BAB 4
IMPLEMENTASI DAN PENGUJIAN SISTEM

4.1 Implementasi Sistem

Tahap implementasi sistem merupakan tahap penerjemahan perancangan berdasarkan hasil analisis kedalam suatu bahasa pemrograman tertentu serta penerapan perangkat lunak yang dibangun pada lingkungan yang sesungguhnya. Adapun pembahasan implementasi terdiri dari perangkat lunak pembangun, perangkat keras pembangun dan implementasi antarmuka.

4.1.1 Implementasi Perangkat Keras

Berikut adalah spesifikasi perangkat keras yang digunakan untuk menguji *game* yang telah dibangun pada tabel 4.1 berikut.

Tabel 4.1 Implementasi Perangkat Keras

No	Perangkat Keras	Spesifikasi
1	Processor	Intel 2.30 GHz
2	Monitor	14 inch
3	Graphic Card	VGA 2 GB
4	Hard disk drive	500 GB
5	Memory	RAM 4 GB
6	Keyboard dan Mouse	Standar
7	Speaker	Standar

4.1.2 Implementasi Perangkat Lunak

Implementasi perangkat lunak digunakan untuk menjalankan aplikasi *game* yang telah dibangun dapat dilihat pada tabel 4.2 berikut.

Tabel 4.2 Implementasi Perangkat Lunak

No	Perangkat Lunak	Spesifikasi
1	Sistem Operasi	Microsoft Windows XP, Windows 7
2	Tool Pembangun	Unity
3	Tool Desain	Adobe Photoshop CS5

Sistem operasi harus sudah terpasang pada komputer untuk memainkan *game* yang telah dibangun. Untuk membuka *project* pada *game* yang dibangun maka pengguna harus menginstal Unity dan komponen yang mendukungnya.

4.1.3 Implementasi Antarmuka

Adapun implementasi antarmuka pada aplikasi *game* yang telah dibangun adalah sebagai berikut.

4.1.3.1 Antarmuka Menu Utama

Antarmuka menu utama merupakan tampilan menu utama pada aplikasi *game*. Implementasi antarmuka menu utama pada *game* yang dibangun dapat dilihat pada gambar 4.1 berikut.



Gambar 4.1 Antarmuka Menu Utama

4.1.3.2 Antarmuka Petunjuk

Pada antarmuka petunjuk akan ditampilkan informasi mengenai petunjuk cara bermain seperti tombol apa saja yang digunakan ketika bermain.

Implementasi antarmuka petunjuk pada *game* yang dibangun dapat dilihat pada gambar 4.2 berikut.



Gambar 4.2 Antarmuka Petunjuk

4.1.3.3 Antarmuka Menu Pilih Permainan

Pada antarmuka menu pilih permainan berisi tentang permainan apa saja yang bisa dimainkan. Implementasi antarmuka menu pilih permainan pada *game* yang dibangun dapat dilihat pada gambar 4.3 berikut.



Gambar 4.3 Antarmuka Pilih Permainan

4.1.3.4 Antarmuka materi *sprint*

Pada antarmuka menu materi *sprint* berisi tentang materi *sprint*. Implementasi antarmuka materi *sprint* pada *game* yang dibangun dapat dilihat pada gambar 4.4 berikut.



Gambar 4.4 Antarmuka Materi *Sprint*

4.1.3.5 Antarmuka Materi Lari Gawang

Pada antarmuka menu materi lari gawang berisi tentang materi lari gawang. Berikut adalah antarmuka materi lari gawang. Implementasi antarmuka materi lari gawang pada *game* yang dibangun dapat dilihat pada gambar 4.5 berikut.



Gambar 4.5 Antarmuka Materi Lari Gawang

4.1.3.6 Antarmuka Materi Lompat Tinggi

Pada antarmuka menu materi lompat tinggi berisi tentang materi lompat tinggi. Berikut adalah antarmuka materi lompat tinggi. Implementasi antarmuka materi lompat tinggi pada *game* yang dibangun dapat dilihat pada gambar 4.6 berikut.



Gambar 4.6 Antarmuka Materi Lompat Tinggi

4.1.3.7 Antarmuka Materi Lompat Jauh

Pada antarmuka menu materi lompat jauh berisi tentang materi lompat jauh. Berikut adalah antarmuka materi lompat jauh. Implementasi antarmuka materi lompat jauh pada *game* yang dibangun dapat dilihat pada gambar 4.7 berikut.



Gambar 4.7 Antarmuka Materi Lompat Jauh

4.1.3.8 Antarmuka Materi Lempar Lembing

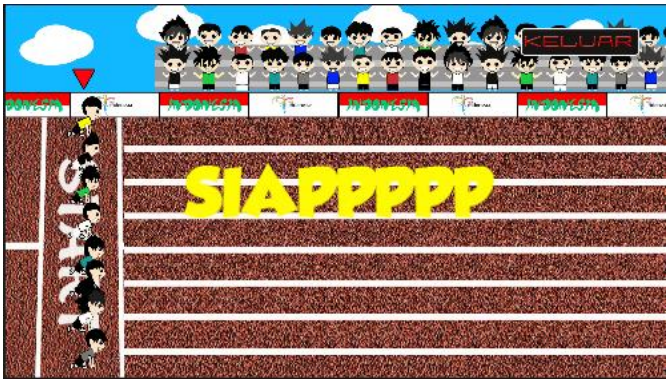
Pada antarmuka menu materi lempar lembing berisi tentang materi lempar lembing. Implementasi antarmuka petunjuk pada *game* yang dibangun dapat dilihat pada gambar 4.8 berikut.



Gambar 4.8 Antarmuka Materi Lempar Lembing

4.1.3.9 Antarmuka Permainan *Sprint*

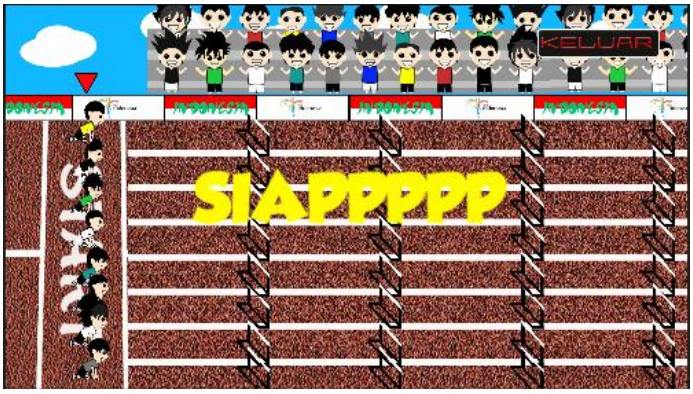
Pada antarmuka menu permainan *sprint* berisi tentang permainan *sprint*. Implementasi antarmuka permainan *sprint* pada *game* yang dibangun dapat dilihat pada gambar 4.9 berikut.



Gambar 4.9 Antarmuka Permainan *Sprint*

4.1.3.10 Antarmuka Permainan Lari Gawang

Pada antarmuka menu permainan lari gawang berisi tentang permainan lari gawang. Implementasi antarmuka permainan lari gawang pada *game* yang dibangun dapat dilihat pada gambar 4.10 berikut.



Gambar 4.10 Antarmuka Permainan Lari Gawang

4.1.3.11 Antarmuka Permainan Lompat Tinggi

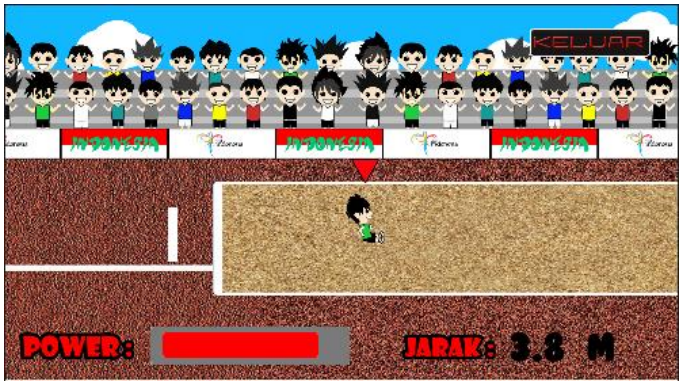
Pada antarmuka menu permainan lompat tinggi berisi tentang permainan lompat tinggi. Implementasi antarmuka permainan lompat tinggi pada *game* yang dibangun dapat dilihat pada gambar 4.11 berikut.



Gambar 4.11 Antarmuka Permainan Lompat Tinggi

4.1.3.12 Antarmuka Permainan Lompat Jauh

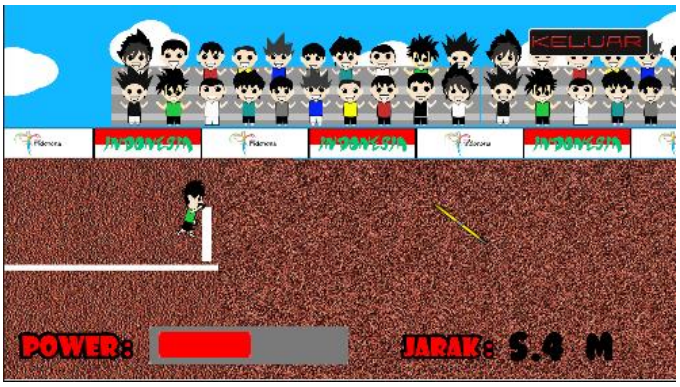
Pada antarmuka menu pilih permainan lompat jauh berisi tentang permainan lompat jauh. Implementasi antarmuka permainan lompat tinggi pada *game* yang dibangun dapat dilihat pada gambar 4.12 berikut.



Gambar 4.12 Antarmuka Permainan Lompat Jauh

4.1.3.13 Antarmuka Permainan Lempar Lembing

Pada antarmuka permainan lempar lembing berisi tentang permainan permainan lempar lembing. Implementasi antarmuka permainan lempar lembing pada *game* yang dibangun dapat dilihat pada gambar 4.13 berikut



Gambar 4.13 Antarmuka Permainan Lempar Lembing

4.2 Pengujian Sistem

Dalam melakukan pengujian terhadap aplikasi yang dibangun dilakukan dengan dua macam teknik, yaitu teknik pengujian *black box* dan teknik pengujian *white box*. Pengujian *black box* berfokus pada menemukan kesalahan program, sedangkan pengujian *white box* berfokus kepada struktur internal (*source code*) program untuk mengetahui apakah masih terjadi *error*. Pengujian dilakukan dengan dua tahap, yaitu tahap pengujian *alpha* (fungsional) dan tahap pengujian

beta.

4.2.1 Hasil Pengujian *(Black Box)*

Kasus dan hasil uji perangkat lunak yang sudah dibangun dengan menggunakan metode *black box* berdasarkan pengamatan yang dilakukan dan menghasilkan kesimpulan dari pengujian tersebut, diantaranya sebagai berikut:

1. Pengujian Menu Utama

Penguijian menu utama dengan kasus dan hasil uji, kemudian diamati dan menghasilkan kesimpulan dapat dilihat pada tabel 4.3 berikut.

Tabel 4.3 Pengujian Menu Utama

Kasus dan Hasil Uji (Uji Data Benar)			
Masukan	**Keluaran yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Pilih menu mulai	Menampilkan menu pilih permainan	Dapat menampilkan layar pilih permainan	[√] Berhasil [ ] Tidak
Pilih menu petunjuk	Menampilkan petunjuk cara bermain	Dapat menampilkan petunjuk cara bermain	[√] Berhasil [ ] Tidak
Pilih menu keluar	Keluar dari aplikasi *game*	Dapat keluar dari aplikasi *game*	[√] Berhasil [ ] Tidak

2. Pengujian Pilih Permainan

Penguijian pilih permainan dengan kasus dan hasil uji, kemudian diamati dan menghasilkan kesimpulan dapat dilihat pada table 4.4 berikut.

Tabel 4.4 Pengujian Pilih Permainan

Kasus dan Hasil Uji (Uji Data Benar)			
Masukan	**Keluaran yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Pilih menu *sprint*	Menampilkan menu materi *sprint*	Dapat menampilkan menu materi *sprint*	[√] Berhasil [ ] Tidak
Pilih menu lari gawang	Menampilkan menu materi lari gawang	Dapat menampilkan menu materi lari gawang	[√] Berhasil [ ] Tidak
Pilih menu lompat tinggi	Menampilkan menu materi lompat tinggi	Dapat menampilkan menu materi lompat tinggi	[√] Berhasil [ ] Tidak
Pilih menu lompat jauh	Menampilkan menu materi lompat jauh	Dapat menampilkan menu materi lompat jauh	[√] Berhasil [ ] Tidak
Pilih menu lempar lembing	Menampilkan menu materi lempar lembing	Dapat menampilkan menu materi lempar lembing	[√] Berhasil [ ] Tidak
Pilih tombol	Menampilkan menu	Dapat menampilkan menu	[√] Berhasil

kembali	utama	utama	[ ] Tidak

3. Pengujian Menu Petunjuk

Pengujian menu petunjuk dengan kasus dan hasil uji, kemudian diamati dan menghasilkan kesimpulan yang dapat dilihat pada tabel 4.5 berikut.

Tabel 4.5 Pengujian Menu Petunjuk

Kasus dan Hasil Uji (Uji Data Benar)			
Masukan	**Keluaran yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Pilih menu petunjuk	Menampilkan menu petunjuk	Dapat menampilkan menu petunjuk	[√] Berhasil [ ] Tidak
Pilih tombol kembali	Menampilkan menu utama	Dapat menampilkan menu utama	[√] Berhasil [ ] Tidak

4. Pengujian Menu Materi *Sprint*

Pengujian menu materi *sprint* dengan kasus dan hasil uji, kemudian diamati dan menghasilkan kesimpulan yang dilihat pada table 4.6 berikut.

Tabel 4.6 Pengujian Menu Materi *Sprint*

Kasus dan Hasil Uji (Uji Data Benar)			
Masukan	**Keluaran yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Pilih tombol mulai	Menampilkan menu permainan *sprint*	Dapat menampilkan menu permainan *sprint*	[√] Berhasil [ ] Tidak
Pilih tombol kembali	Menampilkan menu utama	Dapat menampilkan menu utama	[√] Berhasil [ ] Tidak
Pilih tombol selanjutnya	Menampilkan materi berikutnya	Dapat menampilkan materi berikutnya	[√] Berhasil [ ] Tidak
Pilih tombol sebelumnya	Menampilkan materi sebelumnya	Dapat menampilkan materi sebelumnya	[√] Berhasil [ ] Tidak

5. Pengujian Menu Materi Lari Gawang

Pengujian menu materi lari gawang dengan kasus dan hasil uji, kemudian diamati dan menghasilkan kesimpulan yang dilihat pada table 4.7 berikut.

Tabel 4.7 Pengujian Menu Materi Lari Gawang

Kasus dan Hasil Uji (Uji Data Benar)			
Masukan	**Keluaran yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Pilih tombol mulai	Menampilkan menu permainan lari gawang	Dapat menampilkan menu permainan lari gawang	[√] Berhasil [ ] Tidak
Pilih tombol	Menampilkan menu utama	Dapat menampilkan menu	[√] Berhasil

kembali		utama	[ ] Tidak
Pilih tombol selanjutnya	Menampilkan materi berikutnya	Dapat menampilkan materi berikutnya	[√] Berhasil [ ] Tidak
Pilih tombol sebelumnya	Menampilkan materi sebelumnya	Dapat menampilkan materi sebelumnya	[√] Berhasil [ ] Tidak

6. Pengujian Menu Materi Lompat Tinggi

Pengujian menu materi lompat tinggi dengan kasus dan hasil uji, kemudian diamati dan menghasilkan kesimpulan yang dilihat pada tabel 4.8 berikut.

Tabel 4.8 Pengujian Menu Materi Lompat Tinggi

Kasus dan Hasil Uji (Uji Data Benar)			
Masukan	**Keluaran yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Pilih tombol mulai	Menampilkan menu permainan lompat tinggi	Dapat menampilkan menu permainan lompat tinggi	[√] Berhasil [ ] Tidak
Pilih tombol kembali	Menampilkan menu utama	Dapat menampilkan menu utama	[√] Berhasil [ ] Tidak
Pilih tombol selanjutnya	Menampilkan materi berikutnya	Dapat menampilkan materi berikutnya	[√] Berhasil [ ] Tidak
Pilih tombol sebelumnya	Menampilkan materi sebelumnya	Dapat menampilkan materi sebelumnya	[√] Berhasil [ ] Tidak

7. Pengujian Menu Materi Lompat Jauh

Pengujian menu materi lompat jauh dengan kasus dan hasil uji, kemudian diamati dan menghasilkan kesimpulan yang dilihat pada tabel 4.9 berikut.

Tabel 4.9 Pengujian Menu Materi Lompat Jauh

Kasus dan Hasil Uji (Uji Data Benar)			
Masukan	**Keluaran yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Pilih tombol mulai	Menampilkan menu permainan lompat jauh	Dapat menampilkan menu permainan lompat jauh	[√] Berhasil [ ] Tidak
Pilih tombol kembali	Menampilkan menu utama	Dapat menampilkan menu utama	[√] Berhasil [ ] Tidak
Pilih tombol selanjutnya	Menampilkan materi berikutnya	Dapat menampilkan materi berikutnya	[√] Berhasil [ ] Tidak
Pilih tombol sebelumnya	Menampilkan materi sebelumnya	Dapat menampilkan materi sebelumnya	[√] Berhasil [ ] Tidak

8. Pengujian Menu Materi Lempar Lembing

Pengujian menu materi lempar lembing dengan kasus dan hasil uji, kemudian diamati dan menghasilkan kesimpulan yang dilihat pada tabel 4.10 berikut.

Tabel 4.10 Pengujian Menu Materi Lempar Lembing

Kasus dan Hasil Uji (Uji Data Benar)			
Masukan	**Keluaran yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Pilih tombol mulai	Menampilkan menu permainan lempar lembing	Dapat menampilkan menu permainan lempar lembing	[√] Berhasil [ ] Tidak
Pilih tombol kembali	Menampilkan menu utama	Dapat menampilkan menu utama	[√] Berhasil [ ] Tidak
Pilih tombol selanjutnya	Menampilkan materi berikutnya	Dapat menampilkan materi berikutnya	[√] Berhasil [ ] Tidak
Pilih tombol sebelumnya	Menampilkan materi sebelumnya	Dapat menampilkan materi sebelumnya	[√] Berhasil [ ] Tidak

9. Pengujian Menu Permainan *Sprint*

Pengujian menu permainan *sprint* dengan kasus dan hasil uji, kemudian diamati dan menghasilkan kesimpulan yang dilihat pada tabel 4.11 berikut.

Tabel 4.11 Pengujian Menu Permainan *Sprint*

Kasus dan Hasil Uji (Uji Data Benar)			
Masukan	**Keluaran yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Tekan tombol keyboard kanan	Pemain berlari	Pemain dapat berlari	[√] Berhasil [ ] Tidak
Tekan keyboard kiri	Pemain berlari	Pemain dapat berlari	[√] Berhasil [ ] Tidak
Pilih tombol keluar	Menampilkan materi *sprint*	Dapat menampilkan materi *sprint*	[√] Berhasil [ ] Tidak

10. Pengujian Menu Permainan Lari Gawang

Pengujian menu permainan lari gawang dengan kasus dan hasil uji, kemudian diamati dan menghasilkan kesimpulan yang dilihat pada tabel 4.12 berikut.

Tabel 4.12 Pengujian Menu Permainan Lari Gawang

Kasus dan Hasil Uji (Uji Data Benar)			
Masukan	**Keluaran yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Tekan tombol keyboard kanan	Pemain berlari	Pemain dapat berlari	[√] Berhasil [ ] Tidak
Tekan	Pemain berlari	Pemain dapat berlari	[√] Berhasil

keyboard kiri			[ ] Tidak
Tekan tombol keyboard space	Pemain melompat	Pemain dapat melompat	[√] Berhasil [ ] Tidak
Pilih tombol keluar	Menampilkan materi lari gawang	Dapat menampilkan materi lari gawang	[√] Berhasil [ ] Tidak

11. Pengujian Menu Permainan Lompat Tinggi

Pengujian menu permainan lompat tinggi dengan kasus dan hasil uji, kemudian diamati dan menghasilkan kesimpulan yang dilihat pada tabel 4.13 berikut.

Tabel 4.13 Pengujian Permainan Lompat Tinggi

Kasus dan Hasil Uji (Uji Data Benar)			
Masukan	**Keluaran yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Tekan tombol keyboard kanan	Pemain berlari	Pemain dapat berlari	[√] Berhasil [ ] Tidak
Tekan keyboard kiri	Pemain berlari	Pemain dapat berlari	[√] Berhasil [ ] Tidak
Tekan tombol keyboard space	Pemain melompat	Pemain dapat melompat	[√] Berhasil [ ] Tidak
Pilih tombol keluar	Menampilkan materi lompat tinggi	Dapat menampilkan materi lompat tinggi	[√] Berhasil [ ] Tidak

12. Pengujian Menu Permainan Lompat Jauh

Pengujian menu permainan lompat jauh dengan kasus dan hasil uji, kemudian diamati dan menghasilkan kesimpulan yang dilihat pada table 4.14 berikut.

Tabel 4.14 Pengujian Permainan Lompat Jauh

Kasus dan Hasil Uji (Uji Data Benar)			
Masukan	**Keluaran yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Tekan tombol keyboard kanan	Pemain berlari	Pemain dapat berlari	[√] Berhasil [ ] Tidak
Tekan keyboard kiri	Pemain berlari	Pemain dapat berlari	[√] Berhasil [ ] Tidak
Tekan tombol keyboard space	Pemain melompat	Pemain dapat melompat	[√] Berhasil [ ] Tidak
Pilih tombol keluar	Menampilkan materi lompat jauh	Dapat menampilkan materi lompat jauh	[√] Berhasil [ ] Tidak

13. Pengujian Menu Permainan Lempar Lembing

Pengujian menu permainan lempar lembing dengan kasus dan hasil uji,

kemudian diamati dan menghasilkan kesimpulan yang dilihat pada tabel 4.15 berikut.

Tabel 4.15 Pengujian Permainan Lempar Lembing

Kasus dan Hasil Uji (Uji Data Benar)			
Masukan	**Keluaran yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Tekan tombol keyboard kanan	Pemain berlari	Pemain dapat berlari	[√] Berhasil [ ] Tidak
Tekan keyboard kiri	Pemain berlari	Pemain dapat berlari	[√] Berhasil [ ] Tidak
Tekan tombol keyboard space	Pemain melempar lembing	Pemain dapat melempar lembing	[√] Berhasil [ ] Tidak
Pilih tombol keluar	Menampilkan materi lempar lembing	Dapat menampilkan materi lempar lembing	[√] Berhasil [ ] Tidak

4.2.2 Kasus dan Hasil Pengujian *(White Box)*

Berikut ini adalah kasus menguji perangkat lunak yang telah dibangun menggunakan metode *white box*. Pengujian *white box* menggunakan *flow graph* yang digunakan untuk menggambarkan alur dari algoritma dan *grap matrix* yang digunakan untuk menggenerasi *flow graph*. Adapun pengujian *white-box* adalah sebagai berikut.

1. **Pengujian Algoritma *Fuzzy Logic***

Pengujian algoritma *Fuzzy Logic* merupakan proses dimana akan diuji implementasi *Fuzzy logic* terhadap NPC dalam menentukan kecepatan.

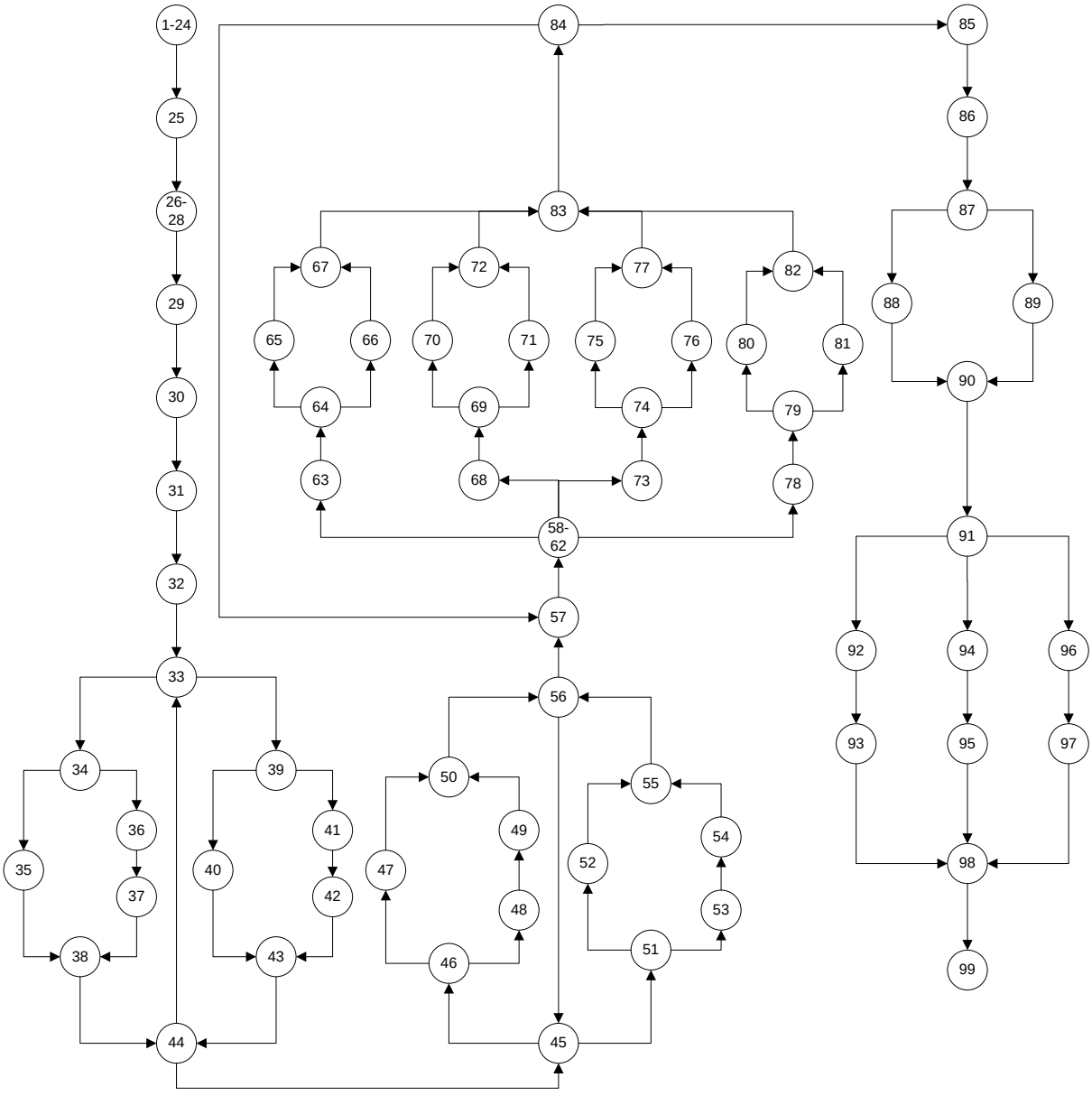
No	Koding
1.	`public float jarak;`
2.	`public float stamina=70;`
3.	`private float dekat;`
4.	`private int dekat1 = 0;`
5.	`private int dekat2 = 45;`
6.	`private int dekat3 = 55;`
7.	`private float jauh;`
8.	`private int jauh1 = 45;`

```
9.  private int jauh2 = 55;
10. private int jauh3 = 100;
11. private float lelah;
12. private int lelah1 = 0;
13. private int lelah2 = 45;
14. private int lelah3 = 55;
15. private float fit;
16. private int fit1 = 45;
17. private int fit2 = 55;
18. private int fit3 = 100;
19. public float lambat;
20. private float lambat1;
21. private float lambat2;
22. public float sedang;
23. public float cepat;
24. public float hasil;
25. void Start () {
26. InvokeRepeating("hitung_jarak",1,1);
27. InvokeRepeating("hitung_stamina",1,1);
28. InvokeRepeating("hitung_min",1,1);
29. }
30. void Update () {
31. stamina = Mathf.Round(this.GetComponent<enemyControl>().stamina);
32. }
33. void hitung_jarak() {
34. if ((dekat1<=jarak) && (jarak<=dekat2)) {
35. dekat = 1; jauh=0;
36. } else if ((dekat2<jarak) && (jarak<=dekat3)) {
37. dekat = -(jarak-dekat3)/(dekat3-dekat2);
38. }
39. if ((jauh1<jarak) && (jarak<jauh2)) {
```

```
40. jauh = ((jarak-jauh1)/(jauh2-jauh1));
41. } else if ((jauh2<=jarak) && (jarak<=jauh3)) {
42. jauh = 1; dekat=0;
43. }
44. }
45. void hitung_stamina() {
46. if ((lelah1<=stamina) && (stamina<=lelah2)) {
47. lelah=1; fit=0;
48. } else if ((lelah2<stamina) && (stamina<=lelah3)) {
49. lelah = -(stamina-lelah3)/(lelah3-lelah2);
50. }
51. if ((fit1<stamina) && (stamina<fit2)) {
52. fit = (stamina-fit1)/(fit2-fit1);
53. } else if ((fit2<=stamina) && (stamina<=fit3)) {
54. fit = 1; lelah=0;
55. }
56. }
57. void hitung_min(){
58. lambat=0;
59. lambat1=0;
60. lambat2=0;
61. sedang =0;
62. cepat=0;
63. if ((dekat>0) && (fit>0)) {
64. if (dekat<=fit) {
65. sedang = dekat;
66. } else sedang = fit;
67. }
68. if ((dekat>0) && (lelah>0)) {
69. if (dekat<=lelah) {
70. lambat = dekat;
```

```
71.  } else lambat1 = lelah;
72.  }
73.  if ((jauh>0) && (fit>0)) {
74.  if (jauh<=fit) {
75.  cepat = jauh;
76.  } else cepat = fit;
77.  }
78.  if ((jauh>0) && (lelah>0)) {
79.  if (jauh<=lelah) {
80.  sedang = jauh;
81.  } else lambat2 = lelah;
82.  }
83.  hasil_fuzzy();
84.  }
85.  void hasil_fuzzy() {
86.  if (lambat1>0) {
87.  if (lambat1>=lambat2) {
88.  lambat=lambat1;
89.  }else lambat=lambat2;
90.  }
91.  hasil = (lambat*20+sedang*40+cepat*60)/(lambat+sedang+cepat);
92.  if (hasil>=60) {
93.  this.GetComponent<enemyRun>().speed=
     this.GetComponent<enemyControl>().maxSpeed;
94.  } else if(hasil>=40) {
95.  this.GetComponent<enemyRun>().speed=
     this.GetComponent<enemyControl>().maxSpeed - 0.3f;
96.  } else if (hasil<40) {
97.  this.GetComponent<enemyRun>().speed=
     this.GetComponent<enemyControl>().maxSpeed - 0.5f;
98.  }
99.  }
```

A. Flow Graph



Gambar 4.14 Flowgraph Algoritma *Fuzzy Logic*

Keterangan :

= Menggambarkan kondisi

⟶ = Menggambarkan aksi

B. Cyclomatic Complexity V(G)

E = 85 N = 70

V(G) = E – N + 2

V(G) = 88 – 70 + 2

V(G) = (18) + 2 = 20

Keterangan : E = Jumlah Aksi

N = Jumlah Kondisi

C. Independent Path

Path 1 = 1-2-3-4-5-6-7-8-9-10-13-19-20-21-22-25-31-32-33-34-35-36-38-54-55-56-57-58-59-61-62-63-64-69-70

Path 2 = 1-2-3-4-5-6-7-8-9-11-12-13-19-20-21-22-25-31-32-33-34-35-36-38-54-55-56-57-58-59-61-62-63-64-69-70

Path 3 = 1-2-3-4-5-6-7-8-14-15-18-19-20-21-22-25-31-32-33-34-35-36-38-54-55-56-57-58-59-61-62-63-64-69-70

Path 4 = 1-2-3-4-5-6-7-8-14-16-17-18-19-20-21-22-25-31-32-33-34-35-36-38-54-55-56-57-58-59-61-62-63-64-69-70

Path 5 = 1-2-3-4-5-6-7-8-14-16-17-18-19-20-21-23-24-25-31-32-33-34-35-36-38-54-55-56-57-58-59-61-62-63-64-69-70

Path 6 = 1-2-3-4-5-6-7-8-14-15-18-19-20-26-27-30-31-32-33-34-35-36-38-54-55-56-57-58-59-61-62-63-64-69-70

Path 7 = 1-2-3-4-5-6-7-8-14-15-18-19-20-26-28-29-30-31-32-33-34-35-36-38-54-55-56-57-58-59-61-62-63-64-69-70

Path 8 = 1-2-3-4-5-6-7-8-14-15-18-19-20-26-28-29-30-31-32-33-34-35-37-38-54-55-56-57-58-59-61-62-63-64-69-70

Path 9 = 1-2-3-4-5-6-7-8-14-15-18-19-20-26-28-29-30-31-32-33-39-40-41-43-54-55-56-57-58-59-61-62-63-64-69-70

Path 10 = 1-2-3-4-5-6-7-8-14-15-18-19-20-26-28-29-30-31-32-33-39-40-42-43-54-55-56-57-58-59-61-62-63-64-69-70

Path 11 = 1-2-3-4-5-6-7-8-14-15-18-19-20-26-28-29-30-31-32-33-44-45-46-48-54-55-56-57-58-59-61-62-63-64-69-70

Path 12 = 1-2-3-4-5-6-7-8-14-15-18-19-20-26-28-29-30-31-32-33-44-45-47-48-54-55-56-57-58-59-61-62-63-64-69-70

Path 13 = 1-2-3-4-5-6-7-8-14-15-18-19-20-26-28-29-30-31-32-33-49-50-51-53-54-55-56-57-58-59-61-62-63-64-69-70

Path 14 = 1-2-3-4-5-6-7-8-14-15-18-19-20-26-28-29-30-31-32-33-49-50-52-53-54-55-56-57-58-59-61-62-63-64-69-70

Path 15 = 1-2-3-4-5-6-7-8-14-15-18-19-20-26-28-29-30-31-32-33-49-50-52-53-54-55-56-57-58-60-61-62-63-64-69-70

Path 16 = 1-2-3-4-5-6-7-8-14-15-18-19-20-26-28-29-30-31-32-33-49-50-52-53-54-55-56-57-58-59-61-62-65-66-69-70

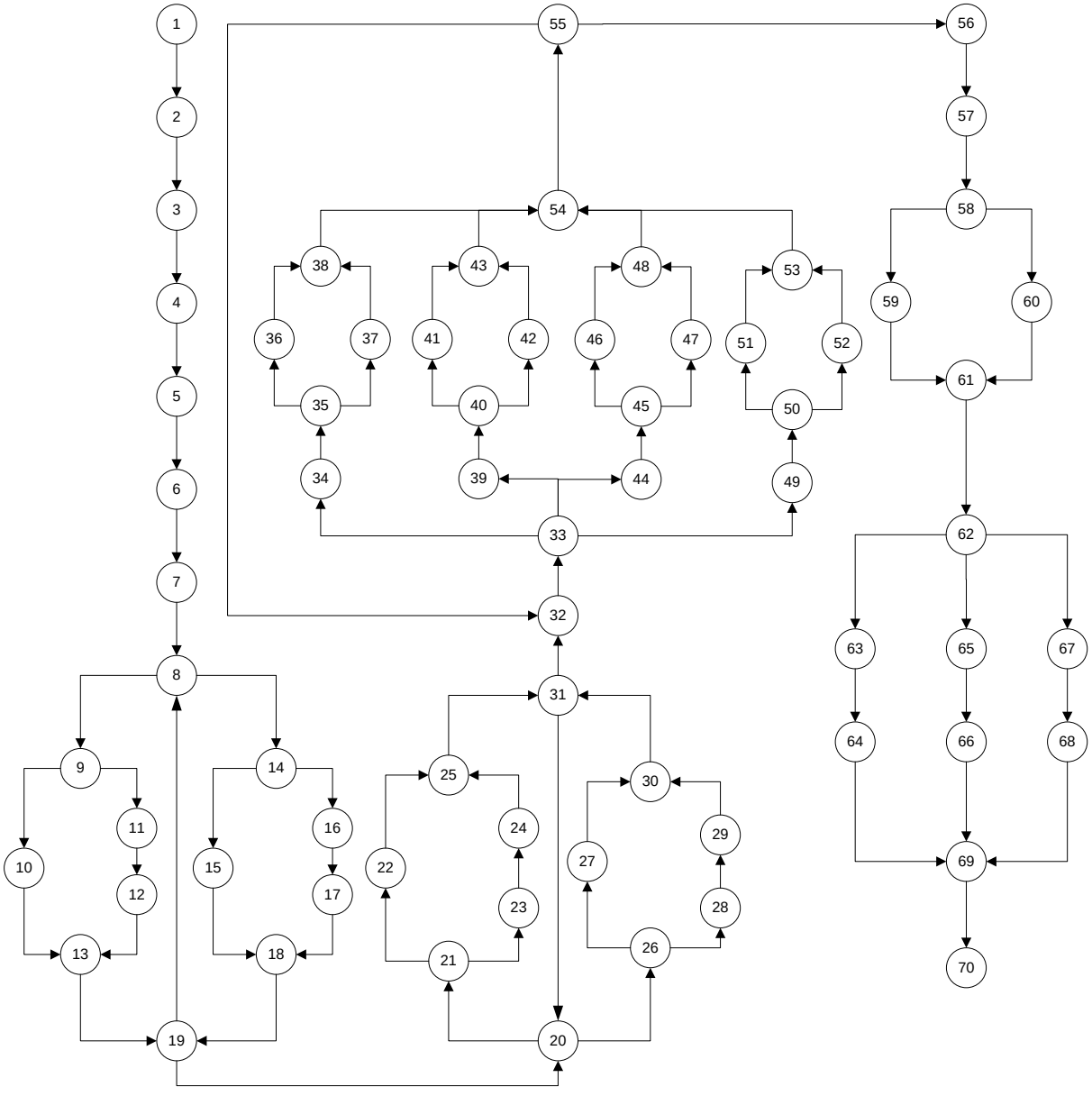
Path 17 = 1-2-3-4-5-6-7-8-14-15-18-19-20-26-28-29-30-31-32-33-49-50-52-53-54-55-56-57-58-59-61-62-67-68-69-70

Path 18 = 1-2-3-4-5-6-7-8-14-15-18-19-8-14-15-18-19-20-26-28-29-30-31-32-33-49-50-52-53-54-55-56-57-58-59-61-62-67-68-69-70

Path 19 = 1-2-3-4-5-6-7-8-14-15-18-19-20-26-28-29-30-31-20-26-28-29-30-31-32-33-49-50-52-53-54-55-56-57-58-59-61-62-67-68-69-70

Path 20 = 1-2-3-4-5-6-7-8-14-15-18-19-20-26-28-29-30-31-32-33-49-50-52-53-54-55-32-33-49-50-52-53-54-55-56-57-58-59-61-62-67-68-69-70

D. Penyederhanaan Flow Graph

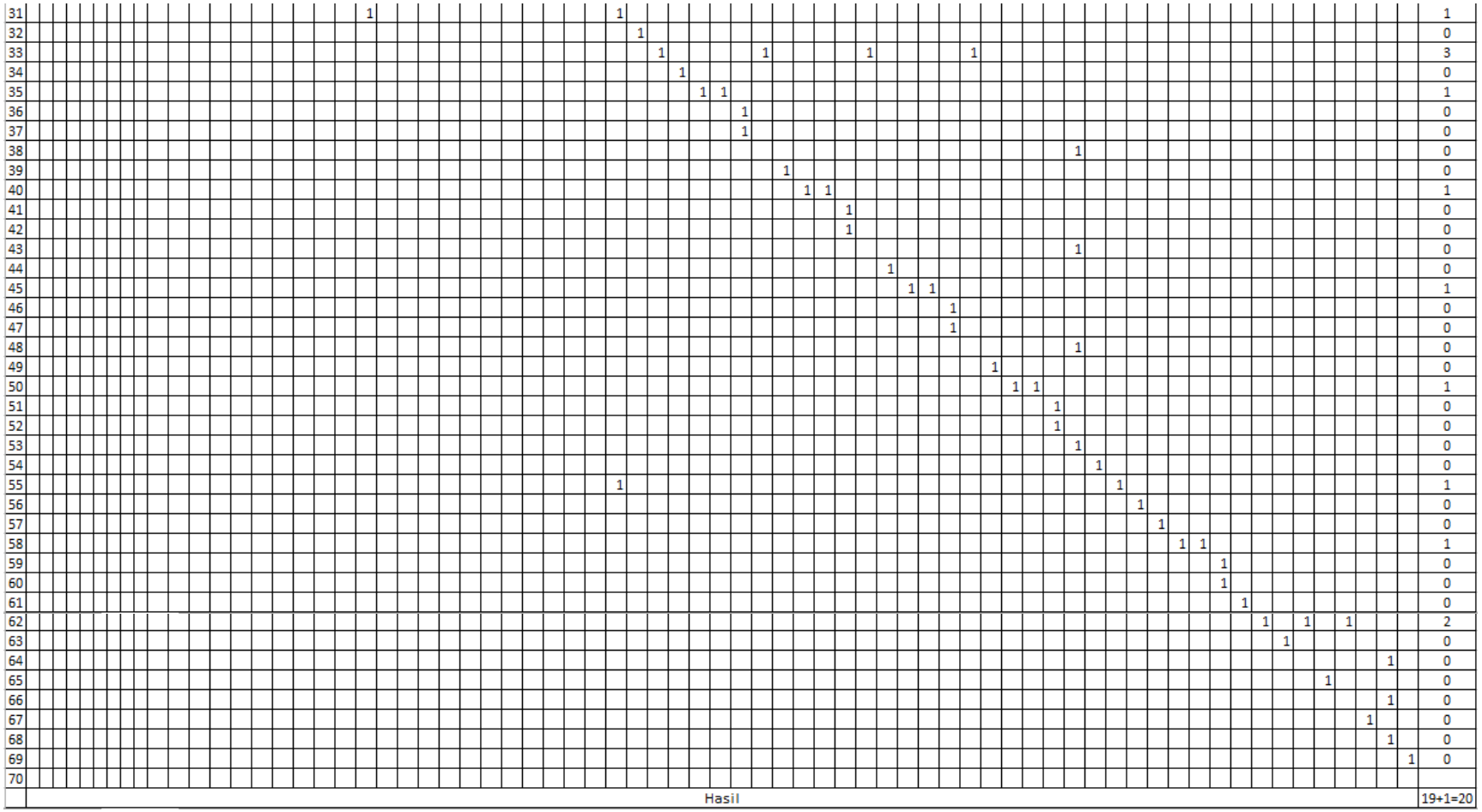


Gambar 4.15 Penyederhanaan Flow Grap

E. Graph Matrik

Tabel 4.16 Graph Matrik

	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	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	Hasil
1		1																																																																						0
2			1																																																																				0
3				1																																																																			0
4					1																																																																		0
5						1																																																																	0
6							1																																																																0
7								1																																																															0
8									1					1																																																									1
9										1	1																																																												1
10													1																																																										0
11												1																																																											0
12													1																																																										0
13																			1																																																				0
14															1	1																																																							1
15																		1																																																					0
16																	1																																																						0
17																		1																																																					0
18																			1																																																				0
19								1												1																																																			1
20																					1					1																																													1
21																						1	1																																																1
22																									1																																														0
23																								1																																															0
24																									1																																														0
25																															1																																								0
26																											1	1																																											1
27																														1																																									0
28																													1																																										0
29																														1																																									0
30																															1																																								0


Hasil
19+1=20

V(G) = jumlah Graph Matrik + 1

V(G) = 19 + 1

V(G) = 20

F. Kesimpulan

Berdasarkan hasil pengujian dihasilkan nilai Cyclomatic Complexity yang tidak sama yaitu 20, maka dapat disimpulkan bahwa pada proses algoritma f*uzzy logic* berjalan dengan baik, karena hasil pengujian memiliki nilai yang sama.

4.2.3 Pengujian *Beta*

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

Kuesioner disebarkan menggunakan teknik *Simple Random Sampling* dimana anggota *sample* dari populasi dilakukan secara acak tanpa memperhatikan strata yang ada dalam populasi. Kuesioner disebarkan kepada 50 orang dari umur 9 tahun ke atas, dari hasil kuesioner tersebut akan dilakukan perhitungan agar dapat diambil kesimpulan terhadap penilaian penerapan sistem yang telah dibangun. Kuesioner terdiri dari 5 pernyataan dengan menggunakan skala 1 sampai 5 (skala likert).

Untuk mencari nilai persentase dari masing-masing jawaban kuesioner digunakan rumus skala likert sebagai berikut :

P = (S/Skorideal) x 100%

Keterangan :

P = Nilai persentase yang dicari

S = Jumlah frekuensi dikalikan dengan nilai yang ditetapkan jawaban

Skorideal = Nilai tertinggi dikalikan dengan jumlah sampel

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

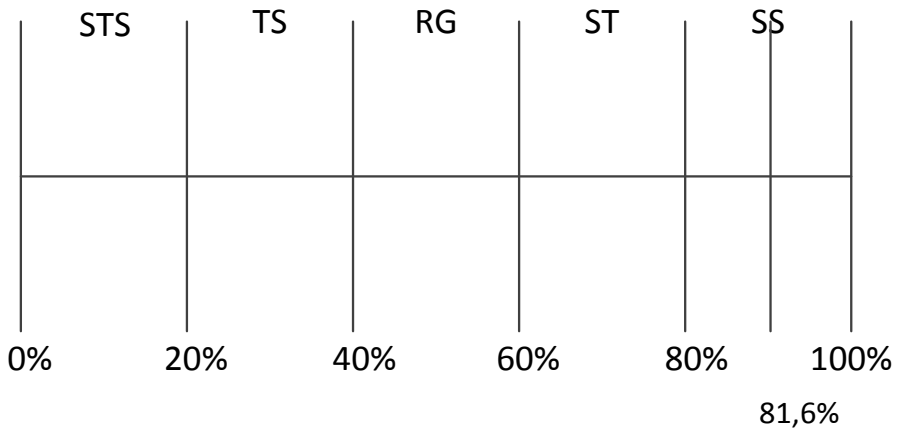
1. Apakah anda setuju *game* olahraga atletik ini menarik untuk dimainkan?

Tabel 4.17 Perhitungan Persentase Pertanyaan ke-1

Pertanyaan	Keterangan	Skor	Frekuensi	S
1	Sangat Setuju	5	9	45
	Setuju	4	38	152
	Ragu-Ragu	3	1	3
	Tidak Setuju	2	2	4
	Sangat Tidak Setuju	1	0	0
Jumlah			50	204

P = (204/250)x100% = 81,6%

Dapat disimpulkan bahwa *game* yang dibangun menarik untuk dimainkan, terhitung 81,6% dari yang diharapkan 100%, atau dapat dikategorikan sebagai sangat setuju, untuk lebih jelasnya dapat dilihat pada gambar 4.16.



Gambar 4.16 Interpretasi Skor Pertanyaan ke-1

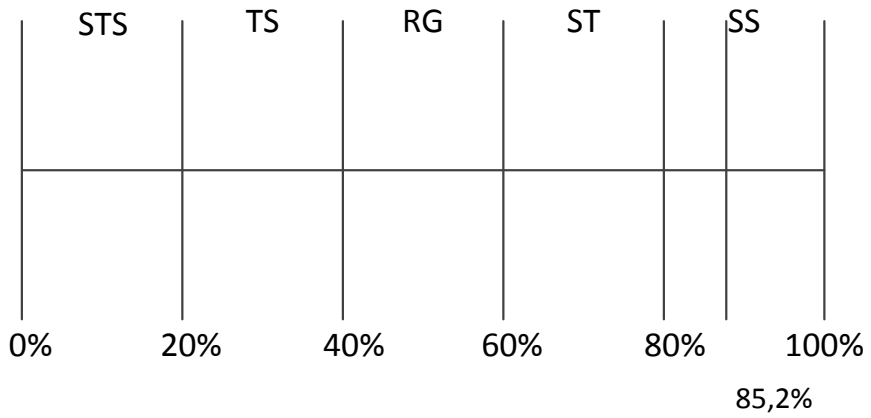
2. Apakah anda setuju *game* olahraga atletik ini memberikan informasi yang baru tentang olahraga atletik?

Tabel 4.18 Perhitungan Persentase Pertanyaan ke-2

Pertanyaan	Keterangan	Skor	Frekuensi	S
2	Sangat Setuju	5	18	90
	Setuju	4	27	108
	Ragu-Ragu	3	5	15
	Tidak Setuju	2	0	0
	Sangat Tidak Setuju	1	0	0
Jumlah			50	213

P = (213/250)x100% = 85,2%

Dapat disimpulkan bahwa *game* yang dibangun memberikan informasi yang baru tentang atletik, terhitung 85,2% dari yang diharapkan 100%, atau dapat dikategorikan sebagai sangat setuju, untuk lebih jelasnya dapat dilihat pada gambar 4.17.



Gambar 4.17 Interpretasi Skor Pertanyaan ke-2

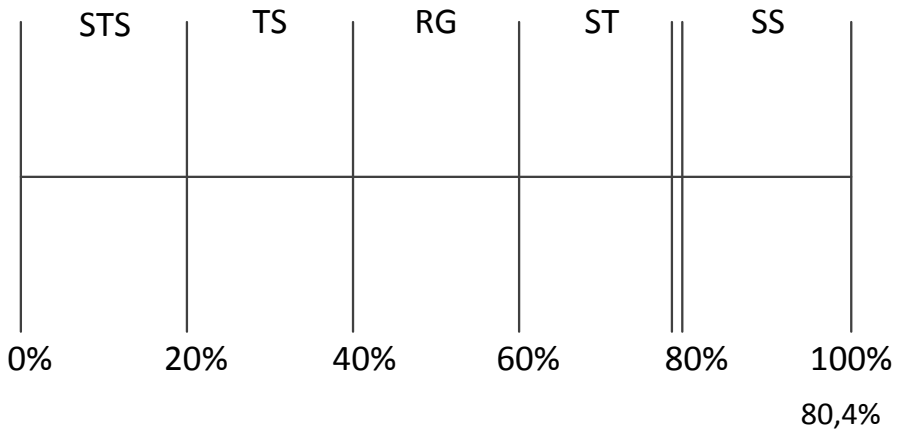
3. Apakah anda *game* olahraga atletik ini sudah cukup jelas dalam pengenalan tentang olahraga atletik?

Tabel 4.19 Perhitungan Persentase Pertanyaan ke-3

Pertanyaan	Keterangan	Skor	Frekuensi	S
3	Sangat Setuju	5	10	50
	Setuju	4	32	128
	Ragu-Ragu	3	7	21
	Tidak Setuju	2	1	2
	Sangat Tidak Setuju	1	0	0
Jumlah			50	201

P = (201/250)x100% = 80,4%

Dapat disimpulkan bahwa *game* yang dibangun sudah cukup jelas dalam pengenalan tentang olahraga atletik, terhitung 84% dari yang diharapkan 100%, atau dapat dikategorikan sebagai sangat setuju, untuk lebih jelasnya dapat dilihat pada gambar 4.18.



Gambar 4.18 Interpretasi Skor Pertanyaan ke-3

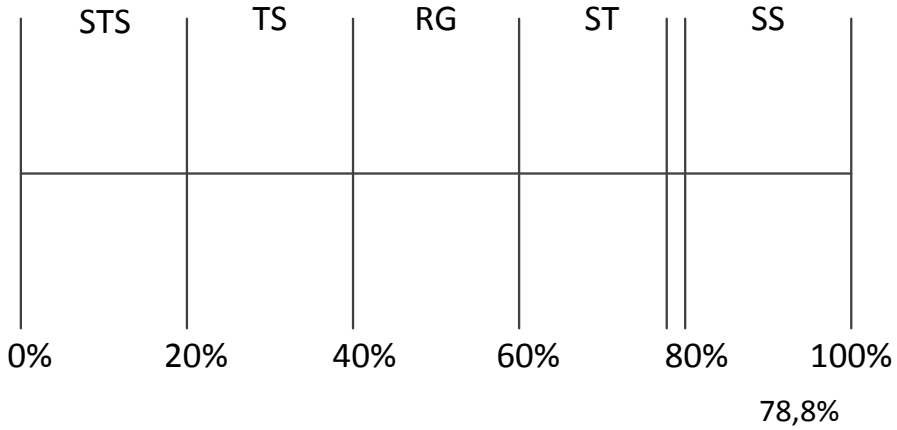
4. Apakah anda setuju *game* olahraga atletik ini sudah cukup jelas dalam penyampaian materi tentang olahraga atletik?

Tabel 4.20 Perhitungan Persentase Pertanyaan ke-4

Pertanyaan	Keterangan	Skor	Frekuensi	S
4	Sangat Setuju	5	17	85
	Setuju	4	15	60
	Ragu-Ragu	3	16	48
	Tidak Setuju	2	2	4
	Sangat Tidak Setuju	1	0	0
Jumlah			50	123

P = (197/250)x100% = 78,8%

Dapat disimpulkan bahwa *game* yang dibangun sudah cukup jelas dalam penyampaian materi tentang olahraga atletik, terhitung 78,8 % dari yang diharapkan 100%, atau dapat dikategorikan sebagai setuju, untuk lebih jelasnya dapat dilihat pada gambar 4.19.



Gambar 4.19 Interpretasi Skor Pertanyaan ke-4

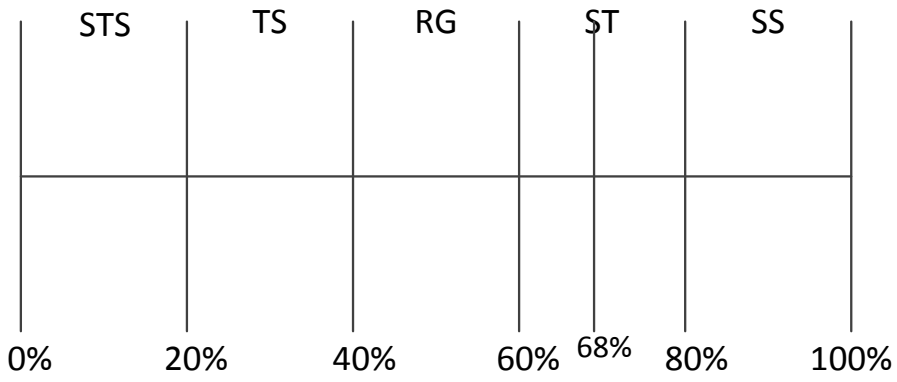
5. Apakah anda setuju *game* olahraga atletik ini sudah sesuai aturannya dengan aslinya?

Tabel 4.21 Perhitungan persentase pertanyaan ke-5

Pertanyaan	Keterangan	Skor	Frekuensi	S
5	Sangat Setuju	5	15	45
	Setuju	4	22	88
	Ragu-Ragu	3	12	36
	Tidak Setuju	2	1	2
	Sangat Tidak Setuju	1	0	0
Jumlah			50	171

P = (171/250)x100% = 68,4%

Dapat disimpulkan bahwa *game* yang dibangun sudah sesuai aturannya dengan aslinya, terhitung 68,4% dari yang diharapkan 100%, atau dapat dikategorikan sebagai setuju, untuk lebih jelasnya dapat dilihat pada gambar 4.20.



Gambar 4.20 Interpretasi Skor Pertanyaan ke-5

4.2.3.1 Kesimpulan Pengujian *Beta*

Berdasarkan hasil pengujian *beta*, maka didapatkan kesimpulan sebagai berikut:

1. Aplikasi *game* yang dibangun menarik untuk dimainkan
2. Aplikasi *game* yang dibangun memberikan informasi yang baru bagi pemainnya
3. Aplikasi *game* yang dibangun dapat mengenalkan olahraga atletik kepada pemainnya

4. Aplikasi *game* yang dibangun memberikan materi yang cukup jelas kepada pemainnya
5. Aplikasi *game* yang dibangun memberikan peraturan yang sesuai kepada pemainnya