BAB 4
IMPLEMENTASI DAN PENGUJIAN

4.1 Implementasi

Tahap implementasi merupakan tahap dimana aplikasi game siap dioperasikan. Pada bagian ini akan dijelaskan mengenai lingkungan implementasi, yang mana terdiri dari implementasi perangkat keras, implementasi perangkat lunak, dan implementasi program serta antarmukanya.

4.1.1 Implementasi Perangkat Keras

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

1. *CPU : Intel I7 Processor* 1,7 GHz
2. *RAM* : 4GB
3. *VGA* : *NVIDIA GeForce* 1 GB
4. *Hard Disk 1 TB*
5. *Keyboard* : 1 buah
6. *Mouse* : 1 buah
7. *Speaker* : 1 buah

4.1.2 Implementasi Perangkat Lunak

Perangkat lunak yang digunakan untuk menjalankan game *Fly Eagle* adalah Windows 7.

4.1.3 Implementasi Aplikasi

File yang ada pada *game fly eagle* dibungkus dalam satu folder *fly eagle*, berikut langkah-langkah untuk menjalankan game tersebut :

1. Buka *folder fly eagle* dengan cara klik *folder*nya 2 kali
2. Akan terdapat banyak *file* dan *sub folder* pada *folder fly eagle*, setelah itu *double* klik pada *flyeaglerun.exe*
3. Akan tampil windows game fly eagle dan itu menandakan game sukses dijalankan.

4.1.4 Implementasi Antar Muka

Tabel 4.1 Implementasi antar muka

No	Menu	Deskripsi	Nama File
1.	Menu utama	Tamplian awal aplikasi	MenuUtama.capx EventMenuUtama.xml
2.	Petunjuk	Tampilan petunjuk cara bermain	Petunjuk.capx EventPetunjuk.xml
3.	Top Score	Tampilan untuk melihat top score permainan	TopScore.capx EventTopScore.xml
4.	Intro	Tampilan cerita dari permainan	Intro.capx EventIntro.xml
5.	Level 1	Tampilan level 1 permainan	Level1.capx Eventlevel1.xml
6.	Level 2	Tampilan level 2 permainan	Level2.capx Eventlevel1.xml
7.	Level 3	Tampilan level 3 permainan	Level3.capx Eventlevel3.xml
8.	Game over	Tampilan jika pemain mengalami game over	GameOver.capx EventGameOver.xml

9.	Tamat	Tampilan jika pemain berhasil menyelesaikan permainan	Tamat.capx EventTamat.xml

4.1.5 Implementasi *Class*

Tabel 4.2 Implementasi *class*

No	Nama *Class*	Nama *File* Fisik
1.	System	system.js
2.	Layout	layout.js
3.	Audio	Audio.js
4.	Player	Player.js
5.	Item	Item.js
6.	ItemSpeed	ItemSpeed.js
7.	Daging	Daging.js
8.	Musuh	Musuh.js
9.	Rintangan	Rintangan.js
10.	Pathfind	Pathfind.js

4.2 Pengujian Sistem

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.

4.2.1 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.

4.2.1.1 Rencana Pengujian

Rencana pengujian yang akan dilakukan pada game fly eagle dapat dilihat pada tabel 4.2.

Tabel 4.3 Rencana Pengujian[11]

No.	Komponen yang diuji	Skenario Pengujian	Jenis Pengujian
1.	Use Case memulai Permainan	Menekan tombol mulai permainan	*Black box*
2.	Use Case melihat petunjuk	Menekan tombol petunjuk	*Black box*
3.	Use case melihat top score	Menekan tombol top score	*Black box*
4.	Use case mengambil daging	Menggerakkan karakter menyentuh daging	*Black box*
5.	Use case mengambil item speed	Menggerakkan karakter menyentuh item speed	*Black box*
6.	Use case membuat karakter melompat	Menekan tombol kanan mouse	*Black box*
7.	Use case menghindari musuh dan rintangan	Menggerakkan karakter mendekati musuh dan rintangan	*Black box*
8.	Jalur musuh burung	Penulusuran algoritma A*	*White box*

4.2.1.2 Hasil Pengujian *Black Box*

Pengujian *Black Box* adalah pengujian yang berfokus pada persyaratan fungsional perangkat lunak, pengujian ini memperhatikan masukan ke sistem dan

respon sistem. Berikut ini adalah kasus untuk menguji perangkat lunak yang sudah dibangun menggunakan metode *Black Box.*

1. Use Case Memulai permainan

Tabel 4.4 Hasil Pengujian memulai permainan

No	Skenario	Test Case	Hasil Yang Diharapkan	Status
1.	Berhasil memulai permainan	User menekan tombol mulai di menu utama	Menampilkan halaman intro	[√] Berhasil [ ] Tidak Berhasil

2. Use Case Melihat petunjuk

Tabel 4.5 Hasil Pengujian melihat petunjuk

No	Skenario	Test Case	Hasil Yang Diharapkan	Status
1.	Berhasil melihat petunjuk	User menekan tombol petunjuk di menu utama	Menampilkan halaman petunjuk	[√] Berhasil [ ] Tidak Berhasil

3. Use Case melihat *top score*

Tabel 4.6 Hasil Pengujian melihat *top score*

No	Skenario	Test Case	Hasil Yang Diharapkan	Status
1.	Berhasil melihat top score	User menekan tombol *top score* di menu utama	Menampilkan halaman top score	[√] Berhasil [ ] Tidak Berhasil

4. Use Case mengambil daging

Tabel 4.7 Hasil Pengujian mengambil daging

No	Skenario	Test Case	Hasil Yang Diharapkan	Status
1.	Berhasil mengambil daging dan menambah score	Kondisi awal : berada dalam permainan Menggerakkan karakter menyentuh daging	Daging menghilang dan score bertambah 1	[√] Berhasil [ ] Tidak Berhasil

5. Use Case mengambil item speed

Tabel 4.8 Hasil Pengujian mengambil item speed

No	Skenario	Test Case	Hasil Yang Diharapkan	Status
1.	Berhasil mengambil item speed dan menambah kecepatan karakter	Kondisi awal : berada dalam permainan Menggerakkan karakter menyentuh item speed	Item speed menghilang dan kecepatan karkter bertambah	[√] Berhasil [ ] Tidak Berhasil

6. Use Case membuat karakter melompat

Tabel 4.9 Hasil Pengujian membuat karakter melompat

No	Skenario	Test Case	Hasil Yang Diharapkan	Status
1.	Berhasil membuat karakter melompat	Kondisi awal : berada dalam permainan Menekan tombol kanan mouse	Karakter melompat	[√] Berhasil [ ] Tidak Berhasil

7. Use Case menghindari musuh dan rintangan

Tabel 4.10 Hasil Pengujian menghindari musuh dan rintangan

No	Skenario	Test Case	Hasil Yang Diharapkan	Status
1.	Berhasil menghindari rintangan	Kondisi awal : berada dalam permainan Menggerakkan karakter menghindari rintangan	Karakter tetap bergerak	[√] Berhasil [ ] Tidak Berhasil
2.	Karakter menyentuh rintangan atau mush	Kondisi awal : berada dalam permainan	Menampilkan halaman game over	[√] Berhasil [ ] Tidak Berhasil

		Menggerakkan karakter menyentuh rintangan dan musuh		

4.2.1.3 Pengujian *White Box*

Berikut ini adalah kasus menguji perangkat lunak yang telah dibangun menggunakan metode *white box*. Pengujian *whitebox* 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 A* (A star)

Pengujian algoritma A* merupakan proses dimana akan diuji implementasi A* terhadap NPC burung dalam melakukan pencarian tujuan.

NO	Coding

1.	`pathfinder.prototype.addCellToOpenList = function (x_, y_, g_)`
	`{`
	`this.closedCache.hasOwnProperty(((x_ << 16) + y_).toString())`
	`return;`
	`var i, len, c;`
	`var curCellCost = this.at(x_, y_);`

2.	`for (i = 0, len = this.openList.length; i < len; i++)` `{`
3.	`c = this.openList[i];` `x_ === c.x && y_ === c.y`

4.	if (this.currentNode.g + g_ + curCellCost < c.g) {

5.	c.parent = this.currentNode;
	c.g = this.currentNode.g + g_ + curCellCost;
	c.h = this.estimateH(c.x, c.y);
	c.f = c.g + c.h;

6.	if (this.openList.length === 1) {
7.	return; }

8.	this.openList.splice(i, 1);
	this.insertToOpenList(c);
	}

9.	return; } }

10.	c = allocNode();
	c.x = x_;
	c.y = y_;
	c.h = this.estimateH(x_, y_);
	c.g = this.currentNode.g + g_ + curCellCost;
	c.f = c.h + c.g;
	c.parent = this.currentNode;

11.	if (!this.openList.length) {

12.	this.openList.push(c);
	return;
	}

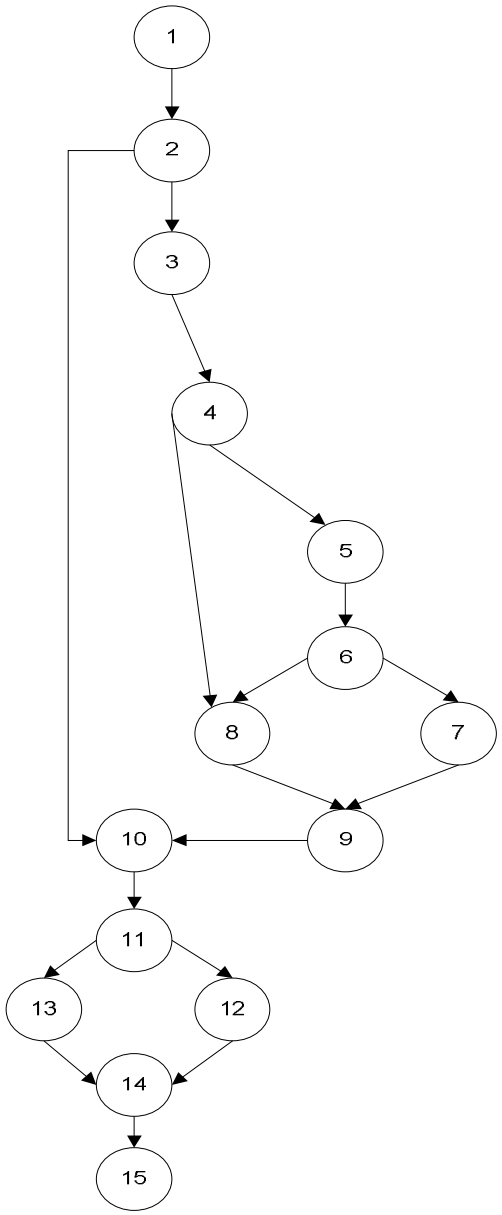
13.	this.insertToOpenList(c); };

14.	function quickAbs(x)
	{
	return x < 0 ? -x : x;
	};

15.	pathfinder.prototype.estimateH = function (x_, y_)
	{
	var dx = quickAbs(x_ - this.targetX);
	var dy = quickAbs(y_ - this.targetY);

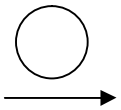
```
                return dx * 10 + dy * 10;
        };
```

2. Flow Graph



Gambar 4.1 *Flow Graph* Algoritma A*

Keterangan :



= Menggambarkan kondisi

= Menggambarkan aksi

3. ***Cyclomatic Complexity*V(G)**

V(G) = E –N + 2

V(G) = 18–15+ 2

V(G) = 5

Keterangan:

E =Jumlah busur pada *flowgraph*

N =Jumlah simpul pada *flowgraph*

V(G) =*Cyclomatic Complexity*

4. ***Independent Path***

Path 1 : 1,2,3,4,5,6,7,9,10,11,12,14,15

Path 2 : 1,2,3,4,5,6,8,9,10,11,13,14,15

Path 3 : 1,2,3,4,5,6,8,9,10,11,12,14,15

Path 4 : 1,2,3,4,8,9,10,11,12,14,15

Path 5 : 1,2,10,11,12,14,15

5. **Hasil Uji**

Tabel 4.11 Hasil Pengujian *White Box*

NO	Path	Yang diharapkan	Hasil uji	Keterangan
1.	1,2,3,4,5,6,7,9,10,11, 12,14,15	Menghasilkan *rute* terpendek dari A*	Menghasilkn *rute* terpendek dari A*	[√] Alur terlewati [ ] Alur tidak terlewati
2.	1,2,3,4,5,6,8,9,10,11, 13,14,15	Memeriksa *node* tetangga dari *node* yang diperiksa	Memeriksa *node* tetangga dari *node* yang diperiksa	[√] Alur terlewati [ ] Alur tidak terlewati
3.	1,2,3,4,5,6,8,9,10,11, 12,14,15	Memeriksa apakah *node* tetangga dari *node* yang diperiksa masuk pada *closed*	Memeriksa apakah *node* tetangga dari *node* yang diperiksa masuk pada *closed*	[√] Alur terlewati [ ] Alur tidak terlewati
4.	1,2,3,4,8,9,10,11,12,1 4,15	Memeriksa *openlist*	Memeriksa *openlist*	[√] Alur terlewati [ ] Alur tidak terlewati
5.	1,2,10,11,12,14,15	Tidak melakukan perulangan	Tidak melakukan perulangan Open List kosong	[√] Alur terlewati [ ] Alur tidak terlewati

6. Kesimpulan

Berdasarkan hasil pengujian yang dilakukan maka bisa diambil kesimpulan bahwa masing-masing *Independent Path* bisa dijalankan dan dieksekusi

4.2.1.4 Kesimpulan Pengujian Alpha

Dari hasil pengujian yang dilakukan, dapat disimpulkan bahwa sistem sudah berjalan sesuai dengan yang diharapkan. Secara fungsional sistem sudah dapat menghasilkan keluaran yang diharapkan.

4.2.2 Pengujian Beta

Pengujian beta merupakan pengujian langsung kepada pengguna untuk mencoba aplikasi yang baru. Pengujian yang dilakukan adalah dengan cara membagikan kuesioner kepada 30 pengguna aplikasi, dengan range usia sesuai dengan tujuan pembuatan *game Fly Eagle*. Berikut adalah pertanyaan dan hasil kuesioner yang telah dibagikan menggunakan rumus skala likert :

$$P = \frac{S}{skorideal} \times 100\% \quad (4.1)$$

Keterangan :

P = Nilai presentase yang dicari

S = Jumlah frekuensi dikalikan dengan nilai yang ditetapkan jawaban

skorideal = Nilai tertinggi dikalikan dengan jumlah sampel

100% = Bilangan tetap

Berikut hasil kuesioner yang telah diambil :

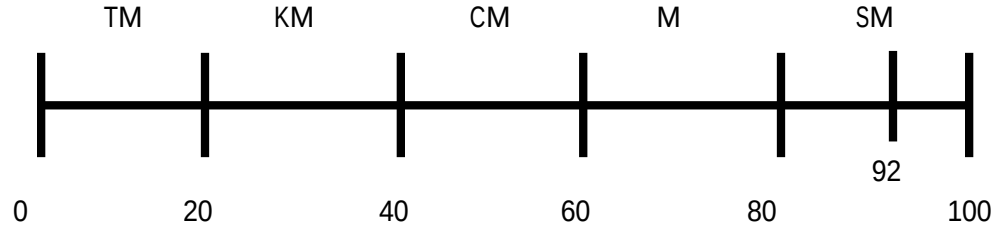
1. Apakah anda setuju variasi rintangan pada *game fly eagle* lebih banyak dan menarik daripada *game flap flap bird*?

Tabel 4.12 Perhitungan kuesioner pertanyaan 1

Pertanyaan	No	Keterangan	Skor	Frekuensi	S
1	1	Sangat setuju	5	18	90
	2	Setuju	4	12	48
	3	Cukup setuju	3	0	0
	4	Kurang setuju	2	0	0
	5	Tidak setuju	1	0	0
Jumlah				**30**	**138**

$$P = \frac{138}{150} \times 100\% = 92\%$$

Berdasarkan hasil persentase diatas maka dapat disimpulkan bahwa penilain variasi ritangan *game fly eagle* lebih banyak dan menarik dari *flap bird* sebesar 92% dari yang diharapkan 100% atau bisa dikategorikan sebagai sangat menarik. Berikut kriteria interpretasi skor :



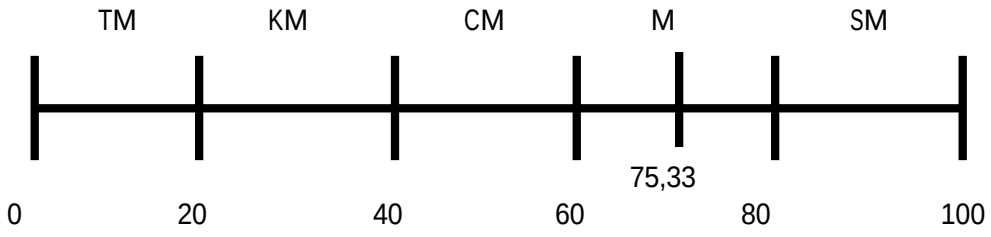
2. Apakah anda setuju tingkat kesulitan setiap level yang ada dalam *game fly eagle* sudah cukup menarik dan lebih baik daripada *game flap flap bird*?

Tabel 4.13 Perhitungan kuesioner pertanyaan 2

Pertanyaan	No	Keterangan	Skor	Frekuensi	S
2	1	Sangat setuju	5	8	40
	2	Setuju	4	9	36
	3	Cukup setuju	3	11	33
	4	Kurang setuju	2	2	4
	5	Tidak setuju	1	0	0
Jumlah				**30**	**113**

$$P = \frac{113}{150} \times 100\% = 75{,}33\%$$

Berdasarkan hasil persentase diatas maka dapat disimpulkan bahwa penilain tingkat kesulitan level *game fly eagle* sudah baik dan lebih menarik dari *flap bird* sebesar 75,33% dari yang diharapkan 100% atau bisa dikategorikan sebagai setuju. Berikut kriteria interpretasi skor :



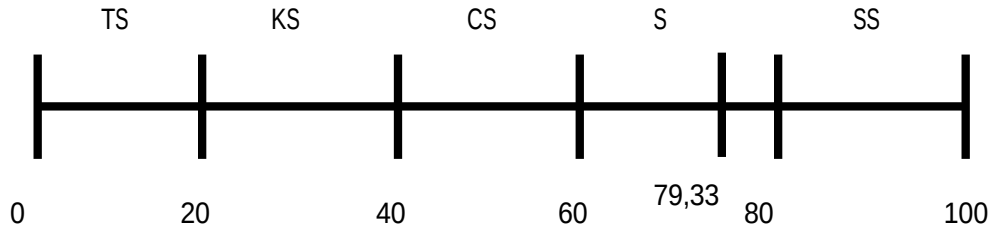
3. Apakah anda setuju dengan adanya musuh yang pintar pada *game fly eagle* membuat game ini lebih menantang dan menarik daripada *game flap flap bird?*

Tabel 4.14 Perhitungan kuesioner pertanyaan 3

Pertanyaan	No	Keterangan	Skor	Frekuensi	S
3	1	Sangat setuju	5	6	30
	2	Setuju	4	17	68
	3	Cukup setuju	3	7	21
	4	Kurang setuju	2	0	0
	5	Tidak setuju	1	0	0
Jumlah				**30**	**119**

$$P = \frac{119}{150} \times 100\% = 79{,}33\%$$

Berdasarkan hasil persentase diatas maka dapat disimpulkan bahwa penilain dengan adanya musuh yang pintar membuat *game fly eagle* lebih menarik dari *flap bird* sebesar 79,33% dari yang diharapkan 100% atau bisa dikategorikan sebagai setuju. Berikut kriteria interpretasi skor :



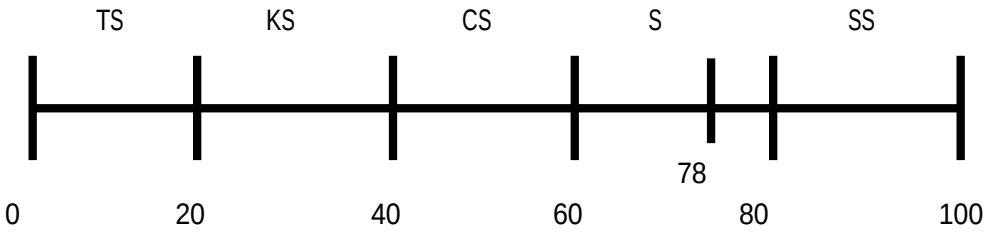
4. Apakah anda setuju anda setuju *game fly eagle* lebih menarik daripada *game flap flap bird*?

Tabel 4.15 Perhitungan kuesioner pertanyaan 4

Pertanyaan	No	Keterangan	Skor	Frekuensi	S
4	1	Sangat setuju	5	7	35
	2	Setuju	4	13	52
	3	Cukup setuju	3	10	30
	4	Kurang setuju	2	0	0
	5	Tidak setuju	1	0	0
Jumlah				**30**	**117**

$$P = \frac{117}{150} \times 100\% = 78\%$$

Berdasarkan hasil persentase diatas maka dapat disimpulkan bahwa penilain *game fly eagle* lebih menarik daripada game *flap flap bird* sebesar 78% dari yang diharapkan 100% atau bisa dikategorikan sebagai setuju. Berikut kriteria interpretasi skor :



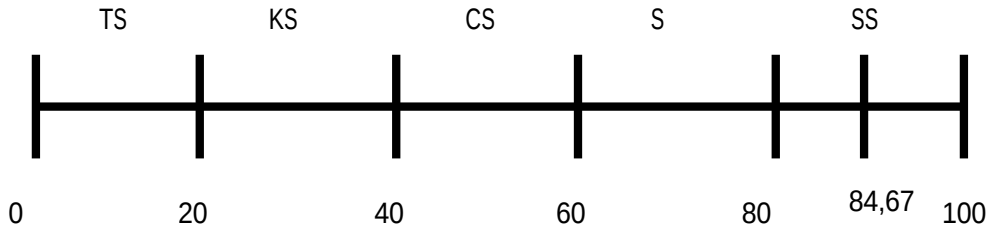
5. Apakah *game fly eag*le menarik untuk dimainkan ?

Tabel 4.16 Perhitungan kuesioner pertanyaan 5

Pertanyaan	No	Keterangan	Skor	Frekuensi	S
5	1	Sangat menarik	5	12	60
	2	Menarik	4	13	52
	3	Cukup menarik	3	5	15
	4	Kurang menarik	2	0	0
	5	Tidak menarik	1	0	0
Jumlah				**30**	**127**

$$P = \frac{127}{150} \times 100\% = 84{,}67\%$$

Berdasarkan hasil persentase diatas maka dapat disimpulkan bahwa penilain game fly eagle menarik untuk dimainkan sebesar 84,67% dari yang diharapkan 100% atau bisa dikategorikan sebagai sangat setuju. Berikut kriteria interpretasi skor :



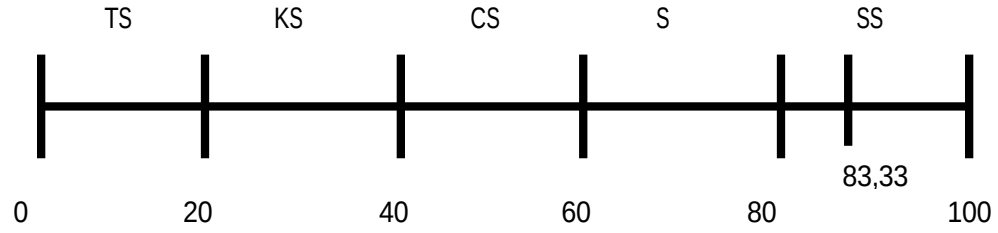
6. Apakah anda setuju aturan dan cara bermain pada *game fly eagle* mudah untuk dipahami?

Tabel 4.17 Perhitungan kuesioner pertanyaan 6

Pertanyaan	No	Keterangan	Skor	Frekuensi	S
6	1	Sangat setuju	5	10	50
	2	Setuju	4	15	60
	3	Cukup setuju	3	5	15
	4	Kurang setuju	2	0	0
	5	Tidak setuju	1	0	0
Jumlah				**30**	**125**

$$P = \frac{125}{150} \times 100\% = 83{,}33\%$$

Berdasarkan hasil persentase diatas maka dapat disimpulkan bahwa penilain aturan dan cara bermain *game fly eagle* mudah dipahami sebesar 83,33% dari yang diharapkan 100% atau bisa dikategorikan sebagai sangat setuju. Berikut kriteria interpretasi skor :



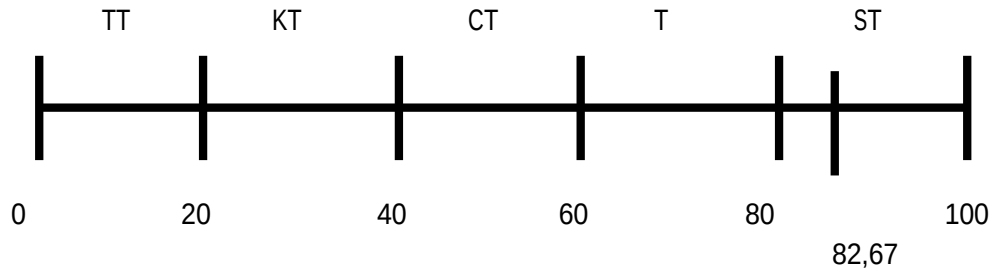
7. Apakah game fly eagle membuat anda tertarik untuk memainkannya kembali?

Tabel 4.18 Perhitungan kuesioner pertanyaan 7

Pertanyaan	No	Keterangan	Skor	Frekuensi	S
7	1	Sangat tertarik	5	6	30
	2	Tertarik	4	22	88
	3	Cukup tertarik	3	2	6
	4	Kurang tertarik	2	0	0
	5	Tidak tertarik	1	0	0
Jumlah				**30**	**124**

$$P = \frac{124}{150} \times 100\% = 82{,}67\%$$

Berdasarkan hasil persentase diatas maka dapat disimpulkan bahwa penilain *game fly eagle* membuat tertarik untuk dimainkan kembali sebesar 82,67% dari yang diharapkan 100% atau bisa dikategorikan sebagai sangat tertarik. Berikut kriteria interpretasi skor :



4.2.2.1 Kesimpulan Pengujian Beta

Dari kuesioner yang dibagikan, dapat disimpulkan bahwa:

1. Sebanyak 92% responden menyatakan sangat setuju variasi rintangan *fly eagle* lebih banyak dan menarik daripada *flap flap bird.*
2. Sebanyak 75,33% responden menyatakan setuju tingkat kesulitan level *game fly eagle* sudah baik dan lebih menarik daripada *flap flap bird.*
3. Sebanyak 79,33% responden menyatakan setuju dengan adanya musuh yang pintar membuat game *fly eagle* lebih menarik dari *flap flap bird.*
4. Sebanyak 78% responden setuju game *fly eagle* lebih menarik dari *flap flap bird.*
5. Sebanyak 84,67% responden sangat setuju bahwa *game fly eagle* menarik untuk dimainkan.
6. Sebanyak 83,33% responden sangat setuju bahwa aturan dalam *game fly eagle* mudah dipahami.
7. Sebanyak 82,67% responden sangat tertarik untuk memainkan *fly eagle* kembali.