BAB 4

IMPLEMENTASI DAN PENGUJIAN

4.1 Implementasi

Proses implementasi adalah tahapan dimana aplikasi yang sudah dibangun dan di rancang lalu di uji kelayakannya. Pemograman pada game ini menggunakan Unity , Selanjutnya dioperasikan sebagaimana mestinya sesuai fungsi dan kelayakannya untuk digunakan.

4.2 Perangkat Pendukung yang Digunakan

Dalam proses pembangunan aplikasi ini, tentunya membutuhkan perangkat keras (hardware). Berikut merupakan penjelasan dari perangkat keras dan perangkat lunak yang digunakan.

4.2.1 Perangkat Keras

Perangkat keras yang digunakan selama implementasi dan pengujian aplikasi ini memiliki spesifikasi sebagai berikut:

1. CPU : Intel Core i3 – 3217U1.8GHz
2. Memory : 4 GB
3. VGA : AMD Radeon HD 6730M 2GB
4. Keyboard
5. Mouse
6. Speaker

4.2.2 Implementasi Class

Implementasi *class* merupakan implementasi dari perancangan *class diagram*yang terbentuk dari *use case diagram* kedalam bentuk file fisik dengan ekstensi".cs". Implementasi *class* bisa dilihat pada tabel 4.1 dibawah ini:

Tabel Error! No text of specified style in document.**.1Implementasi Class**

N	Nama *Class*	Nama File Fisik

o		
1.	Implementasi Fuzzy	SugenoFuzzyScript.cs
2.	Pop Up	AlasPopupScriptinGame.cs
3.	Alat siram	AlatSiramScript.cs
4.	Game manager	GameManager.cs
5.	Hama	HamaScript.cs
6.	Matahari	Matahari.cs
7.	Pestisida	PestisidaScript.cs
8.	Level	PopupMisiLevelScript.cs
9.	Pot	PotScript.cs
10.	Pupuk	PupukScript.cs
11.	Rak	RakScript.cs
12.	Item	ChooseItems.cs
13.	Inventory background	InventoryBackgroundScript.cs
14.	Inventory	InventoryManagerScript.cs
15.	Inventory system	InventorySystemScript.cs
16.	Tanaman bawang	TanamanBawangDaun.cs
17.	Tanaman seledri	TanamanSaledri.cs
18.	Tanaman tomat	TanamanTomat.cs
19.	Manager item	ItemManager.cs
20.	Placement item	ItemPlacement.cs

21.	Item place	PlacableItems.cs
22.	Ujian	UjianManagerScript.cs
23.	Healt bar	AlasHealthBarScript.cs
24.	Aimasi	AnimatedScript.cs
25.	Button	ButtonScript.cs
26.	Botton ujian	ButtonUjianScript.cs
27.	Time	CounterTime.cs
28.	Cover	CoverScript.cs
29.	Dashboard	DashboardManager.cs
30.	Floating healt bar	FloatingHealthBarScript.cs
31.	Game over	GameOverScript.cs
32.	Item animation	ItemAnimationScript.cs
33.	Loading	LoadingScreenScript.cs
34.	Music	MusicManager.cs
35.	Music player	MusicPlayer.cs
36.	Quit	QuitScript.cs

4.3 Implementasi Aplikasi

Cara menjalankan Game Edukasi Merawat Tananaman ini hanya tinggal klik 2 kali pada Aplikasi yang terdapat pada folder Build Game Edukasi Merawat Tanaman.

4.4 Implementasi Antar Muka

4.4.1 Menu Utama



Gambar Error! No text of specified style in document.**.1Tampilan Menu Utama**

Layar ini menampilkan menu utama yang di mana di dalamnya terdapat pilihan Main, Petunjuk Permainan, keluar Dan pengaturan suara.

4.4.2 Tampilan Petunjuk Permainan



Gambar Error! No text of specified style in document.**.2Menu Petunjuk Permainan**

Dalam menu Petunjuk Permainan terdapat cara memilih tanaman dan alat, cara menyimpan pot di rak, cara menanam dancara membasmi hama.

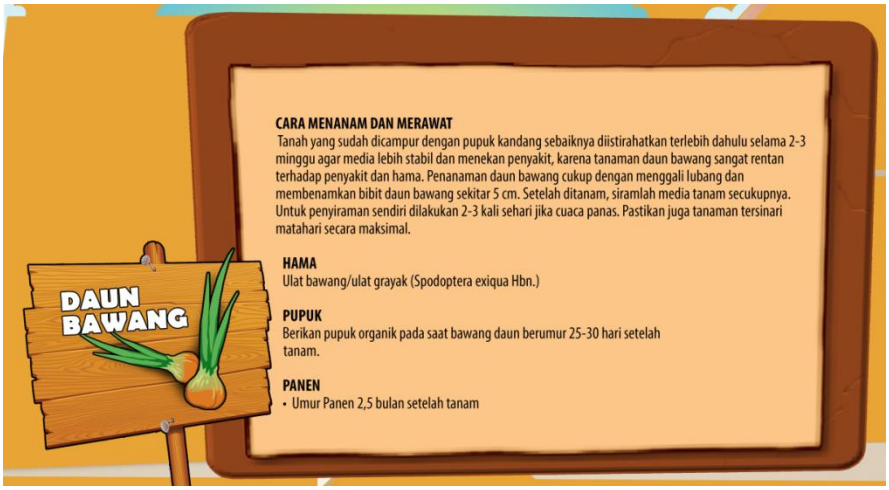
4.4.3 Tampilan Level



Gambar Error! No text of specified style in document.**.3Level Permainan**

Layar ini menampilkan Menu Level permaianan, pemain harus menyelesaikan level 1 untuk membuka kunci ke level selanjutnya.

4.4.4 Tampilan Info Tanaman



Gambar Error! No text of specified style in document.**.4Info Tanaman**

Layar ini menjelaskan tentang info tanaman meliputi Cara menanam dan merawat tanaman, hama tanaman, pupuk yang digunakan dan jangka waktu panen tanaman.

4.4.5 Tampilan Kuis Tanaman



Gambar Error! No text of specified style in document.**.5Kuis Tanaman**

Ini adalah tampilan Kuis Tanaman, Pemain harus menjawab pertanyan untuk memulai permainan.

4.4.6 Tampilan Inventory Permainan



Gambar Error! No text of specified style in document.**.6Inventory Permainan**

Layar ini merupakan menu untukmemilih inventory yang akan digunakan dalam permainan.

4.4.7 Tampilan Loading



Gambar Error! No text of specified style in document.**.7Halaman Loading Permainan**

Pada Layar ini akan di jelaskan sekilas bagaimana cara pemain menanam, menyiram, memupuk, membasmi hama yang akan berada pada game ini.

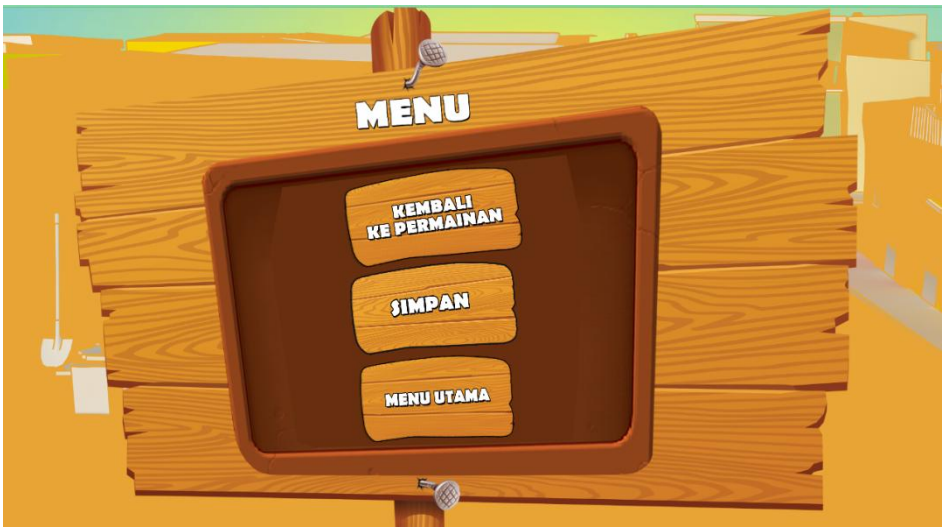
4.4.8 Tampilan Permainan



Gambar Error! No text of specified style in document.**.8Halaman Permainan**

Layar ini menampilkan permainan dalam game ini yang di mana terdapat Rak, slot inventory, menu dan tampilan waktu dala permainn.

4.4.9 Tampilan Menghentikan Permainan Sementara



Gambar Error! No text of specified style in document.**.9Tampilan Menghentikan Permainan Sementara**

Layar ini menampilkan Menu untuk menghentikan permainan sementara, pemain juga dapat langsung ke menu utama atau melanjutkan kembali ke permainan.

4.4.10 Tampilan Permainan Berhasil



Gambar Error! No text of specified style in document.**.10Permainan Berhasil**

Layar ini menampilkan Permainan berhasil menyelesaikan misi merawat 3 tanaman dengan sempurna.

4.4.11 Tampilan Permainan Gagal



Gambar Error! No text of specified style in document.**.11Permainan Gagal**

Berikut adalah tampilan pemain yang gagal menyelesaikan misi apabila salah satu tanaman tidak berhasil dirawat dengan sempurna.

4.5 Pengujian

Pengujian adalah hal yang perlu dilakukan untuk mengetahui kekurangan atau kesalahan pada perangkat lunak yang diuji. Pengujian bermaksud untuk mengetahui perangkat lunak yang dibuat sudah memenuhi kriteria yang sesuai dengan tujuan perancangan perangkat lunak tersebut.

4.5.1 Pengujian Alpha

Pengujian alpha dilakukan pada sisi pengembang yang dianggap pemakai dan merekam semua kesalahan dan masalah pemakaian. Pengujian alpha dilakukan pada sebuah lingkungan yang terkendali.

Pendekatan perancangan kasus uji menggunakan dua metode, yaitu white box dan *Black Box Testing*.*White box* adalah metode perancangan kasus uji yang

menggunakan struktur kontrol program. Black box adalah metode pengujian yang berfokus pada persyaratan fungsional perangkat lunak.

4.5.1.1 Rencana Pengujian

Rencana pengujian yang akan dilakukan pada aplikasi ini selengkapnya terlihat pada Tabel 4.1

Tabel Error! No text of specified style in document.**.2Rencana Pengujian Aplikasi**

Kelas Uji	Butir Uji	Jenis Pengujian
Mulai Permainan	Memulai memainkan	*Blackbox*
Melihat Petunjuk Permainan	Membuka halaman petunjuk	*Blackbox*
Melihat Info Tanaman	Melihat halaman info tanaman	*Blackbox*
Mengatur Suara On/Off	Mengatur suara	*Blackbox*
Keluar Permainan	Melihat tampilan keluar	*Blackbox*
Memilih Level Permainan	Memilih level	*Blackbox*
Memilih Inventory Permainan	Memilih inventory	*Blackbox*
Mengambil Poin Matahari	Mengambil poin matahari	*Blackbox*
Memakai Inventory Permainan	Memakai inventory	*Blackbox*

Membasmi Hama	Membasmi hama	*WhiteBox*
Menghentikan Permainan Sementara	Mengentikan permainan	*Blackbox*
Melihat Tampilan Petunjuk	Melihat tampilan petunjuk	*Blackbox*
Melihat Tampilan Info Tanaman	Melihat tampilan info tanaman	*Blackbox*
Menjawab Kuis	Proses menjawab kuis	*Blackbox*
Melihat Tampilan Kuis	Melihat tampilan kuis	*Blackbox*

4.5.1.2 Pengujian Black Box

Tabel Error! No text of specified style in document..**3Pengujian Black Box**

Skenario Uji	Hasil Yang Diharapkan	Kesimpulan
Mulai Permainan	Sistem Menampilkan Pilihan Level	Diterima [x] Ditolak [ ]
Melihat Petunjuk Permainan	Menampilkanhalamanpetunjuk	Diterima [x] Ditolak [ ]
Melihat Info Tanaman	Menmpilkan halaman info	Diterima [x] Ditolak [ ]
Mengatur Suara On/Off	Suara bisa dirubah	Diterima [x] Ditolak [ ]

Keluar Permainan	Menampilkan halaman keluar	Diterima [x] Ditolak [ ]
MemilihLevel Permainan	Level permainan terpilih	Diterima [x] Ditolak [ ]
Memilih Inventory Permainan	Inventory terpilih	Diterima [x] Ditolak [ ]
Mengambil Poin Matahari	Poin bertambah	Diterima [x] Ditolak [ ]
Memakai Inventory Permainan	Inventory permainan terpakai	Diterima [x] Ditolak [ ]
Membasmi Hama	Darah hama berkurang	Diterima [x] Ditolak [ ]
MenghentikanPermainan Sementara	Menghentikan permainan	Diterima [x] Ditolak [ ]
Melihat Tampilan Petunjuk	Menampilkanhalaman petunjuk	Diterima [x] Ditolak [ ]
MelihatTampilanInfo Tanaman	Tampil halaman info	Diterima [x] Ditolak [ ]
Menjawab Kuis	Menampilkan halaman kuis	Diterima [x] Ditolak [ ]
Melihat Tampilan Kuis	Melihat tampilan kuis	Diterima [x] Ditolak [ ]

4.5.1.3 Pengujian White Box

Berikut ini adalah kasus menguji perangkat lunak yang telah dibangun menggunakan metode white-box berdasarkan tabel 4.1. Pengujian white-box menggunakan flow graph yang digunakan untuk menggambarkan alur dari algoritma dan grap matrix yang digunakan untuk menggenerasi flow graph. Berikut pengujian white-box dapat dilihat pada tabel 4.3

Tabel Error! No text of specified style in document.**.4Pengujian White Box**

o	Coding

	void Start () {
	hamaScript = GetComponent<HamaScript>();
	darahHamaCurr = hamaScript.healthBarEnemy;
	kekuatanHamaCurr = hamaScript.kekuatanEnemy;
	}

	void Update () {
	
	// update input untuk fuzzy - darah dan kekuatan
	darahHamaCurr = hamaScript.healthBarEnemy;
	kekuatanHamaCurr = hamaScript.kekuatanEnemy;
	
	KosongkanMatrixImplikasi();
	KosongkanMatrixAggregasi();
	
	FuzzyRule();
	MinimumFromImplikasi(indexRule, implikasi, implikasiMinimum);

```
            PerhitunganDefuzzifikasi(aggregasiKabur,
aggregasiBertahan, aggregasiMenyerang);
            Keputusan(z);
```

```
       void CetakInfo()
         {
            Debug.Log("darah_sedikit:           "           +
darahHamaFuz_Sedikit);
            Debug.Log("darah_sedang:            "           +
darahHamaFuz_Sedang);
            Debug.Log("darah_banyak:            "           +
darahHamaFuz_Banyak);
            Debug.Log("kekuatan_lemah:           "          +
kekuatanHamaFuz_Lemah);
            Debug.Log("kekuatan_sedang:          "          +
kekuatanHamaFuz_Sedang);
            Debug.Log("kekuatan_kuat:           "           +
kekuatanHamaFuz_Kuat);
            Debug.Log("rule index: " + indexRule);
```

```
       for (int i = 0; i < 2; i++)
            {
                for (int j = 0; j < indexRule; j++)
                {
                    Debug.Log("matrix[" + i + ", " + j + "]
: " + implikasi[i, j]);
                }
            }

            for (int i = 0; i < indexRule; i++)
            {
```

```
                Debug.Log("minimum index-" + i + " : " + implikasiMinimum[i]);
            }
    if (indexKabur > 0)
            {
    for (int i = 0; i < indexKabur; i++)
                {
                    Debug.Log("index kabur-" + i + " : " + himpunanKabur[i]);
                }
            }
    if (indexBertahan > 0)
            {
    for (int i = 0; i < indexBertahan; i++)
                {
                    Debug.Log("index bertahan-" + i + " : " + himpunanBertahan[i]);
                }
            }
    if (indexMenyerang > 0)
            {
0   for (int i = 0; i < indexMenyerang; i++)
                {
                    Debug.Log("index menyerang-" + i + " : " + himpunanMenyerang[i]);
                }
            }
1   void FuzzyRule()
```

```
        {
            // inisialisasi
            indexRule = 0;
            indexKabur = 0;
            indexBertahan = 0;
            indexMenyerang = 0;

            darahHamaFuz_Sedikit = 0;
            darahHamaFuz_Sedang = 0;
            darahHamaFuz_Banyak = 0;

            kekuatanHamaFuz_Lemah = 0;
            kekuatanHamaFuz_Sedang = 0;
            kekuatanHamaFuz_Kuat = 0;
```

```
    if (darahHamaCurr <= 45 && kekuatanHamaCurr <= 45)
            {
                Debug.Log("Rule-1");
```

```
    if (darahHamaCurr <= 20)
                {
                    darahHamaFuz_Sedikit = 1;
                }
```

```
    else
                {
```

```
    if (darahHamaCurr < 45)
                    {
                        darahHamaFuz_Sedikit = (45 - darahHamaCurr) / (45 - 20);
                    }
```

6	else
	{
	darahHamaFuz_Sedikit = 0;
	}
	}

7	if (kekuatanHamaCurr <= 30)
	{
	kekuatanHamaFuz_Lemah = 1;
	}

8	else {

9	if (kekuatanHamaCurr < 45)
	{
	kekuatanHamaFuz_Lemah = (45 - kekuatanHamaCurr) / (45 - 30);
	}

0	else
	{
	kekuatanHamaFuz_Lemah = 0;
	}
	}
	InputToImplikasiMatrix(0, indexRule, darahHamaFuz_Sedikit);
	InputToImplikasiMatrix(1, indexRule, kekuatanHamaFuz_Lemah);
	
	aksi = AksiHama.kabur;

	InputToAggregasi(ReturnWhoisMinimum(darahHamaFuz_Sedikit, kekuatanHamaFuz_Lemah), indexKabur, aksi);
	
	indexRule++;
	indexKabur++;
	
	}

1	if (darahHamaCurr <= 45 && kekuatanHamaCurr <= 70 && kekuatanHamaCurr >= 30) {

2	if (darahHamaCurr <= 20)
	{
	darahHamaFuz_Sedikit = 1;
	}

3	else {

4	if (darahHamaCurr < 45)
	{
	darahHamaFuz_Sedikit = (45 - darahHamaCurr) / (45 - 20);
	}

5	else
	{
	darahHamaFuz_Sedikit = 0;
	}
	}

6	if (kekuatanHamaCurr <= 30) {

```
                    kekuatanHamaFuz_Sedang = 0;
                }
7   else
                {
8   if (kekuatanHamaCurr < 45)
                    {
                        kekuatanHamaFuz_Sedang = (kekuatanHamaCurr - 30) / (45 - 30);
                    }
9   else
                    {
0   if (kekuatanHamaCurr <= 55)
                        {
                            kekuatanHamaFuz_Sedang = 1;
                        }
1   else
                        {
2   if (kekuatanHamaCurr < 70)
                            {
                                kekuatanHamaFuz_Sedang = (70 - kekuatanHamaCurr) / (70 - 55);
                            }
3   else
                            {
                                kekuatanHamaFuz_Sedang = 0;
                            }
                        }
                    }
```

	}
	InputToImplikasiMatrix(0, indexRule, darahHamaFuz_Sedikit);
	InputToImplikasiMatrix(1, indexRule, kekuatanHamaFuz_Sedang);
	
	aksi = AksiHama.kabur;
	
	InputToAggregasi(ReturnWhoisMinimum(darahHamaFuz_Sedikit, kekuatanHamaFuz_Sedang), indexKabur, aksi);
	
	indexRule++;
	indexKabur++;
	
	}

4	if (darahHamaCurr <= 45 && kekuatanHamaCurr >= 55) {

5	if (darahHamaCurr <= 20)
	{
	darahHamaFuz_Sedikit = 1;
	}

6	else {

7	if (darahHamaCurr < 45)
	{
	darahHamaFuz_Sedikit = (45 - darahHamaCurr) / (45 - 20);
	}

	else

```
8                       {
                            darahHamaFuz_Sedikit = 0;
                        }
                    }
9       if (kekuatanHamaCurr <= 55)
                    {
                        kekuatanHamaFuz_Kuat = 0;
                    }
0       else
                    {
1       if (kekuatanHamaCurr < 70)
                        {
                            kekuatanHamaFuz_Kuat                =
(kekuatanHamaCurr - 55) / (70 - 55);
                        }
2       else
                        {
                            kekuatanHamaFuz_Kuat = 1;
                        }
                    }
                    InputToImplikasiMatrix(0,          indexRule,
darahHamaFuz_Sedikit);
                    InputToImplikasiMatrix(1,          indexRule,
kekuatanHamaFuz_Kuat);

                    aksi = AksiHama.kabur;

InputToAggregasi(ReturnWhoisMinimum(darahHamaFuz_Sedikit,
kekuatanHamaFuz_Kuat), indexKabur, aksi);
```

	indexRule++;
	indexKabur++;
	
	}

3	if (darahHamaCurr >= 20 && darahHamaCurr <= 80 && kekuatanHamaCurr <= 45) {

4	if (darahHamaCurr <= 20 || darahHamaCurr >= 80)
	{
	darahHamaFuz_Sedang = 0;
	}

5	else {

6	if (darahHamaCurr < 45)
	{
	darahHamaFuz_Sedang = (darahHamaCurr - 20) / (45 - 20);
	}

7	else {

8	if (darahHamaCurr < 55)
	{
	darahHamaFuz_Sedang = 1;
	}

9	else {
	if (kekuatanHamaCurr <= 30)

```
1                   {
                        kekuatanHamaFuz_Lemah = 1;
                    }
2       else
                    {
3       if (kekuatanHamaCurr < 45)
                        {
                            kekuatanHamaFuz_Lemah = ( 45 -
kekuatanHamaCurr) / (45 - 30);
4       else
                        {
                            kekuatanHamaFuz_Lemah = 0;
                        }
                    }
                    InputToImplikasiMatrix(0,          indexRule,
darahHamaFuz_Sedang);
                    InputToImplikasiMatrix(1,          indexRule,
kekuatanHamaFuz_Lemah);

                    aksi = AksiHama.bertahan;


InputToAggregasi(ReturnWhoisMinimum(darahHamaFuz_Sedang,
kekuatanHamaFuz_Lemah), indexBertahan, aksi);

                    indexBertahan++;
                    indexRule++;
```

```
            }
5   if (darahHamaCurr >= 20 && darahHamaCurr <= 80 &&
kekuatanHamaCurr <= 70 && kekuatanHamaCurr >= 30)
            {
6   if (darahHamaCurr <= 20 || darahHamaCurr >= 80)
                {
                    darahHamaFuz_Sedang = 0;
                }
7   else
                {
8   if (darahHamaCurr < 45)
                    {
                        darahHamaFuz_Sedang = (darahHamaCurr
- 20) / (45 - 20);
                    }
9   else
                    {
0   if (darahHamaCurr < 55)
                        {
                            darahHamaFuz_Sedang = 1;
                        }
1   else
                        {
2   if (kekuatanHamaCurr <= 30)
                {
                    kekuatanHamaFuz_Sedang = 0;
                }
    else
```

3	{

4	if (kekuatanHamaCurr < 45)
	{
	kekuatanHamaFuz_Sedang = (kekuatanHamaCurr - 30) / (45 - 30);
	}

5	else {

6	if (kekuatanHamaCurr <= 55)
	{
	kekuatanHamaFuz_Sedang = 1;
	}

7	else {

8	if (kekuatanHamaCurr < 70)
	{
	kekuatanHamaFuz_Sedang = (70 - kekuatanHamaCurr) / (70 - 55);
	}

9	else
	{
	kekuatanHamaFuz_Sedang = 0;
	}
	}
	}
	}
	InputToImplikasiMatrix(0, indexRule, darahHamaFuz_Sedang);

	InputToImplikasiMatrix(1, indexRule, kekuatanHamaFuz_Sedang);
	
	aksi = AksiHama.bertahan;
	
	InputToAggregasi(ReturnWhoisMinimum(darahHamaFuz_Sedang, kekuatanHamaFuz_Sedang), indexBertahan, aksi);
	
	indexBertahan++;
	indexRule++;
	
	}

0	if (darahHamaCurr >= 20 && darahHamaCurr <= 80 && kekuatanHamaCurr >= 55) {

1	if (darahHamaCurr <= 20 || darahHamaCurr >= 80)
	{
	darahHamaFuz_Sedang = 0;
	}

2	else {

3	if (darahHamaCurr < 45)
	{
	darahHamaFuz_Sedang = (darahHamaCurr - 20) / (45 - 20);
	}

	else

```
4                       {
5       if (darahHamaCurr < 55)
                        {
                            darahHamaFuz_Sedang = 1;
                        }
6       else
                        {
7       if (kekuatanHamaCurr <= 55)
                {
                    kekuatanHamaFuz_Kuat = 0;
                }
8       else
                {
9       if (kekuatanHamaCurr < 70)
                    {
                        kekuatanHamaFuz_Kuat                =
(kekuatanHamaCurr - 55) / (70 - 55);
                    }
0       else
                    {
                        kekuatanHamaFuz_Kuat = 1;
                    }
                }
                InputToImplikasiMatrix(0,          indexRule,
darahHamaFuz_Sedang);
                InputToImplikasiMatrix(1,          indexRule,
kekuatanHamaFuz_Kuat);
```

	aksi = AksiHama.bertahan;
	
	InputToAggregasi(ReturnWhoisMinimum(darahHamaFuz_Sedang, kekuatanHamaFuz_Kuat), indexBertahan, aksi);
	
	indexRule++;
	indexBertahan++;
	
	}

1	if (darahHamaCurr >= 55 && kekuatanHamaCurr <= 45) {

2	if (darahHamaCurr <= 55)
	{
	darahHamaFuz_Banyak = 0;
	}

3	else {

4	if (darahHamaCurr < 80)
	{
	darahHamaFuz_Banyak = (darahHamaCurr - 55) / (80 - 55);
	}

5	else
	{
	darahHamaFuz_Banyak = 1;
	}
	}

	if (kekuatanHamaCurr <= 30)

6	{
	kekuatanHamaFuz_Lemah = 1;
	}

7	else {

8	if (kekuatanHamaCurr < 45)
	{
	kekuatanHamaFuz_Lemah = (45 - kekuatanHamaCurr) / (45 - 30);

9	else
	{
	kekuatanHamaFuz_Lemah = 0;
	}
	}
	InputToImplikasiMatrix(0, indexRule, darahHamaFuz_Banyak);
	InputToImplikasiMatrix(1, indexRule, kekuatanHamaFuz_Lemah);
	
	aksi = AksiHama.menyerang;
	
	InputToAggregasi(ReturnWhoisMinimum(darahHamaFuz_Banyak, kekuatanHamaFuz_Lemah), indexMenyerang, aksi);
	
	indexMenyerang++;
	indexRule++;
	
	}

0	if (darahHamaCurr >= 55 && kekuatanHamaCurr <= 70 && kekuatanHamaCurr >= 30) {

1	if (darahHamaCurr <= 55)
	{
	darahHamaFuz_Banyak = 0;
	}

2	else {

3	if (darahHamaCurr < 80)
	{
	darahHamaFuz_Banyak = (darahHamaCurr - 55) / (80 - 55);
	}

4	else
	{
	darahHamaFuz_Banyak = 1;
	}
	}

5	if (kekuatanHamaCurr <= 30)
	{
	kekuatanHamaFuz_Sedang = 0;
	}

6	else {

7	if (kekuatanHamaCurr < 45)
	{
	kekuatanHamaFuz_Sedang =

```
(kekuatanHamaCurr - 30) / (45 - 30);
                    }
8     else
                    {
9     if (kekuatanHamaCurr <= 55)
                        {
                            kekuatanHamaFuz_Sedang = 1;
                        }
00    else
                        {
01    if (kekuatanHamaCurr < 70)
                            {
                                kekuatanHamaFuz_Sedang = (70
- kekuatanHamaCurr) / (70 - 55);
                            }
02    else
                            {
                                kekuatanHamaFuz_Sedang = 0;
                            }
                        }
                    }
                InputToImplikasiMatrix(0,          indexRule,
darahHamaFuz_Banyak);
                InputToImplikasiMatrix(1,          indexRule,
kekuatanHamaFuz_Sedang);

                aksi = AksiHama.menyerang;
```

```
InputToAggregasi(ReturnWhoisMinimum(darahHamaFuz_Banyak,
kekuatanHamaFuz_Sedang), indexMenyerang, aksi);

                indexMenyerang++;
                indexRule++;

            }
03      if (darahHamaCurr >= 55 && kekuatanHamaCurr >= 55)
            {
04      if (darahHamaCurr <= 55)
                {
                    darahHamaFuz_Banyak = 0;
                }
05      else
                {
06      if (darahHamaCurr < 80)
                    {
                        darahHamaFuz_Banyak = (darahHamaCurr
- 55) / (80 - 55);
                    }
07      else
                    {
                        darahHamaFuz_Banyak = 1;
                    }
                }
        if (kekuatanHamaCurr <= 55)
```

```
08                  {
                        kekuatanHamaFuz_Kuat = 0;
                    }
09      else
                    {
10      if (kekuatanHamaCurr < 70)
                        {
                            kekuatanHamaFuz_Kuat                =
(kekuatanHamaCurr - 55) / (70 - 55);
                        }
11      else
                        {
                            kekuatanHamaFuz_Kuat = 1;
                        }
                    }
                    InputToImplikasiMatrix(0,          indexRule,
darahHamaFuz_Banyak);
                    InputToImplikasiMatrix(1,          indexRule,
kekuatanHamaFuz_Kuat);

                    aksi = AksiHama.menyerang;

InputToAggregasi(ReturnWhoisMinimum(darahHamaFuz_Banyak,
kekuatanHamaFuz_Kuat), indexMenyerang, aksi);

                    indexRule++;
                    indexMenyerang++;
```

	} }

12	void KosongkanMatrixImplikasi()
	{
	for (int i = 0; i < 2; i++)
	{
	for (int j = 0; j < 4; j++)
	{
	implikasi[i, j] = 0;
	}
	}
	}

13	void KosongkanMatrixAggregasi()
	{
	for (int i = 0; i < himpunanKabur.Length; i++)
	{
	himpunanKabur[i] = 0;
	}
	
	for (int i = 0; i < himpunanBertahan.Length; i++)
	{
	himpunanBertahan[i] = 0;
	}
	
	for (int i = 0; i < himpunanMenyerang.Length; i++)
	{

	himpunanMenyerang[i] = 0;
	}
	}

14	void InputToImplikasiMatrix(int varIndex, int indexRule, float value)
	{
	implikasi[varIndex, indexRule] = value;
	}

15	void MinimumFromImplikasi(int indexRule, float[,] imp, float[] min)
	{
	float val = 999;
	
	for (int i = 0; i < indexRule; i++)
	{
	for (int j = 0; j < 2; j++)
	{
	if (imp[j, i] < val)
	{
	val = imp[j, i];
	min[i] = val;
	}
	}
	val = 999;
	}
	}

16	void Aggregasi() {

17	if (aksi == AksiHama.kabur)
	{
	float t = -999;
	for (int i = 0; i < himpunanKabur.Length; i++)
	{
	if (himpunanKabur[i] > t)
	{
	t = himpunanKabur[i];
	aggregasiKabur = himpunanKabur[i];
	}
	}
	t = -999;
	}

18	if (aksi == AksiHama.bertahan)
	{
	//Debug.Log("indexber: " + himpunanBertahan.Length);
	float t = -999;
	for (int i = 0; i < himpunanBertahan.Length; i++)
	{

19	if (himpunanBertahan[i] > t)
	{
	t = himpunanBertahan[i];
	aggregasiBertahan = himpunanBertahan[i];
	}

	}
	t = -999;
	}

20	if (aksi == AksiHama.menyerang)
	{
	float t = -999;
	for (int i = 0; i < himpunanMenyerang.Length; i++)
	{

21	if (himpunanMenyerang[i] > t)
	{
	t = himpunanMenyerang[i];
	aggregasiMenyerang = himpunanMenyerang[i];
	}
	}
	t = -999;
	}
	}

22	void PerhitunganDefuzzifikasi(float kabur, float bertahan, float menyerang) {

23	if ((menyerang + bertahan + kabur) != 0)
	{
	z = ((kabur * batasKabur) + (bertahan * batasBertahan) + (menyerang * batasMenyerang)) / (kabur + bertahan + menyerang);

24	else {

```
                z = 5;
            }
25  void Keputusan(float z)
        {
26  if (z < 2)
            {
                keputusan = "kabur";
                Debug.Log("kep.kabur");
            }
27  else
            {
28  if (z < 5)
                {
                    keputusan = "bertahan";
                    Debug.Log("kep.bertahan");
                }
29  else
                {
                    keputusan = "serang";
                    Debug.Log("kep.serang");
                }
            }
        }
30  void InputToAggregasi(float setValue,int indexAksi,
AksiHama aksiH)
        {
31  if (aksiH == AksiHama.kabur)
            {
```

	himpunanKabur[indexAksi] = setValue; }

32	if (aksiH == AksiHama.bertahan)
	{
	himpunanBertahan[indexAksi] = setValue;
	}

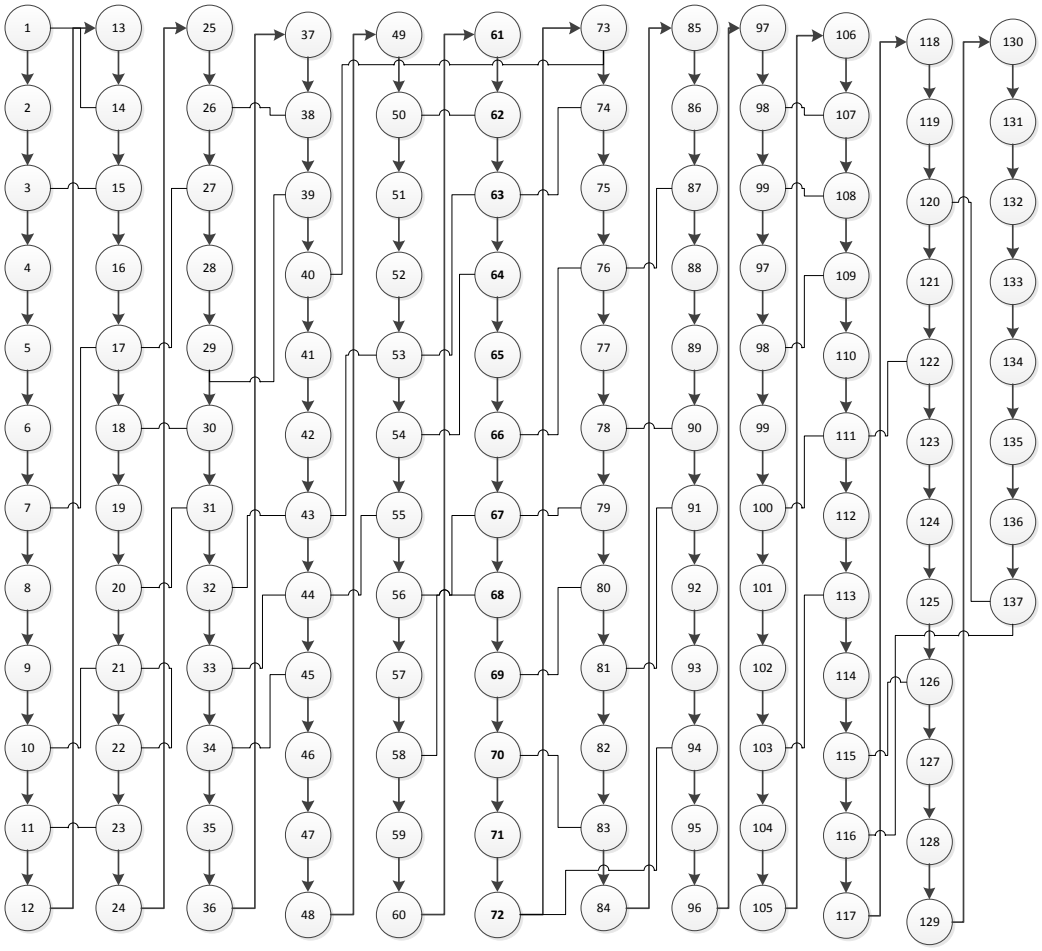
33	if (aksiH == AksiHama.menyerang)
	{
	himpunanMenyerang[indexAksi] = setValue;
	}

34	Aggregasi(); }
35	float ReturnWhoisMinimum(float nilaiDarahHama, float nilaiKekuatanHama) {

36	if (nilaiDarahHama < nilaiKekuatanHama)
	{
	return nilaiDarahHama;
	}

37	else
	{
	return nilaiKekuatanHama;
	}
	}
	}

a. Flowgraph Penelusuran Algoritma Fuzzy Sugeno Pada Hama



Gambar Error! No text of specified style in document.**.12Flow Graph Algoritma Fuzzy Sugeno Pada Hama**

b. Cyclomatic Complexity (VG)

$$V(G) = E - N + P$$

V(G) = 135 – 137 + 0

V(G) = 1 + 0 = 1

Keterangan : E = Jumlah aksi

N = Jumlah Kondisi

P = Jumlah bagian dari graph yang tidak terkoneksi

c. Independent Path

Path 1 = 1 – 2 – 3 – 4 – 5 – 7 – 8 – 17 – 18 – 19 – 20 – 21 – 22 – 23 – 24 – 25 – 26 – 27 – 28 – 29 – 30 – 31 – 32 – 33 – 35 – 36 – 37 – 38 – 39 – 40 – 41 – 41 – 43 – 44 – 45 -46 – 47 – 48 -49 -50 – 51 – 52 – 53 – 54 – 55 -56 - 57 -58 – 59 – 60 – 61 – 62 – 63 – 64 – 65 – 66 – 67 – 68 – 69 – 70 – 71 – 72 – 73 – 74 – 75 – 76 – 77 – 78 – 79 -80 – 81 – 82 – 83 – 84 – 85 – 86 – 87 – 89 – 90 – 91 – 92 – 93 – 94 – 95 – 96 – 97 – 98 -99 – 100 – 101 – 102 – 103 – 104 – 105 – 106 – 107 – 108 – 109 – 110 – 112 – 113 – 114 – 115 – 116 – 117 – 118 – 119 -120 – 121 -122 – 123 -124 -125 – 126 – 127 – 128 – 129 – 130 – 131 – 132 – 134 – 135 – 136 – 137

Path 2 = 1 – 2 – 3 – 4 – 5 – 7 – 8 – 17 – 18 – 19 – 20 – 21 – 22 – 23 – 24 – 25 – 26 – 27 – 28 – 29 – 30 – 31 – 32 – 33 – 35 – 36 – 37 – 38 – 39 – 40 – 41 – 41 – 43 – 44 – 45 -46 – 47 – 48 -49 -50 – 51 – 52 – 53 – 54 – 55 -56 - 57 -58 – 59 – 60 – 61 – 62 – 63 – 64 – 65 – 66 – 67 – 68 – 69 – 70 – 71 – 72 – 73 – 74 – 75 – 76 – 77 – 78 – 79 -80 – 81 – 82 – 83 – 84 – 85 – 86 – 87 – 89 – 90 – 91 – 92 – 93 – 94 – 95 – 96 – 97 – 98 -99 – 100 – 101 – 102 – 103 – 104 – 105 – 106 – 107 – 108 – 109 – 110 – 112 – 113 – 114 – 115 – 116 – 117 – 118 – 119 -120 – 121 -122 – 123 -124 -125 – 126 – 127 – 128 – 129 – 130 – 131 – 132 – 134 – 135 – 136 - 137

Path 3 = 1 – 2 – 3 – 4 – 5 – 7 – 8 – 17 – 18 – 19 – 20 – 21 – 22 – 23 – 24 – 25 – 26 – 27 – 28 – 29 – 30 – 31 – 32 – 33 – 35 – 36 – 37 – 38 – 39 – 40 – 41 – 41 – 43 – 44 – 45 -46 – 47 – 48 -49 -50 – 51 – 52 – 53 – 54 – 55 -56 - 57 -58 – 59 – 60 – 61 – 62 – 63 – 64 – 65 – 66 – 67 – 68 – 69 – 70 – 71 – 72 – 73 – 74 – 75 – 76 – 77 – 78 – 79 -80 – 81 – 82 – 83 – 84 – 85 – 86 – 87 – 89 – 90 – 91 – 92 – 93 – 94 – 95 – 96 – 97 – 98 -99 – 100 – 101 – 102 – 103 – 104 – 105 – 106 – 107 – 108 – 109 – 110 – 112 – 113 – 114 –

115 – 116 – 117 – 118 – 119 -120 – 121 -122 – 123 -124 -125 – 126 – 127 – 128 – 129 – 130 – 131 – 132 – 134 – 135 – 136 - 137

Path 4 = 1 – 2 – 3 – 4 – 5 – 7 – 8 – 17 – 18 – 19 – 20 – 21 – 22 – 23 – 24 – 25 – 26 – 27 – 28 – 29 – 30 – 31 – 32 – 33 – 35 – 36 – 37 – 38 – 39 – 40 – 41 – 41 – 43 – 44 – 45 -46 – 47 – 48 -49 -50 – 51 – 52 – 53 – 54 – 55 -56 - 57 -58 – 59 – 60 – 61 – 62 – 63 – 64 – 65 – 66 – 67 – 68 – 69 – 70 – 71 – 72 – 73 – 74 – 75 – 76 – 77 – 78 – 79 -80 – 81 – 82 – 83 – 84 – 85 – 86 – 87 – 89 – 90 – 91 – 92 – 93 – 94 – 95 – 96 – 97 – 98 -99 – 100 – 101 – 102 – 103 – 104 – 105 – 106 – 107 – 108 – 109 – 110 – 112 – 113 – 114 – 115 – 116 – 117 – 118 – 119 -120 – 121 -122 – 123 -124 -125 – 126 – 127 – 128 – 129 – 130 – 131 – 132 – 134 – 135 – 136 - 137

Path 5 = 1 – 2 – 3 – 4 – 5 – 7 – 8 – 17 – 18 – 19 – 20 – 21 – 22 – 23 – 24 – 25 – 26 – 27 – 28 – 29 – 30 – 31 – 32 – 33 – 35 – 36 – 37 – 38 – 39 – 40 – 41 – 41 – 43 – 44 – 45 -46 – 47 – 48 -49 -50 – 51 – 52 – 53 – 54 – 55 -56 - 57 -58 – 59 – 60 – 61 – 62 – 63 – 64 – 65 – 66 – 67 – 68 – 69 – 70 – 71 – 72 – 73 – 74 – 75 – 76 – 77 – 78 – 79 -80 – 81 – 82 – 83 – 84 – 85 – 86 – 87 – 89 – 90 – 91 – 92 – 93 – 94 – 95 – 96 – 97 – 98 -99 – 100 – 101 – 102 – 103 – 104 – 105 – 106 – 107 – 108 – 109 – 110 – 112 – 113 – 114 – 115 – 116 – 117 – 118 – 119 -120 – 121 -122 – 123 -124 -125 – 126 – 127 – 128 – 129 – 130 – 131 – 132 – 134 – 135 – 136 – 137

Path 6 = 1 – 2 – 3 – 4 – 5 – 7 – 8 – 17 – 18 – 19 – 20 – 21 – 22 – 23 – 24 – 25 – 26 – 27 – 28 – 29 – 30 – 31 – 32 – 33 – 35 – 36 – 37 – 38 – 39 – 40 – 41 – 41 – 43 – 44 – 45 -46 – 47 – 48 -49 -50 – 51 – 52 – 53 – 54 – 55 -56 - 57 -58 – 59 – 60 – 61 – 62 – 63 – 64 – 65 – 66 – 67 – 68 – 69 – 70 – 71 – 72 – 73 – 74 – 75 – 76 – 77 – 78 – 79 -80 – 81 – 82 – 83 – 84 – 85 – 86 – 87 – 89 – 90 – 91 – 92 – 93 – 94 – 95 – 96 – 97 – 98 -99 – 100 – 101 – 102 – 103 – 104 – 105 – 106 – 107 – 108 – 109 – 110 – 112 – 113 – 114 – 115 – 116 – 117 – 118 – 119 -120 – 121 -122 – 123 -124 -125 – 126 – 127 – 128 – 129 – 130 – 131 – 132 – 134 – 135 – 136 - 137

Path 7 = 1 – 2 – 3 – 4 – 5 – 7 – 8 – 17 – 18 – 19 – 20 – 21 – 22 – 23 – 24 – 25 – 26 – 27 – 28 – 29 – 30 – 31 – 32 – 33 – 35 – 36 – 37 – 38 – 39 – 40 –

41 – 41 – 43 – 44 – 45 -46 – 47 – 48 -49 -50 – 51 – 52 – 53 – 54 – 55 -56 - 57 -58 – 59 – 60 – 61 – 62 – 63 – 64 – 65 – 66 – 67 – 68 – 69 – 70 – 71 – 72 – 73 – 74 – 75 – 76 – 77 – 78 – 79 -80 – 81 – 82 – 83 – 84 – 85 – 86 – 87 – 89 – 90 – 91 – 92 – 93 – 94 – 95 – 96 – 97 – 98 -99 – 100 – 101 – 102 – 103 – 104 – 105 – 106 – 107 – 108 – 109 – 110 – 112 – 113 – 114 – 115 – 116 – 117 – 118 – 119 -120 – 121 -122 – 123 -124 -125 – 126 – 127 – 128 – 129 – 130 – 131 – 132 – 134 – 135 – 136 - 137

Path 8 = 1 – 2 – 3 – 4 – 5 – 7 – 8 – 17 – 18 – 19 – 20 – 21 – 22 – 23 – 24 – 25 – 26 – 27 – 28 – 29 – 30 – 31 – 32 – 33 – 35 – 36 – 37 – 38 – 39 – 40 – 41 – 41 – 43 – 44 – 45 -46 – 47 – 48 -49 -50 – 51 – 52 – 53 – 54 – 55 -56 - 57 -58 – 59 – 60 – 61 – 62 – 63 – 64 – 65 – 66 – 67 – 68 – 69 – 70 – 71 – 72 – 73 – 74 – 75 – 76 – 77 – 78 – 79 -80 – 81 – 82 – 83 – 84 – 85 – 86 – 87 – 89 – 90 – 91 – 92 – 93 – 94 – 95 – 96 – 97 – 98 -99 – 100 – 101 – 102 – 103 – 104 – 105 – 106 – 107 – 108 – 109 – 110 – 112 – 113 – 114 – 115 – 116 – 117 – 118 – 119 -120 – 121 -122 – 123 -124 -125 – 126 – 127 – 128 – 129 – 130 – 131 – 132 – 134 – 135 – 136 - 137

Path9 = 1 – 2 – 3 – 4 – 5 – 7 – 8 – 17 – 18 – 19 – 20 – 21 – 22 – 23 – 24 – 25 – 26 – 27 – 28 – 29 – 30 – 31 – 32 – 33 – 35 – 36 – 37 – 38 – 39 – 40 – 41 – 41 – 43 – 44 – 45 -46 – 47 – 48 -49 -50 – 51 – 52 – 53 – 54 – 55 -56 - 57 -58 – 59 – 60 – 61 – 62 – 63 – 64 – 65 – 66 – 67 – 68 – 69 – 70 – 71 – 72 – 73 – 74 – 75 – 76 – 77 – 78 – 79 -80 – 81 – 82 – 83 – 84 – 85 – 86 – 87 – 89 – 90 – 91 – 92 – 93 – 94 – 95 – 96 – 97 – 98 -99 – 100 – 101 – 102 – 103 – 104 – 105 – 106 – 107 – 108 – 109 – 110 – 112 – 113 – 114 – 115 – 116 – 117 – 118 – 119 -120 – 121 -122 – 123 -124 -125 – 126 – 127 – 128 – 129 – 130 – 131 – 132 – 134 – 135 – 136 - 137

Tabel Error! No text of specified style in document..**5Matriks penelusuran Algoritma Fuzzy Pada Hama Tanaman**

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
1		1				1																			
2			1													1									

3				1																		1					
4					1														1								
5							1								1												
6																											
7								1	1	1	1	1	1	1	1	1											
8																	1										
9																	1										
10																	1										
11																	1										
12																	1										
13																	1										
14																	1										
15																	1										
16																	1										
17																		1									
18																			1								
19																				1							
20																					1						
21																						1					
22																							1				
23																								1			
24																									1		
25																											
26																											
27																											
28																											
29						1																					
30									1																		

V(G) = jumlah_matriks + 1

= 6 + 1

= 7

Setelah mendapatkan Cyclomatic Complexity dari graph matriks, perlu dicari nilai Cyclomatic Complexity berdasarkan predicate node (P), yaitu jumlah simpul yang memiliki lebih dari satu busur. Berdasarkan Gambar 4.4, simpul-simpul yang memiliki jumlah busur lebih dari satu ialah 1, 2, 3, 4, 5, dan 7,sehingga Predicate node (P) = 6

Setelah nilai P didapat, untuk mendapatkan nilai Cyclomatic Complexity dari predicate node, rumus dan perhitungan yang digunakan ialah

V(G) = P + 1

= 6 + 1

= 7

Jadi, nilai hasil dari graph matriks dan predicate node masing-masing adalah 7

4.5.2 Pengujian Beta

Pengujian beta merupakan pengujian yang dilakukan secara objektif, dimana pengujian langsung melibatkan para pengguna aplikasi game secara langsung dengan menggunakan kuesioner mengenai kepuasan para pengguna terhadap aplikasi game yang dibangun. metode penilaian pengujian yang digunakan adalah metode kuantitatif berdasarkan data dari pengguna.

Kuesioner diberikan kepada beberapa calon pengguna perangkat lunak yang akan dibangun. Adapun beberapa poin yang diteliti pada Game merawat tanman adalah sebagai berikut:

1. Seberapa jauh aplikasi ini membantu anak untuk mengenal tanman sayuran.
2. Seberapa efektif aplikasi ini membantu anak dalam memahami tentang merawat tanaman.

3. Seberapa efektif aplikasi ini membantu anak dalam memberikan pengetahuan.

Skala Jawaban: I : Iya
T : Tidak
TT : Tidak Tahu
TS : Tidak Setuju
STJ : Sangat Tidak Setuju

Berikut ini adalah beberapa pertanyaan yang diajukan kepada anak adalah sebagai berikut:

Berikut ini adalah beberapa pertanyaan yang diajukan kepada pengguna adalah sebagai berikut:

Tabel Error! No text of specified style in document.**.6Pertanyaan Untuk Anak**

O	**PERTANYAAN**	**JAWABAN**				
				T	**S**	**TJ**
	Apakah dengan Game ini adik – adik mendapatkan pengalaman bercocock tanam ?	0	6			
	Apakah informasi penjelasan tentang tanaman yang disampaikan diawal permainan mudah untuk dimengerti ?	0	1			
	Apakah pertanyaan yang ada didalam game ini membantu adik – adik untuk mengenal tanman sayuran tomat, seledri, daun bawang ?	8	4			
	Apakah game merawat tanaman					

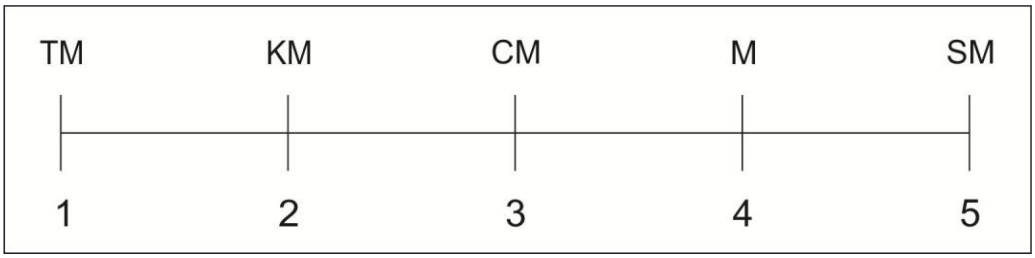
	ini mudah untuk dimainkan ?	2	6			
	Apakah setelah memainkan game ini adik – adik ingin memainkannya kembali ?	7	0			-

4.5.2.1 Kesimpulan Pengujian Beta

Berdasarkan hasil kuesioner yang diberikan kepada anak, dapat dihitung tingkat persetujuan rata-rata responden terhadap masing-masing jawaban. Berikut ini adalah skor yang diberikan untuk jawaban kuisioner yang telah diberikan kepada penilitian para penulis:

1. Jawaban I diberi nilai 5
2. Jawaban T diberi nilai 4
3. Jawaban TT diberi nilai 3
4. Jawaban TS diberi nilai 2
5. Jawaban STJ diberi nilai 1

Secara kontinum dapat digambarkan sebagai berikut:



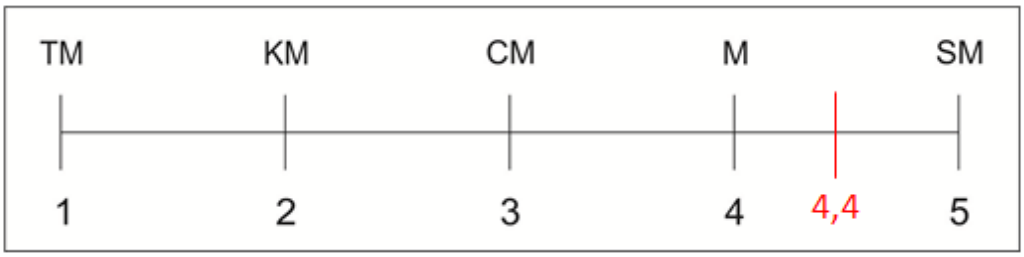
1. Pengolahan Pertanyaan No.1

Tabel Error! No text of specified style in document.**.7Hasil Kuesioner pertanyaan No.1**

o	**Pertanyaan**	**ilai**	**Resp onden**	**kor**	**Rata -Rata Setiap Soal**	**Kep utusan**
	Apakah dengan Game ini adik – adik mendapatkan		20	00	4,4	Bera da diantara skala membantu dan sangat
			16	4		

	pengalaman bercocock tanam ?		4	2		membantu
			0			
			0			
	Jumlah		40	76		

Jika semua responden (40 orang responden) menjawab SM, maka jumlah skor ideal (kriterium) untuk seluruh item adalah 5 (diperoleh dari 200 : 40). Berdasarkan hasil penelitian, jumlah skor yang diperoleh adalah 176. Jadi berdasarkan data tersebut, maka tingkat persetujuan rata-rata responden terhadap pertanyaan nomor (1) adalah (176: 40) = 4,4 dari yang diharapkan (5) Jadi berdasarkan pengolahan data yang diperoleh dari 40 responden, maka 4,4 terletak pada daerah antara membantu dengan sangat membantu. Secara kontinum dapat digambarkan sebagai berikut:



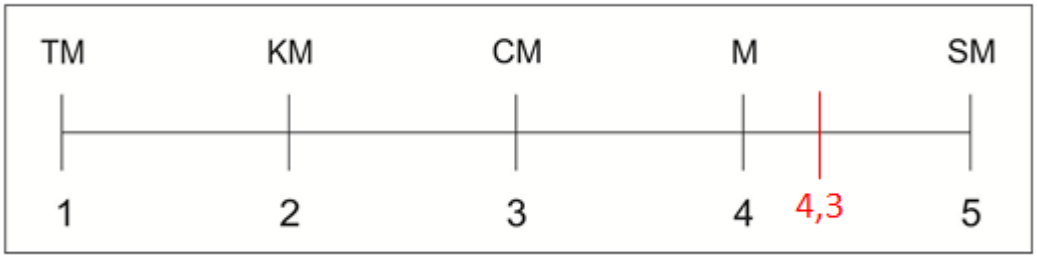
2. Pengolahan Pertanyaan No.2

Tabel Error! No text of specified style in document.**.8Hasil Kuesioner pertanyaan No.2**

o	**Pertanyaan**	**ilai**	**Resp onden**	**kor**	**Rata -Rata Setiap Soal**	**Kep utusan**
	Apakah informasi penjelasan tentang tanaman yang disampaikan		20	00	4,3	Bera da diantara skala membantu dan sangat
			11	4		

	diawal permainan mudah untuk dimengerti ?		9	7		membantu
			0			
			0			
	Jumlah		40	71		

Jika semua responden (40 orang responden) menjawab SM, maka jumlah skor ideal (kriterium) untuk seluruh item adalah 5 (diperoleh dari 200 : 40). Berdasarkan hasil penelitian, jumlah skor yang diperoleh adalah 171. Jadi berdasarkan data tersebut, maka tingkat persetujuan rata-rata responden terhadap pertanyaan nomor (2) adalah (171: 40) = 4,3 dari yang diharapkan (5) Jadi berdasarkan pengolahan data yang diperoleh dari 40 responden, maka 4,3 terletak pada daerah antara membantu dengan sangat membantu. Secara kontinum dapat digambarkan sebagai berikut:



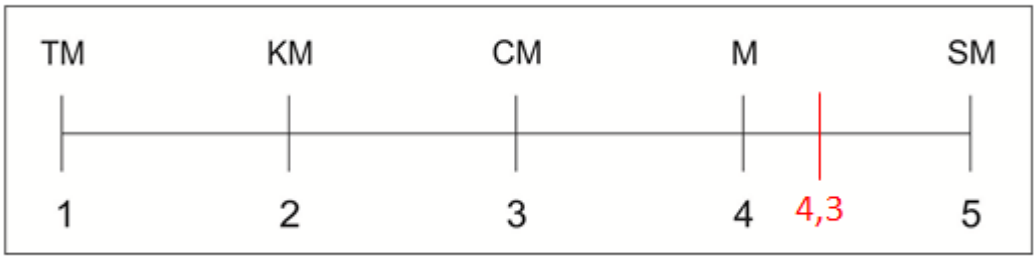
3. Pengolahan Pertanyaan No.3

Tabel Error! No text of specified style in document.**.9Hasil kuesioner pertanyaan No.3**

o	**Pertanyaan**	**ilai**	**Respondent**	**kor**	**Rata-Rata Setiap Soal**	**Keputusan**
	Apakah pertanyaan yang ada didalam game ini membantu adik –		18	0	4,3	Berada diantara skala membantu dan sangat
			14	6		

	adik untuk mengenal tanman sayuran tomat, seledri, daun bawang ?		8	4		membantu
			0			
			0			
	Jumlah		40	70		

Jika semua responden (40 orang responden) menjawab SM, maka jumlah skor ideal (kriterium) untuk seluruh item adalah 5 (diperoleh dari 200 : 40). Berdasarkan hasil penelitian, jumlah skor yang diperoleh adalah 170. Jadi berdasarkan data tersebut, maka tingkat persetujuan rata-rata responden terhadap pertanyaan nomor (3) adalah (170: 40) = 4,3 dari yang diharapkan (5) Jadi berdasarkan pengolahan data yang diperoleh dari 40 responden, maka 4,3 terletak pada daerah antara membantu dengan sangat membantu. Secara kontinum dapat digambarkan sebagai berikut:



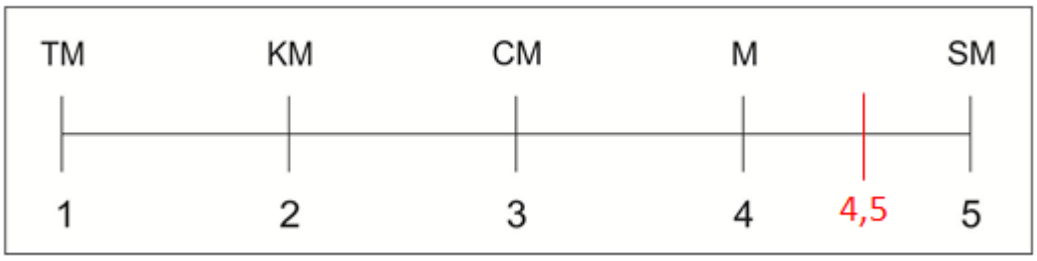
4. Pengolahan Pertanyaan No.4

Tabel Error! No text of specified style in document.**.10Hasil kuesioner pertanyaan No.4**

o	**Pertanyaan**	**ilai**	**Resp onden**	**kor**	**Rata -Rata Setiap Soal**	**Kep utusan**
	Apakah game merawat tanaman ini mudah untuk		22	10	4,5	Bera da diantara skala membantu dan sangat
			16	4		

	dimainkan ?		2			membantu
			0			
			0			
	Jumlah		40	80		

Jika semua responden (40 orang responden) menjawab SM, maka jumlah skor ideal (kriterium) untuk seluruh item adalah 5 (diperoleh dari 200 : 40). Berdasarkan hasil penelitian, jumlah skor yang diperoleh adalah 180. Jadi berdasarkan data tersebut, maka tingkat persetujuan rata-rata responden terhadap pertanyaan nomor (4) adalah (180: 40) = 4,5 dari yang diharapkan (5) Jadi berdasarkan pengolahan data yang diperoleh dari 40 responden, maka 4,5 terletak pada daerah antara membantu dengan sangat membantu. Secara kontinum dapat digambarkan sebagai berikut:



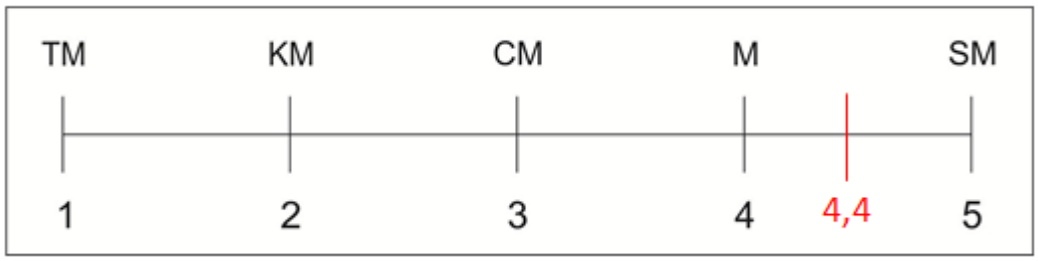
5. Pengolahan Pertanyaan No.5

Tabel Error! No text of specified style in document.**.11Hasil kuesioner pertanyaan No.5**

o	**Pertanyaan**	**ilai**	**Resp onden**	**kor**	**Rata -Rata Setiap Soal**	**Kep utusan**
	Apakah setelah memainkan game ini adik – adik ingin memainkannya		17	5	4,4	Bera da diantara skala membantu dan sangat membantu
			20	0		
			3			

	kembali ?		0			
			0			
	Jumlah		40	74		

Jika semua responden (40 orang responden) menjawab SM, maka jumlah skor ideal (kriterium) untuk seluruh item adalah 5 (diperoleh dari 200 : 40). Berdasarkan hasil penelitian, jumlah skor yang diperoleh adalah 174. Jadi berdasarkan data tersebut, maka tingkat persetujuan rata-rata responden terhadap pertanyaan nomor (5) adalah (174: 40) = 4,4 dari yang diharapkan (5) Jadi berdasarkan pengolahan data yang diperoleh dari 40 responden, maka 4,4 terletak pada daerah antara membantu dengan sangat membantu. Secara kontinum dapat digambarkan sebagai berikut:



Berdasarkan hasil perhitungan diatas, maka dapat dapat diambil kesimpulan dari hasil perhitungan rata-rata tingkat persetujuan responden dari semua pertanyaan yang diajukan kepada beberapa guru dan siswa. Dengan demikian nilai perhitungan tersebut berarti bahwa sistem ini membantu dalam proses belajar mengajar dengan adanya aplikasi ini.