BAB 4
IMPLEMENTASI DAN PENGUJIAN

4.1 Implementasi

Implementasi merupakan tahap dari aplikasi yang sudah dibangun dan dirancang. Selanjutnya dilakukan uji kelayakan terhadap sistem aplikasi yang telah dibangun sehingga dapat memberikan masukan demi berkembangnya sistem aplikasi yang telah dibangun. Adapun sistem aplikasi yang telah dibangun adalah Pembangunan Game Sugie Man Penyelamat *Sugar Glider*. Selain itu, dibutuhkan perangkat keras dan perangkat lunak yang digunakan untuk melakukan implementasi dan pengujian terhadap aplikasi yaitu sebagai berikut :

4.1.1 Implementasi Perangkat Keras

Perangkat keras yang dapat digunakan untuk menjalankan aplikasi *game* ini yaitu sebagai berikut :

Tabel 4.1 Implementasi Perangkat Keras

No	Perangkat Keras	Spesifikasi
1	*Hardisk*	40 Giga
2	*Keyboard*	Standar
3	*Mouse*	Standar
4	*Processor*	*Processor Pentium 4*
5	*RAM*	1 Giga
6	*Speaker*	Standar
7	VGA	VGA *Card 128*MB

4.1.2 Implementasi Perangkat Lunak

Adapun spesifikasi perangkat lunak yang dibutuhkan untuk menjalankan aplikasi Pembangunan Game Sugie Man Penyelamat *Sugar Glider* adalah :

1. Sistem Operasi Windows Xp.
2. Adobe Flash.

4.1.3 Implementasi Instalasi Aplikasi *Game*

Aplikasi Pembangunan Game Sugie Man Penyelamat *Sugar Glider* merupakan sebuah *game* adventure yang mengenalkan dan memberikan

pengetahuan mengenai *Sugar Glider*. Pembangunan *Game* Sugie Man Penyelamat *Sugar Glider* dapat dijalankan pada sistem operasi windows dan harus sudah *terinstal* Adobe Flash. Apabila Adobe Flash belum *terinstal*, maka *Game* Sugie Man Penyelamat *Sugar Glider* tidak dapat dimainkan, dikarenakan file aplikasi *game* ini berekstensi .swf yang dapat berjalan dengan dukungan Adobe Flash.

4.1.4 Implementasi Antarmuka

Antarmuka dapat membuat pengguna merasa betah berada dalam aplikasi atau sebaliknya. Berikut ini merupakan antarmuka yang ada pada aplikasi permainan *Game* Sugie Man Penyelamat *Sugar Glider*:

1. **Antarmuka Menu Permainan**

   Saat pengguna membuka aplikasi pertama kali, maka antarmuka yang ditampilkan oleh sistem adalah antarmuka menu permainan. Berikut ini merupakan antarmuka menu permainan yang dapat dilihat pada gambar 4.1.



Gambar 4.1 Antarmuka Menu Permainan

Gambar diatas merupakan menu permainan pada aplikasi permainan Sugie Man Penyelamat *Sugar Glider*. Terdapat 4 sub menu di dalamnya, yaitu main game yang berfungsi untuk memulai permainan, cara bermain berfungsi untuk melihat petunjuk permainan, keluar berfungsi untuk keluar dari permainan, dan pembuat berfungsi untuk melihat biodata pemilik *game* ini.

2. Antarmuka Intro

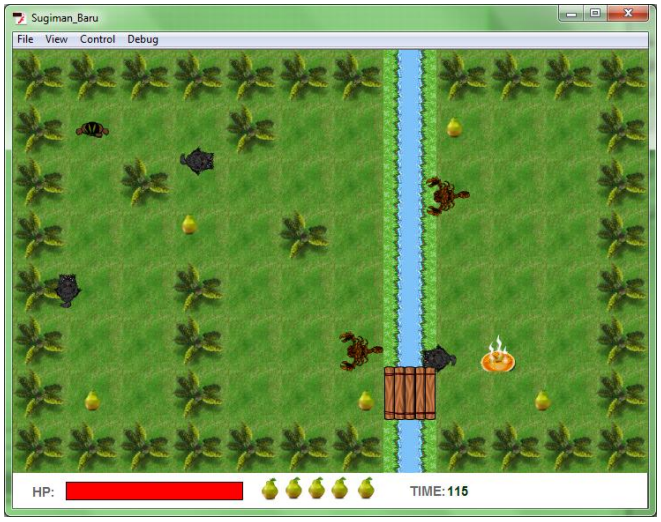
Setelah pengguna memasukan nama pemain, pengguna akan disuguhkan dengan ilustrasi singkat pada permainan Sugie Man Penyelamat *Sugar Glider* (*Intro*). Tujuan dari antarmuka intro ini yaitu untuk memberikan gambaran alur cerita dalam permainan ini sehingga pemain mengetahui misinya dalam permainan, dan juga terdapat informasi mengenai *Sugar Glider*. Antarmuka intro dapat dilihat pada gambar 4.2.



Gambar 4.2 Antarmuka Intro

3. Antarmuka Permainan

Antarmuka *Permainan* merupakan tampilan ketika pemain berada dalam permainan Sugie Man Penyelamat *Sugar Glider*. Terdapat 3 (tiga) *stage* dalam permainan ini, yaitu *stage* 1 hutan, *stage* 2 rawa, dan *stage* 3 rumah bos pemburu liar. Tampilan dari antarmuka permainan dapat dilihat pada gambar berikut.



Gambar 4.3 Antarmuka Permainan level 1



Gambar 4.4 Antarmuka Permainan level 2



Gambar 4.5 Antarmuka Permainan level 3

4. Antarmuka Cara Bermain

Antarmuka cara main merupakan tampilan ketika pemain memilih menu cara bermain pada menu permainan. Tampilan dari antarmuka cara bermain dapat dilihat pada gambar 4.6.



Gambar 4.6 Antarmuka Cara Bermain

5. Antarmuka tentang

Antarmuka tentang merupakan tampilan biodata pembuat. Tampilan dari antarmuka tentang dapat dilihat pada gambar 4.7.



Gambar 4.7 Antarmuka tentang

4.2 Pengujian

Pengujian sistem merupakan hal terpenting yang dilakukan untuk menemukan kekurangan atau kesalahan pada perangkat lunak yang diuji. Pengujian dimaksudkan untuk mengetahui perangkat lunak yang dibuat sudah memenuhi kriteria yang sesuai dengan perancangan perangkat lunak tersebut. Adapun pengujian yang akan dilakukan adalah pengujian alpa dan pengujian beta.

4.2.1 Pengujian Alpa

Pada tahap ini merupakan pengujian program yang dilakukan oleh pembuat game ataupun orang-orang yang terlibat didalamnya. pengujian alpha hanya untuk sirkulasi internal dan masalah *(error)* atau ketidaklengkapan yang terdapat dalam game. Proses pengujian terhadap aplikasi akan menggunakan *black-box testing* dan *white-box testing*.

Pengujian pada *black-box testing* dimaksudkan untuk mengetahui apakah fungsi-fungsi, masukan, dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan.

Pengujian pada *white-box testing* dimaksudkan untuk menguji perangkat lunak dari segi desain dan kode program apakah mampu menghasilkan fungsi-fungsi, masukan, dan keluaran yang sesuai dengan spesifikasi kebutuhan.

4.2.1.1 Skenario Pengujian Aplikasi

Skenario pengujian menjelaskan pengujian terhadap sistem yang ada pada aplikasi game Sugie Man Penyelamat *Sugar Glider*. Skenario pengujian dari aplikasi permainan ini dapat dilihat pada tabel 4.2

Tabel 4.2 Skenario Pengujian Aplikasi

No	Komponen yang di uji	Butir Uji	Jenis Uji
1	Menampilkan Level 1	Menyentuh buah pear	*Black box*
		Menyentuh makanan khas papua	*Black box*
		Menekan navigasi atas	*Black box*
		Menekan navigasi bawah	*Black box*
		Menekan navigasi kanan	*Black box*
		Menekan navigasi kiri	*Black box*
2	Menampilkan Level 2	Menyentuh buah melon	*Black box*
		Menyentuh makanan khas papua	*Black box*
		Menyentuh kadal	*Black box*
		Menekan navigasi atas	*Black box*
		Menekan navigasi bawah	*Black box*
		Menekan navigasi kanan	*Black box*
		Menekan navigasi kiri	*Black box*
3	Menampilkan level 3	Menyentuh buah apel	*Black box*
		Menyentuh makanan khas papua	*Black box*
		Menyentuh sugar glider	*Black box*
		Menekan navigasi atas	*Black box*
		Menekan navigasi bawah	*Black box*
		Menekan navigasi kanan	*Black box*
		Menekan navigasi kiri	*Black box*
4	Gerak Karakter	Menekan navigasi atas	*Black box*
		Menekan navigasi bawah	*Black box*
		Menekan navigasi kanan	*Black box*
		Menekan navigasi kiri	*Black box*
5	Menyentuh Buah	Menyentuh buah pear, melon dan apel	*Black box*
6	Menyentuh Makanan Khas Papua	Menyentuh Makanan Khas Papua	*Black box*
7	Menyelamatkan Hewan Yang Terperangkap	Menyelamatkan Hewan Yang Terperangkap	*Black box*
8	Perhitungan skor	Tampil informasi skor	*Black box*

4.2.1.2 Kasus dan Hasil Pengujian (*Black Box*)

Berikut ini merupakan kasus untuk menguji perangkat lunak yang sudah dibangun dengan menggunakan metode *black box* berdasarkan skenario pengujian aplikasi yang terdapat pada tabel 4.2 yang sudah dijabarkan. Hasil pengujian yang akan dilakukan pada aplikasi ini selengkapnya dapat dilihat pada tabel 4.3.

Tabel 4.3 Skenario hasil pengujian

Kasus Dan Hasil Uji				
No	**Kasus/uji**	**Sekenario Uji**	**Hasil yang diharapkan**	**Hasil Pengujian**
1	Menampilkan Level 1	Menyentuh buah pear	Jumlah buah pear bertambah	[√] Berhasil [ ] Tidak Berhasil
		Menyentuh makanan khas papua	Darah bertambah	[√] Berhasil [ ] Tidak Berhasil
		Menekan navigasi atas	Karakter berjalan ke atas	[√] Berhasil [ ] Tidak Berhasil
		Menekan navigasi bawah	Karakter berjalan ke bawah	[√] Berhasil [ ] Tidak Berhasil
		Menekan navigasi kanan	Karakter berjalan ke kanan	[√] Berhasil [ ] Tidak Berhasil
		Menekan navigasi kiri	Karakter berjalan ke kiri	[√] Berhasil [ ] Tidak Berhasil
2	Menampilkan Level 2	Menyentuh buah melon	Jumlah buah melon bertambah	[√] Berhasil [ ] Tidak Berhasil
		Menyentuh makanan khas papua	Darah bertambah	[√] Berhasil [ ] Tidak Berhasil
		Menyentuh kadal	Kadal menghilang	[√] Berhasil [ ] Tidak Berhasil
		Menekan navigasi atas	Karakter berjalan ke atas	[√] Berhasil [ ] Tidak Berhasil
		Menekan navigasi bawah	Karakter berjalan ke bawah	[√] Berhasil [ ] Tidak Berhasil
		Menekan navigasi kanan	Karakter berjalan ke kanan	[√] Berhasil [ ] Tidak Berhasil
		Menekan navigasi kiri	Karakter berjalan ke kiri	[√] Berhasil [ ] Tidak Berhasil
3	Menampilkan level 3	Menyentuh buah apel	Jumlah buah apel bertambah	[√] Berhasil [ ] Tidak Berhasil
		Menyentuh makanan khas papua	Darah bertambah	[√] Berhasil [ ] Tidak Berhasil

		Menyentuh sugar glider	Sugar glider menghilang	[√] Berhasil [ ] Tidak Berhasil
		Menekan navigasi atas	Karakter berjalan ke atas	[√] Berhasil [ ] Tidak Berhasil
		Menekan navigasi bawah	Karakter berjalan ke bawah	[√] Berhasil [ ] Tidak Berhasil
		Menekan navigasi kanan	Karakter berjalan ke kanan	[√] Berhasil [ ] Tidak Berhasil
		Menekan navigasi kiri	Karakter berjalan ke kiri	[√] Berhasil [ ] Tidak Berhasil
4	Gerak Karakter	Menekan navigasi atas	Karakter berjalan ke atas	[√] Berhasil [ ] Tidak Berhasil
		Menekan navigasi bawah	Karakter berjalan ke bawah	[√] Berhasil [ ] Tidak Berhasil
		Menekan navigasi kanan	Karakter berjalan ke kanan	[√] Berhasil [ ] Tidak Berhasil
		Menekan navigasi kiri	Karakter berjalan ke kiri	[√] Berhasil [ ] Tidak Berhasil
5	Menyentuh Buah	Menyentuh buah pear, melon dan apel	Jumlah buah bertambah pear, melon dan apel	[√] Berhasil [ ] Tidak Berhasil
6	Menyentuh Makanan Khas Papua	Menyentuh Makanan Khas Papua	Darah bertambah	[√] Berhasil [ ] Tidak Berhasil
7	Menyelamatkan Hewan Yang Terperangkap	Menyelamatkan Hewan Yang Terperangkap	Hewan yang terperangkap menghilang	[√] Berhasil [ ] Tidak Berhasil
8	Perhitungan skor	Tampil skor	Menampilkan skor	[√] Berhasil [ ] Tidak Berhasil

Kesimpulan

Berdasarkan hasil pengujian yang dilakukan, dapat disimpulkan bahwa sistem sudah berjalan seperti yang diharapkan dan secara fungsional sudah dapat menghasilkan keluaran yang telah diharapkan.

4.2.1.3 Kasus dan Hasil Pengujian (*White Box*)

Dibawah ini merupakan kasus algoritma logika fuzzy untuk menguji perangkat lunak yang sudah dibangun dengan menggunakan metode *white box*, berdasarkan skenario pengujian aplikasi yang dijabarkan.

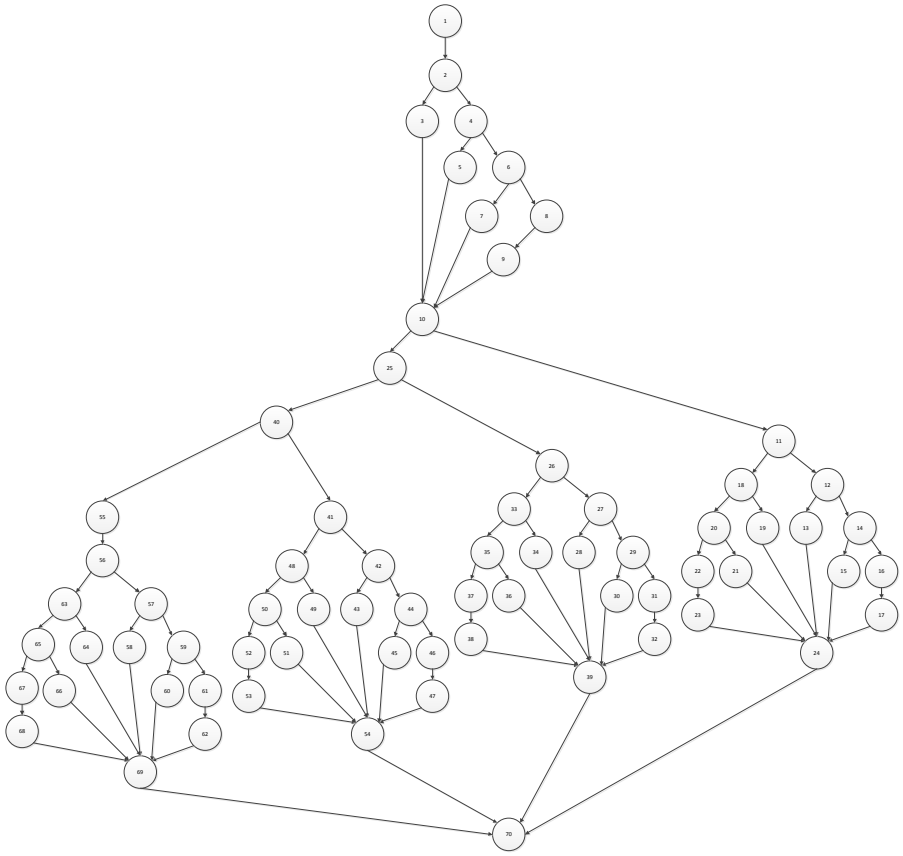
a. Berikut merupakan logika *fuzzy* pada bos pemburu liar level 3:

```
1       keberanian = runFuzzy(getSerangan(), Main.HPBos);
2        if(keberanian > 80){
3           speed = 4;
4       }else if(keberanian > 60){
5            speed = 3;
6        }else if(keberanian > 40){
7           speed = 2;
8        }else if(keberanian > 20){
9           speed = 1; }
10           if(downPressed){
11               if(keberanian > 60){
12                  if(x < Main.player.x){
13                     rightPressed = true;
14                  }else if(x > Main.player.x){
15                     leftPressed = true;
16                  }else if(y > Main.player.y){
17                     upPressed = true; }
18              }else{ if(x > Main.player.x){
19                     rightPressed = true;
20                  }else if(x < Main.player.x){
21                    leftPressed = true;
22                  }else if(y < Main.player.y){
23                     upPressed = true; }}
24          downPressed = false;
25          }else if(rightPressed){
26                     if(keberanian > 60){
27                        if(y > Main.player.y){
28                           upPressed = true;
29                        }else if(y < Main.player.y){
```

```
30                          downPressed = true;
31                         }else if(x > Main.player.x){
32                          leftPressed = true; }
33                       }else{
33                        if(y < Main.player.y){
34                          upPressed = true;
35                        }else if(y > Main.player.y){
36                          downPressed = true;
37                        }else if(x < Main.player.x){
38                          leftPressed = true; }}
39                     rightPressed = false;
40          }else if(upPressed){
41                    if(keberanian > 60){
42                        if(x > Main.player.x){
43                          leftPressed = true;
44                        }else if(x < Main.player.x){
45                          rightPressed = true;
46                        }else if(y < Main.player.y){
47                          downPressed = true; }
48                    }else{ if(x < Main.player.x){
49                          leftPressed = true;
50                        }else if(x > Main.player.x){
51                          rightPressed = true;
52                        }else if(y > Main.player.y){
53                          downPressed = true; } }
54                    upPressed = false;
55          }else if(leftPressed){
56                    if(keberanian > 60){
57                        if(y < Main.player.y){
58                          downPressed = true;
59                        }else if(y > Main.player.y){
```

```
60                          upPressed = true;
61                      }else if(x < Main.player.x){
62                          rightPressed = true; }
63                  }else{ if(y > Main.player.y){
64                          downPressed = true;
65                      }else if(y < Main.player.y){
66                          upPressed = true;
67                      }else if(x > Main.player.x){
68                          rightPressed = true; }}
69                  leftPressed = false; }
70      END
```

b. *Flow Graph* Logika *Fuzzy*



Gambar 4.8 *Flow Graph Logika Fuzzy*

Cyclomatic Complexity V(G)

V(G) = E – N + 2

Ket :

E = Jumlah busur pada flowgraph

N = Jumlah simpul pada

flowgraph

Dik : E = 105

N = 70

V(G) = E – N + 2

= 105 – 70 + 2

= 37

Independent path

Independent Path adalah jalur yang melintasi atau melalui program dimana sekurang-kurangnya terdapat proses perintah yang baru atau kondisi yang baru. Berikut ini merupakan *Independent Path* dari logika fuzzy diatas:

Path 1 : 1-2-3-10

Path 2 : 1-2-4-5-10

Path 3 : 1-2-4-6-7-10

Path 4 : 1-2-4-6-8-10

Path 5 :1-2-4-6-8-9-10

Path 6 : 10-11-12-13-24-70

Path 7 : 10-11-12-14-15-24-70

Path 8 : 10-11-12-14-16-24-70

Path 9 : 10-11-12-14-16-17-24-70

Path 10 : 10-11-18-20-22-24-70

Path 11 : 10-11-18-20-22-23-24-70

Path 12 : 10-11-18-20-21-24-70

Path 13 : 10-11-18-19-24-70

Path 14 : 10-25-26-33-35-37-39-70

Path 15 : 10-25-26-33-35-37-38-39-70

Path 16 : 10-25-26-33-35-36-39-70

Path 17 : 10-25-26-33-34-39-70

Path 18 : 10-25-26-27-28-39-70

Path 19 : 10-25-26-27-29-30-39-70

Path 20 : 10-25-26-27-29-31-39-70

Path 21 : 10-25-26-27-31-32-39-70

Path 22 : 25-40-41-42-44-46-47-54-70

Path 23 : 25-40-41-42-44-46-54-70

Path 24 : 25-40-41-42-44-45-54-70

Path 25 : 25-40-41-42-43-54-70

Path 26 : 25-40-41-48-50-52-53-54-70

Path 27 : 25-40-41-48-50-52-54-70

Path 28 : 25-40-41-48-50-51-54-70

Path 29 : 25-40-41-48-49-54-70

Path 30 : 40-55-70

Path 31 : 40-55-56-63-65-67-68-69-70

Path 32 : 40-55-56-63-65-67-69-70

Path 33 : 40-55-56-63-65-66-69-70

Path 34 : 40-55-56-63-64-69-70

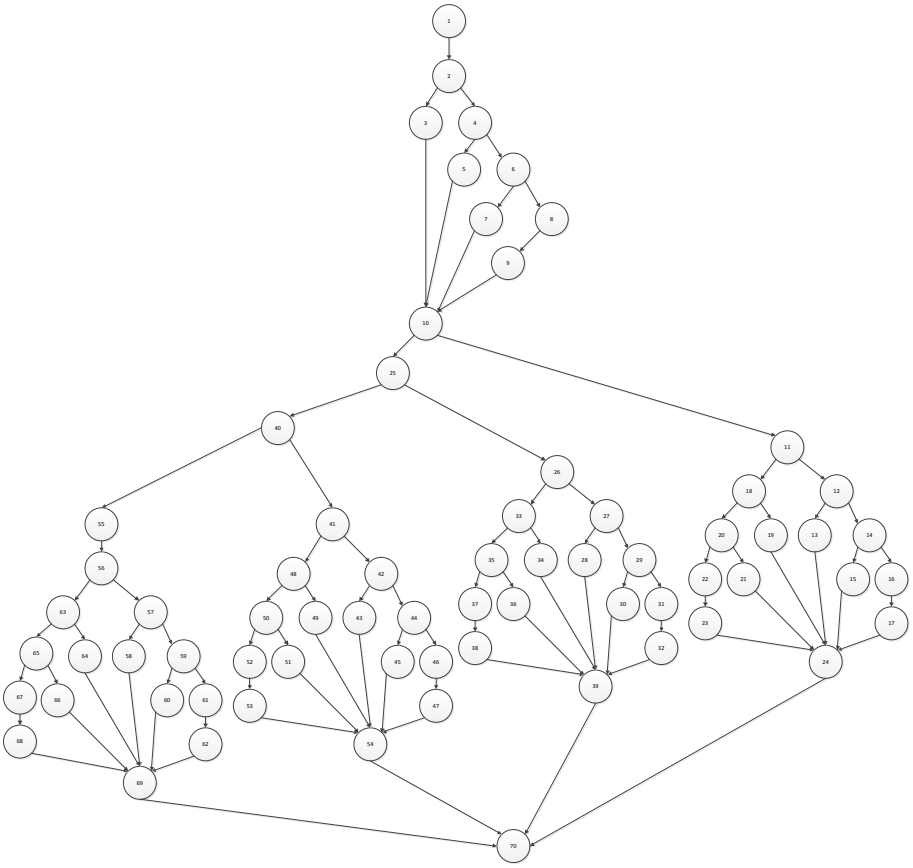
Path 35 : 40-55-56-57-58-69-70

Path 36 : 40-55-56-57-59-60-69-70

Path 37 : 40-55-56-57-59-61-69-70

Path 38 : 40-55-56-57-59-61-62-69-70

Graph Matriks



Gambar 4.9 Graph Matriks

Setelah flow graph diberi nomor baru, lalu dibuat tabel graph matriks seperti pada tabel 4.7 berikut:

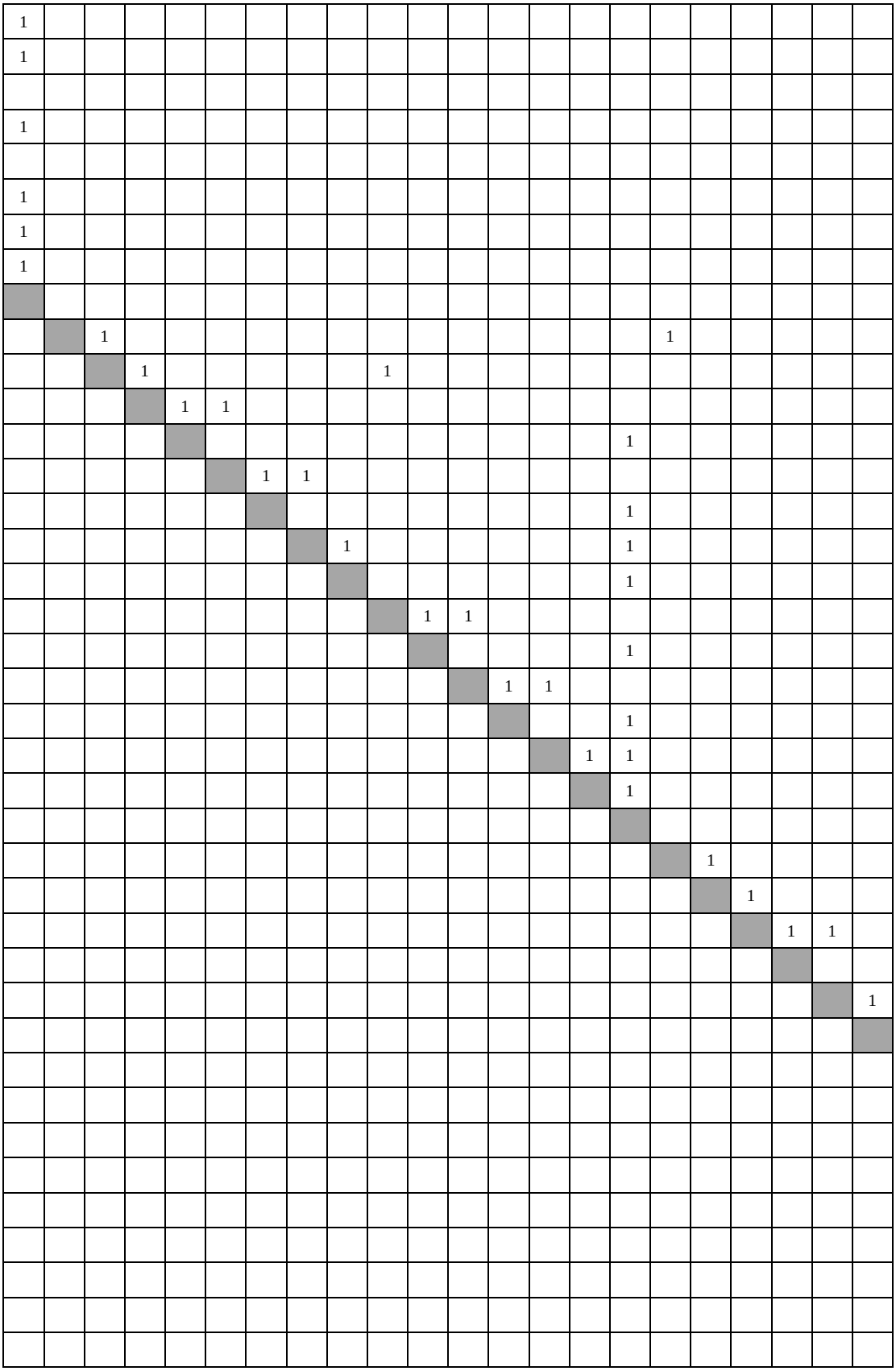
Tabel 4.4 *graph matrices testing*

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

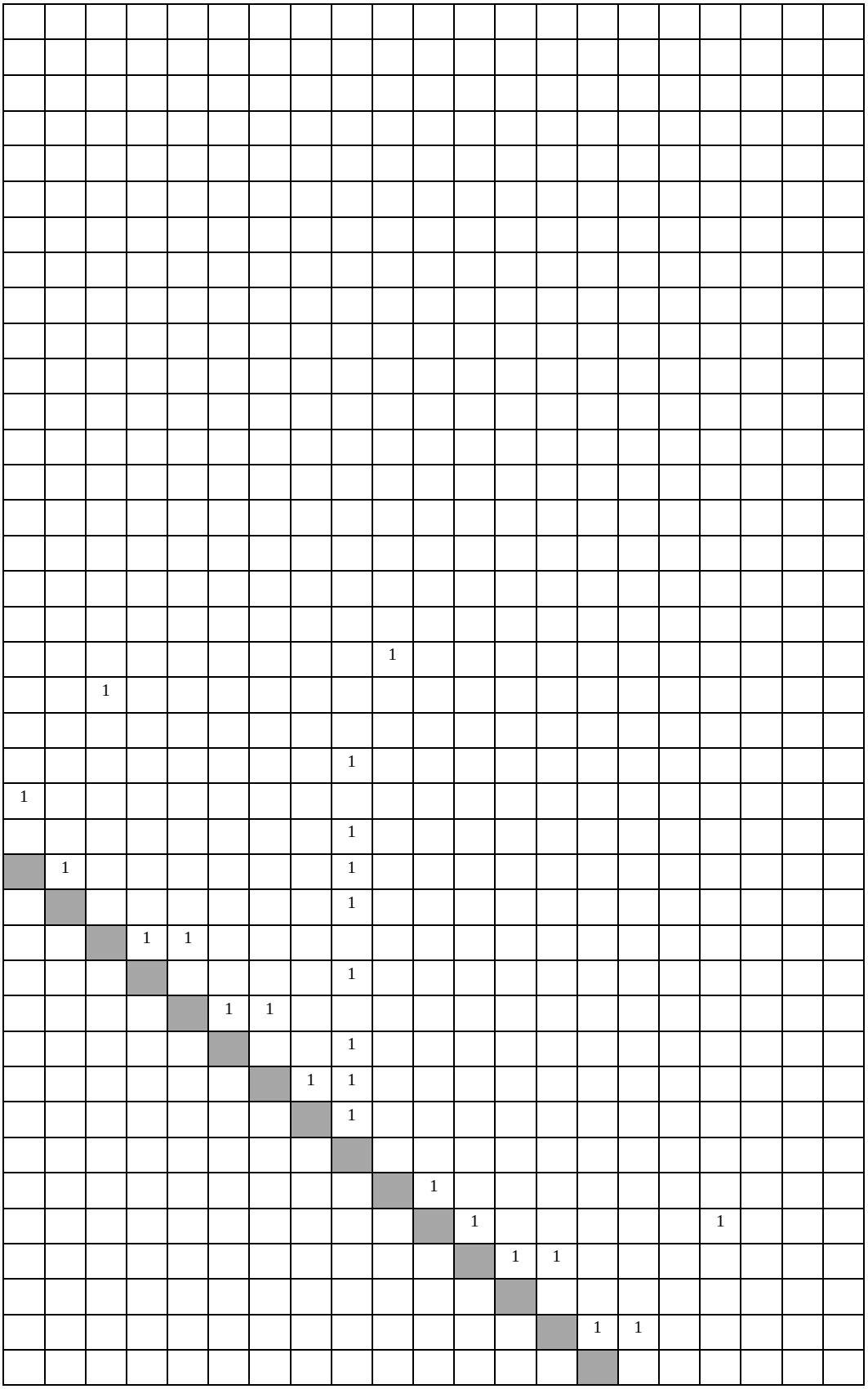
11											1						1						
12												1	1										
13																							
14														1	1								
15																							
16																1							
17																							
18																		1	1				
19																							
20																				1	1		
21																							
22																							1
23																							
24																							
25																							
26																							
27																							
28																							
29																							
30																							
31																							
32																							
33																							
34																							
35																							
36																							
37																							
38																							
39																							
40																							
41																							
42																							
43																							
44																							
45																							
46																							
47																							
48																							

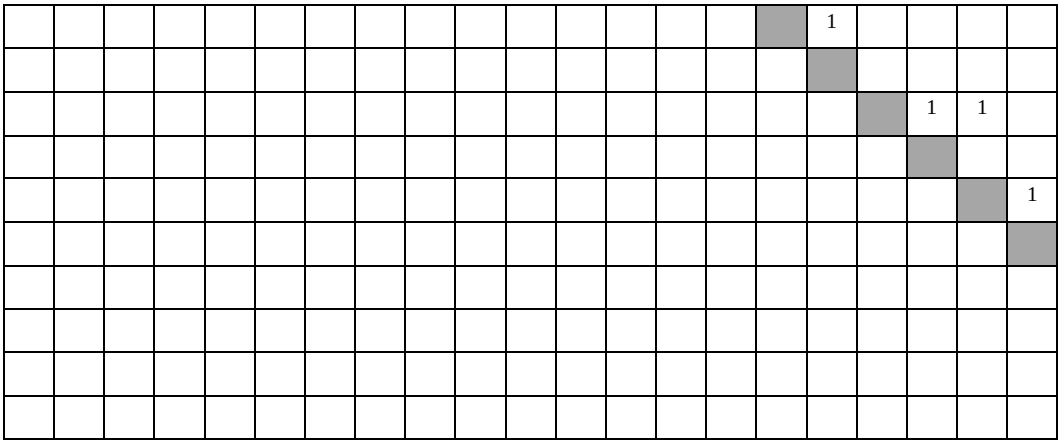
49																						
50																						
51																						
52																						
53																						
54																						
55																						
56																						
57																						
58																						
59																						
60																						
61																						
62																						
63																						
64																						
65																						
66																						
67																						
68																						
69																						
70																						

24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45
	1																				
1																					
1																					



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**	**Sum**
				0
				1
				0
				1
				0
				1
				0
				1
				0
				1
				1
				1
				0
				1
				0
				1
				0
				1
				0
				1
				0
				1
				0
			1	0
				1
				1
				1

				0
				1
				0
				1
				0
				1
				0
				1
				0
				1
				0
			1	0
				1
				1
				1
				0
				1
				0
				1
				0
				1
				0
				1
				0
				1
				0
			1	0
			1	1
				1
				1
		1		0
				1
		1		0
		1		1
		1		0
				1
		1		0
1				1
		1		0

	1	1		1
		1		0
			1	0
				0
				36

$V_{(G)}$ = Jumlah Graph Matriks + 1

= 36 + 1

= 37

Setelah di lakukan pengujian hasil yang didapat bisa dilihat pada tabel berikut:

Path no	Node (n)	Hasil yang diharapkan	Hasil sesuai uji kasus	Keterangan
1	Path 1 : 1-2-3-10	Menghasilkan nilai speed=4	Nilai speed=4	[√] alur terlewati [ ] alur tidak terlewati
2	Path 2 : 1-2-4-5-10	Menghasilkan nilai speed=3	Nilai speed=3	[√] alur terlewati [ ] alur tidak terlewati
3	Path 3 : 1-2-4-6-7-10	Menghasilkan nilai speed=2	Nilai speed=2	[√] alur terlewati [ ] alur tidak terlewati
4	Path 4 : 1-2-4-6-8-10	Menghasilkan nilai speed=1	Nilai speed=1	[√] alur terlewati [ ] alur tidak terlewati
5	Path 5 :1-2-4-6-8-9-10	Menghasilkan nilai speed=0	Nilai speed=0	[√] alur terlewati [ ] alur tidak terlewati
6	Path 6 : 10-11-12-13-24-70	Menghasilkan nilai rightPressed=true dan downPressed=false	Nilai rightPressed=true dan downPressed=false	[√] alur terlewati [ ] alur tidak terlewati
7	Path 7 : 10-11-12-14-15-24-70	Menghasilkan nilai leftPressed=true dan downPressed=false	Nilai leftPressed=true dan downPressed=false	[√] alur terlewati [ ] alur tidak terlewati
8	Path 8 : 10-11-12-14-16-24-70	Menghasilkan nilai upPressed,leftPressed,upPressed dan downPressed = false	Nilai upPressed,leftPressed,upPressed dan downPressed =	[√] alur terlewati [ ] alur tidak terlewati

			false	
9	Path 9 : 10-11-12-14-16-17-24-70	Menghasilkan nilai upPressed=true dan downPressed=false	Nilai upPressed=true dan downPressed=false	[√] alur terlewati [ ] alur tidak terlewati
10	Path 10 : 10-11-18-20-22-24-70	Menghasilkan nilai upPressed,leftPressed,upPressed dan downPressed = false	Nilai upPressed,leftPressed,upPressed dan downPressed = false	[√] alur terlewati [ ] alur tidak terlewati
11	Path 11 : 10-11-18-20-22-23-24-70	Menghasilkan nilai upPressed=true dan downPressed=false	Nilai upPressed=true dan downPressed=false	[√] alur terlewati [ ] alur tidak terlewati
12	Path 12 : 10-11-18-20-21-24-70	Menghasilkan nilai leftPressed=true dan downPressed=false	Nilai leftPressed=true dan downPressed=false	[√] alur terlewati [ ] alur tidak terlewati
13	Path 13 : 10-11-18-19-24-70	Menghasilkan nilai rightPressed=true dan downPressed=false	Nilai rightPressed=true dan downPressed=false	[√] alur terlewati [ ] alur tidak terlewati
14	Path 14 : 10-25-26-33-35-37-39-70	Menghasilkan nilai downPressed,leftPressed,upPressed dan rightPressed = false	Nilai downPressed,leftPressed,upPressed dan rightPressed = false	[√] alur terlewati [ ] alur tidak terlewati
15	Path 15 : 10-25-26-33-35-37-38-39-70	Menghasilkan nilai leftPressed=true dan rightPressed =false	Nilai leftPressed=true dan rightPressed =false	[√] alur terlewati [ ] alur tidak terlewati
16	Path 16 : 10-25-26-33-35-36-39-70	Menghasilkan nilai downpressed =true dan rightPressed =false	Nilai downpressed =true dan rightPressed =false	[√] alur terlewati [ ] alur tidak terlewati
17	Path 17 : 10-25-26-33-34-39-	Menghasilkan nilai rightPressed=true dan rightPressed	Nilai rightPressed=true dan	[√] alur terlewati [ ] alur tidak terlewati

	70	=false	rightPressed =false	
18	Path 18 : 10-25-26-27-28-39-70	Menghasilkan nilai upPressed =true dan rightPressed =false	Nilai upPressed =true dan rightPressed =false	[√] alur terlewati [ ] alur tidak terlewati
19	Path 19 : 10-25-26-27-29-30-39-70	Menghasilkan nilai downPressed =true dan rightPressed =false	Nilai downPressed =true dan rightPressed =false	[√] alur terlewati [ ] alur tidak terlewati
20	Path 20 : 10-25-26-27-29-31-39-70	Menghasilkan nilai downPressed,leftPressed,upPressed dan upPressed = false	Nilai downPressed,leftPressed,upPressed dan upPressed = false	[√] alur terlewati [ ] alur tidak terlewati
21	Path 21 : 10-25-26-27-31-32-39-70	Menghasilkan nilai leftPressed=true dan rightPressed =false	Nilai leftPressed=true dan rightPressed =false	[√] alur terlewati [ ] alur tidak terlewati
22	Path 22 : 25-40-41-42-44-46-47-54-70	Menghasilkan nilai downpressed =true dan upPressed =false	Nilai downpressed =true dan upPressed =false	[√] alur terlewati [ ] alur tidak terlewati
23	Path 23 : 25-40-41-42-44-46-54-70	Menghasilkan nilai downPressed,leftPressed,upPressed dan upPressed = false	Nilai downPressed,leftPressed,upPressed dan upPressed = false	[√] alur terlewati [ ] alur tidak terlewati
24	Path 24 : 25-40-41-42-44-45-54-70	Menghasilkan nilai rightPressed=true dan upPressed=false	Nilai rightPressed=true dan upPressed=false	[√] alur terlewati [ ] alur tidak terlewati
25	Path 25 : 25-40-41-42-43-54-70	Menghasilkan nilai leftPressed=true dan upPressed =false	Nilai leftPressed=true dan upPressed =false	[√] alur terlewati [ ] alur tidak terlewati
26	Path 26 : 25-40-41-48-50-52-53-54-70	Menghasilkan nilai downPressed =true dan upPressed =false	Nilai downPressed =true dan upPressed	[√] alur terlewati [ ] alur tidak terlewati

			=false	
27	Path 27 : 25-40-41-48-50-52-54-70	Menghasilkan nilai downPressed,leftPressed,upPressed dan upPressed = false	Nilai downPressed,leftPressed,upPressed dan upPressed = false	[√] alur terlewati [ ] alur tidak terlewati
28	Path 28 : 25-40-41-48-50-51-54-70	Menghasilkan nilai rightPressed=true dan upPressed =false	Nilai rightPressed=true dan upPressed =false	[√] alur terlewati [ ] alur tidak terlewati
29	Path 29 : 25-40-41-48-49-54-70	Menghasilkan nilai leftPressed=true dan upPressed =false	Nilai leftPressed=true dan upPressed =false	[√] alur terlewati [ ] alur tidak terlewati
30	Path 30 : 40-55-70	Menghasilkan nilai rightPressed=true dan upPressed=false	Nilai rightPressed=true dan upPressed=false	[√] alur terlewati [ ] alur tidak terlewati
31	Path 31 : 40-55-56-63-65-67-68-69-70	Menghasilkan nilai rightPressed=true dan leftPressed =false	Nilai rightPressed=true dan leftPressed =false	[√] alur terlewati [ ] alur tidak terlewati
32	Path 32 : 40-55-56-63-65-67-69-70	Menghasilkan nilai downPressed,leftPressed,upPressed dan upPressed = false	Nilai downPressed,leftPressed,upPressed dan upPressed = false	[√] alur terlewati [ ] alur tidak terlewati
33	Path 33 : 40-55-56-63-65-66-69-70	Menghasilkan nilai upPressed =true dan leftPressed =false	Nilai upPressed =true dan leftPressed =false	[√] alur terlewati [ ] alur tidak terlewati
34	Path 34 : 40-55-56-63-64-69-70	Menghasilkan nilai downpressed =true dan leftPressed =false	Nilai downpressed =true dan leftPressed =false	[√] alur terlewati [ ] alur tidak terlewati
35	Path 35 : 40-55-56-57-58-69-70	Menghasilkan nilai downPressed =true dan leftPressed =false	Nilai downPressed =true dan leftPressed =false	[√] alur terlewati [ ] alur tidak terlewati

36	Path 36 : 40-55-56-57-59-60-69-70	Menghasilkan nilai upPressed =true dan leftPressed =false	Nilai upPressed =true dan leftPressed =false	[√] alur terlewati [ ] alur tidak terlewati
37	Path 37 : 40-55-56-57-59-61-69-70	Menghasilkan nilai downPressed,leftPressed,upPressed dan upPressed = false	Nilai downPressed,leftPressed,upPressed dan upPressed = false	[√] alur terlewati [ ] alur tidak terlewati
38	Path 38 : 40-55-56-57-59-61-62-69-70	Menghasilkan nilai rightPressed=true dan leftPressed =false	Nilai rightPressed=true dan leftPressed =false	[√] alur terlewati [ ] alur tidak terlewati

Kesimpulan

Dari hasil pengujian pada setiap metode, dihasilkan nilai *Cyclomatic Complexity* V(G) yang sama yaitu 37. Maka dapat disimpulkan bahwa pengujian *white box* pada proses logika fuzzy berjalan dengan baik, karena setiap pengujian menghasilkan nilai yang sama.

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 Pretest posttest

Pretest dan Postest merupakan pengujian terhadap pengguna untuk mengetahui tingkat pengenalan informasi tentang *sugar glider* sebelum dan sesudah menggunakan aplikasi. Hasil yang didapat akan menjadi bahan pendukung kuesioner.

1. Pretest

Pada pengujian *pretest* pengguna akan diberikan soal pilihan ganda dengan materi yang sama dengan aplikasi *game* yang akan dimainkan. Soal yang terdapat pada pengujian *pretest* telah ditentukan sebelumnya dan setiap pengguna akan

diberikan soal yang sama. Dalam pengerjaannya pengguna akan diuji satu persatu sebelum menggunakan aplikasi *game* Sugie Man Penyelamat *Sugar Glider*. Kriteria penilaian dalam *pretest* pada setiap soalnya jika benar bernilai 20, jika salah bernilai 0.

2. Posttest

Pada pengujian *posttest* pengguna akan diberikan soal yang sama dengan soal *pretest*. Soal yang terdapat pada pengujian *posttest* telah ditentukan sebelumnya dan setiap pengguna akan diberikan soal yang sama. Dalam pengerjaannya pengguna akan diuji satu persatu setelah menggunakan aplikasi *game* Sugie Man Penyelamat *Sugar Glider*. Kriteria penilaian dalam *posttest* pada setiap soalnya jika benar bernilai 20, jika salah bernilai 0.

3. Hasil Pretest Posttest

Hasil pengujian pretest posttest akan dijelaskan sebagai berikut.

Tabel 4.5 Hasil Pretest Posttest

No Responden	Pretest	Posttest
1	0	60
2	20	100
3	0	100
4	0	100
5	0	100
6	0	80
7	0	100
8	0	100
9	0	100
10	0	100
11	0	100
12	0	100
13	0	100
14	0	100
15	40	60
16	0	60
17	0	100
18	40	80
19	20	60
20	20	100
21	0	100
22	20	100

23	0	100
24	0	80
25	20	80
26	0	80
27	20	80
28	0	100
29	0	100
30	20	80

Data diatas akan diolah secara statistik dengan manggunakan metode sebagai berikut:

a. Uji Normalitas

Dilakukan untuk menyelidiki apakah data yang dikumpulkan mengikuti dugaan mengikuti distribusi normal atau tidak. Untuk melakukan uji normalitas dapat dengan memakai uji sebagai berikut:

1. Kolmogorov-Smirnov, merupakan uji normalitas untuk sampel besar. Pada SPSS, jika dipilih tingkat signifikansi α= 0,05 < nilai sig SPSS, maka dapat dikatakan bahwa data mengikuti distribusi normal.
2. Shapiro-Wilks, merupakan uji normalitas untuk sampel kecil sampai dengan jumlah 2000. Pada SPSS, jika dipilih tingkat signifikansi α= 0,05 < nilai sig SPSS, maka dapat dikatakan bahwa data mengikuti distribusi normal.

Pada uji normalitas tersebut disusun hipotesis yaitu:

H0: sampel berasal dari populasi yang berdistribusi normal.

H1: sampel berasal dari populasi yang berdistribusi tidak normal.

Dari hasil pengolahan data tabel 4.14 dengan menggunakan SPSS 22, maka didapatkan hasil seperti berikut:

Tabel 4.6 Tests of Normality

	Kolmogorov-Smirnov[a]			Shapiro-Wilk		
	Statistic	df	Sig.	Statistic	df	Sig.
Pretest	,424	30	,000	,628	30	,000
Posttest	,386	30	,000	,681	30	,000

a. Lilliefors Significance Correction

Dari tabel diatas terlihat bahwa pada kolom kolmogorov-smirnov (KS) nilai α= 5% = 0,05 < sig.= 0,000, maka H0 ditolak, dan kolom saphiro-wilks (SW) nilai α= 5% = 0,05 < sig.= 0,000, maka H0 ditolak.

Dari uji KS dan SW dapat disimpulkan bahwa populasi data berasal dari distribusi tidak normal. Sehingga tidak dapat dilanjutkan dengan uji homogenitas, tetapi dilakukan pengujian statistik *non-parametrik* dengan uji Mann-Whitney pada SPSS 22.

b. Uji Hipotesis

Uji hipotesis dilakukan untuk mengetahui peningkatan setelah memainkan *game* Sugie Man Penyelamat *Sugar Glider*, yang dilihat dari hasil nilai pretest dan posttest. Karena tidak terdistribusi normal maka tidak dapat dilanjutkan dengan uji homogenitas, tetapi dilakukan pengujian statistik *non-parametrik* dengan uji Mann-Whitney.

a) Merumuskan hipotesis pengujian sebagai berikut:

H_0 : game sugie man penyelamat sugar glider tidak berpengaruh terhadap peningkatan informasi terhadap responden.

H_1 : game sugie man penyelamat sugar glider berpengaruh terhadap peningkatan informasi terhadap responden.

b) Melihat nilai *t* pada baris dengan taraf signifikansi 5% (α= 0,05)

Tabel 4.7 Test Statistics[a]

	PrePros
Mann-Whitney U	,000
Wilcoxon W	465,000
Z	-6,928
Asymp. Sig. (2-tailed)	,000

a. Grouping Variable: Grup

Dari tabel diatas terlihat bahwa nilai signifikan (2-tailed)= 0,000 < 0,05, maka H0 ditolak. Dapat disimpulkan bahwa game sugie man penyelamat sugar glider berpengaruh terhadap peningkatan informasi tentang sugar glider bagi responden.

4.2.2.2 Kuesioner

Kuesioner disebarkan seperti pada Pretest posttest dan juga ke pada orang yang sama. menghitung tanggapan dengan menggunakan pola skala likert. Kuesioner disebarkan kepada 30 orang dari usia 11 tahun keatas, dari hasil kuesioner tersebut akan dilakukan perhitungan agar dapat diambil kesimpulan terhadap penilaian penerapan sistem yang telah dibangun. Kuesioner terdiri dari 5 pertanyaan dengan menggunakan teknik skala likert dengan skala 1 sampai 5.

Untuk mencari nilai persentase dari masing-masing jawaban kuesioner digunakan rumus (2.3).

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

1. Apakah *game* Sugie Man Penyelamat *Sugar Glider* sudah membantu anda dalam mengetahui informasi seputar *sugar glider*?

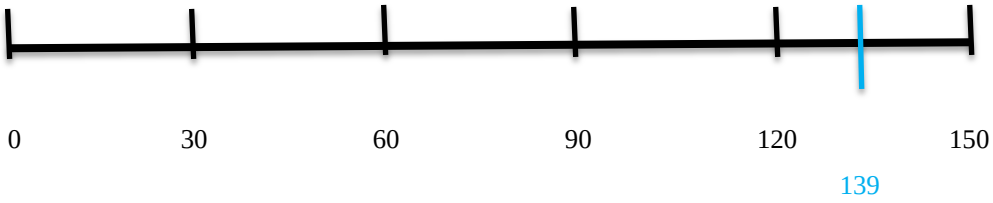
Tabel 4.8 Perhitungan persentase pertanyaan ke-1

Pertanyaan	Keterangan	Skor	Frekuensi	Total Skor
1	Sangat Setuju	5	19	95
	Setuju	4	11	44
	Kurang Setuju	3	-	-
	Tidak Setuju	2	-	-
	Sangat Tidak Setuju	1	-	-
Jumlah			30	139

Maka diperoleh perhitungan sebagai berikut:

$$P = \frac{139}{5 \times 30} X\ 100\% = 92{,}6\ \%$$

Dapat disimpulkan bahwa penilaian terhadap *game* Sugie Man Penyelamat *Sugar Glider* sudah membantu responden dalam mengetahui informasi seputar *sugar glider* adalah 92,6% dari yang diharapkan 100%,atau dapat dikategorikan sebagai sangat setuju, untuk lebih jelasnya dapat dilihat pada gambar 4.38.



Gambar 4.10 Hasil skor pertanyaan ke-1

2. Apakah instruksi atau tata cara memainkan *game* ini mudah untuk dimengerti?

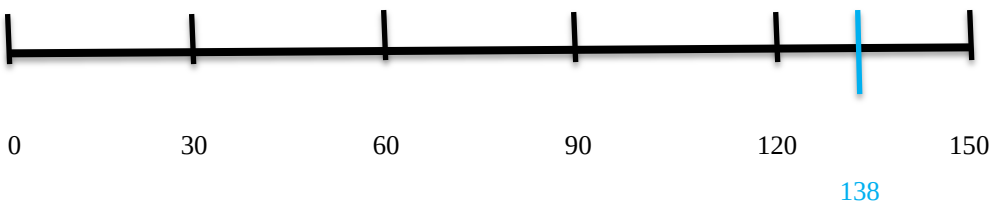
Tabel 4.9 Perhitungan persentase pertanyaan ke-2

Pertanyaan	Keterangan	Skor	Frekuensi	Total Skor
2	Sangat Setuju	5	18	90
	Setuju	4	12	48
	Kurang Setuju	3	-	-
	Tidak Setuju	2	-	-
	Sangat Tidak Setuju	1	-	-
Jumlah			30	138

Maka diperoleh perhitungan sebagai berikut:

$$P = \frac{138}{5 \times 30} X\ 100\% = 92\ \%$$

Dapat disimpulkan bahwa penilaian terhadap tata cara memainkan *game* Sugie Man Penyelamat *Sugar Glider* mudah dimengerti adalah 92% dari yang diharapkan 100%,atau dapat dikategorikan sebagai sangat setuju, untuk lebih jelasnya dapat dilihat pada gambar 4.37.



Gambar 4.11 Hasil skor pertanyaan ke-2

3. Apakah kualitas *backsoundnya* sudah baik?

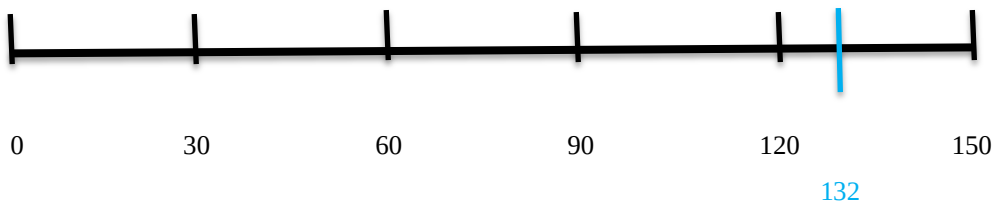
Tabel 4.11 Perhitungan persentase pertanyaan ke-3

Pertanyaan	Keterangan	Skor	Frekuensi	Total Skor
3	Sangat Setuju	5	15	75
	Setuju	4	12	48
	Kurang Setuju	3	3	9
	Tidak Setuju	2	-	-
	Sangat Tidak Setuju	1	-	-
Jumlah			30	132

Maka diperoleh perhitungan sebagai berikut:

$$P = \frac{132}{5 \times 30} X\ 100\% = 88\%$$

Dapat disimpulkan bahwa penilaian *backsound* pada *game* Sugie Man Penyelamat *Sugar Glider* adalah 88% dari yang diharapkan 100%, atau dapat dikategorikan sebagai sangat setuju, untuk lebih jelasnya dapat dilihat pada gambar 4.40.



Gambar 4.12 Hasil skor pertanyaan ke-3

4. Apakah *backgroundnya* sudah sesuai dengan alur cerita permainannya?

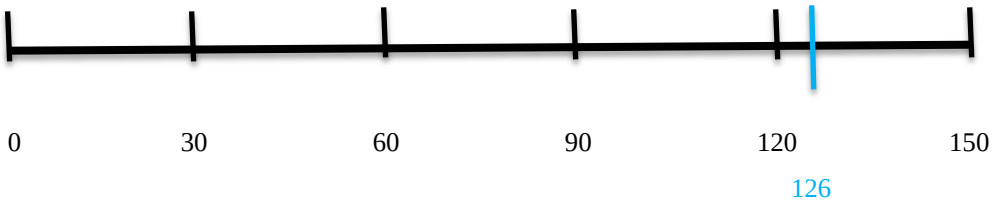
Tabel 4.12 Perhitungan persentase pertanyaan ke-4

Pertanyaan	Keterangan	Skor	Frekuensi	Total Skor
4	Sangat Setuju	5	10	50
	Setuju	4	16	64
	Kurang Setuju	3	4	12
	Tidak Setuju	2	-	-
	Sangat Tidak Setuju	1	-	-
Jumlah			30	126

Maka diperoleh perhitungan sebagai berikut:

$$P = \frac{126}{5 \times 30} X\ 100\% = 84\%$$

Dapat disimpulkan bahwa penilaian terhadap *background* yang ada pada *game* Sugie Man Penyelamat *Sugar Glider* adalah 84% dari yang diharapkan 100%,atau dapat dikategorikan sebagai sangat setuju, untuk lebih jelasnya dapat dilihat pada gambar 4.39.



Gambar 4.13 Hasil skor pertanyaan ke-4

5. Apakah *game* Sugie Man Penyelamat *Sugar Glider* ini membuat anda tertarik untuk memainkannya kembali?

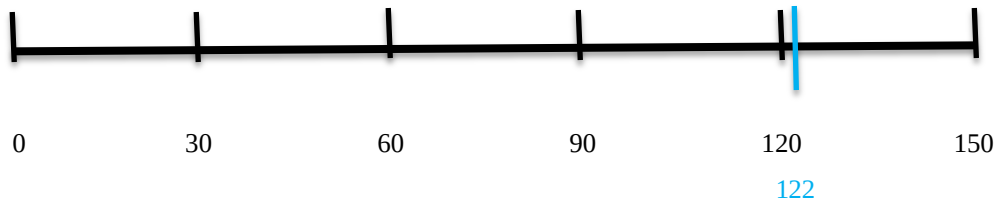
Tabel 4.13 Perhitungan persentase pertanyaan ke-5

Pertanyaan	Keterangan	Skor	Frekuensi	Total Skor
5	Sangat Setuju	5	10	50
	Setuju	4	12	48
	Kurang Setuju	3	8	24
	Tidak Setuju	2	-	-
	Sangat Tidak Setuju	1	-	-
Jumlah			30	122

Maka diperoleh perhitungan sebagai berikut:

$$P = \frac{122}{5 \times 30} X\ 100\% = 81{,}3\%$$

Dapat disimpulkan bahwa penilaian terhadap ketertarikan pada *game* Sugie Man Penyelamat *Sugar Glider* adalah 81,3% dari yang diharapkan 100%, atau dapat dikategorikan sebagai sangat setuju, untuk lebih jelasnya dapat dilihat pada gambar 4.41.



Gambar 4.14 Hasil skor pertanyaan ke-5

4.2.2.3 Kesimpulan kuesioner

Dari hasil pengujian yang dilakukan dengan cara menyebarkan kuesioner dapat disimpulkan bahwa :

1. *Game* Sugie Man Penyelamat *Sugar Glider* dapat memberikan informasi tentang *sugar glider.*
2. *Game* Sugie Man Penyelamat *Sugar Glider user friendly*.
3. *Game* Sugie Man Penyelamat *Sugar Glider* dapat membuat responden untuk memainkannya kembali.

4.2.2.4 Kesimpulan pengujian beta

1. Berdasarkan pretest dan posttest, game sugie man penyelamat sugar glider dapat meningkatkan informasi sugar glider.
2. Berdasarkan kuesioner yang dibagikan, *game* Sugie Man Penyelamat *Sugar Glider* dapat memberikan informasi tentang *sugar glider, game* Sugie Man Penyelamat *Sugar Glider user friendly, game* Sugie Man Penyelamat *Sugar Glider* dapat membuat responden untuk memainkannya kembali. Maka dapat disimpulkan *game* ini sebagai sarana alternatif agar penyampaian informasi lebih menyenangkan.