BAB 4

IMPLEMENTASI DAN PENGUJIAN

4.1 Implementasi

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 *virtual mirror eyeglasses* yang dijalankan menggunakan *flash player.*

4.1.1 Implementasi Perangkat Keras

Perangkat keras yang dapat digunakan untuk menjalankan aplikasi ini membutuhkan grafis video yang cukup tinggi. Rekomendasi spesifikasi minimum perangkat keras untuk menjalankan aplikasi *virtual mirror eyeglasses* ini adalah sebagai berikut :

1. *Processor* dengan kecepatan minimal 1,8 GHz
2. *RAM* minimal 1 GB
3. VGA *Card* minimal 128 MB
4. Webcam minimal 1 Megapixel
5. Koneksi internet minimal 152 Kbps

4.1.2 Implementasi Perangkat Lunak

Adapun spesifikasi perangkat lunak yang dibutuhkan untuk menjalankan aplikasi *virtual mirror eyeglasses* ini adalah :

1. *Microsoft Windows* 7
2. Library *Beyond Reality Face*
3. Adobe Flash Professional CS5.5
4. Autodesk 3D Studio Max 2010
5. Adobe Flash Player minimal versi 10

4.1.3 Implementasi Aplikasi

Implementasi aplikasi ini merupakan sebuah langkah penggunaan aplikasi sehingga dapat berjalan sesuai dengan tahap perancangan.

1. Eksekusi file

   Untuk dapat menjalankan aplikasi ini dapat dengan mengeksekusi file *swf* atau dengan menggunakan web browser yang telah mendukung minimal flash player 10.

2. Memberikan akses kepada kamera

   Setelah aplikasi dijalankan pengguna, kamera melalui sistem meminta akses kepada pengguna agar dapat mengizinkan pemakaian kamera, karena aplikasi ini tidak akan berjalan tanpa kamera. Adapun method yang diperlukan dalam pemanggilan kamera dan aksesnya adalah sebagai berikut :

```
/** Init the video and camera handling. */
public function initVideoHandling() : void {
      _cameraManager = new CameraManager(this);
      _videoManager = new VideoManager();
```

```
/** Initializes the Camera. */
public function initCamera() : void {
      _camera = Camera.getCamera();
      if(_camera != null) {
            _timerHardwareCheck.stop();

      _camera.removeEventListener(StatusEvent.STATUS,
onStatus);

_camera.addEventListener(StatusEvent.STATUS, onStatus);
            if(_handler)
_handler.onCameraActive(_camera);
      } else {
_timerHardwareCheck.removeEventListener(TimerEvent.TIMER,
onCheckRecordMedia);
_timerHardwareCheck.addEventListener(TimerEvent.TIMER,
onCheckRecordMedia);
_timerHardwareCheck.start();
}
}
```

3. Memilih model kacamata

   Setelah pengguna memberikan akses kepada kamera, pengguna baru dapat berinteraksi dengan tampilan aplikasi. Pada aplikasi ini terdapat pemilihan model kacamata. Pada saat pengguna menekan pada

icon/tombol model kacamata yang diinginkan maka sistem akan menjalankan function onClickedLoadModel seperti berikut :

```
private function onClickedLoadModel(event : MouseEvent) :
void {
      switch(event.currentTarget){
            case    _layout._bt0:    _flare3D.model    =
_modelBlack; break;
            case    _layout._bt1:    _flare3D.model    =
_modelRed; break;
      }
}
```

Saat pengguna memilih model kacamata, sistem juga melakukan proses pendeteksian wajah pengguna yang sebelumnya wajah pengguna harus tepat berada di depan kamera agar pendeteksian oleh sistem dapat berjalan.

```
public function initBRF() : void {
      _brfManager = new BeyondRealityFaceManager(stage);
      _brfManager.addEventListener(Event.INIT,
onInitBRF);
      _leftEyePoint = new Point();
      _rightEyePoint = new Point();
}
```

Setelah inisialisasi library di atas selanjutnya library telah siap digunakan dan pendeteksian tersedia.

```
public function onReadyBRF(event : Event = null) : void {
      _faceShapeVertices=BRFUtils.getFaceShapeVertices(_b
rfManager.faceShape);
      _faceShapeTriangles                              =
BRFUtils.getFaceShapeTriangles();
```

```
      _faceDetectionROI=_brfManager.vars.faceDetectionVar
s.faceDetectionROI;
      _leftEyeDetectionROI=_brfManager.vars.faceDetection
Vars.leftEyeDetectionROI;
      _rightEyeDetectionROI=_brfManager.vars.faceDetectio
nVars.rightEyeDetectionROI;

      _brfReady = true;
}
```

Lalu dipanggil method onVideoUpdate agar dapat terus mendeteksi wajah.

```
public function onVideoUpdate() : void {
      if(_brfReady) {
            var start : int = getTimer();
            _brfManager.update();
            _stats.input = getTimer() - start;
            showResult();
      }
}
```

Seperti yang terdapat pada method di atas terjadi pemanggilan method showResult. Method ini menghasilkan daerah wajah yang terdeteksi sehingga model kacamata dapat menyesuaikan posisi dan ukurannya terhadap wajah yang terdeteksi.

```
public function showResult(showAll : Boolean = false) :
void {
      _draw.clear();
      if(showAll       ||       _brfManager.task       ==
BRFStatus.FACE_DETECTION) {
            drawROIs();
            drawLastDetectedFaces(0x66ff00);
```

```
            drawLastDetectedFace(0xff7900, 3.0, 0.9);

      }
      if(showAll       ||       _brfManager.task       ==
BRFStatus.FACE_ESTIMATION) {

BRFUtils.getFaceShapeVertices(_brfManager.faceShape);
            drawShape();
      }
}
```

4. Membuat screenshot dan menyimpannya

   Pengguna dapat mengambil screenshot dan menyimpan gambar hasil sreenshot tersebut. Dengan menggunakan method onClickedSnapshot dan onClickedDownload, sistem dapat mengambil screenshot dari video data yang diterima dari kamera dan pengguna dapat menyimpannya ke direktori pengguna. Berikut adalah method yang dipakai untuk dapat mengambil screenshot dengan menghentikan update video kamera.

```
private function onClickedSnapshot(event : MouseEvent) :
void {
      _videoManager.stop();
      _layout._btCam.visible = false;
      _layout._btWebcam.visible = true;
      _layout._btDownload.visible = true;
}
```

   Setelah update video kamera berhenti maka pengguna dapat mengambil screenshot dan menyimpannya. Berikut adalah method yang dipakai untuk menyimpan hasil screenshot.

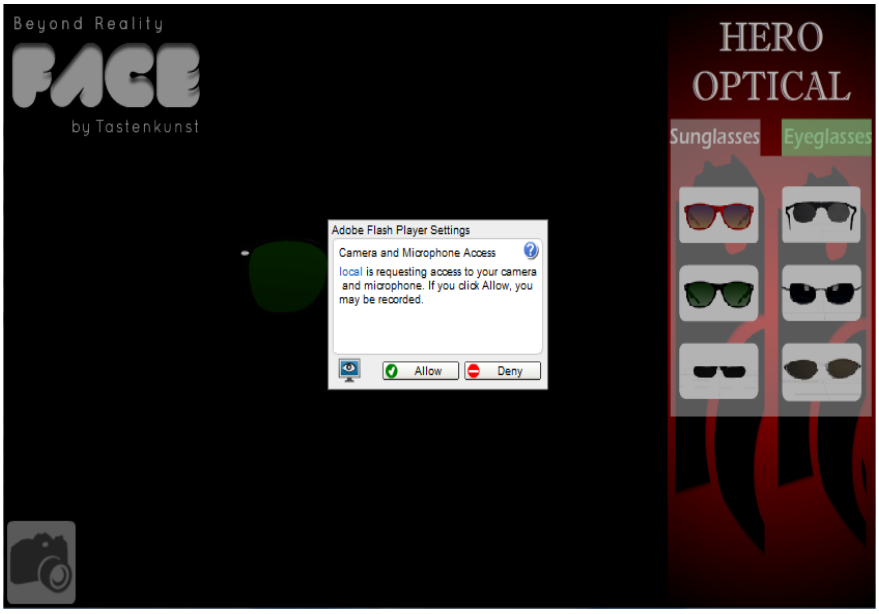
```
private function onClickedDownload(event : MouseEvent) :
void {
                    var    bmd    :    BitmapData    =    new
BitmapData(640, 480, false, 0xffffff);
                    _layout.visible = false;
                    _stats.visible = false;
                    bmd.draw(stage);
                    _layout.visible = true;
                    _stats.visible = true;
                    var     image     :     ByteArray     =
PNGEncoder.encode(bmd);
                    var    fr    :    FileReference    =    new
FileReference();
                    fr.save(image, "brf_snapshot.png");
            }
```

4.1.4 Implementasi Antarmuka

Dari perancangan antarmuka yang telah dibuat pada bab sebelumnya, maka tahap selanjutnya yaitu mengimplementasikannya menjadi sebuah tampilan. Berikut ini merupakan antarmuka yang ada pada aplikasi *virtual mirror eyeglasses* :

1. Antarmuka Streaming Kamera

   Saat pengguna membuka aplikasi pertama kali, maka antarmuka yang ditampilkan oleh sistem adalah antarmuka streaming kamera. Berikut ini merupakan tampilan dari antarmuka streaming kamera :



Gambar 4.1 Antarmuka Streaming Kamera

Gambar di atas merupakan tampilan awal dimana sistem meminta kepada pengguna memberikan akses pada kamera agar kamera dapat berjalan sehingga dapat mengambil gambar yang dibutuhkan untuk proses selanjutnya.

2. Antarmuka Pilih Model Kacamata

Setelah pengguna memberikan akses pada kamera maka pengguna dihadapkan pada antarmuka pilih model kacamata. Pada tampilan antarmuka ini terdapat pilihan model kacamata. Berikut ini merupakan tampilan dari antarmuka pilih model kacamata :



Gambar 4.2 Antarmuka Pilih Model Kacamata

Seperti yang terlihat pada gambar di atas, pengguna memilih model kacamata yang telah tersedia, kemudian kacamata yang dipilih muncul ditengah-tengah tampilan antarmuka.

3. Antarmuka Capture Image

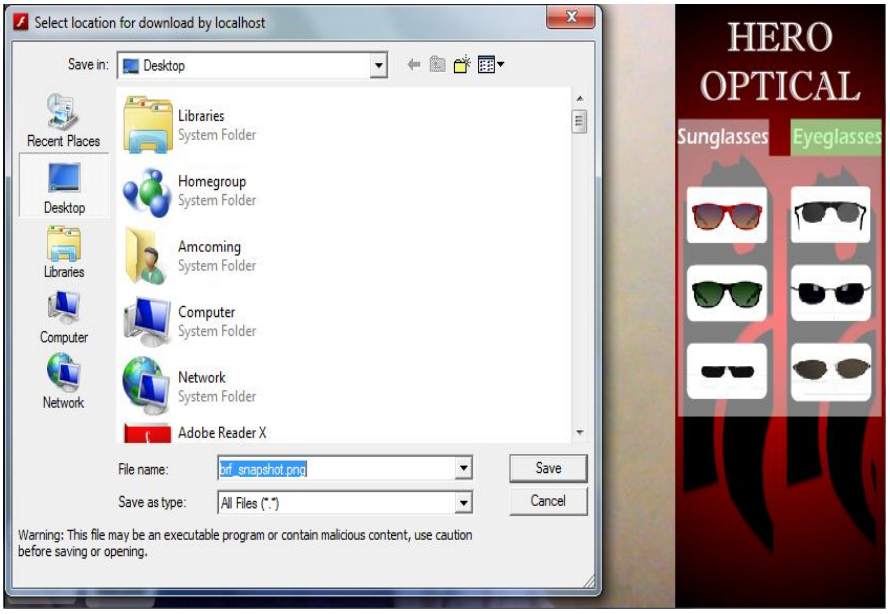
   Setelah sistem mendeteksi wajah pengguna dan model kacamata yang diinginkan sesuai dengan posisi wajah, pengguna dapat meng-*capture* hasil tangkapan kamera. Berikut ini merupakan tampilan dari antarmuka capture image :



Gambar 4.3 Antarmuka Capture Image

4. Antarmuka Simpan Snapshot

   Setelah pengguna menekan tombol capture image, sistem dapat menyimpan hasil capture image tersebut ke direktori pengguna dengan menekan satu tombol simpan snapshot. Berikut ini merupakan tampilan dari antarmuka simpan snapshot :



Gambar 4.4 Antarmuka Simpan Snapshot

4.2 Pengujian

Ada dua macam pengujian yang akan dilakukan, yaitu pengujian performansi dan *black box testing*. Pengujian performansi yaitu dengan melakukan percobaan-percobaan dalam kondisi-kondisi tertentu seperti pengaruh kecepatan gerakan wajah, posisi wajah pada saat pendeteksian, jarak wajah terhadap webcam, dan pengaruh pencahayaan terhadap deteksi wajah. Pengujian pada *black box testing* yaitu menemukan kesalahan yang terdapat pada program.

4.2.1 Pengujian Performansi

Pengujian performansi dilakukan dengan serangkaian percobaan-percobaan dalam kondisi-kondisi tertentu yang dapat mempengaruhi keefektifan kinerja sistem pendeteksian wajah.

4.2.1.1 Pengujian Kecepatan Gerakan Wajah

Pada pengujian ini dilakukan untuk menguji kemampuan senstivitas *fram*e *per second* (*fps*) webcam dalam menangkap dan mendeteksi gerakan wajah yang bergerak. Uji coba terhadap deteksi pergerakan ini dilakukan dengan cara \ melakukan perubahan gerakan wajah dari suatu posisi ke posisi yang lain dalam *window interface*.

Tabel 4.1 Uji Kecepatan GerakanWajah

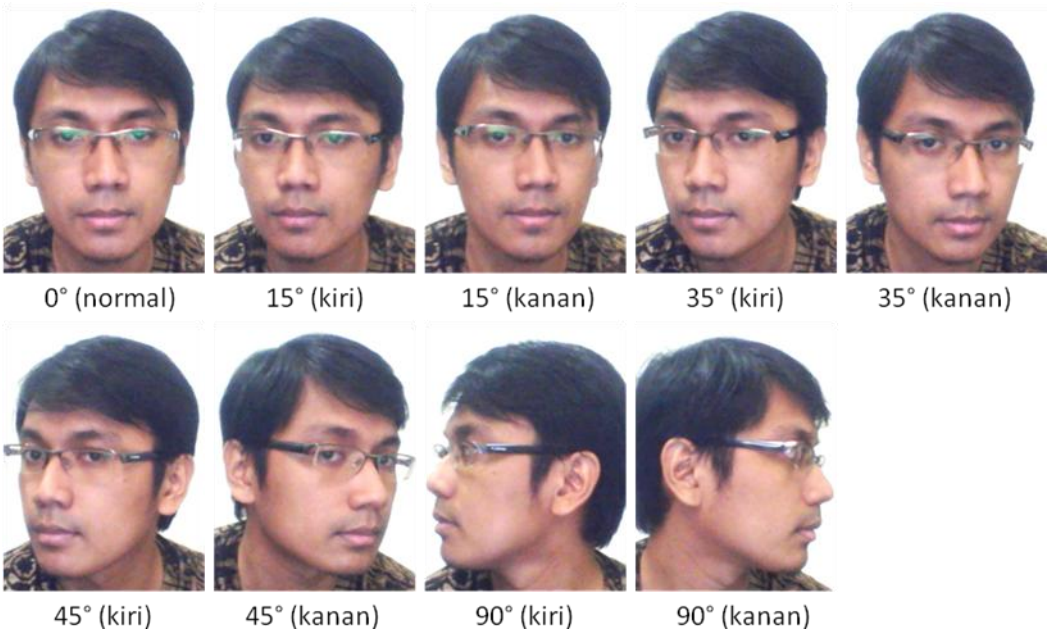
Subyek	Diam	Sedang	Cepat
1.	Terdeteksi	Terdeteksi	Terdeteksi
2.	Terdeteksi	Terdeteksi	Terdeteksi
3.	Terdeteksi	Terdeteksi	Terdeteksi
4.	Terdeteksi	Terdeteksi	Delay 1s
Rata-rata	100%	100%	75%

Tabel diatas menjelaskan bahwa semakin cepat gerakan yang dilakukan subyek maka sensitifitas dari sistem face detection akan semakin rendah, begitu pula sebaliknya jika gerakan wajah semakin lambat. Pada pengujian deteksi wajah dengan parameter kecepatan gerak pengguna ini ada beberapa faktor yang perlu diperhatikan antara lain: seberapa cepat kamera mengambil *frame* untuk tiap detiknya. Semakin tinggi nilai rata-rata *frame* perdetiknya maka pendeteksian wajah dengan gerakan cepat pun dapat dilakukan.

Hal ini dapat diatasi dengan penggunaan kamera yang dapat mengambil *frame* sebanyak mungkin dalam satu detiknya (nilai fps tinggi). Selain itu faktor yang perlu diperhatikan berikutnya adalah seberapa baiknya processor yang di gunakan, serta seberapa besar memori komputer yang dipakai (*memory usage*) serta posisi wajah harus di depan kamera.

4.2.1.2 Pengujian Deteksi Posisi Wajah

Deteksi wajah pada tugas akhir ini menggunakan proses *haar cascade*. Pengujian pada tahap ini dilakukan dengan beberapa variasi posisi sudut ke arah kiri dan ke kanan yang dibentuk oleh wajah, diantaranya adalah : 0° (normal), 15°, 35°, 45°, 90° seperti ditunjukkan seperti gambar berikut :



Gambar 4.5 Berbagai posisi pengujian deteksi wajah

Pengujian ini dilakukan dengan jarak uji : 40 cm.

Tabel 4.2 Pengujian Deteksi Posisi Wajah

Subyek	Normal	15°	35°	45°	90°
1.	Terdeteksi	Terdeteksi	Terdeteksi	Gagal	Gagal
2.	Terdeteksi	Terdeteksi	Terdeteksi	Gagal	Gagal
3.	Terdeteksi	Terdeteksi	Terdeteksi	Gagal	Gagal
4.	Terdeteksi	Terdeteksi	Terdeteksi	Gagal	Gagal
Rata-rata	100%	100%	100%	0%	0%

Pada pengujian ini dapat dianalisa bahwa kondisi posisi wajah berpengaruh untuk deteksi wajah, dimana saat wajah posisi 0° (normal) proses

deteksi berjalan dengan baik hingga pada posisi 35°. Sedangkan untuk posisi 45° dan 90° sistem sama sekali tidak dapat melakukan proses deteksi.

4.2.1.3 Pengujian Jarak Wajah Terhadap Webcam

Parameter yang juga sangat berpengaruh pada bagian pendeteksian wajah kali ini adalah jarak antara si pengguna dengan kamera yang digunakan. Namun pada pengujian kali ini hanya dilakukan pada user yang pendeteksiannya tepat pada area wajah yang tegak lurus dan ketinggiannya sejajar dengan tinggi webcam. Berikut adalah tabel pendeteksian wajah berdasarkan jarak user dengan kamera.

Tabel 4.3 Pengujian Jarak Wajah Terhadap Webcam.

Subyek	Jarak Wajah (cm)						
	20	40	60	80	100	120	>120
1	√	√	√	√	D1	X	X
2	√	√	√	√	√	X	X
3	√	√	√	√	√	X	X
4	√	√	√	√	√	X	X
Rata-rata (%)	100	100	100	100	75	0	0

Keterangan :

Tanda “√” menunjukkan wajah terdeteksi oleh sistem

Tanda “X” menunjukkan wajah tidak terdeteksi oleh sistem

D1 = Delay 1 second

Dari pengujian diatas dapat dianalisa dan disimpulkan bahwa semakin besar ukuran wajah dan semakin dekat wajah ke kamera, maka sistem deteksi wajah akan bekerja lebih baik apabila dilakukan pada jarak yang agak jauh dari posisi kamera (20 cm - 100 cm). Nilai optimum dari pengujian deteksi obyek berdasarkan jarak kamera adalah 20-80 cm.

4.2.1.4 Pengaruh Cahaya Terhadap Deteksi Wajah

Pengujian kali ini yaitu tentang pengaruh terhadap proses deteksi wajah. Dalam pengujian pengaruh pencahayaan ini menggunakan jenis lampu neon dan pijar bermerek Philips.

Tabel 4.4 Pengujian Pengaruh Cahaya terhadap Sistem

Subyek	Jenis Lampu				
	LP 15W	LP 40W	LN 8W	LN 14W	LN 23W
1	Terdeteksi	Terdeteksi	Terdeteksi	Terdeteksi	Terdeteksi
2	Terdeteksi	Terdeteksi	delay 1s	Terdeteksi	Terdeteksi
3	Terdeteksi	Terdeteksi	delay 1s	Terdeteksi	Terdeteksi
4	Terdeteksi	Terdeteksi	delay 1s	Terdeteksi	Terdeteksi
Rata-rata (%)	100	100	100	100	100

Keterangan :

LP = Lampu Pijar

LN = Lampu Neon

W = Watt

4.2.2 Pengujian Alpha

Pengujian *fungsional* yang digunakan untuk menguji sistem yang baru adalah metode pengujian *alpha*. Pengujian *alpha* dilakukan dengan menggunakan metode *black box*. Pengujian *black box* berfokus pada persyaratan fungsional perangkat lunak.

1. Rencana Pengujian

   Rencana pengujian yang akan dilakukan dapat dilihat pada tabel berikut.

Tabel 4.5 Rencana Pengujian Aplikasi

No	Komponen yang diuji	Butir Uji	Jenis Uji
1	Kamera	Memberikan akses kepada kamera	*Black box*
		Capture video secara realtime	*Black box*
2	Interaksi pengguna	Memilih model kacamata	*Black box*
		Menekan tombol capture image	*Black box*
		Menekan tombol simpan snapshot	*Black box*
		Menekan tombol kembali	*Black box*
3	Deteksi	Mendeteksi wajah pengguna	*Black box*
4	Model	Menampilkan model 3D	*Black box*
		Menyesuaikan ukuran dan letak model pada wajah	*Black box*

2. Kasus dan Hasil Pengujian

Berikut adalah kasus untuk menguji perangkat lunak yang sudah dibangun menggunakan metode *black box* berdasarkan Tabel Rencana Pengujian yang telah dibuat.

Tabel 4.6 Kasus dan Hasil Uji

No	Kasus/diuji	Skenario uji	Hasil yang diharapkan	Hasil pengujian
1	Kamera	Memberikan akses kepada kamera	Kamera bekerja setelah pengguna memilih tombol *allow* kamera	[√]Berhasil [ ]Tidak Berhasil
			Kamera tidak bekerja ketika pengguna memilih tombol *deny* kamera	[√]Berhasil [ ]Tidak Berhasil
2	Interaksi pengguna	Memilih model kacamata	pengguna berhasil memilih model kacamata	[√]Berhasil [ ]Tidak Berhasil
		Menekan tombol capture image	Menampilkan hasil capture oleh kamera	[√]Berhasil [ ]Tidak Berhasil
		Menekan tombol simpan snapshot	Pengguna dapat menyimpan gambar hasil capture kamera	[√]Berhasil [ ]Tidak Berhasil

		Menekan tombol kembali	Menampilkan kembali hasil streaming kamera pada layar	[√]Berhasil [ ]Tidak Berhasil
3	Deteksi Wajah	Mendeteksi wajah pengguna	Sistem dapat mendeteksi wajah pengguna melalui webcam	[√]Berhasil [ ]Tidak Berhasil
4	Model	Menampilkan model 3D	Menampilkan model 3D yang telah dipilih oleh pengguna	[√]Berhasil [ ]Tidak Berhasil
		Menyesuaikan ukuran dan letak model pada wajah	Posisi dan ukuran model 3D sesuai dengan wajah yang terdeteksi	[√]Berhasil [ ]Tidak Berhasil

4.2.2.1 Kesimpulan Hasil Pengujian Alpha

Berdasarkan hasil pengujian alpha (fungsional) dengan kasus uji di atas dapat ditarik kesimpulan bahwa pembangunan aplikasi virtual mirror eyeglasses tidak terdapat kesalahan proses dan secara fungsional mengeluarkan hasil yang sesuai dengan yang diharapkan.

4.2.3 Pengujian Beta

Untuk mengetahui tanggapan dan penilaian pengguna terhadap aplikasi ini, telah disebarkan kuesioner kepada 20 responden. Analisis kuantitatif terhadap hasil pengujian dengan kuesioner ini menggunakan Skala Likert [15]. Berikut adalah pertanyaan dan hasil kuesioner yang telah dibagikan.

Hasil pengujian *Beta* adalah sebagai berikut :

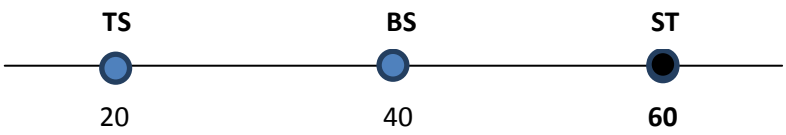
1. Apakah dengan menggunakan aplikasi ini dapat memudahkan anda mencoba model kacamata?

Kategori Jawaban	Tidak Setuju	Biasa Saja	Setuju
Frekuensi Jawaban	0	0	20
Skor	1	2	3
Jumlah Skor	0	0	60

Berdasarkan skor yang telah diterapkan rata-rata jawaban dapat dihitung sebagai berikut :

Jumlah skor ideal untuk seluruh item = 3 x 20 = 60 (seandainya semua jawaban Setuju). Jumlah skor yang didapat dari penelitian = 0+0+60 = **60**. Jadi berdasarkan data itu maka tingkat persetujuan terhadap aplikasi ini dapat memudahkan pengguna mencoba model kacamata sebesar = (60 : 60) x 100% = **100%** dari yang diharapkan (100%).

Secara kontinum dapat digambarkan sebagai berikut :



Jadi berdasarkan data yang diperoleh dari 20 orang responden maka rata-rata 60 terletak pada daerah **setuju**.

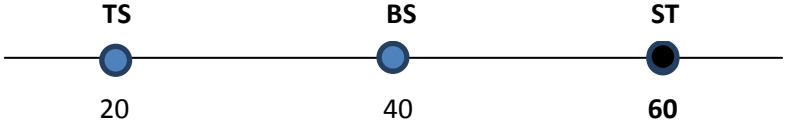
2. Dikarenakan jarak toko yang jauh, apakah dengan menggunakan aplikasi ini dapat menghemat waktu anda jika hanya ingin mencoba kacamata?

Kategori Jawaban	Tidak Setuju	Biasa Saja	Setuju
Frekuensi Jawaban	0	0	20
Skor	1	2	3
Jumlah Skor	0	0	60

Berdasarkan skor yang telah diterapkan rata-rata jawaban dapat dihitung sebagai berikut :

Jumlah skor ideal untuk seluruh item = 3 x 20 = 60 (seandainya semua jawaban Setuju). Jumlah skor yang didapat dari penelitian = 0+0+60 = **60**. Jadi berdasarkan data itu maka tingkat persetujuan terhadap aplikasi ini dapat menghemat waktu pengguna sebesar = (60 : 60) x 100% = **100%** dari yang diharapkan (100%).

Secara kontinum dapat digambarkan sebagai berikut :



Jadi berdasarkan data yang diperoleh dari 20 orang responden maka rata-rata 60 terletak pada daerah **setuju**.

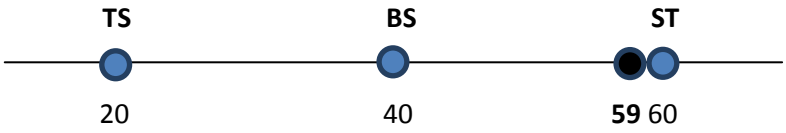
3. Apakah aplikasi ini mudah dalam penggunaannya?

Kategori Jawaban	Tidak Setuju	Biasa Saja	Setuju
Frekuensi Jawaban	0	1	19
Skor	1	2	3
Jumlah Skor	0	2	57

Berdasarkan skor yang telah diterapkan rata-rata jawaban dapat dihitung sebagai berikut :

Jumlah skor ideal untuk seluruh item = 3 x 20 = 60 (seandainya semua jawaban Setuju). Jumlah skor yang didapat dari penelitian = 0+2+57 = **59**. Jadi berdasarkan data itu maka tingkat persetujuan terhadap aplikasi ini mudah digunakan sebesar = (59 : 60) x 100% = **98%** dari yang diharapkan (100%).

Secara kontinum dapat digambarkan sebagai berikut :



Jadi berdasarkan data yang diperoleh dari 20 orang responden maka rata-rata 59 terletak pada daerah **setuju**.

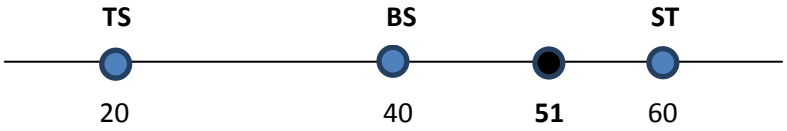
4. Apakah model kacamata virtual yang disediakan pada aplikasi mirip dengan kacamata aslinya?

Kategori Jawaban	Tidak Setuju	Biasa Saja	Setuju
Frekuensi Jawaban	2	5	13
Skor	1	2	3
Jumlah Skor	2	10	39

Berdasarkan skor yang telah diterapkan rata-rata jawaban dapat dihitung sebagai berikut :

Jumlah skor ideal untuk seluruh item = 3 x 20 = 60 (seandainya semua jawaban Setuju). Jumlah skor yang didapat dari penelitian = 2+10+39 = **51**. Jadi berdasarkan data itu maka tingkat persetujuan terhadap model pada aplikasi ini mirip dengan aslinya sebesar = (51 : 60) x 100% = **85%** dari yang diharapkan (100%).

Secara kontinum dapat digambarkan sebagai berikut :



Jadi berdasarkan data yang diperoleh dari 20 orang responden maka rata-rata 51 terletak pada daerah **setuju**.

5. Apakah menggunakan aplikasi ini menjadi salah satu daya tarik anda sehingga berminat ingin membeli?

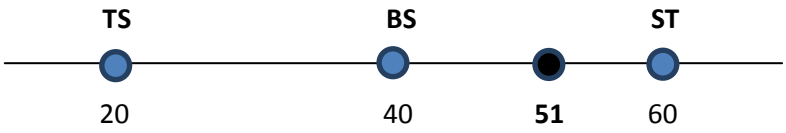
Kategori Jawaban	Tidak Setuju	Biasa Saja	Setuju
Frekuensi Jawaban	1	7	12
Skor	1	2	3
Jumlah Skor	1	14	36

Berdasarkan skor yang telah diterapkan rata-rata jawaban dapat dihitung sebagai berikut :

Jumlah skor ideal untuk seluruh item = 3 x 20 = 60 (seandainya semua jawaban Setuju). Jumlah skor yang didapat dari penelitian =

1+14+36 = **51**. Jadi berdasarkan data itu maka tingkat persetujuan terhadap aplikasi sebagai daya penarik minat pengguna sebesar = (51 : 60) x 100% = **85%** dari yang diharapkan (100%).

Secara kontinum dapat digambarkan sebagai berikut :



Jadi berdasarkan data yang diperoleh dari 20 orang responden maka rata-rata 51 terletak pada daerah **setuju**.

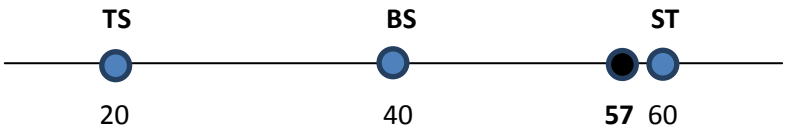
6. Apakah tampilan antarmuka aplikasi menarik?

Kategori Jawaban	Tidak Setuju	Biasa Saja	Setuju
Frekuensi Jawaban	0	3	17
Skor	1	2	3
Jumlah Skor	0	6	51

Berdasarkan skor yang telah diterapkan rata-rata jawaban dapat dihitung sebagai berikut :

Jumlah skor ideal untuk seluruh item = 3 x 20 = 60 (seandainya semua jawaban Setuju). Jumlah skor yang didapat dari penelitian = 0+6+51 = **57**. Jadi berdasarkan data itu maka tingkat persetujuan terhadap antarmuka aplikasi menarik sebesar = (57 : 60) x 100% = **95%** dari yang diharapkan (100%).

Secara kontinum dapat digambarkan sebagai berikut :



Jadi berdasarkan data yang diperoleh dari 20 orang responden maka rata-rata 57 terletak pada daerah **setuju**.

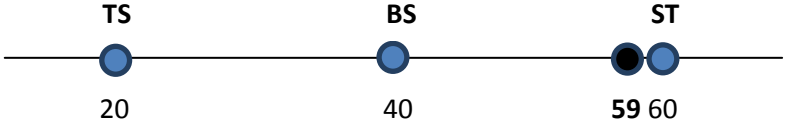
7. Apakah anda selalu menggunakan fitur ‘ambil screenshot’ setiap kali anda menggunakan aplikasi ini?

Kategori Jawaban	Tidak Setuju	Biasa Saja	Setuju
Frekuensi Jawaban	0	1	19
Skor	1	2	3
Jumlah Skor	0	2	57

Berdasarkan skor yang telah diterapkan rata-rata jawaban dapat dihitung sebagai berikut :

Jumlah skor ideal untuk seluruh item = 3 x 20 = 60 (seandainya semua jawaban Setuju). Jumlah skor yang didapat dari penelitian = 0+2+57 = **59**. Jadi berdasarkan data itu maka tingkat persetujuan terhadap antarmuka aplikasi menarik sebesar = (59 : 60) x 100% = **98%** dari yang diharapkan (100%).

Secara kontinum dapat digambarkan sebagai berikut :



Jadi berdasarkan data yang diperoleh dari 20 orang responden maka rata-rata 59 terletak pada daerah **setuju**.

4.2.3.1 Kesimpulan Pengujian Beta

Berdasarkan hasil kuesioner (pengujian *beta*) yang dilakukan maka dapat disimpulkan bahwa :

1. Aplikasi *virtual mirror eyeglasses* ini dapat memudahkan konsumen dalam mencoba kacamata tanpa harus membeli.
2. Dengan adanya aplikasi ini konsumen tidak harus datang ke toko untuk hanya sekedar mencoba kacamata, sehingga sangat menghemat waktu konsumen.
3. Model kacamata belum terlalu mirip dengan aslinya karena faktor hardware dan kemampuan desain modelnya yang terbatas.
4. Tidak sedikit juga konsumen yang tertarik ingin membeli setelah menggunakan aplikasi ini.
5. Pengguna tertarik menggunakan aplikasi ini karena kebanyakan pengguna masih belum mengetahui teknologi *augmented reality*.