BAB 4
IMPLEMENTASI DAN PENGUJIAN SISTEM

4.1 Implementasi Sistem

Tahap implementasi merupakan tahap penciptaan perangkat lunak dan juga tahap kelanjutan dari kegiatan perancangan aplikasi, tahap ini merupakan tahap dimana aplikasi siap untuk dioperasikan. Aplikasi ini diimplementasikan dengan nama Tradisional Kelereng dengan file extension-nya .Apk. Aplikasi yang diuji bisa langsung di-*install* dan tidak memerlukan *software* pendukung.

4.1.1 Perangkat Pengujian Aplikasi

Terdapat dua jenis perangkat yang digunakan dalam pengujian aplikasi ini, yaitu perangkat keras dan perangkat lunak. Spesifikasi perangkat keras yang digunakan selama pengujian aplikasi ini memiliki spesifikasi sebagai berikut :

1. *Processor* 800 MHz
2. RAM 512MB
3. Memory 2 GB
4. *Display 320 x 480 Pixels.*

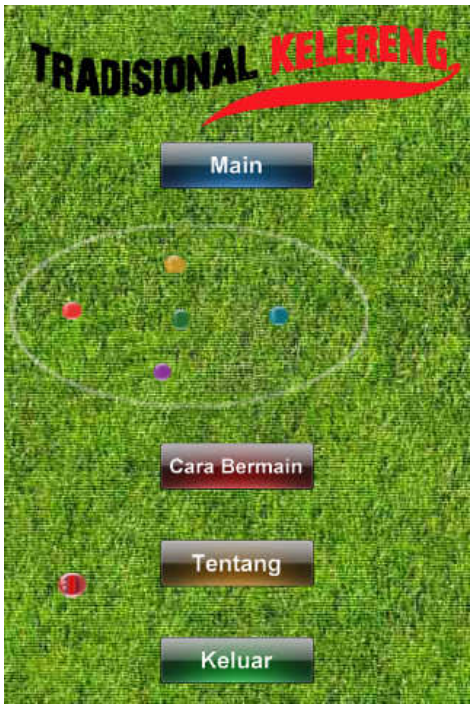
Sedangkan Spesifikasi minimum perangkat lunak (*software*) yang digunakan untuk implementasi Game Tradisional kelereng adalah perangkat *mobile* Android 2.3 Gingerbread.

4.1.2 Implementasi Antarmuka

Proses implementasi antar muka ini berfungsi untuk melihat hasil dari perancangan antar muka yang telah dirancang sebelumnya. Implementasi ini menampilkan tampilan menu utama, cara bermain, tentang, tampilan *level* 1 sampe 3.

1. Tampilan Menu Utama

Layar ini berfungsi menampilkan menu utama dimana terdapat lima pilihan yaitu main, cara bermain, tentang dan keluar. Tampilan menu utama dapat dilihat pada gambar 4.1.



Gambar 4.1 Tampilan Menu Utama

2. Tampilan Cara Bermain

Layar ini berfungsi menampilkan cara bermain game tradisional kelereng, dengan menekan tombol cara bermain pada tampilan menu utama, untuk kembali ke menu utama bisa menekan tombol kembali. Tampilan cara bermain dapat dilihat pada gambar 4.2.



Gambar 4.2 Tampilan Cara Bermain

3. Tampilan Tentang

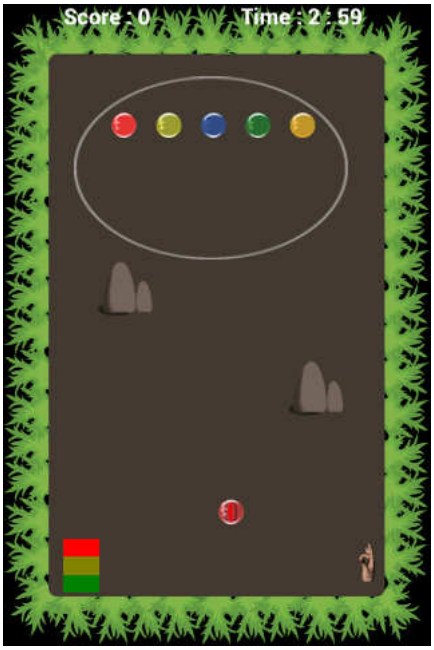
Layar ini berfungsi menampilkan tentang pembuat game tradisional kelereng, dengan menekan tombol tentang pada tampilan menu utama, untuk kembali ke menu utama bisa menekan tombol kembali. Tampilan tentang dapat dilihat pada gambar 4.3.



Gambar 4.3 Tampilan Tentang

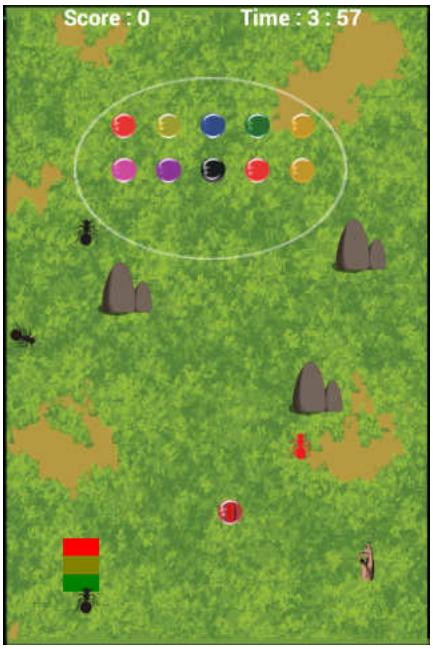
4. Tampilan *Level*

Layar ini berfungsi menampilkan *level* 1 sampai *level* 3 dari game tradisional kelereng. Tampilan tiap *level* dapat dilihat pada gambar 4.4, gambar 4.5 dan gambar 4.6.



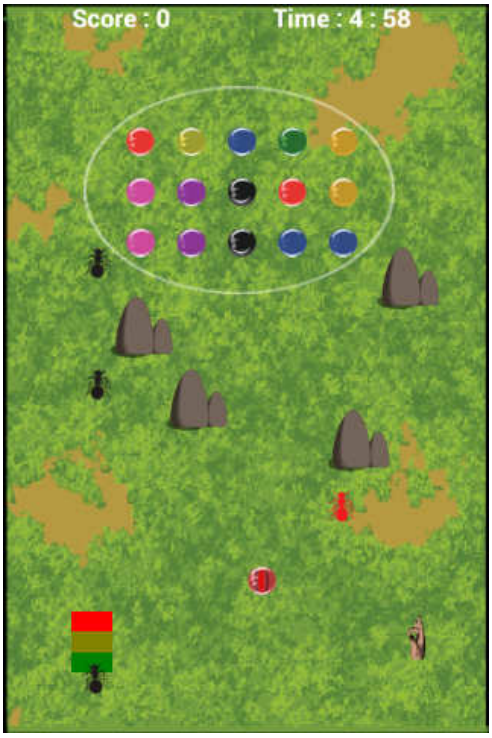
Gambar 4.4 Tampilan *Level* 1

Pemain menekan tombol main pada tampilan menu utama maka akan tampil permainan *level* 1. Tampilan permainan *level* 1 dapat dilihat pada gambar 4.4.



Gambar 4.5 Tampilan *Level* 2

Pemain dapat menyelesaikan permainan pada *level* 1 maka pemain bisa melanjutkan permainan ke *level* 2. Tampilan *level* 2 dapat dilihat pada gambar 4.5.



Gambar 4.6 Tampilan *Level* 3

Pemain dapat menyelesaikan permainan pada *level* 2 maka pemain bisa melanjutkan permainan ke *level* 3. Tampilan *level* 3 dapat dilihat pada gambar 4.6.

5. Tapilan Menang

Layar ini berfungsi sebagai pesan untuk pemain bahwa pemain telah berhasil menyelesaikan permainan dan terdapat tombol next untuk melanjutkan ke *level* berikutnya dan tombol kembali untuk kembali ke menu utama. Tampilan menang dapat dilihat pada gambar 4.6.



Gambar 4.7 Tampilan Menang

6. Tamplan Kalah

Layar ini berfungsi sebagai pesan untuk pemain bahwa pemain telah gagal menyelesaikan permainan dan terdapat tombol coba lagi untuk mengulang *level* tersebut dan tombol kembali untuk kembali ke menu utama. Tampilan kalah dapat dilihat di gambar 4.8.



Gambar 4.8 Tampilan Kalah

4.2 Pengujian Sistem

Pengujian sistem dilakukan bertujuan untuk menemukan kesalahan atau kekurangan pada perangkat lunak yang dibangun. Pengujian pada sistem yang dibangun dilakukan melalui dua tahap yaitu pengujian *Black Box* dan pengujian *betha.*

4.2.1 Rencana Pengujian

Rencana pengujian yang akan dilakukan pada aplikasi ini selengkapnya terlihat pada tabel 4.1.

Tabel 4. 1 Rencana Pengujian

No	Komponen yang Diuji	Detail Pengujian	Jenis Uji
1	Menu	Memilih tombol Main	*Black Box*
		Memilih tombol Cara Bermain	*Black Box*
		Memilih tombol Tentang	*Black Box*
2	Button	Menekan tombol Keluar	*Black Box*
		Menekan tombol Kembali	*Black Box*
		Menekan tombol Keluar	*Black Box*
		Menekan tombol Next	*Black Box*
		Menekan Tombol Coba Lagi	*Black Box*
3	Permainan	Touch Drag Objek Kelereng Utama	*Black Box*
		Touch Objek Tangan	*Black Box*
4	Algoritma	Greedy *Best Firt Search*	*White Box*

1. Pengujian *Black Box*

Berdasarkan hasil pengujian dan yang telah disusun, maka dapat dilakukan pengujian sebagai berikut. Pengujian yang dilakukan meliputi pengujian tampilan pembua, dapat dilihat pada table 4.2.

Tabel 4. 2 Kasus dan Hasil Uji

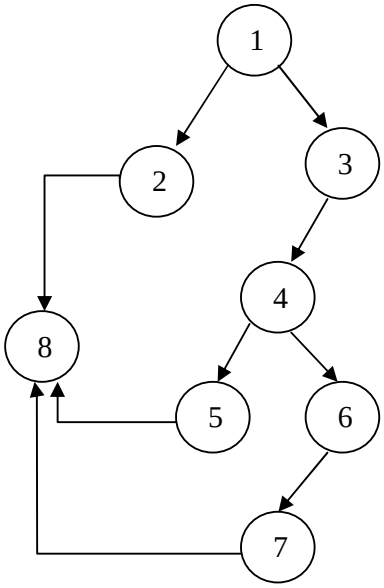
Kasus dan hasil uji (data normal)				
No	Kasus/diuji	Skenario uji	Hasil yang diharapkan	Hasil pengujian
1	Menu	Memilih tombol Main	Menampikan tampilan *Level* 1	[√] Berhasil [ ] Tidak Berhasil
		Memilih tombol Cara Bermain	Menampilkan tampilan Cara Bermain	[√] Berhasil [ ] Tidak Berhasil
		Memilih tombol Tentang	Menampilkan tampilan Tentang	[√] Berhasil [ ] Tidak Berhasil
2	Button	Menekan tombol Keluar	Keluar Game	[√] Berhasil [ ] Tidak Berhasil
		Mnekan tombol Kembali	Kembali Ke Menu Utama	[√] Berhasil [ ] Tidak Berhasil
		Menekan tombol Next	Melanjutkan ke *level* selanjutnya	[√] Berhasil [ ] Tidak Berhasil
		Menekan tombol Coba Lagi	Mengulang permainan	[√] Berhasil [ ] Tidak Berhasil
3	Permainan	Touch Drag objek Kelereng utama	Menampilkan garis arah	[√] Berhasil [ ] Tidak Berhasil
		Touch objek Tangan	Objek kelereng utama bergerak	[√] Berhasil [ ] Tidak Berhasil

1. Pengujian *White Box* Algoritma Greedy *Best First Search*

   Berikut ini merupakan algoritma Greedy *Best First Search* pada *level* 3 yang diterapkan pada objek semut merah yang menggunakan bahasa pemrograman java.

```
1.  private LinkedList<AnimatedSprite> bSemutM;
    bSemutM = new LinkedList<AnimatedSprite>();
2. if (A_node != null)
3. float ratioX = (xCoor / Math.abs(xCoor))
    * Math.min(1, Math.abs(xCoor / yCoor));
    float ratioY = (yCoor / Math.abs(yCoor))
    * Math.min(1, Math.abs(yCoor / xCoor));
4. for(int i = 0; i < A_node.getLength(); i++){
   mCurrentnode.to(A_node.getTileColumn(i)*tilewidth,
   A_node.getTileRow(i) * tileheight);
5. mMoveModifier=new  PathModifier(mSpeed  *  A_node.getLength(),
   mCurrentnode, new PathModifier.IPathModifierListener()
6. switch(A_node.getDirectionToNextStep(pWaypointIndex))
7. mWaypointIndex = pWaypointIndex;
8. isWalking = false;
   A_node = null;
   player.stopAnimation();}});
   player.registerEntityModifier(mMoveModifier);
```

Flow Graph Algoritma Greedy *Best First Search*



Gambar 4.9 *Flowgraph* Algoritma Greedy *Best First Search*

Keterangan :

= Menggambarkab kondisi

= Menggambarkan aksi

2. *Cyclomatic Complexity* V(G)

Cyclomatic complexity akan digunakan dalam menguji path yang nantinya akan digunakan untuk mencari jumlah path dalam satu dan juga sebagai penentuan jalur objek semut. Apabila menggunakan basis path, nilai yang dihitung untuk *Cyclomatic* complexity menentukan jalur indpenden dalam basis set suatu program dan memberi batas atas untuk jumlah uji coba yang harus dikerjakan bahwa seluruh perintah sekurang-kurangnya telah dikerjakan sekali. Berikut ini adalah perhitungan *Cyclomatic* complexity yang nantinya akan digunakan sebagai jumlah aksi pada independent path.

V(G) = E – N + 2

Di mana E adalah edge, dan N adalah node. Jadi :

V(G) = 9 edge – 8 node + 2 = 3

Keterangan :

E = Jumlah aksi

N = Jumlah Kondisi.

3. *Independent Path*

Dari hasil perhitungan *cyclomatic complexity* terdapat 3 *independent path*, yaitu :

Path 1 = 1 - 2 - 8

Path 2 = 1 - 3 - 4 – 5 - 8

Path 3 = 1 - 3 - 4 – 6 – 7 – 8.

Tabel 4. 3 Graph Matriks Jalur Semut (Greedy *Best First Search*)

N	1	2	3	4	5	6	7	N(E)-1
1		1	1					2-1=1
2							1	1-1= 0
3				1				1-1=0
4					1	1		2-1=1
5							1	1-1= 0
6							1	1-1= 0
7								0
Sum(E)+1								2+1=3

Keterangan:

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

Tabel 4. 4 Hasil Pengujian *Whitebox*

No	Kasus uji	Yang diharapkan	Hasil Sesuai Kasus Uji	Keterangan
1	- `float ratioX = (xCoor / Math.abs(xCoor))* Math.min(1, Math.abs(xCoor / yCoor));float ratioY = (yCoor / Math.abs(yCoor))* Math.min(1, Math.abs(yCoor / xCoor));.getTileHeight();`	Mencari F dengan nilai H	Mencari F dengan nilai H	[√] Alur terlewati [ ] Alur tidak terlewati
2	- `for(int i = 0; i < A_node.getLength(); i++){` `mCurrentnode.to(A_node.getTileColumn(i)* tilewidth, A_node.getTileRow(i) * tileheight);`	Masukkan semua *neighbor node* ke dalam open dan hitung nilai F dan H.	*Current node* dan nilai F,H masuk ke dalam open	[√] Alur terlewati [ ] Alur tidak terlewati

3	- `isWalking = false;`
	`A_node = null;`
	`player.stopAnimation();}});`
	`player.registerEntityModifier(mMoveModifier);`
		Nilai H yang didapat lebih kecil dari nilai H sebelumnya. Update ulang nilai F,H	Nilai F,H, diupdate ulang	[√] Alur terlewati [ ] Alur tidak terlewati

Berdasarkan pengujian yang dilakukan, maka bisa diambil kesimpulan bahwa masing-masing *independent path* yang dihasilkan bisa dijalankan dan dieksekusi serta tidak ada perulangan yang tak berhingga.

4.2.2 Pengujian Beta

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 30 responden, dari hasil kuesioner tersebut akan dilakukan perhitungan agar dapat diambil kesimpulan terhadap penilaian penerapan sistem yang telah dibangun. Kuesioner terdiri dari 6 pernyataan dengan menggunakan Sistem penskoran dengan skala pengukuran yang digunakan adalah skala likert, Berikut skor untuk jawaban kuesioner dapat dilihat pada tabel 4.5.

Tabel 4. 5 Skor Jawaban Pada Kuesioner

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

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

$$P = \frac{S}{skorideal} x100\%$$

Keterangan:

P = Nilai presentase yang dicari

S = Jumlah frekuensi dikalikan dengan nilai yang di tetapkan jawaban

Skor ideal = Nilai tertinggi dikalikan dengan jumlah sampel.

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

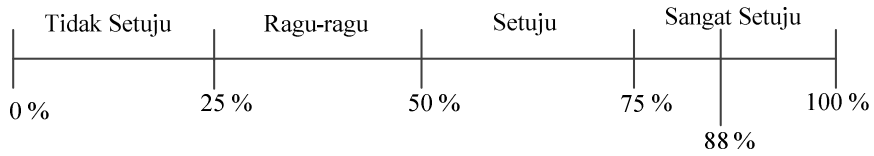
1. Apakah anda setuju game Tradisional Kelereng ini dapat membantu anda menjadi lebih fokus memprediksi bidikan?

Tabel 4. 6 Perhitungan persentase pertanyaan ke-1

Pertanyaan	Keterangan	Skor	Frekuensi	S
1.	Sangat setuju	4	16	64
	Setuju	3	14	42
	Ragu-ragu	2	0	0
	Tidak setuju	1	0	0
Jumlah			30	106

$$\frac{106}{120} x100 = 88\%$$

Berdasarkan hasil presentase nilai pada tabel 4.6, maka dapat disimpulkan bahwa penilaian terhadap pertanyaan tentang setuju tidakkah unsur dasar permainan kelereng dijadikan sebuah game adalah 80% dari yang diharapkan 100%, atau dapat dikategorikan sebagai sangat setuju. Berikut kriteria interprestasi skor :



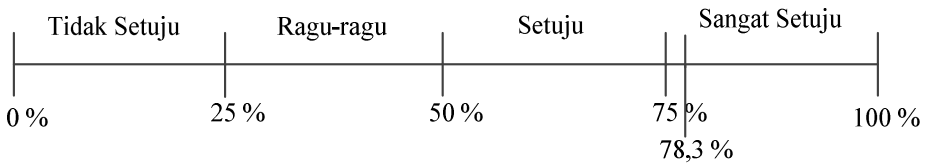
2. Apakah dengan bermain game Tradisional Kelereng ini anda dapat mengenal permainan kelereng?

Tabel 4. 7 Perhitungan persentase pertanyaan ke-2

Pertanyaan	Keterangan	Skor	Frekuensi	S
2.	Sangat setuju	4	18	72
	Setuju	3	12	36
	Ragu-ragu	2	0	0
	Tidak setuju	1	0	0
Jumlah			30	108

$$P = \frac{108}{120} x100 = 90\%$$

Berdasarkan hasil presentase nilai pada tabel 4.7, maka dapat disimpulkan bahwa game Tradisional Kelereng ini masyarakat akan lebih mengenal permainan tradisional kelereng adalah 78,3% dari yang diharapkan 100%, atau dapat dikategorikan sebagai sangat setuju. Berikut kriteria interprestasi skor :



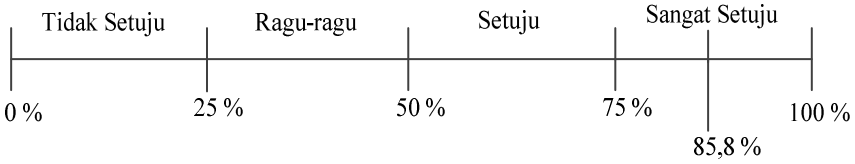
3.Apakah anda setuju game Tradisional Kelereng ini mudah untuk dimainkan?

Tabel 4. 8 Perhitungan persentase pertanyaan ke-3

Pertanyaan	Keterangan	Skor	Frekuensi	S
3.	Sangat setuju	4	14	56
	Setuju	3	15	45
	Ragu-ragu	2	1	2
	Tidak setuju	1	0	0
Jumlah			30	103

$$P = \frac{103}{120} x100 = 85{,}8\%$$

Berdasarkan hasil presentase nilai pada tabel 4.8, maka dapat disimpulkan bahwa penilaian terhadap sesuai tidaknya game Tradisional Kelereng ini dengan unsur dasar peraturan permainan kelereng adalah 85,8% dari yang diharapkan 100%, atau dapat dikategorikan setuju. Berikut kriteria interprestasi skor :



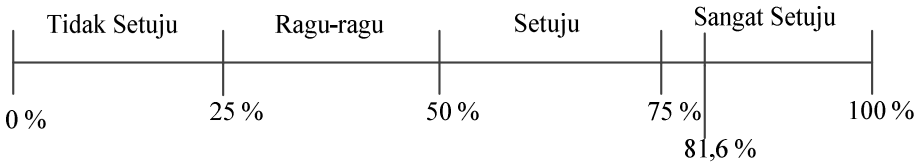
4. Apakah menurut anda tampilan antarmuka game Tradisional Kelereng ini menari.

Tabel 4. 9 Perhitungan persentase pertanyaan ke-5

Pertanyaan	Keterangan	Skor	Frekuensi	S
4.	Sangat setuju	4	13	52
	Setuju	3	12	36
	Ragu-ragu	2	5	10
	Tidak setuju	1	0	0
Jumlah			30	98

$$P = \frac{98}{120} x100 = 81{,}6\%$$

Berdasarkan hasil presentase nilai pada tabel 4.9 , maka dapat disimpulkan bahwa penilaian terhadap mudah tidaknya permainan game Tradisional Kelereng ini adalah 81,6% dari yang diharapkan 100%, atau dapat dikategorikan sebagai sangat setuju. Berikut kriteria interprestasi skor :



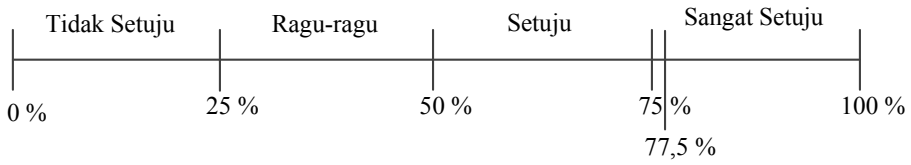
5. Dengan adanya hambatan berupa semut ini, apakah tantangan yang diberikan pada game Tradisional Kelereng ini cukup menantang?

Tabel 4. 10 Perhitungan persentase pertanyaan ke-6

Pertanyaan	Keterangan	Skor	Frekuensi	S
5.	Sangat setuju	4	10	40
	Setuju	3	13	39
	Ragu-ragu	2	7	14
	Tidak setuju	1	0	0
Jumlah			30	93

$$P = \frac{93}{120} x100 = 77{,}5\%$$

Berdasarkan hasil presentase nilai pada tabel 4.10, maka dapat disimpulkan bahwa penilaian terhadap setujukah game tradisional kelereng ini dibangun pada perangkat *mobile* adalah 77,5% dari yang diharapkan 100%, atau dapat dikategorikan sangat setuju. Berikut kriteria interprestasi skor :



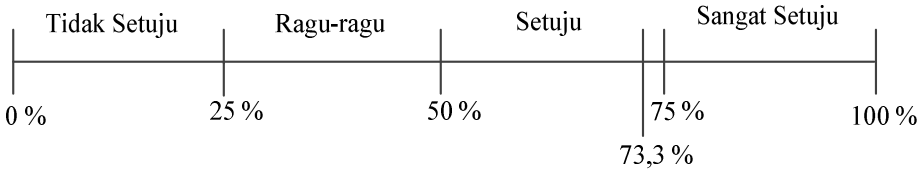
6. Apakah konsep kelereng dan semut pada game Tradisional Kelereng ini baik ?

Tabel 4. 11 Perhitungan persentase pertanyaan ke-7

Pertanyaan	Keterangan	Skor	Frekuensi	S
6.	Sangat setuju	4	8	32
	Setuju	3	12	36
	Ragu-ragu	2	10	20
	Tidak setuju	1	0	0
Jumlah			30	88

$$P = \frac{88}{120} x100 = 73{,}3\%$$

Berdasarkan hasil presentase nilai pada tabel 4.11 maka dapat disimpulkan bahwa penilaian terhadap terhiburkah anda setelah memainkan game Tradisional Kelereng ini adalah 73,3% dari yang diharapkan 100%, atau dapat dikategorikan sebagai setuju. Berikut kriteria interprestasi skor :



Hasil dari pengujian beta dapat disimpulkan bahwa :

1. Pemain dapat lebih fokus memprediksi bidikan setelah memainkan game ini
2. Pemain dapat mengetahui permainan kelereng setelah memainkan game ini.
3. Pemain merasa sangat mudah dalam memainkan *game* ini.
4. Pemain beranggapan antarmuka *game* ini sudah menarik.
5. Pemain merasa tertantang dengan adanya hambatan semut yang dinamis
6. Pemain merasa terbantu dalam mengenal permainan tradisional kelereng.