BAB IV

IMPLEMENTASI DAN PENGUJIAN

IV.1. Implementasi

Implementasi merupakan tahap penciptaan perangkat lunak dan juga tahap kelanjutan dari kegiatan perancangan aplikasi, pada tahap ini aplikasi harus siap untuk dijalankan. Aplikasi ini diimplementasikan dengan nama Takedown The Terrorists dengan *file extension*-nya *.exe*.

IV.1.1 Implementasi Perangkat Keras

Perangkat keras yang dapat digunakan untuk menjalankan aplikasi *game* ini harus memiliki spesifikasi yang telah ditentukan oleh pembuat yang meliputi beberapa komponen perangkat keras sebagai kebutuhan sistem minimum agar dapat menjalankan *game Takedown The Terrorists*. Berikut ini adalah kebutuhan sistem perangkat keras minimum agar dapat menjalankan *game Takedown The Terrorists* adalah sebagai berikut :

1. Processor Dual Core 1,7 Ghz
2. RAM 2 GB
3. VGA 512 MB
4. Harddisk 2 GB
5. Mouse dan Keyboard

IV.1.2 Implementasi Perangkat Lunak

Adapun spesifikasi perangkat lunak yang dibutuhkan untuk menjalankan *game Takedown The Terrorists* adalah :

1. Sistem operasi minimal windows XP 32-bit
2. DirectX 9.0

IV.1.3 Implementasi Installasi Aplikasi Game

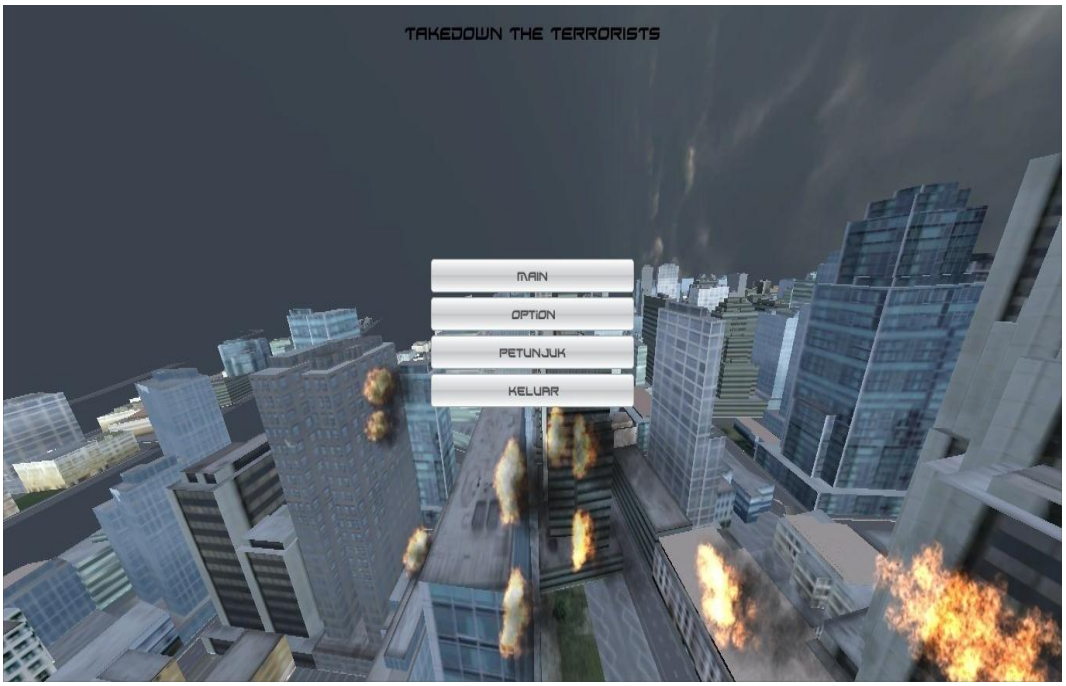
Untuk memainkan *game Takedown The Terrorists*, pemain terlebih dahulu harus memiliki *file Takedown The Terrorists .exe*.

IV.1.4 Implementasi Antarmuka

Antarmuka dapat membuat pengguna merasa nyaman dalam penggunaan fitur-fitur dan tampilan yang ada pada aplikasi ditinjau dari warna, bentuk dan kemudahan dalam penggunaan aplikasinya. Berikut ini merupakan antarmuka yang ada pada aplikasi *game Takedown The Terrorists.*

IV.1.4.1 Antarmuka Menu Utama

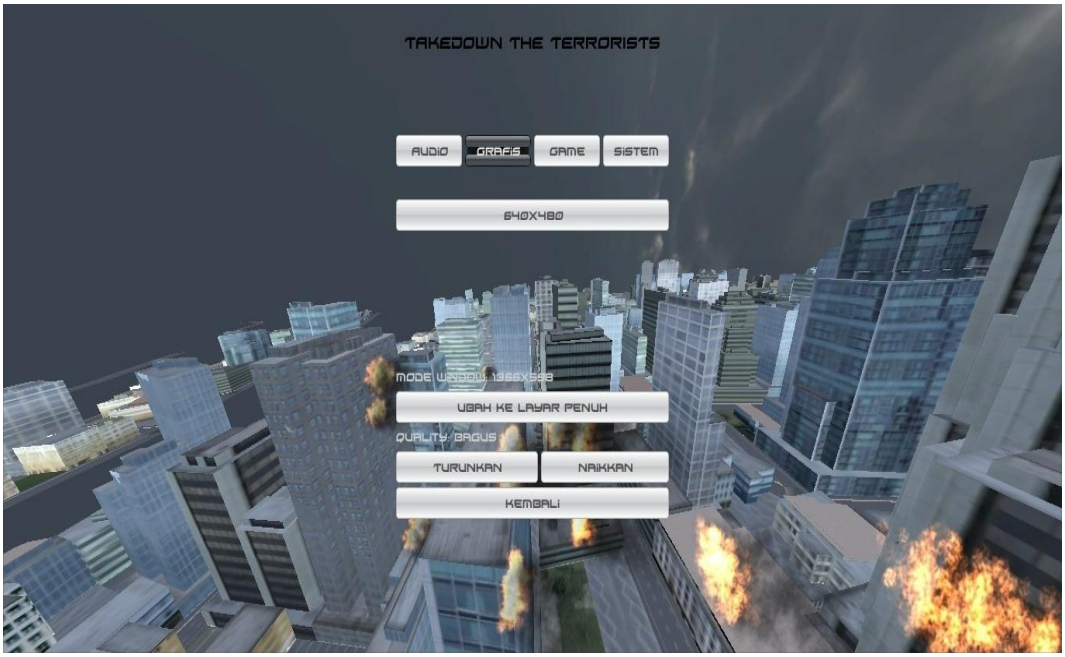
Antarmuka menu utama merupakan antarmuka dimana saat pengguna membuka aplikasi pertama kali maka antarmuka yang muncul ialah antarmuka menu utama. Berikut ini merupakan tampilan dari antarmuka menu utama



Gambar IV.1 Antarmuka Menu Utama

IV.1.4.2 Antarmuka Menu Option

Antarmuka yang ditampilkan ketika pengguna ingin masuk ke pengaturan game, pengaturan yang disajikan ialah pengaturan gambar, suara sensitivitas mouse, dan untuk mengetahui hardware yang digunakan pada komputer pemain



Gambar IV.2 Antarmuka Menu Option

IV.4.1.3 Antarmuka Menu Main

Antarmuka yang ditampilkan oleh sistem adalah antarmuka *scene* video berupa video cerita awal dari *game* pada *level* pertama. Selanjutnya akan masuk ke antarmuka permainan. Jika seluruh misi telah diselesaikan maka akan masuk ke antarmuka *ending scene* berupa video akhir cerita dari *game*. Berikut ini merupakan tampilan dari antarmuka menu main.



Gambar IV.3 Antarmuka *scene video level 1*

Gambar IV.3 merupakan antarmuka *scene video* pada *level* pertama yang terdapat pada *game Takedown The Terrorists*.



Gambar IV.4 Antarmuka *scene level 1*

Gambar IV.4 merupakan antarmuka permainan *level* pertama yang terdapat pada *game Takedown The Terrrorists.*



Gambar IV.5 Antarmuka *scene level 2*

Gambar IV.5 merupakan antarmuka permainan *level* kedua yang terdapat pada *game Takedown The Terrrorists*



Gambar IV.6 Antarmuka *scene level 3*

Gambar IV.6 merupakan antarmuka permainan *level* ketiga yang terdapat pada *game Takedown The Terrrorists*

IV.1.4.4 Antarmuka Menu Petunjuk

Antarmuka Menu Petunjuk menampilkan ketika pada saat pengguna memilih menu petunjuk, maka antarmuka yang ditampilkan oleh sistem adalah antarmuka informasi kontrol yang digunakan pada *game Takedown The Terrorists.*



Gambar IV.7 Antarmuka Petunjuk

IV.2 Pengujian

Pada umumnya ada 2 teknik pengujian yaitu *white box testing* dan *black box testing***.** Pengujian yang menggunakan teknik *white box* lebih menitik beratkan pada struktur internal (*source code*) dari program, tetapi lebih kepada menemukan kesalahan yang ada di dalam program. Selain *white box testing* ada pengujian lain yang dilakukan pada *game Takedown The Terrorists* yaitu pengujian *black box testing.* Cara pengujian dilakukan melalui pengujian *beta.*

IV.2.1 Skenario Pengujian *Black box*

Skenario pengujian *blacbox* menjelaskan skenario pengujian sistem yang ada di dalam *game Takedown The Terrorists* bisa dilihat pada tabel IV.1

Tabel IV.1 Rencana Pengujian

Kelas Uji	Butir Uji	Jenis Pengujian
Menu Utama	Masuk menu main	*Black Box*
	Masuk menu option	*Black Box*
	Masuk menu petunjuk	*Black Box*
	Keluar	*Black Box*
Menu Main	Menampilkan menu main	*Black Box*
Menu Option	Menampilkan menu option	*Black Box*
Menu Petunjuk	Menampilkan menu petunjuk	*Black Box*
Keluar	Keluar permainan	*Black Box*
Permainan	Menggerakkan karakter	*Black Box*
	Menembakan senjata	*Black Box*
	Mengganti senjata	*Black Box*
	Menampilkan menu pause	*Black Box*
Mengambil darah	Menambahkan nilai darah	*Black Box*
Menembak	Menembakkan senjata	*Black Box*
Mengganti Senjata	Mengganti senjata 1	*Black Box*
	Mengganti senjata 2	*Black Box*
Memberhentikan permainan	Menghentikan permainan	*Black Box*
	Menampilkan menu pause	*Black Box*
Mengambil petunjuk	Menampilkan petunjuk permainan	*Black Box*
Mengambil peluru	Menambahkan peluru senjata	*Black Box*
Menghitung skor	Menambahkan Skor	*Black Box*

IV.2.2. Kasus dan Hasil Pengujian *Blackbox*

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 IV.1 yang sudah dijabarkan. Hasil

pengujian yang akan dilakukan pada aplikasi ini selengkapnya dapat dilihat pada tabel IV.1

Tabel IV.2 Pengujian Black Box

No	Kasus yang akan diuji	Skenario dan Hasil Uji		
		Skenario Uji	**Hasil Yang Diharapkan**	**Kesimpulan**
1	Menu Utama	Klik Menu Main	Sistem menampilkan halaman *video scene*	[√] Berhasil [ ] Tidak Berhasil
		Klik Menu Option	Sistem menampilkan halaman menu Option	[√] Berhasil [ ] Tidak Berhasil
		Klik Menu Petunjuk	Sistem menampilkan halaman menu Petunjuk	[√] Berhasil [ ] Tidak Berhasil
		Klik Menu Keluar	Menutup atau Keluar dari aplikasi	[√] Berhasil [ ] Tidak Berhasil
2	Menu Main	Klik Menu Main	Sistem menampilkan halaman *video scene*	[√] Berhasil [ ] Tidak Berhasil
3	Menu Option	Klik Menu Option	Sistem menampilkan halaman menu Option	[√] Berhasil [ ] Tidak Berhasil
4	Menu Petunjuk	Klik Menu Petunjuk	Sistem menampilkan halaman menu Petunjuk	[√] Berhasil [ ] Tidak Berhasil
5	Menu Keluar	Klik Menu Keluar	Menutup atau Keluar dari aplikasi	[√] Berhasil [ ] Tidak Berhasil
6	Permainan	Tekan tombol W *keyboard*	Karakter pemain bergerak kedepan	[√] Berhasil [ ] Tidak Berhasil
		Tekan tombol A *keyboard*	Karakter pemain bergerak kekiri	[√] Berhasil [ ] Tidak Berhasil
		Tekan tombol S *keyboard*	Karakter pemain bergerak kebelakang	[√] Berhasil [ ] Tidak Berhasil
		Tekan tombol D *keyboard*	Karakter pemain bergerak kekanan	[√] Berhasil [ ] Tidak Berhasil
		Tekan tombol W *keyboard*	Karakter pemain bergerak kedepan	[√] Berhasil [ ] Tidak Berhasil
		Klik kiri *mouse*	Karakter pemain menembak	[√] Berhasil [ ] Tidak Berhasil
		Tekan tombol “esc” *keyboard*	Menampilkan *Pause Screen*	[√] Berhasil [ ] Tidak Berhasil
		Tekan tombol 1 *keyboard*	Karakter memakai senjata 1	[√] Berhasil [ ] Tidak Berhasil
		Tekan tombol 2 *keyboard*	Karakter memakai senjata 2	[√] Berhasil [ ] Tidak Berhasil
		Tekan tombol F *keyboard*	Karakter memakai grenade launcher	[√] Berhasil [ ] Tidak Berhasil
		Tekan tombol “spasi” *keyboard*	Karakter melompat	[√] Berhasil [ ] Tidak Berhasil
		Tekan tombol	Karakter berjalan	[√] Berhasil

		"shift" *keyboard*		[ ] Tidak Berhasil
7	Mengambil darah	Mengarahkan karakter hingga menyentuh icon darah menggunakan *mouse* dan *keyboard*	Bar darah pada permainan akan bertambah jika darah pemain memang tidak penuh dan *icon* darah menghilang	[√] Berhasil [ ] Tidak Berhasil
8	Menembak	Mengarahkan *pointer* / *cursor* senjata menggunakan *keyboard* dan *mouse* lalu menekan tombol kiri *mouse* untuk mengeluarkan peluru / *mouse*	Senjata yang dipegang karakter pada *game* akan mengeluarkan peluru dan suara tembakan secara bersama-sama	[√] Berhasil [ ] Tidak Berhasil
9	Mengganti Senjata	Menekan tombol 1 pada *keyboard* untuk menggunakan senjata 1 dan menekan tombol 2 pada *keyboard* untuk menggunakan senjata 2	Senjata 1 akan digunakan apablia pemain menekan tombol 1 dan Senjata 2 akan digunakan apablia pemain menekan tombol 2	[√] Berhasil [ ] Tidak Berhasil
10	Memberhenti-kan permainan	Menekan tombol "esc" pada *keyboard*	*Game* akan mengeluarkan *screen pause*	[√] Berhasil [ ] Tidak Berhasil
11	Mengambil petunjuk	Mengarahkan karakter hingga menyentuh icon panah menggunakan *mouse* dan *keyboard*	*Game* akan menampilkan *pop up* berupa teks, gambar dan suara	[√] Berhasil [ ] Tidak Berhasil
12	Mengambil Peluru	Mengarahkan karakter hingga menyentuh icon peluru menggunaakan mouse dan keyboard	*GUI bullets* bertambah	[√] Berhasil [ ] Tidak Berhasil
13	Menghitung skor	Karakter menembak musuh hingga musuh mati	*Score* ditambahkan +100 untuk setiap musuh yang mati	[√] Berhasil [ ] Tidak Berhasil

IV.2.3 Kasus dan Hasil Pengujian (*White Box Testing*)

1. Pengujian *White Box* Algoritma A*

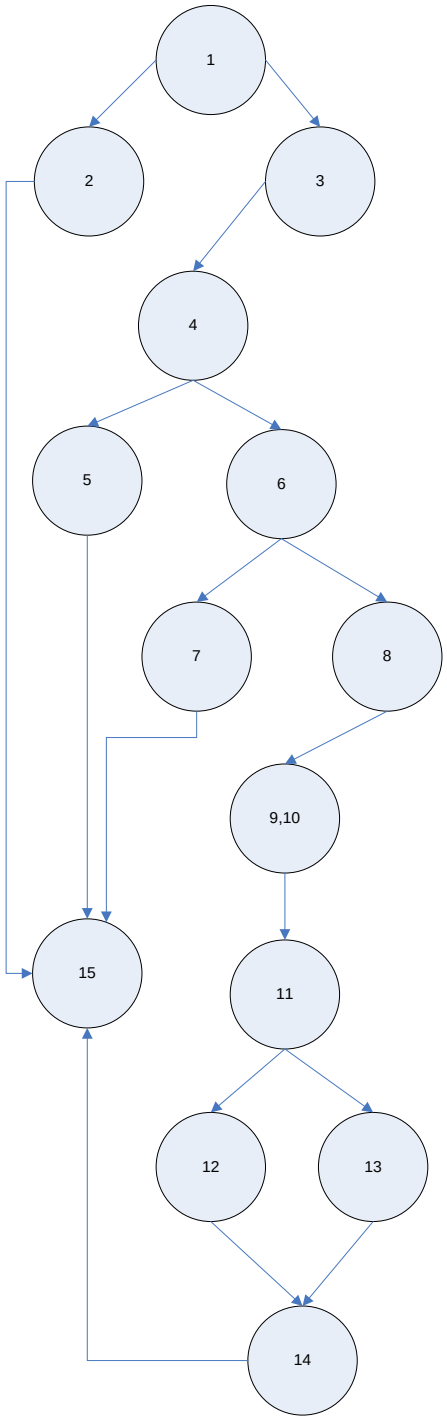
   Berikut ini merupakan algoritma A* yang penerapannya menggunakan bahasa pemrograman C#:

```
1. public int f {
2. get {
3. return g+h;
4. }
5. }
6. if (heuristic == Heuristic.None) {
7. nodeR.h = 0;}
8. else if (heuristic == Heuristic.Manhattan) {
9. nodeR.h = (uint)Mathfx.RoundToInt ((Abs (position.x-
   targetPosition.x) + Abs (position.y-targetPosition.y) + Abs
   (position.z-targetPosition.z))* scale);}
10. else if (heuristic == Heuristic.DiagonalManhattan) {
11. int xDistance = Abs (position.x-targetPosition.x);
12.  int zDistance = Abs (position.z-targetPosition.z);
13.  if (xDistance > zDistance) {
14.  nodeR.h = (uint)(14*zDistance + 10*(xDistance-
     zDistance))/10;}
15.  else {nodeR.h = (uint)(14*xDistance + 10*(zDistance-
     xDistance))/10;}
16.  nodeR.h = (uint)Mathfx.RoundToInt (nodeR.h * scale);}
17.  get {return nodeR.g+nodeR.h;}
18. public
19. virtual
20. void UpdateG () {
21. g = parent.g+cost+penalty;
22. for (int i=0;i<connections.Length;i++) {
23. if (connections[i].parent == this && connections[i].pathID
    == pathID) {
24. connections[i].UpdateAllG (open);
25. }
26. }
27. }
28. public virtual int[] InitialOpen (BinaryHeap open, Int3
    targetPosition, Int3 position, Path path, bool doOpen) {
29. return BaseInitialOpen
    (open,targetPosition,position,path,doOpen);
30. }
31. public int[] BaseInitialOpen (BinaryHeap open, Int3
    targetPosition, Int3 position, Path path, bool doOpen) {
32. if (connectionCosts == null) {
33. return null;
```

```
34. }
35. int[] costs = connectionCosts;
36. connectionCosts = new int[connectionCosts.Length];
37. for (int i=0;i<connectionCosts.Length;i++) {
38. connectionCosts[i] = (connections[i].position-
    position).costMagnitude;
39. }
40. if (!doOpen) {
41. for (int i=0;i<connectionCosts.Length;i++) {
42. Node other = connections[i];
43. for (int q = 0;q < other.connections.Length;q++) {
44. if (other.connections[q] == this) {
45. other.connectionCosts[q] = connectionCosts[i];
46. break;
47. }
48. }
49. }
50. }
51. //int[] tmp = connectionCosts;
52. //Should we open the node and reset the distances after
    that or only calculate the distances and don't reset them
53. if (doOpen) {
54. Open (open,targetPosition,path);
55. connectionCosts = costs;
56. /*for (int i=0;i<connectionCosts.Length;i++) {
57. for (int q = 0;q < connections[i].connections.Length;q++) {
58. if (connections[i].connections[q] == this) {
59. connections[i].connectionCosts[q] = connectionCosts[i];
60. break;
61. }
62. }
63. }*/
64. }
65. return costs;
66. }
```

a. *Flow Graph* Algoritma A*

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



Gambar IV.8 *Flow Graph* Algoritma A*

Keterangan:

 = Menggambarkan kondisi

 = Menggambarkan aksi

b. *Cyclomatic Complexity* V(G)

Cyclomatic complexity akan digunakan dalam menguji path yang nantinya akan digunakan untuk mencari jumlah path dalam satu dan juga sebagai penentuan jalur atau path pencarian NPC terhadap player. Apabila menggunakan basis path, nilai yang dihitung untuk Cyclomatic complexity menentukan jalur indpenden dalam basis set suatu program dan memberi batas atas untuk jumlah uji coba yang harus dikerjakan bahwa seluruh perintah sekurang-kurangnya telah dikerjakan sekali.

Berikut ini adalah perhitungan Cyclomatic complexity yang nantinya akan digunakan sebagai jumlah aksi pada independent path.

E = 17, N = 14

V(G) = E – N + 2

V(G) = 17 – 14 + 2

V(G) = 3 + 2 = 5

Keterangan

E = Jumlah aksi

N = Jumlah Kondisi

c. *Independent Path*

Independent Path adalah jalur yang melintasi atau melalui program dimanan sekurang-kurangnya terdapat proses perintah yang baru atau kondisi yang baru.

Path 1 = 1 - 2 - 15

Path 2 = 1 - 3 - 4 – 5 - 15

Path 3 = 1 - 3 - 4 – 6 – 7 - 15

Path 4 = 1 – 3 – 4 – 6 – 8 – 9 – 10 – 13 – 14 - 15

Path 5 = 1 – 3 – 4 – 6 – 8 – 9 – 10 – 12 – 14 – 15

d. *Graph Matriks* Algoritma A*

Graph metrik ini akan membantu uji coba independent path. Karena matrik persegi empat ini mempunyai ukuran yang sama dengan jumlah node pada flowgraph. Masing-masing baris dan kolom mempunyai hubugan dengan node yang telah ditentukan.

Graph matriks untuk metode procedural content generation dapat dilihat pada Tabel IV.3 di bawah ini.:

Tabel IV.3 *Graph Matriks* Algoritma A*

N	1	2	3	4	5	6	7	8	9	10	11	12	13	14	N(E)-1
1		1	1												2-1=1
2														1	1-1=0
3				1											1-1=0
4					1	1									2-1=1
5														1	1-1=0
6							1	1							2-1=1
7														1	1-1=0
8									1						1-1=0
9										1					1-1=0
10											1	1			2-1=1
11													1		1-1=0
12													1		1-1=0
13														1	1-1=0
14															1-1=0
Sum(E)+1															4+1=5

Keterangan:

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

e. Hasil Uji Algoritma A*

Tabel IV.4 Hasil UjiAlgoritma A*

No	Kasus uji	Yang diharapkan	Hasil Sesuai Kasus Uji	Keterangan
1	- `Conections == null` (posisi simpul berada diluar *path*)	Tidak dilakukan pencarian F	Rute tidak ditemukan	[√] Alur terlewati [ ] Alur tidak terlewati
2	- `Conections == null dan heuristic == Heurictic.none` (posisi simpul berada ditemukan)	Mencari F dengan nilai heuristik ditemukan	Rute ditemukan	[√] Alur terlewati [ ] Alur tidak terlewati
3	- `Connections ≠ null dan heurictic == Heurictic.Manhattan` (posisi *node*terhadap simpul terpilih)	Mencari F dengan nilai H menggunakan *manhattandistance*	Mencari F dengan nilai H menggunakan *manhattandistance*	[√] Alur terlewati [ ] Alur tidak terlewati
4	- `Connections ≠ null heuristic == Heuristic.DiagonalManhattan` - `int xDistance = Abs (position.x- targetPosition.x); int zDistance = Abs (position.z- targetPosition.z)` (posisi *node* berada di *openlist*)	Masukkan semua *neighbor node* ke dalam open dan hitung nilai F dan H.	*Current node* dan nilai F,G,H masuk ke dalam open	[√] Alur terlewati [ ] Alur tidak terlewati
5	- `connections ≠ null, nodeR.h = (uint)(14*xDistance + 10*(zDistance- xDistance))/10` - `nodeR.h = uint)Mathfx.RoundToInt (nodeR.h * scale)` - `get return nodeR.g+nodeR.h`	Nilai G yang didapat lebih kecil dari nilai G sebelumnya. Update ulang nilai F,G,H	Nilai F,G,H, diupdate ulang	[√] Alur terlewati [ ] Alur tidak terlewati

f. Kesimpulan

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

IV.2.4 Kuesioner

Kuesioner ini disebarkan kepada 30 orang yang berusia antara 13 keatas, dari hasil kuesioner tersebut akan dilakukan perhitungan agar dapat diambil

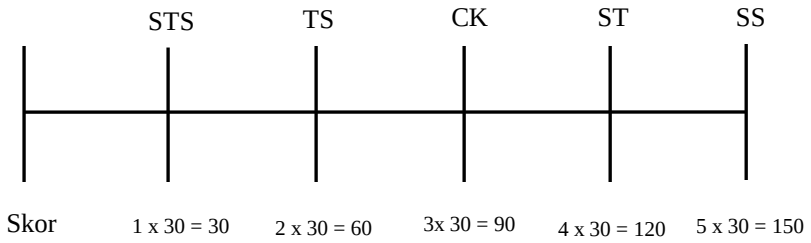
kesimpulan terhadap jawaban responden tentang penilaian sistem yang telah dibangun. Kuesioner terdiri dari 10 pernyataan dengan menggunakan sistem penskoran dengan skala pengukuran yang digunakan adalah skala likert, berikut skor untuk jawaban kuesioner

Tabel IV.4 Skor Jawaban Pada Kuesioner

Skala Jawaban	Keterangan	Skor
SS	Sangat Setuju	5
ST	Setuju	4
CK	Cukup	3
TS	Tidak Setuju	2
STS	Sangat Tidak Setuju	1

Data yang diperoleh dari pemberian kuesioner kepada responden dapat dianalisis dengan menghitung rata-rata jawaban berdasarkan skor yang didapat dari responden. Analisis yang dapat dilakukan adalah sebagai berikut.

Jumlah skor ideal (kriterium) untuk seluruh item = 5 x jumlah responden Rata-rata presentase = (jumlah skor : jumlah skor ideal) x 100%. Jumlah skor ideal = 4 x 30 = 120 secara kontinum dapat digambarkan sebagai berikut:



Berikut ini adalah daftar pertanyaan yang akan diajukan kepada responden :

1. Apakah anda setuju bahwa game ini bersifat menghibur?
2. Apakah anda merasa jenuh ketika memainkan game ini?
3. Apakah anda setuju tentang informasi yang disampaikan tentang beberapa bangunan bersejarah yang ada di Indonesia cukup lengkap?

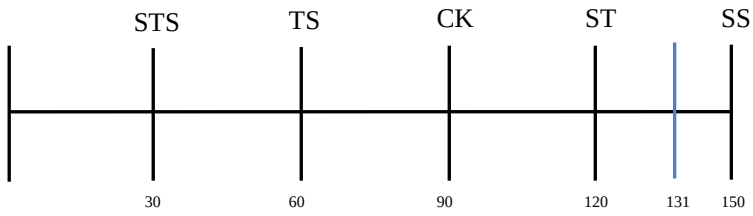
4. Apakah anda setuju game Takedown The Terrorists bisa melatih ketangkasan dan pola pikir dalam menyelesaikan permasalahan yang adad di dalam game?
5. Apakah anda setuju bahwa penyampaian informasi tentang gedung bersejarah di dalam game ini mudah diingat?
6. Apakah anda setuju game Takedown The Terrorists mudah untuk dimainkan?
7. Apakah anda setuju antarmuka di dalam game ini cukup baik?
8. Apakah anda setuju grafik yang disajikan dalam game cukup baik?
9. Apakah anda setuju fitur yang ada di dalam game ini cukup lengkap?
10. Apakah anda terpacu untuk memainkan game ini kembali?

Berikut ini adalah hasil skor masing-masing jawaban yang sudah dihitung nilainya dengan menggunakan rumus diatas.

1. Apakah anda setuju bahwa *game* ini bersifat menghibur?

Tabel IV.6 Perhitungan persentase pertanyaan ke-1

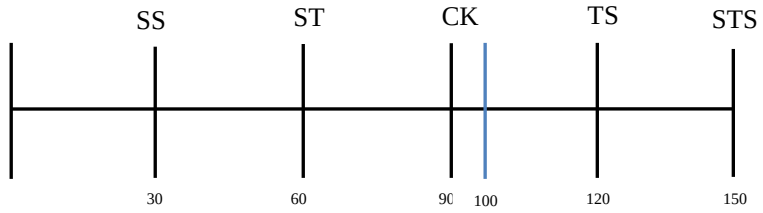
Pertanyaan	Keterangan	Skor	Frekuensi	S
1.	Sangat setuju	5	6	24
	Setuju	4	20	60
	Cukup	3	3	9
	Tidak setuju	2	1	2
	Sangat tidak setuju	1	0	0
Jumlah			30	131



2. Apakah anda merasa jenuh ketika memainkan *game* ini?

Tabel IV.7 Perhitungan persentase pertanyaan ke-2

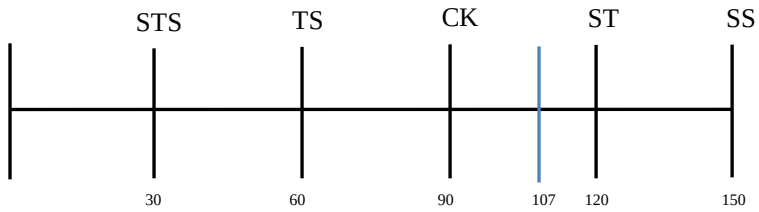
Pertanyaan	Keterangan	Skor	Frekuensi	S
2.	Sangat setuju	1	0	0
	Setuju	2	6	12
	Cukup	3	13	39
	Tidak setuju	4	6	24
	Sangat tidak setuju	5	5	25
Jumlah			30	100



3. Apakah anda setuju tentang informasi yang disampaikan tentang beberapa bangunan bersejarah yang ada di Indonesia cukup lengkap?

Tabel IV.8 Perhitungan persentase pertanyaan ke-3

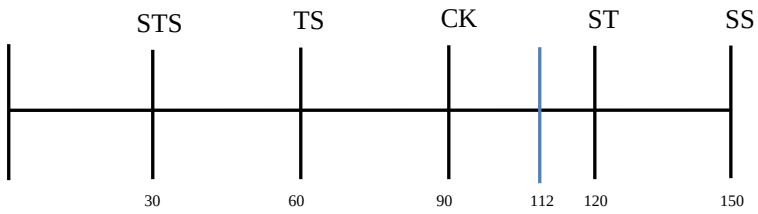
Pertanyaan	Keterangan	Skor	Frekuensi	S
3.	Sangat setuju	5	1	5
	Setuju	4	19	76
	Cukup	3	6	18
	Tidak setuju	2	4	8
	Sangat tidak setuju	1	0	0
Jumlah			30	107



4. Apakah anda setuju game *Takedown The Terrorists* bisa melatih ketangkasan dan pola pikir dalam menyelesaikan permasalahan yang adad di dalam game?

Tabel IV.9 Perhitungan persentase pertanyaan ke-4

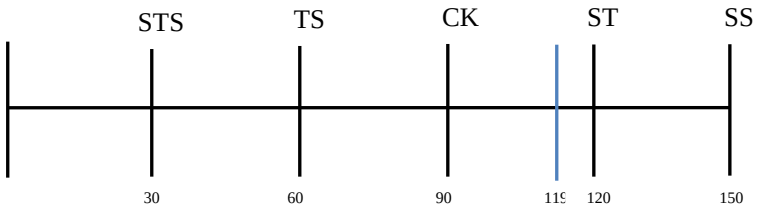
Pertanyaan	Keterangan	Skor	Frekuensi	S
4.	Sangat setuju	5	2	10
	Setuju	4	20	80
	Cukup	3	7	21
	Tidak setuju	2	1	2
	Sangat tidak setuju	1	0	0
Jumlah			30	112



5. Apakah anda setuju bahwa penyampaian inforrmasi tentang gedung bersejarah di dalam *game* ini mudah diingat

Tabel IV.10 Perhitungan persentase pertanyaan ke-5

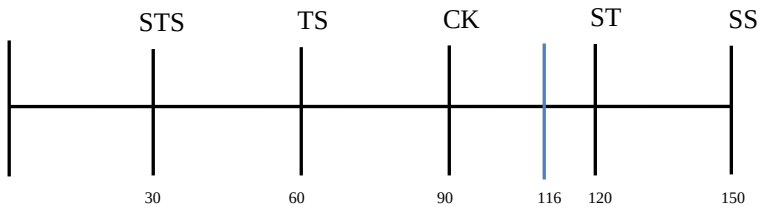
Pertanyaan	Keterangan	Skor	Frekuensi	S
5	Sangat setuju	5	3	15
	Setuju	4	24	96
	Cukup	3	2	6
	Tidak setuju	2	1	2
	Sangat tidak setuju	1	0	0
Jumlah			30	119



6. Apakah anda setuju *game Takedown The terrorists* Mudah Dimainkan?

Tabel IV.11 Perhitungan persentase pertanyaan ke-6

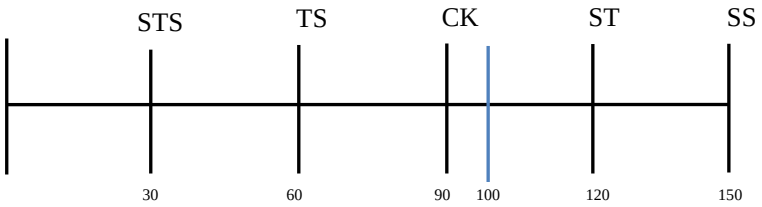
Pertanyaan	Keterangan	Skor	Frekuensi	S
6	Sangat setuju	5	3	15
	Setuju	4	22	88
	Cukup	3	3	9
	Tidak setuju	2	2	4
	Sangat tidak setuju	1	0	0
Jumlah			30	116



7. Apakah anda setuju antarmuka di dalam *game* ini cukup baik?

Tabel IV.12 Perhitungan persentase pertanyaan ke-7

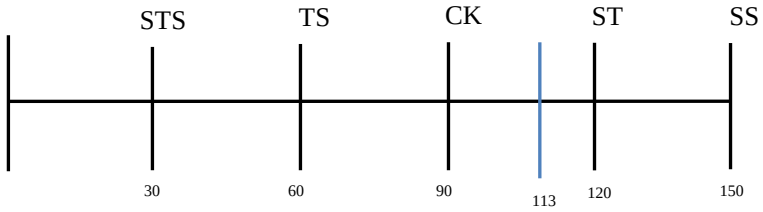
Pertanyaan	Keterangan	Skor	Frekuensi	S
7	Sangat setuju	5	0	0
	Setuju	4	19	76
	Cukup	3	11	33
	Tidak setuju	2	0	0
	Sangat tidak setuju	1	0	0
Jumlah			30	99



8. Apakah anda setuju grafik yang disajikan dalam *game* ini cukup baik?

Tabel IV.13 Perhitungan persentase pertanyaan ke-8

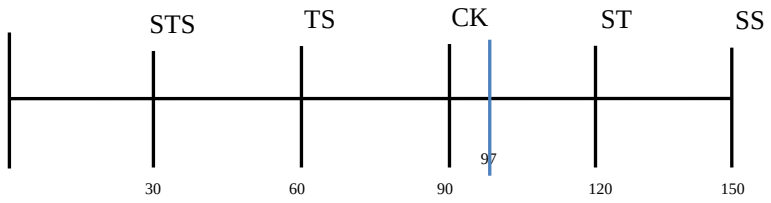
Pertanyaan	Keterangan	Skor	Frekuensi	S
8	Sangat setuju	5	2	10
	Setuju	4	19	76
	Cukup	3	9	27
	Tidak setuju	2	0	0
	Sangat tidak setuju	1	0	0
Jumlah			30	113



9. Apakah anda setuju fitur yang ada di dalam game ini cukup lengkap?

Tabel IV.14 Perhitungan persentase pertanyaan ke-9

Pertanyaan	Keterangan	Skor	Frekuensi	S
9	Sangat setuju	5	1	5
	Setuju	4	14	56
	Cukup	3	12	36
	Tidak setuju	2	3	6
	Sangat tidak setuju	1	0	0
Jumlah			30	97



10. Apakah anda terpacu untuk memainkan game ini kembali?

Tabel IV.15 Perhitungan persentase pertanyaan ke-10

Pertanyaan	Keterangan	Skor	Frekuensi	S
10	Sangat setuju	5	2	10
	Setuju	4	21	84
	Cukup	3	6	18
	Tidak setuju	2	1	2
	Sangat tidak setuju	1	0	0
Jumlah			30	110

