BAB IV

IMPLEMENTASI DAN PENGUJIAN

4.1 Implementasi

Implementasi ini merupakan terjemahan perancangan yang berdasarkan hasil analisis pada bab sebelumnya ke dalam bahasa pemprograman yang dapat dimengerti oleh komputer. Bahasa pemprograman dalam sistem ini menggunakan bahasa *java*, sedangkan untuk desain 3D digunakan aplikasi blender.

4.1.1 Perangkat Lunak Pembangunan

Perangkat lunak pembangunan menjelaskan perangkat lunak yang digunakan untuk membangun aplikasi *game rhythm mobile* menggunakan *augmented reality*. Adapun perangkat lunak yang digunakan untuk membangun aplikasi *game rhythm mobile* menggunakan *augmented reality* adalah sebagai berikut :

1. Windows 7 Profesional sebagai sistem operasi.
2. Eclipse SDK versi 3.7.2 sebagai perangkat lunak pembangunan
3. Sistem operasi android versi 2.3.3 (Gingerbread)
4. Stepmania versi 3.9b untuk membuat *beat* pada *game rhythm*.
5. Blender versi 2.63 untuk pembuatan objek 3D

4.1.2 Perangkat Keras Pembangunan

Perangkat Keras pembangunan menjelaskan perangkat keras yang digunakan untuk membangun aplikasi *game rhythm mobile* menggunakan *augmented reality*. Adapun perangkat keras yang digunakan untuk membangun *game rhythm mobile* menggunakan *augmented reality* adalah sebagai berikut :

1. Processor Intel Core 2 Duo
2. RAM 2 GB
3. Harddisk 250 GB
4. Monitor 14”
5. Keyboard dan Mouse

4.1.3 Implementasi Antarmuka

Implementasi antarmuka berisi pemaparan setiap tampilan aplikasi *game rhythm mobile* menggunakan *augmented reality* yang dibangun beserta *file xml* sebagai *layout* dari masing – masing antarmuka. Adapun gambar dari tampilan implementasi antarmuka aplikasi *game rhythm mobile* menggunakan *augmented reality* tertera pada halaman lampiran A implementasi antarmuka.

Tabel 4-1 Implementasi antarmuka

No	Halaman	Deskripsi	*File*
1	Splash	Halaman yang berisi tata letak untuk menu utama	Splash.xml
2	Instruction layout	Halaman yang berisi informasi dan tata letak untuk tampilan petunjuk	instruction_layout.xml
3	GUI*Game*	Berisi fungsi yang akan menampilkan halaman *game*	GUI*Game*.java

4.2 Pengujian Perangkat Lunak

Pengujian sistem merupakan hal terpenting yang dilakukan untuk menemukan 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. Adapun metode pengujian yang digunakan pada perangkat lunak ini adalah metode pengujian *black box dan* white box. Pengujian *black box* berfokus pada persyaratan fungsional perangkat lunak, sedangkan *white box* pengujian dengan melihat ke dalam modul untuk meneliti kode-kode program yang ada, dan menganalisis apakah ada eksalahan atau tidak. Metode pengujian *black box* sendiri terdiri dari dua tahapan pengujian yaitu tahap pengujian alpha dan tahap pengujian beta.

4.2.1 Pengujian *White Box*

Pengujian *white box* merupakan metode perancangan *test case* yang menggunakan struktur control dari perancangan procedural dalam mendapatkan *test case.* Adapun metode yang digunakan dalam pengujian *white box* ini adalah metode *basis path.* Metode *basis path* mengijinkan pendesain kasus uji untuk mendapatkan perkiraan lojik yang kompleks dari desain procedural dan menggunakan perkiraan ini untuk mendefenisikan aliran eksekusi.

a. Pengujian *white box tracking marker*

Berikut kode *tracking marker* :

```
1.  for (int cosId = 1; cosId <= mMobileSDK.getNumberOfDefinedCoordinateSystems();
    cosId++) {
2.                                          float[] trackingPose = new float[16];
3.                                          final Pose trackingValues =
    mMobileSDK.getTrackingValues(cosId);
4.                                          if (trackingValues.getQuality() > 0) {
5.                                                  if (mIsCosTracking.get(cosId) != true) {
6.                                                          mIsCosTracking.put(cosId, true);
7.                                          mTrackingCallback.onPatternDetected(cosId);}
8.                                          } else if (mIsCosTracking.get(cosId) != false) {
9.                                                  mIsCosTracking.put(cosId, false);
10.                                                 mTrackingCallback.onTrackingLost(cosId);}
11. }
```

Berdasarkan kode *tracking marker* maka diagram alirnya adalah sebagai berikut :

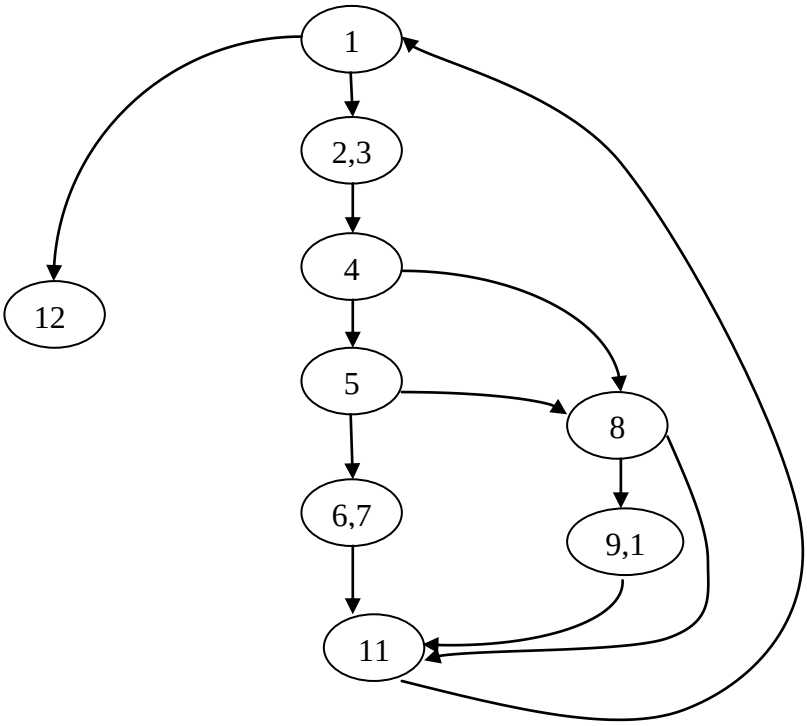


Diagram 4-1 Diagram alir *tracking beat*

Dari diagram alir *tracking marker*, dapat dihitung *cyclomatic complexity*, yakni :

V(G) = Jumlah Busur – Jumlah simpul + 2

V(G) = 12 – 9 + 2 = 5

V(G) = Predicate node + 1

V(G) = 4 + 1 = 5

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

Path 1 = 1-2-3-4-5-6-7-11-1-13

Path 2 = 1-2-3-4-8-9-10-11-1-13

Path 3 = 1-2-3-4-5-8-9-10-11-1-13

Path 4 = 1-2-3-4-8-11-1-13

Path 5 = 1-13

b. Pengujian *white box parser beat*

Berikut kode *parser beat*

```
1.  for (int i = 0; i < line.length(); i++) {
2.      if (isSupportedNoteType(nt)) {
3.          if (!noteAdded)
4.              addNote = true;
5.          if (!jumps && !activeHolds.isEmpty())
6.              addNote = false;
7.          if (nt.equals(NoteType.HOLD_END) && activeHolds.contains(i))
8.              addNote = true;
9.          if (addNote) {
10.             int pitch;
11.             int fraction;
12.             int noteTime = (int)(time + timeIncrease - offset);
13.             float[] coords;
14.             DataNote n = new
        DataNote(nt,fraction,pitch,noteTime,beat,coords);
15.             nd.addNote(n);
16.             osu_num++;
17.             noteAdded = true;
18.             if (osu) {
19.                 coords = rand.nextCoords(lineIndex, lineCount);
20.                 pitch = osu_num;
21.                 fraction = osu_fraction;
22.             } else {
23.                 coords = new float[4];
24.                 pitch = i;
25.                 fraction = parseFraction(lineIndex, lineCount);
26.                 if (nt.equals(NoteType.HOLD_START)) {
27.                     activeHolds.add(i);
```

```
28.                                                     } else if (nt.equals(NoteType.HOLD_END)
                    && activeHolds.contains(i)) {
29.
          activeHolds.remove(activeHolds.indexOf(i));
30.                                             } else if (randomize) {
31.                                                     pitch = rand.nextPitch(jumps);
                                                }
32.                                     }
                                }
                        }
33.     }
```

Berdasarkan kode *parser beat* maka diagram alirnya adalah sebagai berikut :

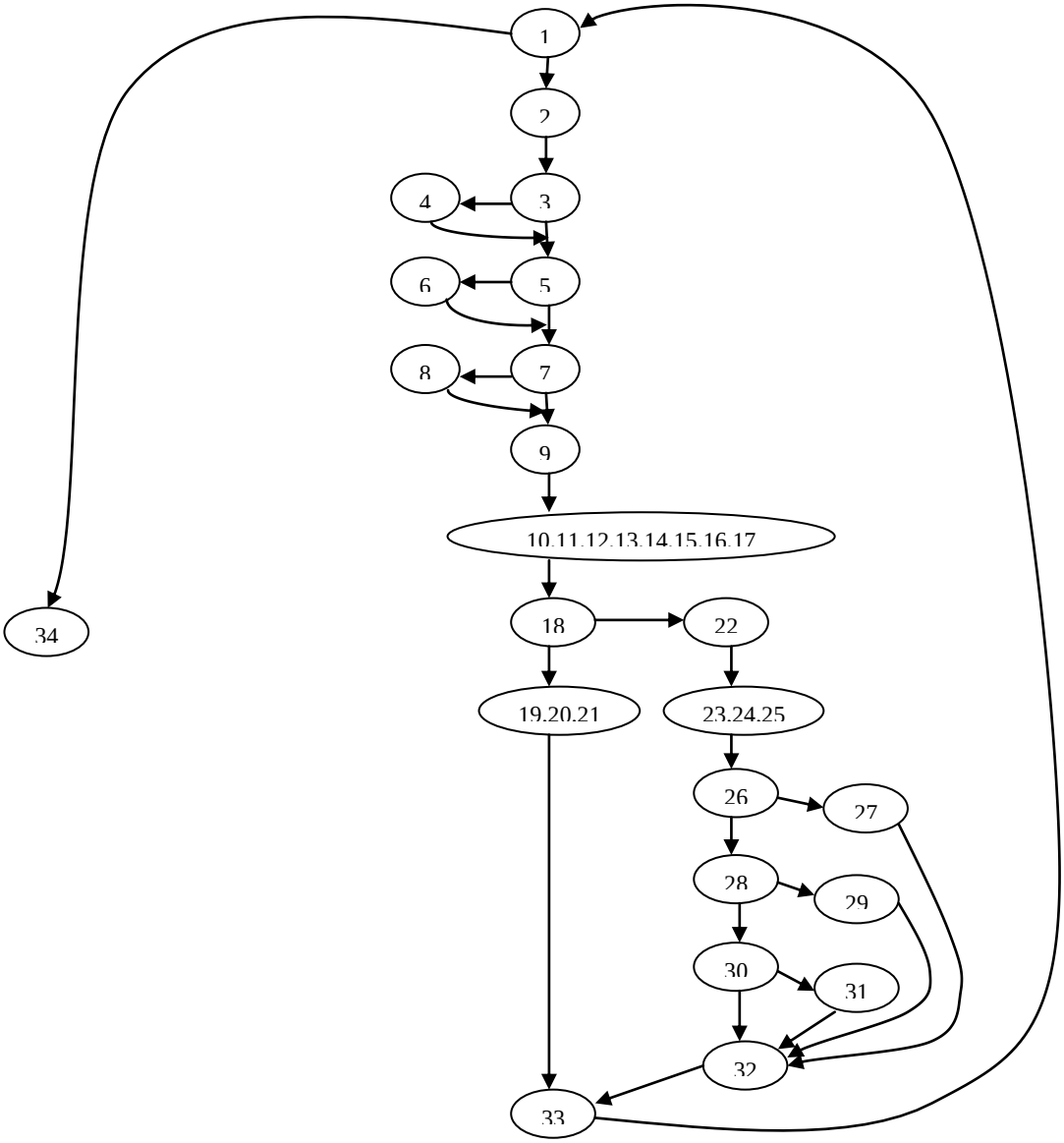


Diagram 4-2 Diagram alir *parser beat*

Dari diagram alir *parser beat*, dapat dihitung *cyclomatic complexity*, yakni :

V(G) = Jumlah Busur – Jumlah simpul + 2

V(G) = 31 – 23 + 2 = 9

V(G) = Predicate node + 1

V(G) = 8 + 1 = 9

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

Path 1 = 1-34

Path 2 = 1-2-3-4-5-7-9-10-11-12-13-14-15-16-17-18-19-20-21-33-1-34

Path 3 = 1-2-3-5-6-7-9-10-11-12-13-14-15-16-17-18-19-20-21-33-1-34

Path 4 = 1-2-3-4-7-8-9-10-11-12-13-14-15-16-17-18-19-20-21-33-1-34

Path 5 = 1-2-3-5-7-9-10-11-12-13-14-15-16-17-18-19-20-21-33-1-34

Path 6 = 1-2-3-5-7-9-10-11-12-13-14-15-16-17-18-22-23-24-25-26-28-30-32-33-1-34

Path 7 = 1-2-3-5-7-9-10-11-12-13-14-15-16-17-18-22-23-24-25-26-27-32-33-1-34

Path 8 = 1-2-3-5-7-9-10-11-12-13-14-15-16-17-18-22-23-24-25-26-28-29-32-33-1-34

Path 9 = 1-2-3-5-7-9-10-11-12-13-14-15-16-17-18-22-23-24-25-26-28-30-31-32-33-1-34

4.2.2 Pengujian *Black box*

4.2.2.1 Pengujian *Alpha*

Pengujian fungsional yang digunakan menguji sistem. Pengujian berfokus pada persyaratan fungsional perangkat lunak. Pihak pembangun mendampingi serta mencatat kesalahan – kesalahan maupun permasalahan yang dirasakan oleh pengguna. Pengujian alpha yang dilakukan pada aplikasi *game rhythm mobile* menggunakan *augmented reality* dibagi tiga bagian, yaitu rencana pengujian, kasus dan hasil pengujian, dan kesimpulan pengujian.

4.2.2.2 Rencana Pengujian

Rencana pengujian yang akan dilakuakn pada aplikasi ini selengkapnya terlihat pada tabel 4-1 skenario pengujian.

Tabel 4-2 Rencana Pengujian

Kelas Uji	Detail Pengujian	Jenis pengujian
Mulai	Validasi *augmented reality* (*box* 3D)	*Black box*
	Permainan *game* dan skor	*Black box*
Pengaturan	Validasi field pengaturan	*Black box*
	Mengatur tingkat kesulitan dan kecepatan	*Black box*
Petunjuk	Menampilkan petunjuk *game*	*Black box*
Keluar	Menampilkan notifikasi	*Black box*

4.2.2.3 Kasus dan Hasil Pengujian Alpha

Kasus dan hasil pengujian berisi pemaparan dari rencana pengujian yang telah disusun pada scenario pengujian. Pengujian ini dilakukan secara *black box* dengan hanya memperhatikan masukan ke dalam sistem dan keluaran dari masukan tersebut.

1. Pengujian Mulai

Pengujian ini dilakukan untuk memulai aplikasi *game* yang telah dibuat. Sebelum memainkan *game*, pengguna harus memilih satu *box* 3D yang berisi lagu secara *live view* dari kamera terhadap marker (cover CD/DVD music). Untuk Pengujian target terhadap marker dapat dilihat pada gambar 4-1.



Gambar 4-1 Posisi target dari sudut pandang

Sedangkan untuk pengujian jarak antara kamera dan marker, pantulan cahaya dan sudut kemiringan kamera dapat dilihat pada tabel 4-2.

Tabel 4-3 Pengujian jarak kamera, sudut kemiringan kamera dan pencahayaan

Hasil Pengujian hubungan jarak antara kamera dan marker, pantulan cahaya dan sudut kemiringan kamera			
Jarak (cm)	Sudut kemiringan Kamera	Pencahayaan (cahaya lampu, sinar matahari, berawan, gelap)	Tingkat keberhasilan
4	0°	Cahaya lampu, gelap	Tidak Berhasil
4	0°	Sinar matahari, berawan	Tidak Berhasil
8	0°	Cahaya lampu, gelap	Tidak Berhasil
8	0°	Sinar matahari, berawan	Berhasil

12	0°	Cahaya lampu, gelap	Tidak Berhasil
12	0°	Sinar matahari, berawan	Berhasil
27	0°	Cahaya lampu, gelap	Tidak Berhasil
27	0°	Sinar matahari, berawan	Berhasil
50	0°	Cahaya lampu, gelap	Tidak Berhasil
50	0°	Sinar matahari, berawan	Berhasil
4	45°	Cahaya lampu, gelap	Tidak Berhasil
4	45°	Sinar matahari, berawan	Tidak Berhasil
8	45°	Cahaya lampu, gelap	Tidak Berhasil
8	45°	Sinar matahari, berawan	Berhasil
12	45°	Cahaya lampu, gelap	Tidak Berhasil
12	45°	Sinar matahari, berawan	Berhasil
27	45°	Cahaya lampu, gelap	Tidak Berhasil
27	45°	Sinar matahari, berawan	Tidak Berhasil
50	45°	Cahaya lampu, gelap	Tidak Berhasil
50	45°	Sinar matahari, berawan	Berhasil
4	70°	Cahaya lampu, gelap	Tidak Berhasil
4	70°	Sinar matahari, berawan	Tidak Berhasil
8	70°	Cahaya lampu, gelap	Tidak Berhasil
8	70°	Sinar matahari, berawan	Berhasil
12	70°	Cahaya lampu, gelap	Tidak Berhasil
12	70°	Sinar matahari, berawan	Berhasil
27	70°	Cahaya lampu, gelap	Tidak Berhasil
27	70°	Sinar matahari, berawan	Berhasil
50	70°	Cahaya lampu, gelap	Tidak Berhasil
50	70°	Sinar matahari, berawan	Berhasil

Hasil pengujian terhadap marker dan kamera agar augmented reality (*box* 3D) muncul dapat dilihat pada tabel 4-2. Pengujian dilakukan dengan jarak 4 cm, 8 cm, 12 cm, 27 cm , 50 cm dan dipengaruhi pantulan cahaya dan kemiringan kamera. Jarak minimum marker terhadap kamera 8 cm dan jarak maksimum marker terhadap kamera yaitu 50 cm dengan sudut minimum 0° dan sudut maksimum 70°. Jarak minimum dan maksimum serta sudut kemiringan kamera minimum – maksimum dipengaruhi dengan cahaya, dimana dalam pengujian apabila ada pantulan cahaya maka *augmented reality* (*box* 3D) tidak akan muncul.

Setelah memilih satu lagu dari *augmented reality* (*box* 3D) maka pengguna dapat memulai *game* dan diakhir permainan akan ditampilkan skor.

Table 4-1 Kasus dan hasil uji menu mulai

Kasus dan hasil uji (data normal)			
Data masukan	Yang diharapkan	Pengamatan	Kesimpulan
Klik menu "Mulai"	Ketika meng-klik mulai menampilkan *augmented reality* (*box* 3D) yang berisi lagu, kemudian permainan *game rhythm* dimulai sesuai dengan *beat* yang keluar saat lagu dimulai dan saat *game* selesai dimainkan maka akan menampilkan skor.	*Augmented reality* (*box* 3D) muncul, permainan *game* dapat dilakukan dan skor ditampilkan.	Diterima

2. Pengujian Pengaturan

Pengujian ini dilakukan untuk mengatur tingkat kesulitan dan kecepatan pada *game*. Saat melakukan perubahan pada tingkat kesulitan, *game* akan memuat *beat* sesuai tingkat kesulitan yang dipilih dengan kategori tingkat kesulitan didalam *file* .sm dari lagu tersebut. Tingkat kesulitan terdiri dari *beginner, easy, medium, hard, challenge*. Untuk tingkat kecepatan pada *game* akan memuat *beat* sesuai dengan tingkat kecepatan yang dipilih. Perubahan yang terjadi ketika tingkat kecepatan diubah, maka *game* akan mengalikan kecepatan normal *beat* (100BPM) dengan tingkat kecepatan yang dipilih. Sebagai contoh jika memilih tingkat kecepatan 5.00x maka waktu turun atau kemunculan *beat* akan menjadi lebih cepat sebanyak 5 kali.

Table 4-2 Kasus dan hasil uji menu pengaturan

Kasus dan hasil uji (data normal)			
Data masukan	Yang diharapkan	Pengamatan	Kesimpulan
Klik menu "Pengaturan"	Dapat mengatur tingkat kesulitan dan kecepatan *beat* pada *game*	Permainan dimulai sesuai dengan pengaturan yang telah dipilih	Diterima

3. Pengujian Petunjuk

Pengujian ini dilakukan untuk menampilkan informasi dan petunjuk yang ada di dalam permainan.

Table 4-3 Kasus dan hasil uji menu petunjuk

Kasus dan hasil uji (data normal)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Klik menu "Petunjuk"	Melihat informasi dan cara untuk memainkan *game rhythm*	Tampil petunjuk permainan	Ditampilkan

4. Pengujian keluar

Pengujian ini dilakukan untuk menampilkan notifikasi apakah keluar dari *game* atau tidak.

Table 4-4 Kasus dan hasil uji menu keluar

Kasus dan hasil uji (data normal)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Klik menu "Keluar"	Menampilkan informasi notifikasi apakah ingin keluar dari *game* atau tidak	Menampilkan notifikasi konfirmasi	Diterima

4.2.2.4 Kesimpulan Pengujian Alpha

Berdasarkan hasil pengujian aplikasi yang dilakukan maka dapat disimpulkan bahwa aplikasi *game rhythm mobile* menggunakan *augmented reality* yang dibangun sudah berjalan sesuai dengan yang diharapkan.

4.2.2.5 Pengujian Beta

Pengujian beta dilakukan di lingkungan pengguna tanpa kehadiran pihak pembangun aplikasi. Pengujian ini merupakan pengujian yang bersifat langsung di lingkungan yang sebenarnya. Pengguna melakukan penilaian terhadap aplikasi dengan menggunakan media kuesioner. Dari hasil kuesioner tersebut maka dapat ditarik kesimpulan apakah aplikasi yang dibangun telah sesuai dengan tujuan atau tidak.

Pengujian beta dilakukan terhadap pengguna dengan menggunakan kuesioner. Untuk mengetahui tanggapan dan penilaian pengguna terhadap aplikasi ini, telah disebarkan kuesioner kepada 20 responden. Kuesioner ini disebarkan menggunakan teknik sampling yaitu *Simple Random Sampling* yang disebarkan kepada banyak pengguna. Dari hasil kuesioner tersebut akan dilakukan perhitungan agar dapat diambil kesimpulan terhadap penilaian penerapan aplikasi yang dibangun. Berikut adalah pertanyaan dan hasil kuesioner yang telah dibagikan dengan menggunakan rumus :

$$Y = \frac{P}{Q} \times 100\ \%$$

Keterangan : P = Banyaknya jawaban responden tiap soal

Q = Jumlah responden

Y = Nilai presentase

Berikut data yang didapat berdasarkan kuesioner pada bagian pertama :

1. 95% (19 responden) berumur antara 19 – 31 tahun
2. 50% (10 responden) adalah laki – laki
3. 50% (10 responden) adalah wanita.

Berikut ini adalah hasil presentasi masing – masing nilai jawaban kuesioner yang diujikan kepada 20 responden dan telah dihitung dengan rumus.

1. Apakah petunjuk pada *game rhythm* android dengan *augmented reality* sudah membantu dalam memainkan *game*?

Tabel 4-4 Jawaban kuesioner untuk pertanyaan no 15

Jawaban	Responden	Presentase (%)
Sangat membantu	7	25%
Membantu	11	55%
Ragu – ragu	4	20%
Total responden	**20**	**100%**

2. Apakah setelah memainkan *game rhythm* android dengan *augmented reality* anda lebih tertarik dengan *game* yang berjenis casual ?

Tabel 4-5 Jawaban kuesioner untuk pertanyaan no 16

Jawaban	Responden	Presentase (%)
Tidak tertarik	-	-
Ragu – ragu	3	15%
Tertarik	15	75%
Sangat tertarik	2	10%
Total responden	**20**	**100%**

3. Apakah anda puas dengan keseluruhan bagian aplikasi *game rhythm* android dengan *augmented reality* ini ?

Tabel 4-6 Jawaban kuesioner untuk pertanyaan no 17

Jawaban	Responden	Presentase (%)
Kurang puas	1	5%
Puas	12	60%
Sangat puas	7	35%
Total responden	**20**	**100%**

4. Apakah menurut anda teknologi yang dikembangkan penulis mudah di aplikasikan ?

Tabel 4-7 Jawaban kuesioner untuk pertanyaan no 18

Jawaban	Responden	Presentase (%)
Mudah	17	85%
Biasa saja	2	10%
Sulit	1	5%
Total responden	**20**	**100%**

5. Menurut anda apakah cara ini kedepannya dapat membantu untuk meningkatkan tingkat penjualan CD/DVD musik ?

Tabel 4-8 Jawaban kuesioner untuk pertanyaan no 19

Jawaban	Responden	Presentase (%)
Ya	18	90%
Ragu – ragu	1	5%
Tidak	1	5%
Total responden	**20**	**100%**

6. Apakah dengan adanya aplikasi *game rhythm* android dengan augmented reality, anda akan tertarik untuk membeli CD/DVD musik ?

Tabel 4-9 Jawaban kuesioner untuk pertanyaan no 20

Jawaban	Responden	Presentase (%)
Ya	13	65%
Ragu –ragu	5	25%
Tidak	2	10%
Total responden	**20**	**100%**

Dari hasil jawaban responden dapat disimpulkan bahwa aplikasi yang dibuat mudah digunakan karena adanya petunjuk pada *game rhythm* android

dengan *augmented reality* yang membantu korespondensi untuk memainkan *game rhythm* android dengan augmented reality. Dan diharapkan kedepannya *game rhythm* android dengan *augmented reality* menjadi solusi alternatif untuk meningkatkan penjualan CD/DVD musik.