BAB 4
IMPLEMENTASI DAN PENGUJIAN

4.1 Implementasi

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 Sejarah Medan Area dengan *file extension*-nya *.exe.*

4.1.1 Implementasi Perangkat Keras

Perangkat keras yang dapat digunakan untuk menjalankan aplikasi *game* ini harus memiliki spesifikasi yang cukup tinggi. Sebagai ilustrasi sistem minimum agar dapat menjalankan *game* Sejarah Medan Area adalah sebagai berikut :

1. Processor minimal kecepatan 2.0 Ghz
2. RAM 2 GB
3. VGA 1 GB
4. Harddisk 10 GB
5. *Mouse* dan *Keyboard*

4.1.2 Implementasi Perangkat Lunak

Adapun spesifikasi perangkat lunak yang dibutuhkan untuk menjalankan aplikasi Sejarah Perjuangan Medan Area adalah :

1. Sistem operasi Windows
2. *Graphic card*
3. DirectX 9 keatas

4.1.3 Implementasi Instalasi Aplikasi Game

Untuk dapat memainkan *game* Sejarah Perjuangan Medan Area, pemain terlebih harus mempunyai *file* Sejarah Medan Area.*exe*. Untuk mulai memainkan

game, pemain hanya perlu menjalankan *file* Sejarah Medan Area.*exe* yang telah ada dalam perangkat keras.

4.1.4 Implementasi Kelas

Pada tahap pembangunan sistem, kelas-kelas yang dirancang pada tahap analisis akan diimplementasikan sesuai dengan kebutuhan pada tahap analisis. Pembangunan *game* FPS 3D Sejarah Medan Area ini dibangun menggunakan aplikasi *untiy*3.5 menggunakan bahasa *JavaScirpt* dan *C#.* Implementasikelasnya adalah sebagai berikut.

4.1.5 Implementasi Antarmuka

Antarmuka dapat membuat pengguna merasa nyaman dan bertahan lama berada dalam aplikasi atau sebaliknya. Berikut ini merupakan antarmuka yang ada pada aplikasi *game* Sejarah Medan Area.

1. Antarmuka Menu Utama

Saat pengguna membuka aplikasi pertama kali, maka antarmuka yang ditampilkan oleh sistem adalah antarmuka menu utama. Berikut ini merupakan tampilan dari antarmuka menu utama :



Gambar 4.1 Antarmuka Menu Utama

Gambar 4.1 di atas merupakan menu utama pada *game* Sejarah Medan Area. Terdapat 3 (tiga) sub menu di dalamnya, yaitu menu Mulai yang berfungsi untuk memulai permainan, menu Petunjuk yang berfungsi untuk melihat informasi kontrol yang digunakan ketika memainkan *game*, dan menu Keluar yang berfungsi untuk keluar dari aplikasi.

2. Antarmuka Menu Mulai

Saat pengguna memilih menu Mulai, maka antarmuka yang ditampilkan oleh sistem adalah antarmuka *scene video* berupa *video* cerita awal dari terjadinya pertempuran di dalam *game* pada *stage* pertama. Selanjutnya akan masuk ke antarmuka petunjuk misi dan kemudian akan masuk pada antarmuka permainan. Jika seluruh misi telah terselesaikan maka akan masuk ke antarmuka *scene stage* 1 berupa *video* dan kemudian masuk ke antarmuka petunjuk misi. Berikut ini merupakan tampilan dari antarmuka menu mainkan.



Gambar 4.2 Antarmuka *scene video stage* 1

Gambar 4.2 merupakan antarmuka *scene video* pada *stage* pertama yang terdapat pada *game* Sejarah Medan Area.



Gambar 4.3 Antarmuka Petunjuk misi

Gambar 4.3 merupakan antarmuka petunjuk misi yang terdapat pada *game* Sejarah Medan Area.



Gambar 4.4 Antarmuka *stage* 1 awal mulai game

Gambar 4.4 merupakan antarmuka permainan yang terdapat pada *game* Sejarah Medan Area. Gambar tersebut memperlihatkan posisi pemain ketika masuk ke dalam permainan.



Gambar 4.5 Antarmuka permainan di depan jalan Bali.

Gambar 4.5 menunjukan posisi pemain ketika akan menghadapi musuh di daerah jalan Bali.

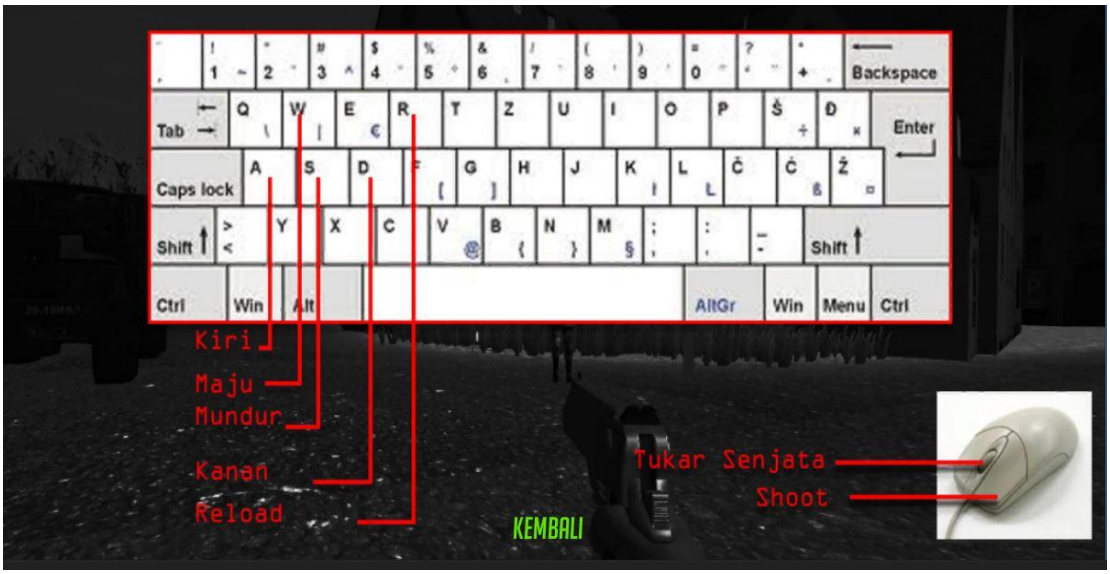


Gambar 4.6 Antarmuka Permainan di *stage* 2 didepan gardu

Gambar 4.6 menunjukan posisi pemain ketika berada di depan gardu fixed bounderius medan area.

3. Antarmuka Menu Petunjuk

Saat pengguna memilih menu petunjuk, maka antarmuka yang ditampilkan oleh sistem adalah antarmuka informasi kontrol yang digunakan pada *game* Sejarah Perjuangan Medan Area.



Gambar 4.7 Antarmuka menu Petunjuk.

4.2 Pengujian

Pada umunya ada 2 (dua) macam teknik pengujian yaitu *white box testing* dan *black box testing*. Pengujian yang menggunakan teknik *white box testing* lebih menitikberatkan pada struktur internal (*source code*) program, dimana setiap baris kode yang akan di*compile* atau 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* Sejarah Perjuangan Medan Area digunakan pengujian *black box*

testing. Cara pengujian dilakukan melalui dua tahap, yaitu pengujian *alpha* (fungsional) dan pengujian *beta*.

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

Skenario pengujian menjelaskan pengujian terhadap sistem yang ada pada aplikasi *game* Sejarah Perjuangan Medan Area. Skenario pengujian yang akan dilakukan pada aplikasi ini selengkapnya dapat dilihat pada table 4.1:

Tabel 4.1 Skenario Pengujian Aplikasi

No	Komponen yang di uji	Skenario Pengujian	Jenis Pengujian
1.	Menu Mainkan	Klik menu Mainkan	*Black box*
2.	Menu Petunjuk	Klik menu Mainkan	*Black box*
		Klik menu Kembali	*Black box*
3.	Menu Keluar	Klik menu Keluar	*Black box*
4.	Berjalan	Tekan tombol W *keyboard*	*Black box*
		Tekan tombol A *keyboard*	*Black box*
		Tekan tombol D *keyboard*	*Black box*
		Tekan tombol S *keyboard*	*Black box*
5	Mengganti Senjata	Menggunakan *Scroll* pada	*Black box*

No	Komponen yang di uji	Skenario Pengujian	Jenis Pengujian
		mouse	
6.	Mengambil obat	Mengarahkan karakter hingga menyentuh *icon* obat menggunakan *mouse* dan *keyboard*	*Black box*
7.	Mengambil granat	Mengarahkan karakter hingga menyentuh *icon granat* menggunakan *mouse* dan *keyboard*	*Black box*
8.	Menembak	Mengklik tombol kiri pada mouse.	*Black box*
9.	Melempar granat	Menekan tombol kiri pada mouse	*Black box*
10.	Memberhentikan Permainan	Menekan tombol ESC pada *keyboard*	*Black box*
11.	Mengambil petunjuk	Mengarahkan karakter hingga menyentuh *icon* intruksi menggunakan *mouse* dan *keyboard*	*Black box*
12.	Algortima A* (A *star*)	Pencarian rute atau jalan terbaik untuk NPC	*White Box*
13.	Algoritma *Fuzzy Logic*	Perilaku NPC	*White 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. Hasil pengujian yang akan dilakukan pada aplikasi ini selengkapnya dapat dilihat pada tabel 4.2

Tabel 4.2 Skenario Pengujian Black Box

No	Kasus yang akan di uji	Skenario dan Hasil Uji		
		Skenario Uji	**Hasil yang Diharapkan**	**Kesimpulan**
1.	Menu Mulai	Klik Menu Mulai	Sistem menampilkan halaman *Scene Video*	[√] Berhasil [ ] Tidak Berhasil
2.	Menu Petunjuk	Klik Menu Petunjuk	Sistem menampilkan petunjuk berupa informasi kontrol yang digunakan dalam *game*	[√] Berhasil [ ] Tidak Berhasil
		Klik Kembali	Sistem menampilkan menu utama	[√] Berhasil [ ] Tidak Berhasil
3.	Menu	Klik Menu	Menutup atau	[√] Berhasil

No	Kasus yang akan di uji	Skenario dan Hasil Uji		
		Skenario Uji	**Hasil yang Diharapkan**	**Kesimpulan**
	Keluar	Keluar	Keluar dari aplikasi	[ ] Tidak Berhasil
4.	Berjalan	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
5.	Mengganti senjata	Mengunakan scroll pada mouse	Senjata berganti, dari pistol ke senapan, dari senapan ke pistol, berulang-ulang.	[√] Berhasil [ ] Tidak Berhasil

No	Kasus yang akan di uji	Skenario dan Hasil Uji		
		Skenario Uji	**Hasil yang Diharapkan**	**Kesimpulan**
6.	Mengambil obat	Mengarahkan karakter hingga menyentuh *icon* obat menggunakan *mouse* dan *keyboard*	*Bar* darah pada permainan akan bertambah jika darah pemain memang tidak penuh dan *icon* obat menghilang	[√] Berhasil [ ] Tidak Berhasil
7.	Mengambil granat	Mengarahkan karakter hingga menyentuh *icon* granat menggunakan *mouse* dan *keyboard*	Jumlah granat pada permainan akan bertambah dan *icon* granat menghilang	[√] Berhasil [ ] Tidak Berhasil
8.	Menembak	Mengarahkan *pointer* / *cursor* senjata menggunakan *keyboard* dan *mouse* lalu menekan tombol kiri	Senjata yang dipegang karakter pada *game* akan mengeluarkan peluru dan suara tembakan secara bersama-sama	[√] Berhasil [ ] Tidak Berhasil

No	Kasus yang akan di uji	Skenario dan Hasil Uji		
		Skenario Uji	**Hasil yang Diharapkan**	**Kesimpulan**
		mouse untuk mengeluarkan peluru / *mouse*		
9.	Melempar granat	Memakai senjata granat kemudian mengarahkan *pointer* / *cursor* senjata menggunakan *keyboard* dan *mouse* lalu menekan tombol kiri *mouse* untuk membuang granat / *mouse*	Granat di lempar dan akan menghasilkan ledakan	[√] Berhasil [ ] Tidak Berhasil
10.	Memberhenti-kan permainan	Menekan tombol ESC pada *keyboard*	*Game* akan mengeluarkan *screen pause*	[√] Berhasil [ ] Tidak Berhasil
11.	Mengambil	Mengarahkan	*Game* akan	[√] Berhasil

No	Kasus yang akan di uji	Skenario dan Hasil Uji		
		Skenario Uji	**Hasil yang Diharapkan**	**Kesimpulan**
	intruksi petunjuk	karakter hingga menyentuh *icon* intruksi menggunakan *mouse* dan *keyboard*	menampilkan *pop up* berupa teks dan suara	[ ] Tidak Berhasil

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

1. Pengujian *White Box* Algoritma A*

Pengujian algiritma A* merupakan proses dimana akan diuji terhadap penerapan implementasi A* terhadap objek musuh dalam melakukan pencarian karakter. Berikut ini merupakan algoritma A* yang penerapannya menggunakan bahasa pemrograman C# :

Berikut ini merupakan pengujian algoritma A star.

```
1. startNode = new PathNode(null, endNode, startTile, 0);
2.                addNodeToOpenList(startNode);
3.           while (openList.Count > 0)
        {
      PathNode currentNode  =  openList[openList.Count 1];
4.      if (currentNode.IsEqualToNode(endNode))
                        {
                            List<Vector2> bestPath = new
      List<Vector2>();
```

```
5.                          while (currentNode != null)
         {
           bestPath.Insert(0,currentNode.GridLocation);
      currentNode = currentNode.ParentNode;
                           }
6.                        return bestPath;
                        }
                          openList.Remove(currentNode);
                 nodeCosts.Remove(currentNode.GridLocation);
       foreach  (PathNode  possibleNode  in
      findAdjacentNodes(currentNode, endNode))
                        {
7.      if
      (nodeStatus.ContainsKey(possibleNode.GridLocation))
         {
8.   if
             (nodeStatus[possibleNode.GridLocation]
      == NodeStatus.Closed)
      {
         continue;
                     }
9.       if
      (nodeStatus[possibleNode.GridLocation]
      == NodeStatus.Open)
        {
10.   if
     (possibleNode.TotalCost  >=
      nodeCosts[possibleNode.GridLocation])
      {
       continue;
         }
        }
       }
```

```
11.     addNodeToOpenList(possibleNode);
        }
12.             nodeStatus[currentNode.GridLocation]  =
        NodeStatus.Closed;
       }
```

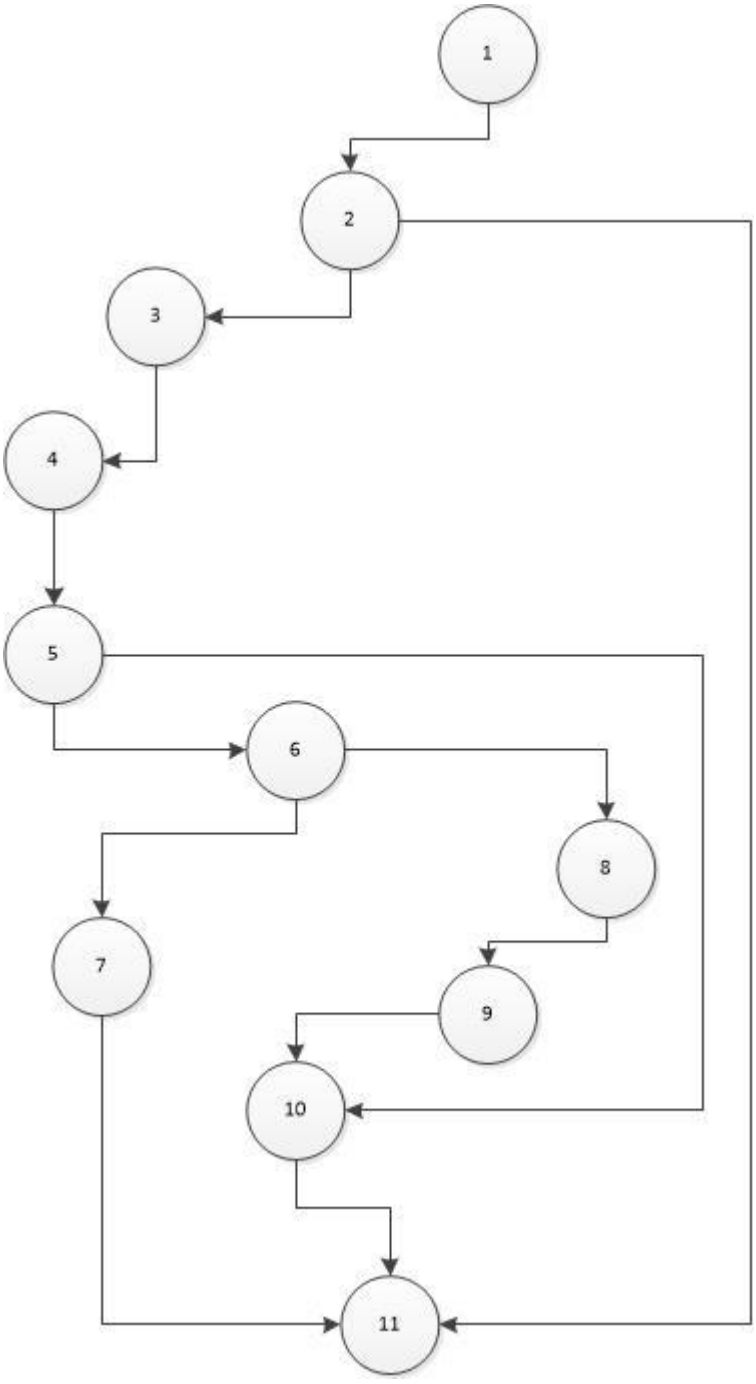
b. *Flow Graph* Algoritma A*

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

Keterangan:

 = Menggambarkan kondisi

 = Menggambarkan aksi



Gambar 4.8 *Flow Graph* Algoritma A*

c. *Cyclomatic Complexity* V(G)

Cyclomatic complexity akan digunakan dalam menguji path yang nantinya akan digunakan untuk implementasi A* dalam mencari rute 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 = 14, N = 12

V(G) = E – N + 2

V(G) = 14 – 12 + 2

V(G) = 2 + 2 = 4

Keterangan

E = Jumlah aksi

N = Jumlah Kondisi

d. *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 – 11

Path 2 = 1 – 2 – 3 – 4 – 5 – 10 – 11

Path 3 = 1 – 2 – 3 – 4 – 5 – 6 – 8 – 9 – 10 – 11

Path 4 = 1 – 2 – 3 – 4 – 5 – 6 – 7 – 11

e. Graph Matriks

Tabel 4.3 Graph Matriks Hasil Uji Algoritma A*

	1	2	3	4	5	6	7	8	9	10	11
1		1									
2			1								1

3				1							
4					1						
5						1				1	
6							1	1			
7											1
8									1		
9										1	
10											1
11											

Keterangan :

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

f. Hasil Uji Algoritma A*

Tabel 4. 4 Hasil Uji Algoritma A*

No	Kasus uji	Yang diharapkan	Pengamatan	Kesimpulan
1	- `startNode = new PathNode (Null, endNode, startTile, 0);` - `addNodeToOpenList (startNode);` - `while (openList.Count > 0) {` - `nodeStatus[curren`	Rute pemain dapat ditelusuri	Menelusuri rute pemain	Path dikerjakan

No	Kasus uji	Yang diharapkan	Pengamatan	Kesimpulan
	tNode.GridLocation] NodeStatus.Closed; }			
2	- startNode = new PathNode (Null, endNode, startTile, 0); - addNodeToOpenList (startNode); - while (openList.Count > 0){ PathNode currentNode – openList[openList .Count – 1]; - if (currentNode.IsEqualToNode(endNode)) { List<Vector2> bestPath = new List<Vector2>(); - while (currentNode !=null) { bestPath.Insert(0,currentNode.GridLocation);	Rute pemain dapat ditelusuri	Menelusuri rute pemain	Path dikerjakan

No	Kasus uji	Yang diharapkan	Pengamatan	Kesimpulan

	currentNode = currentNode.ParentNode;}
	- return bestPath;
	}
	openList.Remove(currentNode);
	nodeCosts.Remove(currentNode.GridLocation); foreach (PathNode possibleNode in findAdjacentNodes (currentNode, endNode)){
	- if(nodeStatus.ContainsKey(possibleNode.GridLocation)){
	- if (nodeStatus[possibleNode.GridLocation==NodeStatus.Closed{
	continue;
	}
	- (nodeStatus[possibleNode.GridLocation]==NodeStatus.O			

No	Kasus uji	Yang diharapkan	Pengamatan	Kesimpulan
	pen {			

3	- startNode = new PathNode (Null, endNode, startTile, 0);
	- addNodeToOpenList (startNode);
	- while (openList.Count > 0){
	PathNode currentNode – openList[openList .Count – 1];
	- if (currentNode.IsEqualToNode(endNode))
	{
	List<Vector2> bestPath = new List<Vector2>();
	- while (currentNode !=null)
	{
	bestPath.Insert(0 ,currentNode.GridLocation);
	currentNode = currentNode.Paren
		Rute pemain dapat ditelusuri	Menelusuri rute pemain	Path dikerjakan

No	Kasus uji	Yang diharapkan	Pengamatan	Kesimpulan

	tNode;}
	- return bestPath;
	}
	openList.Remove(currentNode);
	nodeCosts.Remove(currentNode.GridLocation); foreach (PathNode possibleNode in findAdjacentNodes (currentNode, endNode)){
	- if(nodeStatus.ContainsKey(possibleNode.GridLocation)){
	- if (nodeStatus[possibleNode.GridLocation]==NodeStatus.Open){
	- if (possibleNode.TotalCost >={
	continue;
	}
	}
	}
	- addNodeToOpenList (possibleNod			

No	Kasus uji	Yang diharapkan	Pengamatan	Kesimpulan
	e);} - nodeStatus[currentNode.GrildLocation] NodeStatus.Closed;}			

4	- startNode = new PathNode (Null, endNode, startTile, 0);
	- addNodeToOpenList (startNode);
	- while (openList.Count > 0){
	PathNode currentNode – openList[openList.Count – 1];
	- if (currentNode.IsEqualToNode(endNode))
	{
	List<Vector2> bestPath = new List<Vector2>();
	- while (currentNode
		Rute pemain tidak dapat ditelusuri	Rute pemain tidak ditemukan	Path dikerjakan

No	Kasus uji	Yang diharapkan	Pengamatan	Kesimpulan

	!=null)
	{
	bestPath.Insert(0,currentNode.GridLocation);
	currentNode = currentNode.ParentNode;}
	- return bestPath;
	}
	openList.Remove(currentNode);
	nodeCosts.Remove(currentNode.GridLocation); foreach (PathNode possibleNode in findAdjacentNodes (currentNode, endNode)){
	- addNodeToOpenList (possibleNode);}
	- nodeStatus[currentNode.GrildLocation] NodeStatus.Closed;}			

g. 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

2. Pengujian White Box Algoritma Fuzzy Logic

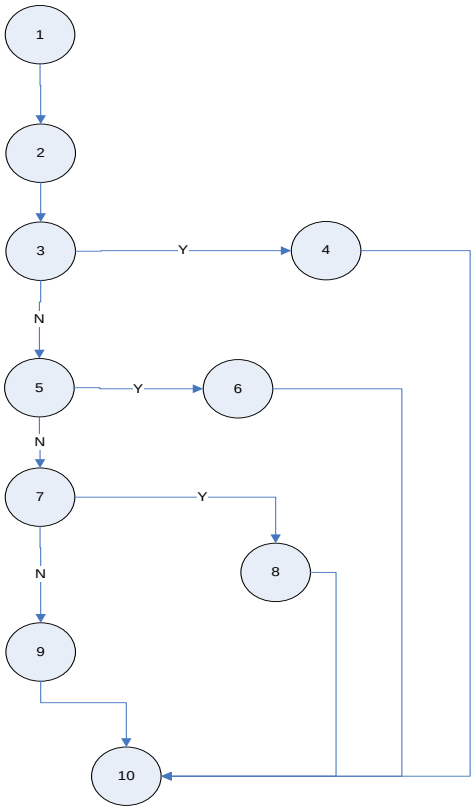
Berikut ini merupakan algoritma *fuzzy logic* yang penerapannya terhadap tingkah laku NPC menggunakan bahasa pemrograman C#:

```
1.  jarak = (target.position -
    myTransform.position).magnitude;
2.  rules rule = new rules(jarak,hitPoints);
    defuzzyfikasi def = new defuzzyfikasi();
    rule.periksaRule();
    def. hasilDefuzzy();
3.  if((balikKeMarkas.position -
    myTransform.position).magnitude < 1.0){
4.  StartCoroutine(tambahDarah());
    }
5.  else if(def.hasilDefuzzyfikasi <= 10.0f) {
6.  balikKeMarkas();
    myTransform.position += myTransform.forward * moveSpeed
    *Time.deltaTime;
    anim.animation.CrossFade(walkAnim);
    playWalkSounds();
    }
7.  else if(def.hasilDefuzzyfikasi > 10.0f &&
    def.hasilDefuzzyfikasi <= 20.0f){
8.  bertahan();
    anim.animation.CrossFade(attackAnim2);
    audio.Stop();
```

```
    }
9.  else if(def.hasilDefuzzyfikasi > 20.0f &&
    def.hasilDefuzzyfikasi <= 30.0f){
10. menyerang();
    damage = Random.Range(20, 30);
    anim.animation.CrossFade(walkAnim);
    }
```

a. *Flow Graph* Algoritma *Fuzzy Logic*

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



Gambar 4.9 *Flow Graph* Algoritma *Fuzzy Logic*

Keterangan:



= Menggambarkan kondisi

 = Menggambarkan aksi

b. *Cyclomatic Complexity* V(G)

E = 12, N = 10

V(G) = E – N + 2

V(G) = 12 – 10 + 2

V(G) = 2 + 2 = 4

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 = 1-2-3-5-7-9-10

Path 2 = 1-2-3-4-10

Path 3 = 1-2-3-5-6-10

Path 4 = 1-2-3-5-7-8-10

d. Hasil Uji Algoritma *Fuzzy Logic*

Tabel 4.5 Hasil Uji Algoritma *Fuzzy Logic*

No	Kasus Uji	Yang diharapkan	Pengamatan	Kesimpulan
1	- balikKeMarkas.position - myTransform.positio	Hasil defuzzy > 30 dan <= 40 NPC menyerang	NPC menyerang	Defuzzyfikasi dikerjakan

	n.m agnitude < 1.0 def.hasilDefuzzyfikasi <= 10.0f def.hasilDefuzzyfikasi > 10.0f && def.hasilDefuzzyfikasi <= 20.0f - def.hasilDefuzzyfikasi > 20.0f && def.hasilDefuzzyfikasi <= 30.0f - def.hasilDefuzzyfikasi > 30.0f && def.hasilDefuzzyfikasi <= 40.0f			
2	- balikKeMarkas.position myTransform.position.m agnitude < 1.0	NPC menambah hitpoints saat kembali ke markas	NPC menambah hitpoints	Defuzzyfikasi dikerjakan
3	- balikKeMarkas.position myTransform.position.m agnitude < 1.0	Hasil defuzzy <= 10 NPC melarikan diri ke markas	NPC melarikan diri ke markas	Defuzzyfikasi dikerjakan

	- def.hasilDefuzzyfikasi <= 10.0f int q = 0;q < other.connections.Length ; q++ other.connections[q] == this connections[i].walkabl e && connections[i].area != area !doOpen Int i=0;i<connectionCosts.			
4	- balikKeMarkas.position myTransform.position.m agnitude < 1.0 - def.hasilDefuzzyfikasi <= 10.0f - def.hasilDefuzzyfikasi > 10.0f && def.hasilDefuzzyfika	Hasil defuzzy > 10 dan <= 20 NPC bertahan	NPC bertahan	Defuzzyfikasi dikerjakan

	si <= 20.0f			

e. Kesimpulan

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

4.2.1.4 Kesimpulan Pengujian Alpha

Dari hasil pengujian yang dilakukan, dapat disimpulkan bahwa sistem sudah dapat berjalan sesuai dengan yang diharapkan beserta proses penanganan kesalahannya. Jadi, secara fungsional sistem sudah dapat menghasilkan keluaran yang diharapkan.

4.2.2 Pengujian *Beta*

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

4.2.2.1 Kuesioner

Kuesioner disebarkan menggunakan teknik *Purposive sampling* dimana anggota *sample* dari populasi dipilih berdasarkan pengalaman yang di miliki oleh responden dan hasil. Kuesioner disebarkan kepada 40 responden dengan usia 17 tahun (setara anak SMA kelas XII), dari hasil kuesioner tersebut akan dilakukan perhitungan agar dapat diambil kesimpulan terhadap penilaian penerapan 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, dapat di lihat pada tabel 4.7.

Tabel 4.6 Skor Jawaban Pada Kuesioner

Skala Jawaban	Keterangan	Skor
SS	Sangat Setuju	4
ST	Setuju	3
RG	Ragu-ragu	2
TS	Tidak Setuju	1

Data hasil kuisioner yang telah diberikan dapat dicari presentasinya dari masing-masing jawaban dengan menggunakan rumus :

$$P = \frac{S}{skorideal} x100\% \quad (4.1)$$

Keterangan:

P = Nilai presentase yang dicari

S = Jumlah frekuensi dikalikan dengan nilai yang di tetapkan jawaban

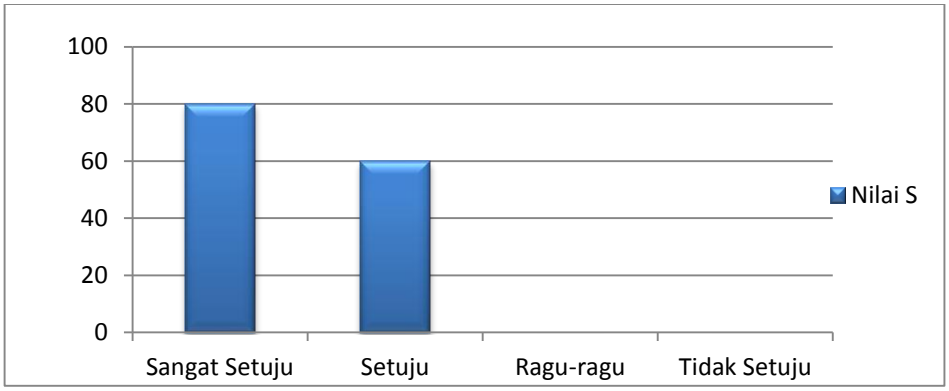
Skor ideal = Nilai tertinggi dikalikan dengan jumlah sampel

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

1. Apakah anda setuju sejarah Medan Area perlu untuk dipelajari dan diketahui?

Tabel 4.7 Tabel perhitungan persentase pertanyaan ke-1

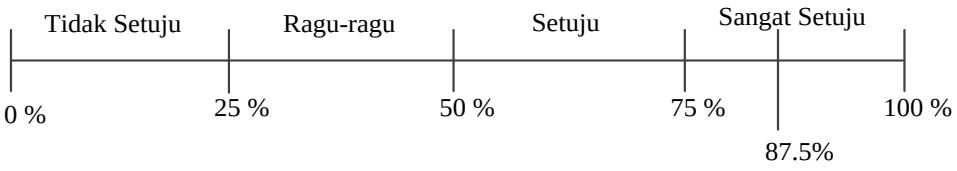
Pertanyaan	Keterangan	Skor	Frekuensi	S
1	Sangat setuju	4	20	80
	Setuju	3	20	60
	Ragu-ragu	2	0	0
	Tidak setuju	1	0	0
Jumlah			40	140



Gambar 4.10 Grafik skor pertanyaan ke-1

$$P = \frac{140}{160} x100 = 87.5\%$$

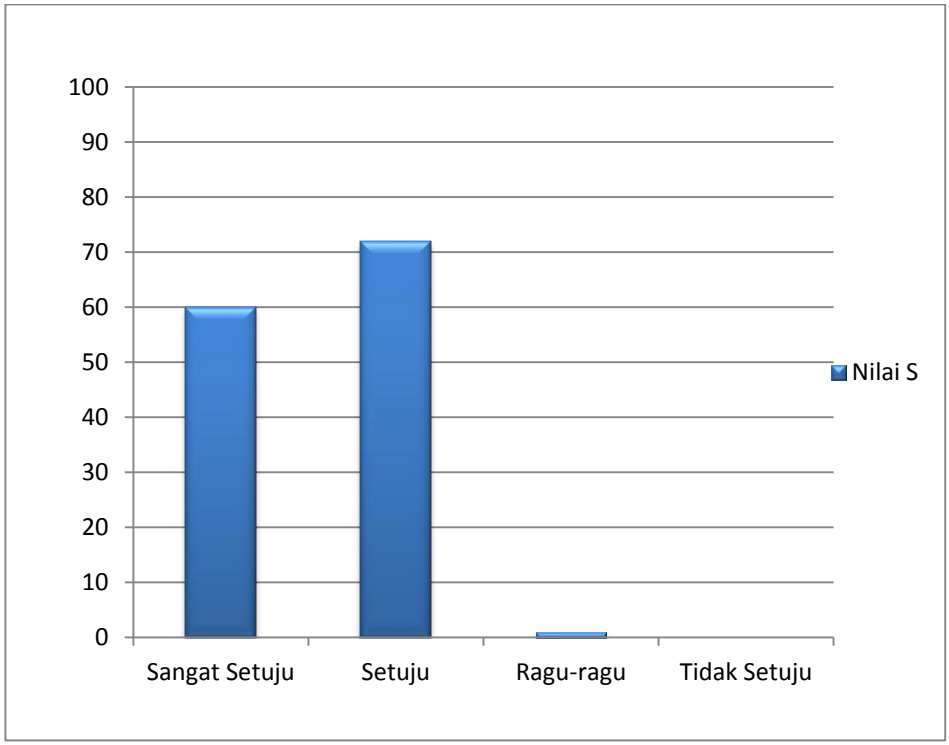
Berdasarkan hasil presentase nilai di atas, maka dapat disimpulkan bahwa penilaian terhadap pertanyaan Apakah anda setuju sejarah Medan Area perlu untuk dipelajari dan diketahui adalah 87.5% dari yang diharapkan 100%, atau dapat dikategorikan sangat setuju. Berikut kriteria interprestasi skor :



2. Apakah anda setuju media pembelajaran dalam bentuk game?

Tabel 4.8 Tabel perhitungan persentase pertanyaan ke-2

Pertanyaan	Keterangan	Skor	Frekuensi	S
2	Sangat setuju	4	15	60
	Setuju	3	24	72
	Ragu-ragu	2	1	2
	Tidak setuju	1	0	0
Jumlah			40	134

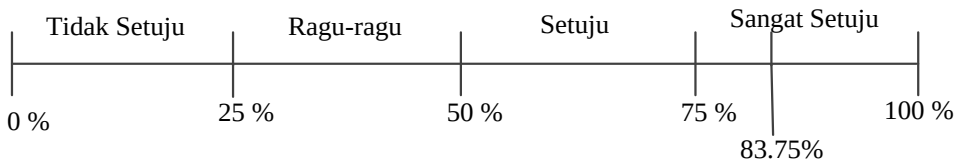


Gambar 4.11 Grafik skor pertanyaan ke-2

$$P = \frac{134}{160} x100 = 83.75\%$$

Berdasarkan hasil presentase nilai di atas, maka dapat disimpulkan media pembelajaran dalam bentuk game adalah 83.75% dari yang diharapkan

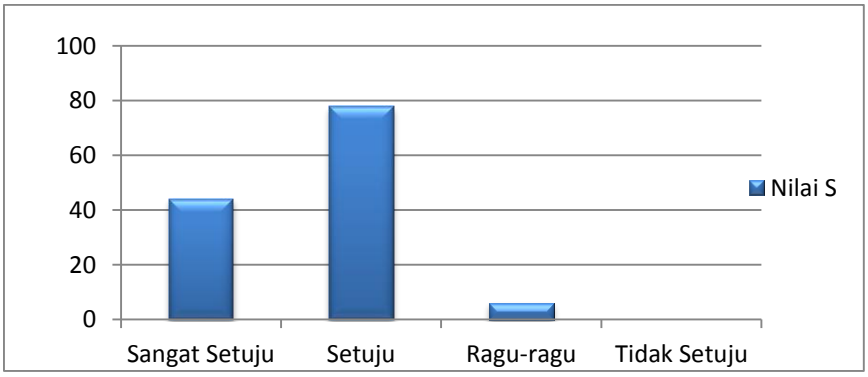
100%, atau dapat dikategorikan sebagai sangat setuju. Berikut kriteria interprestasi skor :



3. Bagaimana menurut anda materi yang disampaikan berupa nama tokoh, dan tempat peristiwa perang medan area?

Tabel 4.9 Tabel perhitungan persentase pertanyaan ke-3

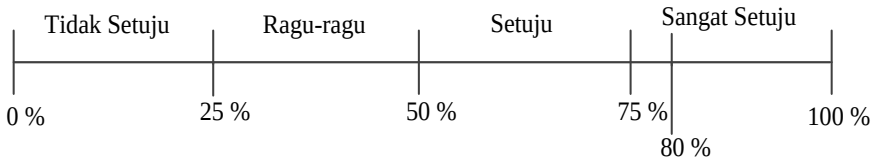
Pertanyaan	Keterangan	Skor	Frekuensi	S
3	Sangat setuju	4	11	44
	Setuju	3	26	78
	Ragu-ragu	2	3	6
	Tidak setuju	1	0	0
Jumlah			40	128



Gambar 4.12 Grafik skor pertanyaan ke-3

$$P = \frac{128}{160} x100 = 80\%$$

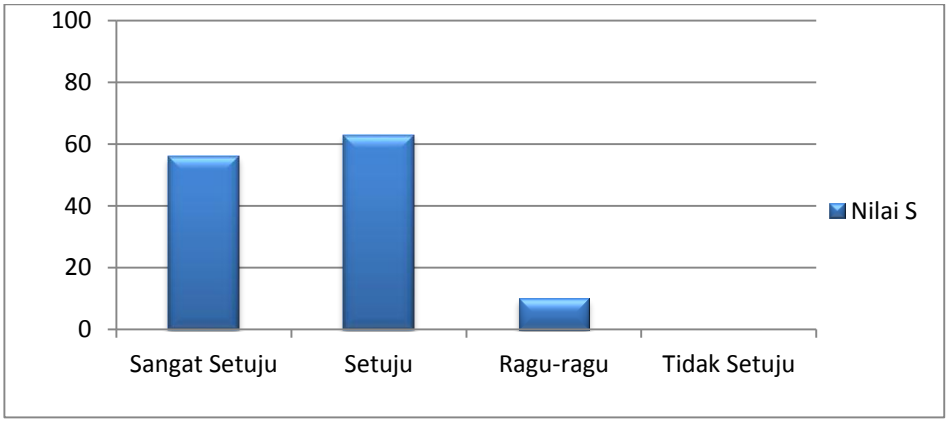
Berdasarkan hasil presentase nilai di atas, maka dapat disimpulkan bahwa sejarah medan area di jadikan sebuah game adalah 80% dari yang diharapkan 100%, atau dapat dikategorikan sebagai sangat setuju. Berikut kriteria interprestasi skor :



4. Apakah anda setuju materi yang disampaikan memberikan pengetahuan untuk pelajar kelas XII mengenai medan area?

Tabel 4.10 Perhitungan persentase pertanyaan ke-4

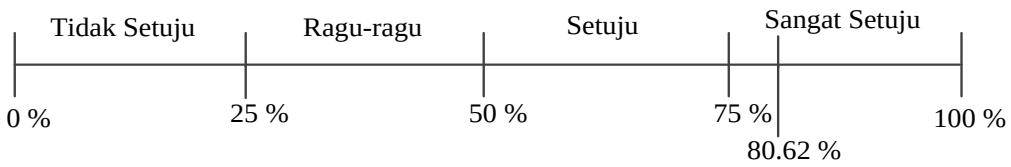
Pertanyaan	Keterangan	Skor	Frekuensi	S
4	Sangat setuju	4	14	56
	Setuju	3	21	63
	Ragu-ragu	2	5	10
	Tidak setuju	1	0	0
Jumlah			40	129



Gambar 4.13 Grafik skor pertanyaan ke-4

$$P = \frac{129}{160} x100 = 80.62\%$$

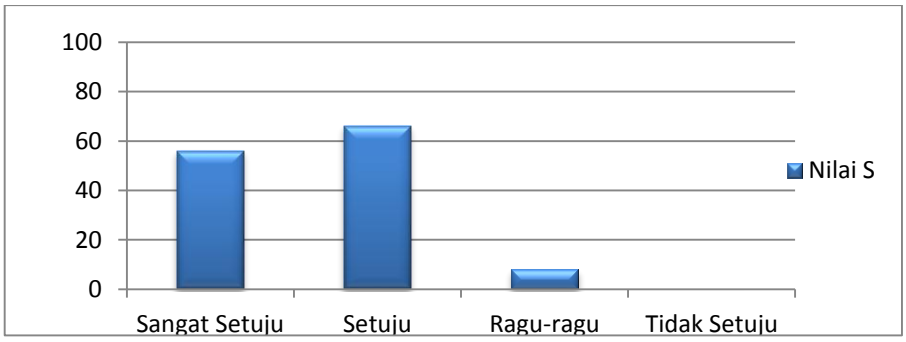
Berdasarkan hasil presentase nilai di atas, maka dapat disimpulkan setuju dengan bermain game sejarah medan area, anda dapat mengetahui sejarah medan area adalah 80.62% dari yang diharapkan 100%, atau dapat dikategorikan sebagai setuju. Berikut kriteria interprestasi skor :



5. Apakah materi yang telah disampaikan mudah untuk di ingat?

Tabel 4.11 Perhitungan persentase pertanyaan ke-5

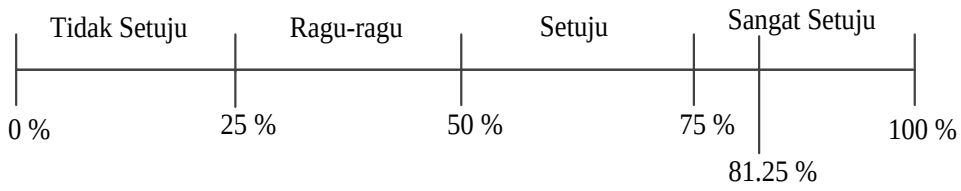
Pertanyaan	Keterangan	Skor	Frekuensi	S
5	Sangat setuju	4	14	56
	Setuju	3	22	66
	Ragu-ragu	2	4	8
	Tidak setuju	1	0	0
Jumlah			40	130



Gambar 4.14 Grafik skor pertanyaan ke-5

$$P = \frac{130}{160} x100 = 81.25\%$$

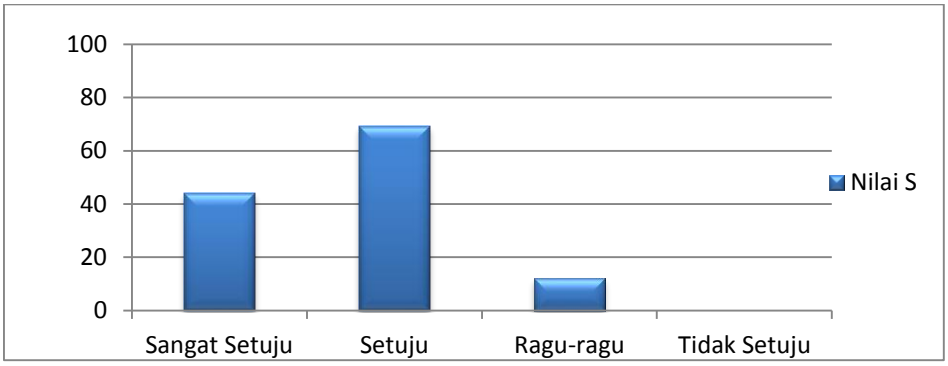
Berdasarkan hasil presentase nilai di atas, maka dapat disimpulkan bahwa materi berupa pengenalan nama tokoh dan tempat peristiwa medan area pada game ini mudah untuk di ingat adalah 81.25% dari yang diharapkan 100%, atau dapat dikategorikan sebagai setuju. Berikut kriteria interprestasi skor :



6. Apakah setelah memainkan game sejarah medan area anda lebih mengetahui nama tokoh dan tempat terjadinya sejarah medan area?

Tabel 4.12 Perhitungan persentase pertanyaan ke-6

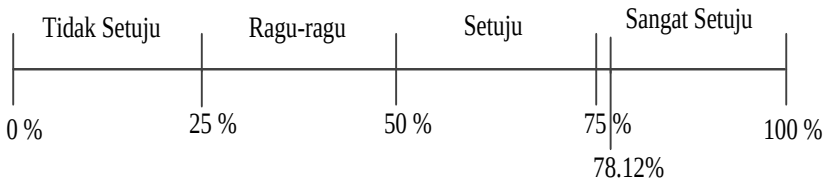
Pertanyaan	Keterangan	Skor	Frekuensi	S
6	Sangat setuju	4	11	44
	Setuju	3	23	69
	Ragu-ragu	2	6	12
	Tidak setuju	1	0	0
Jumlah			40	125



Gambar 4.15 Grafik skor pertanyaan ke-6

$$P = \frac{125}{160} x100 = 78.12\%$$

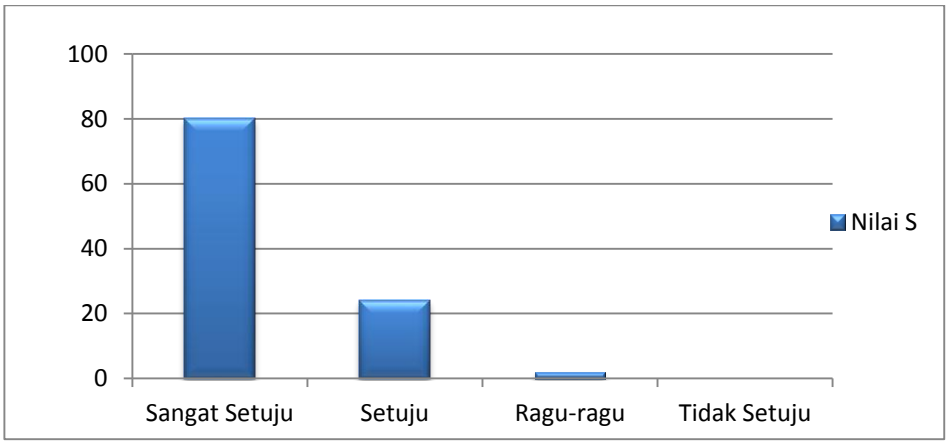
Berdasarkan hasil presentase nilai di atas, maka dapat disimpulkan game sejarah perjuangan Medan Area mudah untuk dimainkan adalah 78.12% dari yang diharapkan 100%, atau dapat dikategorikan sangat setuju. Berikut kriteria interprestasi skor :



7. Apakah anda setuju grafis 3D yang mengekpresikan genre first person shooter memberikan kebebasan pada karakter?

Tabel 4.13 Perhitungan persentase pertanyaan ke-7

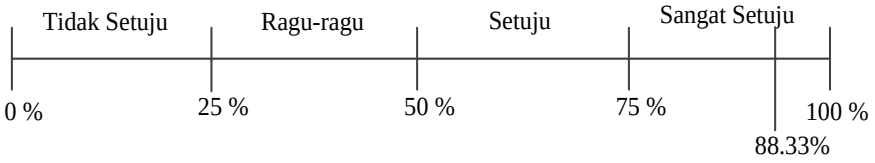
Pertanyaan	Keterangan	Skor	Frekuensi	S
7	Sangat setuju	4	20	80
	Setuju	3	8	24
	Ragu-ragu	2	1	2
	Tidak setuju	1	0	0
Jumlah			40	106



Gambar 4.16 Grafik skor pertanyaan ke-7

$$P = \frac{106}{160} x100 = 88.33\%$$

Berdasarkan hasil presentase nilai di atas, maka dapat disimpulkan bahwa penilaian terhadap bangunan yang terdapat di dalam game sejarah medan area sudah hampir mirip dengan sejarah medan area adalah 88.33% dari yang diharapkan 100%, atau dapat dikategorikan setuju. Berikut kriteria interprestasi skor :

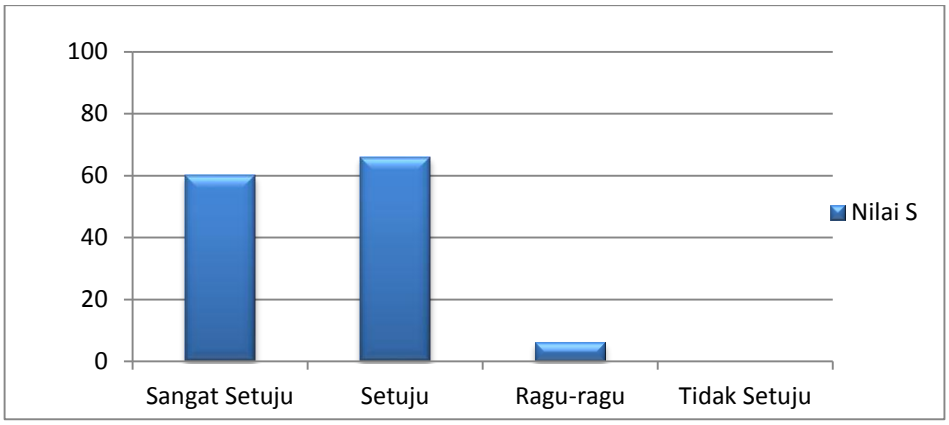


8. Apakah tampilan keseluruhan game sejarah medan area ini menarik untuk dimainkan?

Tabel 4.14 Perhitungan persentase pertanyaan ke-8

Pertanyaan	Keterangan	Skor	Frekuensi	S
8	Sangat setuju	4	15	60

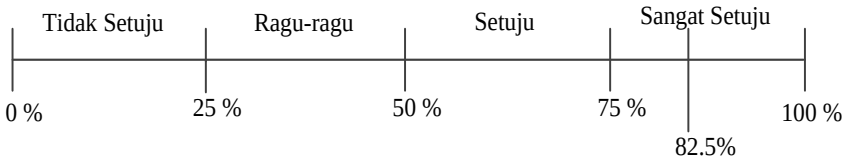
	Setuju	3	22	66
	Ragu-ragu	2	3	6
	Tidak setuju	1	0	0
Jumlah			40	132



Gambar 4.17 Grafik skor pertanyaan ke-8

$$P = \frac{132}{160} x100 = 82.5\%$$

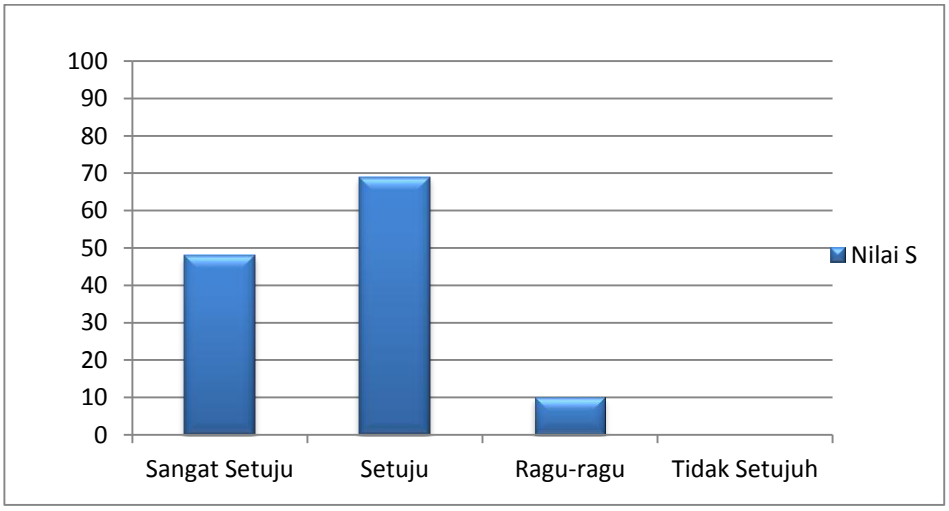
Berdasarkan hasil presentase nilai di atas, maka dapat disimpulkan bahwa tampilan grafis 3D, game sejarah medan area menampilkan nuansa permainan yang lebih nyata?adalah 82.5% dari yang diharapkan 100%, atau dapat dikategorikan setuju. Berikut kriteria interprestasi skor :



9. Apakah cara memainkan game medan area ini mudah untuk di mengerti?

Tabel 4.15 Perhitungan persentase pertanyaan ke-9

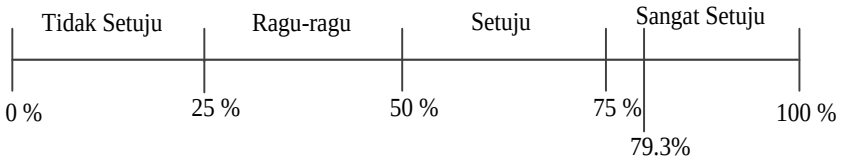
Pertanyaan	Keterangan	Skor	Frekuensi	S
9	Sangat setuju	4	12	48
	Setuju	3	23	69
	Ragu-ragu	2	5	10
	Tidak setuju	1	0	0
Jumlah			40	127



Gambar 4.18 Grafik skor pertanyaan ke-9

$$P = \frac{127}{160} x100 = 79.37\%$$

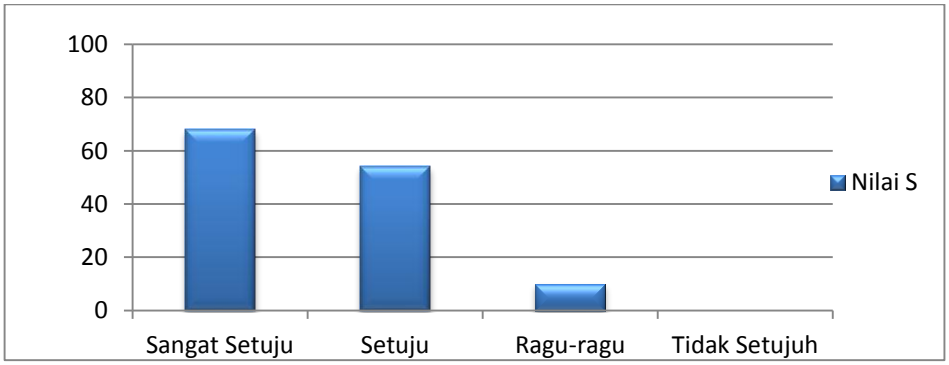
Berdasarkan hasil presentase nilai di atas, maka dapat disimpulkan bahwa game ini memberikan sisi moral yang baik buat generasi muda adalah 73,3% dari yang diharapkan 100%, atau dapat dikategorikan setuju. Berikut kriteria interprestasi skor :



10. Apakah anda setuju setelah memainkan game ini anda merasa terhibur?

Tabel 4.16 Perhitungan persentase pertanyaan ke-10

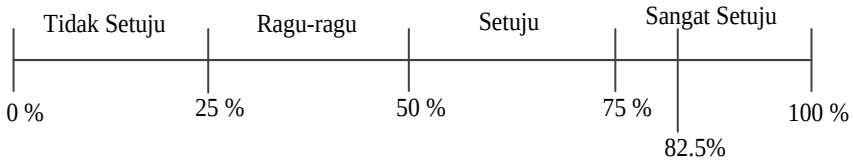
Pertanyaan	Keterangan	Skor	Frekuensi	S
10	Sangat setuju	4	17	68
	Setuju	3	18	54
	Ragu-ragu	2	5	10
	Tidak setuju	1	0	0
Jumlah			30	132



Gambar 4.19 Grafik skor pertanyaan ke-10

$$P = \frac{132}{160} x100 = 82.5\%$$

Berdasarkan hasil presentase nilai di atas, maka dapat disimpulkan bahwa game ini anda merasa terhibur adalah 82.5% dari yang diharapkan 100%, atau dapat dikategorikan setuju. Berikut kriteria interprestasi skor :



4.2.3 Kesimpulan Pengujian Beta

Berdasarkan hasil kuesioner yang didapatkan dari pengujian beta bahwa responden sangat setuju sejarah medan area perlu dipelajari dan diketahui serta media pembelajarannya dalam bentuk game. Dimana di dalam game sejarah medan area terdapat materi pengenalan nama tokoh dan lokasi pertempuran sejarah medan area. Dimana tampilan yang diberikan dalam game berupa grafis 3D yang nantinya diharapkan dapat membuat game lebih menarik sehingga pemain terhibur dalam memainkan game medan area ini.