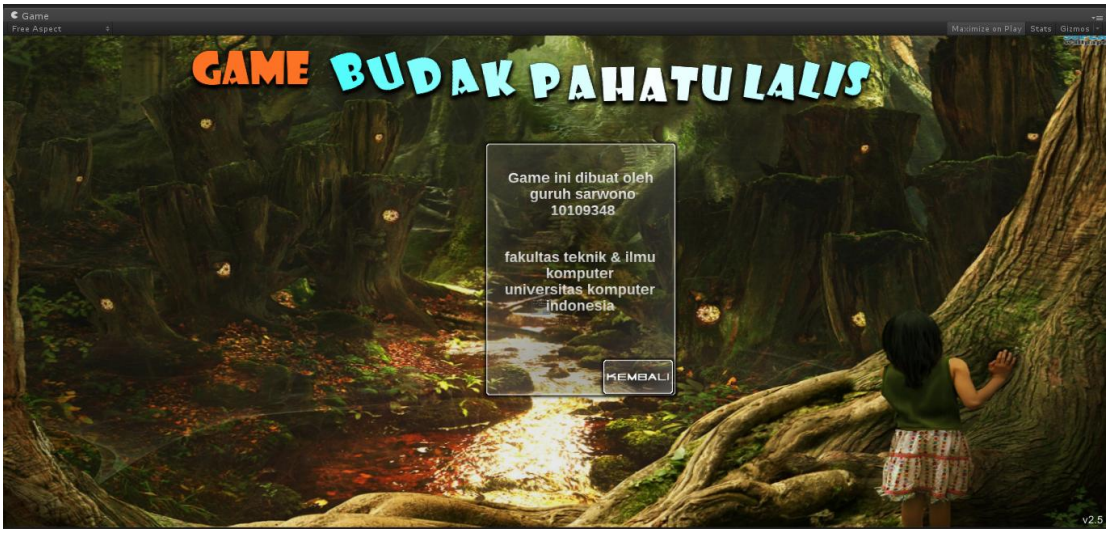
Menu petunjuk merupakan menu untuk membaca aturan atau perintah yang ada didalam *game* agar semua misi tercapai



Gambar 4.6 Menu Petunjuk

7. **Antarmuka Menu *Credits***

Menu *Credits* merupakan menu untuk menampilkan informasi mengenai pembuat *game* budak pahatu lalis



Gambar 4.7 Menu Credits

4.2 Pengujian

Dalam pengujian pada umumnya terdapat dua macam teknik pengujian yaitu *black box sesting* dan *white box testing*. Pengujian yang menggunakan teknik *white box testing* lebih menitikberatkan pada struktur internal (*source code*) program, di mana setiap baris kode yang akan di *compile* dicek satu persatu untuk mengetahui apakah masih terjadi *error*. Pengujian pada *black box testing* tidak terlalu memperhatikan struktur internal (*source code*) dari program, tetapi lebih kepada menemukan kesalahan program. Pada game Budak Pahatu Lalis ini digunakan pengujian *black box testing* dan pengujian *white box testing*

4.2.1 Pengujian *Alpha*

Pengujian dilakukan terhadap aplikasi untuk memastikan bahwa aplikasi dapat berjalan dengan benar sesuai dengan kebutuhan dan tujuan yang diharapkan. Pengujian *alpha* berfokus pada persyaratan fungsional perangkat lunak

4.2.1.1 Skenario Pengujian Aplikasi (*Black Box Testing*)

Skenario pengujian menjelaskan pengujian terhadap sistem yang ada pada aplikasi *game* budak pahatu lalis. Skenario pengujian yang akan dilakukan pada aplikasi ini selengkapnya dapat dilihat pada tabel 4.1

Tabel 4.1 Skenario Pengujian Aplikasi

No	Komponen yang Diuji	Skenario Pengujian	Jenis Pengujian
1.	Menu Utama	Memilih Tombol Mulai Permainan	*Black box*
		Memilih Tombol Petunjuk	*Black box*
		Memilih Tombol *Credits*	*Black box*

No	Komponen yang Diuji	Skenario Pengujian	Jenis Pengujian
		Memilih Tombol Keluar	*Black box*
2.	Menu Mulai Permainan	Memilih Tombol Mulai *Stage* 1	*Black box*
4.	Menu Petunjuk	Memilih Tombol Kembali	*Black box*
5.	Menu *Credits*	Memilih Tombol Kembali	*Black box*
6.	Menu Keluar	Memilih Tombol Ya	*Black box*
		Memilih Tombol Tidak	*Black box*
7.	*Scene Game*	Menu *Stage* 1	*Black box*
		Menu *Stage* 2	*Black box*
		Menu *Stage* 3	*Black box*
		Menu *Pause*	*Black box*
8	Gerak Karakter Player	TekanTombol W *keyboard*	*Black box*
		Tekan Tombol S *keyboard*	*Black box*
		Tekan Tombol A *keyboard*	*Black box*
		Tekan Tombol D *keyboard*	*Black box*
		Tekan Tombol *space* keyboard	*Black box*
		Tekan Tombol Kiri *Mouse*	*Black box*
9.	Algoritma	A*(A *Star*)	*Black box*

4.2.1.2 Kasus Dan Hasil Pengujian (*Black Box Testing*)

Berikut ini merupakan kasus untuk menguji perangkat lunak yang sudah dibangun dengan menggunakan metode *black box testing* berdasarkan skenario pengujian aplikasi yang terdapat pada tabel 4.1 yang sudah dijabarkan. Berikut ini pemaparan dari setiap butir pengujian yang terdapat pada skenario pengujian.

Pengujian Mulai Permainan

1. Pengujian Menu Utama dengan kasus dan hasil pengujian yang menghasilkan kesimpulan dapat dilihat pada tabel 4.2

Tabel 4.2 Skenario Pengujian Menu Utama

Kasus dan Pengujian			
Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**

Kasus dan Pengujian			
Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Tombol Mulai Permainan	Masuk ke *Intro Scene*	Tombol dapat berfungsi sesuai dengan yang diharapkan	(✓) Diterima ( ) Ditolak
Tombol Petunjuk	Menampilkan petunjuk *game*	Tombol dapat berfungsi sesuai dengan yang diharapkan	(✓) Diterima ( ) Ditolak
Tombol *Credits*	Menampilkan informasi *game*	Tombol dapat berfungsi sesuai dengan yang diharapkan	(✓) Diterima ( ) Ditolak
Tombol Keluar	Keluar dari aplikasi	Tombol dapat berfungsi sesuai dengan yang diharapkan	(✓) Diterima ( ) Ditolak

2. Pengujian Mulai Permainan dengan kasus dan hasil pengujian yang menghasilkan kesimpulan dapat dilihat pada tabel 4.3

Tabel 4.3 Skenario Pengujian Mulai Permainan

Kasus dan Pengujian			
Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Mulai *Stage* 1	Menampilkan Permainan *stage* 1	Tombol dapat berfungsi sesuai dengan yang diharapkan	(✓) Diterima ( ) Ditolak

3. Pengujian Petunjuk dengan kasus dan hasil pengujian yang menghasilkan kesimpulan dapat dilihat pada tabel 4.4

Tabel 4.4 Skenario Pengujian Petunjuk

Kasus dan Pengujian			
Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Tombol Kembali	Menampilkan menu utama	Tombol dapat berfungsi sesuai dengan yang diharapkan	(✓) Diterima ( ) Ditolak

4. Pengujian *Credits* dengan kasus dan hasil pengujian yang menghasilkan kesimpulan dapat dilihat pada tabel 4.5

Tabel 4.5 Skenario Pengujian *Credits*

Kasus dan Pengujian			
Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Tombol Kembali	Menampilkan menu utama	Tombol dapat berfungsi sesuai dengan yang diharapkan	(✓) Diterima ( ) Ditolak

5. Pengujian Keluar dengan kasus dan hasil pengujian yang menghasilkan kesimpulan dapat dilihat pada tabel 4.6

Tabel 4.6 Skenario Pengujian Keluar

Kasus dan Pengujian			
Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Tombol Kembali	Menampilkan menu utama	Tombol dapat berfungsi sesuai dengan yang diharapkan	(✓) Diterima ( ) Ditolak

6. Pengujian Menu *Stage* 1 dengan kasus dan hasil pengujian yang menghasilkan kesimpulan dapat dilihat pada tabel 4.7

Tabel 4.7 Skenario Pengujian Menu *Stage* 1

Kasus dan Pengujian			
Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Tombol Mulai Permainan *Stage* 1	Masuk ke permainan *stage* 1	Tombol dapat berfungsi sesuai dengan yang diharapkan	(✓) Diterima ( ) Ditolak

7. Pengujian Menu *Stage* 2 dengan kasus dan hasil pengujian yang menghasilkan kesimpulan dapat dilihat pada tabel 4.8

Tabel 4.8 Skenario Pengujian Menu *Stage* 2

Kasus dan Pengujian			
Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Tombol Mulai Permainan *Stage* 2	Masuk ke permainan *stage* 2	Tombol dapat berfungsi sesuai dengan yang diharapkan	(✓) Diterima ( ) Ditolak

8. Pengujian Menu *Stage* 3 dengan kasus dan hasil pengujian yang menghasilkan kesimpulan dapat dilihat pada tabel 4.9

Tabel 4.9 Skenario Pengujian Menu *Stage* 3

Kasus dan Pengujian			
Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Tombol Mulai Permainan *Stage* 3	Masuk ke permainan *stage* 3	Tombol dapat berfungsi sesuai dengan yang diharapkan	(✓) Diterima ( ) Ditolak

9. Pengujian Menu *Pause* dengan kasus dan hasil pengujian yang menghasilkan kesimpulan dapat dilihat pada tabel 4.10

Tabel 4.10 Skenario Pengujian Menu *Pause*

Kasus dan Pengujian			
Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Tombol *Escape* Pada *Keyboard*	Menampilkan menu *pause* dan menghentikan permainan	Tombol dapat berfungsi sesuai dengan yang diharapkan	(✓) Diterima ( ) Ditolak

10. Pengujian Gerakan Karakter Berjalan Maju dengan kasus dan hasil pengujian yang menghasilkan kesimpulan dapat dilihat pada tabel 4.11

Tabel 4.11 Skenario Pengujian Gerakan Karakter Berjalan Maju

Kasus dan Pengujian			
Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Tombol W Pada *Keyboard*	Menampilkan gerakan karakter berjalan maju	Tombol dapat berfungsi sesuai dengan yang diharapkan	(✓) Diterima ( ) Ditolak

11. Pengujian Gerakan Karakter Berjalan Mundur dengan kasus dan hasil pengujian yang menghasilkan kesimpulan dapat dilihat pada tabel 4.12

Tabel 4.12 Skenario Pengujian Gerakan Karakter Berjalan Mundur

Kasus dan Pengujian			
Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Tombol S Pada *Keyboard*	Menampilkan gerakan karakter berjalan mundur	Tombol dapat berfungsi sesuai dengan yang diharapkan	(✓) Diterima ( ) Ditolak

12. Pengujian Gerakan Karakter Berjalan Ke Kanan dengan kasus dan hasil pengujian yang menghasilkan kesimpulan dapat dilihat pada tabel 4.13

Tabel 4.13 Skenario Pengujian Gerakan Karakter Berjalan Ke Kanan

Kasus dan Pengujian			
Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Tombol D Pada *Keyboard*	Menampilkan gerakan karakter berjalan ke kanan	Tombol dapat berfungsi sesuai dengan yang diharapkan	(✓) Diterima ( ) Ditolak

13. Pengujian Gerakan Karakter Berjalan Ke Kanan dengan kasus dan hasil pengujian yang menghasilkan kesimpulan dapat dilihat pada tabel 4.14

Tabel 4.14 Skenario Pengujian Gerakan Karakter Berjalan Ke Kiri

Kasus dan Pengujian			
Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Tombol A Pada *Keyboard*	Menampilkan gerakan karakter berjalan ke kanan	Tombol dapat berfungsi sesuai dengan yang diharapkan	(✓) Diterima ( ) Ditolak

14. Pengujian Gerakan Karakter Meloncat dengan kasus dan hasil pengujian yang menghasilkan kesimpulan dapat dilihat pada tabel 4.15

Tabel 4.15 Skenario Pengujian Gerakan Karakter Meloncat

Kasus dan Pengujian			
Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Tombol *Space* Pada *Keyboard*	Menampilkan gerakan karakter meloncat	Tombol dapat berfungsi sesuai dengan yang diharapkan	(✓) Diterima ( ) Ditolak

15. Pengujian Gerakan Karakter Memukul dengan kasus dan hasil pengujian yang menghasilkan kesimpulan dapat dilihat pada tabel 4.16

Tabel 4.16 Skenario Pengujian Gerakan Karakter Memukul

Kasus dan Pengujian			
Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Tombol Kiri (*Fire* 1) Pada *Mouse*	Menampilkan gerakan karakter meloncat	Tombol dapat berfungsi sesuai dengan yang diharapkan	(✓) Diterima ( ) Ditolak

4.2.1.3 Kasus Dan Hasil Pengujian (*White Box Testing*)

Black box testing dilakukan untuk memastikan semua perintah dan kondisi dieksekusi minimal sekali. Pengujian *white box* disini digunakan untuk menguji Algoritma yang dipakai didalam *game* budak pahatu lalis dengan menggunakan *flow graph* untuk menggambarkan alur dari algoritma dan *grap matrix* untuk menggenerasi *flow graph*. Adapun pengujian *white-box* adalah sebagai berikut.

1. **Pengujian Algoritma *A** (A *Star*)**

   Pengujian algoritma *A** merupakan proses dimana akan diuji implementasi A* yang diterapkan pada karakter musuh dalam melakukan pencarian jalan terhadap karakter utama. Berikut merupakan *sourcecode* dari Algortima *A** yang dibangun dengan bahasa pemrograman C#.

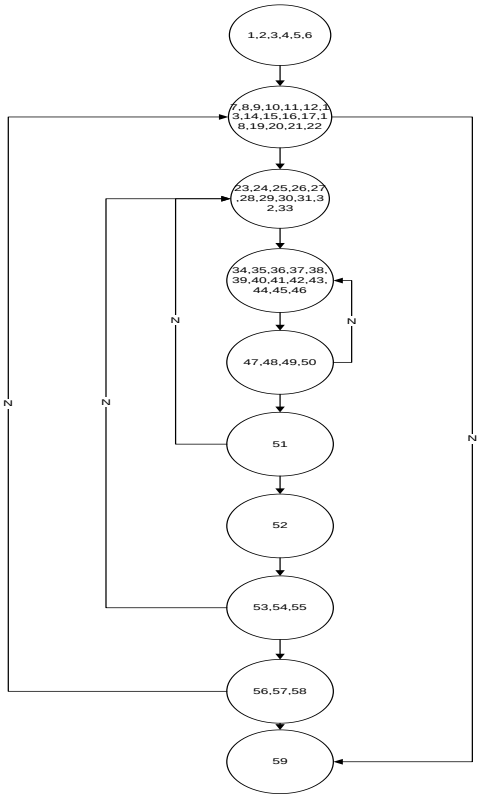
```
1. public Matrix translate (float _X, float _Y, float _Z){
2. m[12] = _X;
3. m[13] = _Y;
4. m[14] = _Z;
5. return this;
6. }
7. public virtual int[] InitialOpen (BinaryHeapM open, Int3
8. targetPosition, Int3 position, Path path, bool doOpen) {
9. return BaseInitialOpen
10.(open,targetPosition,position,path,doOpen);
11.}
12.public void Close () {
13.if (readerStream != null) {
14.readerStream.Close ();
15.}
16.if (writerStream != null) {
17.if (anchors != null) { 118
18.SerializeAnchors ();
19.}
20.writerStream.Close ();
21.}
22.}
23.if (doOpen) {
24.connectionCosts = costs;
25.}
```

```
26. if (connectionCosts == null) {
27. return null;
28. }
29. int[] costs = connectionCosts;
30. connectionCosts = new int[connectionCosts.Length];
31. for (int i=0;i<connectionCosts.Length;i++) {
32. connectionCosts[i] = (connections[i].position-
    position).costMagnitude;
33. }
34. if (!doOpen) {
35. for (int i=0;i<connectionCosts.Length;i++) {
36. Node other = connections[i];
37. if (other.connections != null) {
38. for (int q = 0;q < other.connections.Length;q++) {
39. if (other.connections[q] == this) {
40. other.connectionCosts[q] = connectionCosts[i];
41. break;
42. }
43. }
44. }
45. }
46. }
47. if (connections[i].walkable && connections[i].area !=
    area) {
48. stack.Push (connections[i]);
49. connections[i].area = area;
50. }
51. if (doOpen) {
52. nodeR.g = nodeR.parent.g+nodeR.cost+penalty +
    nodeRunData.path.GetTagPenalty(tags);
53. 119

54. public virtual void UpdateAllG (NodeRun nodeR,
    NodeRunData nodeRunData) {
55. BaseUpdateAllG (nodeR, nodeRunData);
56. }
57. if (Open == null){
58. return null;
59. }
60. return path;
```

2. *Flow Graph*

Berikut merupakan *flowgraph* berdasarkan *sourcecode* diatas yang terdapat pada gambar 4.8



Gambar 4.8 *Flow Graph* Algoritma *A

Keterangan:

= Menggambarkan kondisi

= Menggambarkan aksi

3. *Cyclomatic Complexity* V(G)

V(G) = E – N + 2

V(G) = 13 – 12 + 2

V(G) = 3

Keterangan:

E = Jumlah busur pada *flowgraph*

N = Jumlah simpul pada *flowgraph*

V(G) = *Cyclomatic Complexity*

4. Independent Path

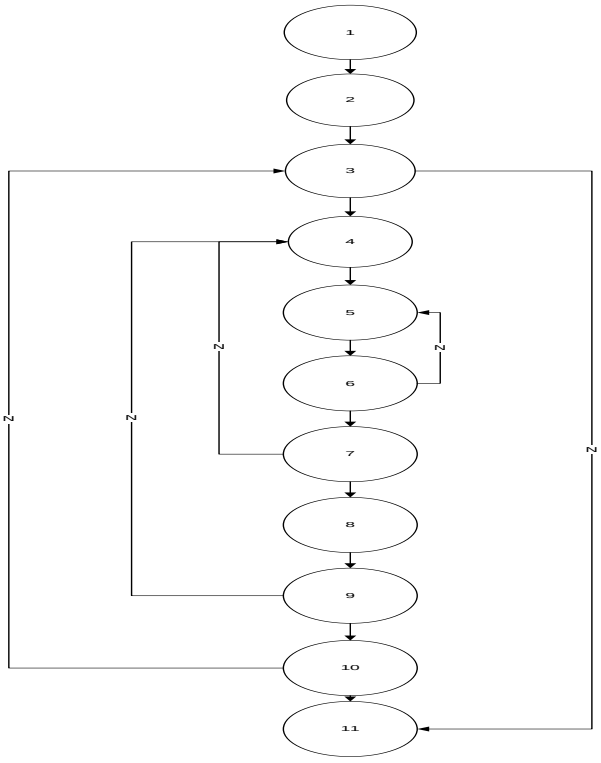
Path 1: 1-18

Path 2: 1-2-3-4-5-9-10-11-12-13-14-15-16-17-1-18

Path 3: 1-2-3-4-5-6-7-8-16-17-1-18

5. Penyederhanaa *Flow Graph*

Berikut merupakan hasil penyederhanaan *flowgraph* yang terdapat pada gambar 4.9



Gambar 4.9 Penyederhanaan *Flow Graph* Algoritma *A**

6. *Graph Matrices Testing*

	1	2	3	4	5	6	7	8	9	10	11	12	Hasil
1		1										1	1
2			1										0
3				1									0
4					1		1						1
5						1							0
6										1			0
7								1					0
8									1				0
9										1			0
10											1		0
11	1												0
12													0
Hasil													2

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

V(G) = 2 + 1

V(G) = 3

7. Kesimpulan

Berdasakan hasil pengujian dihasilkan nilai *cyclomatic complexity* yang sama yaitu 3, maka didapat kesimpulan bahwa pada proses algoritma A* berjalan dengan baik, karena hasil pengujian memiliki nilai yang sama.

4.2.2 Pengujian Beta

Pengujian *beta* merupakan pengujian yang dilakukan secara objektif, dimana pengujian langsung melibatkan para pengguna aplikasi permainan secara langsung dengan menggunakan kuesioner mengenai kepuasan para pengguna terhadap aplikasi permainan yang dibangun. Pengujian dilakukan terhadap 30 responden dengan usia 10 dan 11 tahun atau anak-anak kelas 5 dan 6 (sekolah dasar) dan setiap responden diberikan 5 pertanyaan untuk kemudian dilakukan penilaian dengan memilih 5 pilihan yang telah disediakan untuk setiap pertanyaannya.

Skala yang digunakan dalam pengukuran kuesioner adalah skala *guttman*, yaitu skala yang menginginkan tipe jawaban tegas**,** dan jawaban yang digunakan dari setiap kuisioner terdiri dari 4 skala seperti sangat setuju, setuju, tidak setuju, sangat tidak setuju

4.2.2.1 Rumusan Pengujian Beta

Berdasarkan data hasil kuesioner tersebut, dicari persentase masing – masing jawaban dengan menggunakan rumus:

R = P/Q * 100 %

Keterangan:

P : Banyaknya jawaban dari responden

Q : Jumlah Responden

R : Nilai Presentasi

4.2.2.2 Hasil Kuisioner

Berikut ini adalah daftar pertanyaan kuesioner yang diberikan, serta hasil yang didapat setelah nilainya dihitung.

1. Apakah menurut Adik-adik game ini telah sesuai dengan cerita rakyat kisah budak pahatu lalis?
   A. Sangat Sesuai
   B. Sesuai

C. Tidak Sesuai

D. Sangat Tidak Sesuai

Kategori Jawaban	Sangat Sesuai	Sesuai	Tidak Sesuai	Sangat Tidak Sesuai
Frekuensi jawaban	8	21	1	0
Persentase jawaban	27%	69%	4%	0

2. Apakah Adik-adik setuju jika game ini dapat menjadi media pengganti cerita rakyat berjudul kisah budak pahatu lalis sehingga dapat mengangkat kembali cerita rakyat berjudul kisah budak pahatu lalis ?

   A Sangat Setuju

   B Setuju

   C Tidak Setuju

   D Sangat Tidak Setuju

Kategori Jawaban	Sangat Setuju	Setuju	Tidak Setuju	Sangat Tidak Setuju
Frekuensi jawaban	17	11	2	0
Persentase jawaban	57%	36%	7%	0

3. Apakah Adik-adik merasa mendapatkan manfaat berupa pesan moral dari game ini?

   A Sangat Merasa

   B Merasa

   C Tidak Merasa

   D Sangat Tidak Merasa

Kategori Jawaban	Sangat Merasa	Merasa	Tidak Merasa	Sangat Tidak Merasa
Frekuensi jawaban	13	14	3	0

Persentase jawaban	43%	47%	10%	0

4. Apakah Adik-adik setuju jika game ini dapat membuat Adik-adik menyukai cerita rakyat berjudul kisah budak pahatu lalis?

   A Sangat Setuju

   B Setuju

   C Tidak Setuju

   D Sangat Tidak Setuju

Kategori Jawaban	Sangat Setuju	Setuju	Tidak Setuju	Sangat Tidak Setuju
Frekuensi jawaban	6	19	4	1
Persentase jawaban	20%	63%	13%	3%

5. Apakah Adik-adik setuju jika game ini dapat membuat Adik-adik jadi mengetahui cerita rakyat berjudul kisah budak pahatu lalis setelah bermain game ini?

   A Sangat Setuju

   B Setuju

   C Tidak Setuju

   D Sangat Tidak Setuju

Kategori Jawaban	Sangat Setuju	Setuju	Tidak Setuju	Sangat Tidak Setuju
Frekuensi jawaban	7	18	4	1
Persentase jawaban	23%	60%	13%	4%

4.2.2.3 Kesimpulan Pengujian Beta

Setelah melakukan pengujian maka didapatlah kesimpulan berdasarkan hasil pengujian beta yang dilakukan dengan menyebar kuisioner kepada anak-anak kalangan sekolah dasar. Berikut kesimpulan dari pengujian beta

1. Game budak pahatu lalis telah sesuai dengan cerita rakyat aslinya yaitu kisah budak pahatu lalis
2. Game ini dapat menjadi media pengganti cerita rakyat kisah budak pahatu lalis, sehingga apabila banyak yang memainkan dapat mengangkat kembali cerita rakyat tersebut
3. Game ini dapat memberikan pesan moral kepada anak-anak kalangan sekolah dasa
4. Game ini dapat membuat anak-anak kalangan sekolah dasar menyukai cerita rakyat ini
5. Game ini dapat membuat anak-anak kalangan sekolah dasar mengetahui cerita rakyat asli kisah budak pahatu lalis