BAB IV
IMPLEMENTASI DAN PENGUJIAN SISTEM

4.1. Implementasi Sistem

Implementasi sistem merupakan tahap kelanjutan setelah aplikasi selesai melalui tahap analisis dan perancangan sistem. Pada tahap ini aplikasi mulai di implementasikan dalam bentuk lingkungan implementasi, implementasi api, implementasi basis data serta implementasi antar muka.

4.1.1. Lingkungan Implementasi

Lingkungan implementasi mencakup spesifikasi kebutuhan yang dibutuhkan untuk perangkat yang akan digunakan oleh sistem. Berikut ini merupakan lingkungan implementasi yang harus dipenuhi:

1. Spesifikasi Kebutuhan Perangkat Keras

   Spesifikasi untuk perangkat keras yang digunakan untuk dapat mengoperasikan sistem dengan baik adalah sebagai berikut:

Tabel 4.1 Kebutuhan Perangkat Keras

No.	*Spesifikasi Perangkat Keras*
1	Koneksi data seluler HSDPA atau 4G LTE
2	Dapat mengaktifkan GPS
3	RAM 1 Gb

2. Spesifikasi Kebutuhan Perangkat Lunak

   Spesifikasi untuk perangkat lunak yang digunakan untuk dapat menggunakan sistem dengan baik adalah sebagai berikut:

Tabel 4.2 Kebutuhan Perangkat Lunak

No.	Spesifikasi Perangkat Lunak
1	Sistem operasi android versi 5.0 Lollipop

4.1.2. Implementasi API *Add Brand*

Berikut ini merupakan implementasi API *Add Brand* yang digunakan dalam proses tambah data *brand* pada sistem:

Tabel 4.3 Implementasi API *Add Brand*

```
<?php
include ("koneksi.php");
$brand_name= $_POST['brand_name'];
$brand_location= $_POST['brand_location'];
$id_kategori= $_POST['id_kategori'];
$latitude= $_POST['latitude'];
$longitude= $_POST['longitude'];
$imsrc = base64_decode($_POST['base64']);
$img= $_POST['img'];
header('Content-Type: bitmap; charset=utf-8');
    /*Lokasi penyimpanan gambar yang diupload*/
$path = "gambar/$img";
$handle = fopen($path, 'wb');
$numbytes = fwrite($handle, $imsrc);
$q = mysql_query("INSERT INTO `brand`(`nama`, `foto`,
`id_kategori`, `lokasi`, `lat_brand`, `long_brand`) VALUES
('$brand_name', '$img', '$id_kategori', '$brand_location',
'$latitude', '$longitude')");
?>
```

Berikut ini merupakan implementasi API *Add Category* yang digunakan dalam proses menambah data *category* pada sistem:

Tabel 4.4 Implementasi API Add Category

```
<?php
include ("koneksi.php");
$q = mysql_query("SELECT * FROM kategori");
$response["KATEGORI"] = array();
while ($a = mysql_fetch_array($q)){
 $output = array();
 $output["id"] = $a["id"];
 $output["nama"] = $a["nama"];
 array_push($response["KATEGORI"], $output);
}
echo json_encode($response);
?>
```

Berikut ini merupakan implementasi API Koneksi yang digunakan dalam proses menghubungkan *client* dengan database:

Tabel 4.5 Implementasi API Koneksi

```
<?php
$db = mysql_connect("localhost","angpaoweb_user","pa55_us3r") or
die("Error connecting to database: ".mysql_error());
mysql_select_db("angpaoweb_brandstory", $db) or
die(mysql_error());
?>
```

4.1.3. Implementasi API *List Brand*

Berikut ini merupakan implementasi API *List Brand* untuk menampilkan data *brand*:

Tabel 4.6 Implementasi API *List Brand*

```
<?php
include ("koneksi.php");
$q = mysql_query("SELECT b.id, b.nama, b.foto, k.nama as
nama_kategori FROM brand b, kategori k WHERE b.id_kategori =
k.id ORDER BY b.id DESC");
$response["BRAND"] = array();
while ($a = mysql_fetch_array($q)){
 $output = array();
 $output["id"] = $a["id"];
 $output["nama"] = $a["nama"];
 $output["foto"] = $a["foto"];
 $output["nama_kategori"] = $a["nama_kategori"];
 array_push($response["BRAND"], $output);
}
echo json_encode($response);
?>
```

4.1.4. Implementasi API *List Promo*

Berikut ini merupakan implementasi API *List Promo* untuk menampilkan data promo:

Tabel 4.7 Implementasi API *List Promo*

```
<?php
include ("koneksi.php");
$id = $_POST['id'];
$q = mysql_query("SELECT * FROM promo WHERE id_brand = '$id'");
$response["PROMO"] = array();
while ($a = mysql_fetch_array($q)){
 $output = array();
 $output["id"] = $a["id"];
 $output["id_brand"] = $a["id_brand"];
 $output["isi_promo"] = $a["isi_promo"];
```

```
 $output["gambar"] = $a["gambar"];
 array_push($response["PROMO"], $output);
}
echo json_encode($response);
?>
```

4.1.5. Implementasi API *Logo Detection*

Berikut ini merupakan implementasi API *Logo Detection* untuk mengenali sebuah product dari logo:

Tabel 4.8 Implementasi API *Logo Detection*

```
<?php
$BASE_64 = $_POST['BASE_64'];
$request_json = '{
"requests": [
{
"image": {
"source": {
"imageUri": "http://angpao.web.id/brandstory_api/gambar/nike1.jpg"
}
},
"features": [
{
"type": "LOGO_DETECTION",
"maxResults": 200
}
]
}
]
}';
$ch = curl_init();
$headers = array(
'Content-Type: application/json',

);
curl_setopt($ch, CURLOPT_URL,
'https://vision.googleapis.com/v1/images:annotate?key=AIzaSyApBEi7
-asp0qQI8l1LZuEEZtSUdhuILrw');
curl_setopt($ch, CURLOPT_HTTPHEADER, $headers);
curl_setopt($ch, CURLOPT_HEADER, 0);
$body = '{"requests":  [{ "features":  [ {"type":
"LABEL_DETECTION"}], "image": {"source": { "gcsImageUri":
"gs://vision-sample-images/4_Kittens.jpg"}}}]}';
curl_setopt($ch, CURLOPT_CUSTOMREQUEST, "POST");
curl_setopt($ch, CURLOPT_POSTFIELDS,$request_json);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
// Timeout in seconds
```

```
curl_setopt($ch, CURLOPT_TIMEOUT, 30);
$result=curl_exec($ch);
// Closing
curl_close($ch);
// Will dump a beauty json :3
//var_dump(json_encode($result, true));
echo $result;
?>
```

4.1.6. Implementasi API *Web Detection*

Berikut ini merupakan implementasi API *Web Detection* untuk mengenali sebuah product dari website yang di akses:

Tabel 4.9 Implementasi API *Web Detection*

```
<?php
$BASE_64 = $_POST['BASE_64'];
$request_json = '{
"requests": [
{
"image":{
"content":"'.$BASE_64.'"
},
"features": [
{
"type": "WEB_DETECTION",
"maxResults": 200
}
]
}
]
}';

$ch = curl_init();
$headers = array(
'Content-Type: application/json',
);
curl_setopt($ch, CURLOPT_URL,
'https://vision.googleapis.com/v1/images:annotate?key=AIzaSyApBEi7
-asp0qQI8l1LZuEEZtSUdhuILrw');
curl_setopt($ch, CURLOPT_HTTPHEADER, $headers);
curl_setopt($ch, CURLOPT_HEADER, 0);
$body = '{"requests":  [{ "features":  [ {"type":
"LABEL_DETECTION"}], "image": {"source": { "gcsImageUri":
"gs://vision-sample-images/4_Kittens.jpg"}}}]}';
curl_setopt($ch, CURLOPT_CUSTOMREQUEST, "POST");
curl_setopt($ch, CURLOPT_POSTFIELDS,$request_json);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
```

```
// Timeout in seconds
curl_setopt($ch, CURLOPT_TIMEOUT, 30);
$result=curl_exec($ch);
// Closing
curl_close($ch);
// Will dump a beauty json :3
//var_dump(json_encode($result, true));
echo $result;
?>
```

4.1.7. Implementasi Basis Data

Berikut ini merupakan implementasi dari pembuatan tabel dalam basis data untuk sistem yang dibangun:

1. Tabel *brand*

   Adapun perintah *sql* yang digunakan untuk membuat tabel *brand* adalah sebagai berikut:

Tabel 4.10 Implementasi Basis Data Tabel brand

```
CREATE TABLE `brand` (
  `id` int(11) NOT NULL,
  `nama` varchar(50) NOT NULL,
  `foto` varchar(100) NOT NULL,
  `id_kategori` int(11) NOT NULL,
  `lokasi` varchar(200) NOT NULL,
  `lat_brand` double NOT NULL,
  `long_brand` double NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
ALTER TABLE `brand`
  ADD PRIMARY KEY (`id`);
ALTER TABLE `brand`
  MODIFY    `id`    int(11)    NOT    NULL    AUTO_INCREMENT,
AUTO_INCREMENT=14;
```

2. Tabel *kategori*

   Adapun perintah *sql* yang digunakan untuk membuat tabel *kategori* adalah sebagai berikut:

Tabel 4. 11 Implementasi Basis Data Tabel kategori

```
CREATE TABLE `kategori` (
  `id` int(11) NOT NULL,
  `nama` varchar(50) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
ALTER TABLE `kategori`
```

```
  ADD PRIMARY KEY (`id`);
ALTER TABLE `kategori`
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=4;
```

3. Tabel *promo*

Adapun perintah *sql* yang digunakan untuk membuat tabel *promo* adalah sebagai berikut:

Tabel 4.12 Implementasi Basis Data Tabel promo

```
CREATE TABLE `promo` (
  `id` int(11) NOT NULL,
  `id_brand` int(11) NOT NULL,
  `isi_promo` text NOT NULL,
  `gambar` varchar(200) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
ALTER TABLE `promo`
  ADD PRIMARY KEY (`id`);
ALTER TABLE `promo`
  MODIFY     `id`     int(11)     NOT     NULL     AUTO_INCREMENT,
AUTO_INCREMENT=2;COMMIT;
```

4.1.8. Implementasi Antarmuka

Implementasi antarmuka merupakan tahap pembangunan di mana pada tahap ini dibangun tampilan antarmuka aplikasi dengan ekstensi *xml*. File tersebut merupakan tampilan nyata aplikasi dari desain aplikasi yang sebelumnya telah dibuat dalam tahap analisis antarmuka. Serta dilengkapi dengan file *class* mendukung file-file *xml* dan menjadikan aplikasi dapat berjalan sesuai dengan yang diinginkan. Di bawah ini merupakan daftar dari implementasi antarmuka serta *class* yang dibangun:

Tabel 4.13 Implementasi Antarmuka

No.	Nama Antarmuka	Nama *Class*
1	*Activity_main.xml*	*MainActivity*
2	*Content_main.xml*	*MainActivity*
3	*Activity_brand.xml*	*BrandActivity*
4	*Activity_camera.xml*	*CameraActivity*
5	*Activity_kategori.xml*	*KategoriActivity*
6	*Activity_promo.xml*	*PromoActivity*

4.2. Pengujian Sistem

Pengujian sistem merupakan tahap di mana aplikasi yang telah selesai dibangun akan diuji coba untuk menemukan kekurangan-kekurangan serta kesalahan yang masih terdapat pada aplikasi guna mendapatkan hasil aplikasi yang dapat berjalan sesuai dengan yang diharapkan pada tujuan awal penelitian.

4.2.1. Rencana Pengujian

Pengujian sistem yang akan dilakukan adalah pengujian fungsional dengan menggunakan metode *blackbox* dan penyebaran kuesioner secara *online* yang bertujuan untuk melakukan pengujian langsung oleh pengguna di mana pengguna dapat menggunakan aplikasi dan mengisi kuesioner yang telah disediakan agar didapatkan kekurangan atau kesalahan sistem yang mungkin terjadi saat pengguna menggunakan aplikasi yang telah dibangun.

4.2.2. Skenario Pengujian

Skenario pengujian berisikan tahapan yang akan dilakukan dalam pengujian sistem. Skenario pengujian sistem memuat skenario pengujian secara fungsional yang akan dilakukan dengan pengujian *blackbox* dan skenario pengujian *beta*.

1. Skenario Pengujian *Blackbox*

   Skenario pengujian *blackbox* yang akan dilakukan adalah sebagai berikut:

Tabel 4.14 Skenario Pengujian Blackbox

No	Kelas Pengujian	Poin Pengujian	Jenis Pengujian	Metode
1	Menambah data brand	Admin dapat menambah data brand	*Blackbox*	*Equivalence Partitioning*
2	Mengubah data brand	Admin dapat mengubah data brand	*Blackbox*	*Equivalence Partitioning*
3	Menghapus data brand	Admin dapat menghapus data brand	*Blackbox*	*Equivalence Partitioning*
4	Menambah data kategori	Admin dapat menambah data kategori	*Blackbox*	*Equivalence Partitioning*
5	Mengubah data kategori	Admin dapat mengubah data kategori	*Blackbox*	*Equivalence Partitioning*

6	Menghapus data kategori	Admin dapat menghapus data kategori	*Blackbox*	*Equivalence Partitioning*
7	Mencari data brand	User dapat mencari data brand	*Blackbox*	*Equivalence Partitioning*
8	Melihat data brand	User dapat melihat data brand	*Blackbox*	*Equivalence Partitioning*
9	Melihat store terdekat	User melihat store terdekat	*Blackbox*	*Equivalence Partitioning*
10	Melakukan *take photo by camera*.	User melakukan *take photo by camera*.	*Blackbox*	*Equivalence Partitioning*
11	Melihat webview	User dapat melihat webview	*Blackbox*	*Equivalence Partitioning*
12	Melihat rute	User dapat melihat rute	*Blackbox*	*Equivalence Partitioning*

2. Skenario Pengujian Beta

Pengujian Beta dilakukan dengan menyebarkan kuesioner kepada 40 responden. Kuesioner akan berisikan 8 pertanyaan yang kemudian akan diolah dengan menggunakan metode Skala Likert. Berikut ini adalah pertanyaan yang akan dibagikan kepada responden:

1. Apakah aplikasi dapat menampilkan data brand?
2. Apakah aplikasi dapat melakukan pencarian data brand sesuai inputan?
3. Aakah aplikasi dapat memberikan data store terdekat?
4. Apakah aplikasi dapat mengambil foto untuk di proses?
5. Apakah aplikasi dapat menampilkan webview?
6. Apakah aplikasi data menampilkan rute?
7. Apakah aplikasi bermanfaat untuk pengguna?
8. Apakah aplikasi sesuai dengan kebutuhan pengguna?

Pertanyaan di atas memiliki bobot jawaban dengan penilaian sebagai berikut:

Nilai Bobot	Jawaban
1	Sangat Tidak Setuju
2	Tidak Setuju
3	Cukup
4	Setuju
5	Sangat Setuju

Kemudian dari kuesioner yang telah berhasil diisi oleh responden, dilakukan perhitungan untuk menemukan persentase jawaban responden dengan rumus analisis deskriptif persentase sebagaimana berikut:

$$p = \frac{f}{N} * 100\%$$

Di mana, p: Persentase

f: Frekuensi jawaban

N: Jumlah responden

4.2.3. Hasil Pengujian

Berdasarkan pengujian yang telah dilakukan sesuai dengan skenario pengujian yang telah dibuat sebelumnya maka didapatkan hasil pengujian sebagai berikut ini:

4.2.3.1. Hasil Pengujian *Blackbox*

Hasil pengujian *blackbox* yang dilakukan akan dimasukkan ke dalam sebuah tabel di mana pada tabel tersebut dapat terlihat kesimpulan hasil yang didapatkan dari pengujian yang dilakukan apakah berhasil atau tidak.

1. Hasil Pengujian *Blackbox* Menambah data brand

Berikut ini merupakan hasil pengujian menambah data brand dengan metode *blackbox*.

Tabel 4.15 Hasil Pengujian Blackbox Menambah Data Brand(Data Benar)

Kasus dan Hasil Uji (Data Benar)			
Data Masukkan	Hasil yang diharapkan	Pengamatan	Kesimpulan
Nama: (nama brand) Foto: (foto brand) id_kategori: (id kategori) lokasi: (lokasi store) lat_brand: (latitude store) long_brand: (longitude store)	Sistem dapat menyimpan data brand ke database.	Muncul pesan "Data berhasil disimpan." Sistem menyimpan data brand ke database.	[√] Diterima [ ] Ditolak

Tabel 4.16 Hasil Pengujian Blackbox Menambah Data Brand(Data Salah)

Kasus dan Hasil Uji (Data Salah)			
Data Masukkan	Hasil yang diharapkan	Pengamatan	Kesimpulan
Nama: (kosong/salah) Foto: (kosong/salah) id_kategori: ((kosong/salah) lokasi: (kosong/salah) lat_brand: (kosong/salah) long_brand: (kosong/salah)	Sistem menampilkan pesan kesalahan	Tampil pesan "Data inputan salah". Data tidak disimpan ke database.	[√] Diterima [ ] Ditolak

2. Hasil Pengujian *Blackbox* Mengubah data Brand

Berikut ini merupakan hasil pengujian Mengubah data Brand dengan metode *blackbox*.

Tabel 4. 17 Hasil Pengujian *Blackbox* Mengubah data Brand (Data Benar)

Kasus dan Hasil Uji (Data Benar)			
Data Masukkan	Hasil yang diharapkan	Pengamatan	Kesimpulan
Nama: (nama brand) Foto: (foto brand) id_kategori: (id kategori) lokasi: (lokasi store) lat_brand: (latitude store) long_brand: (longitude store)	Sistem dapat menyimpan perubahan data brand ke database.	Muncul pesan "Data berhasil dirubah." Sistem menyimpan perubahan ke database	[√] Diterima [ ] Ditolak

Tabel 4. 18 Hasil pengujian *blackbox* Mengubah data Brand (Data Salah)

Kasus dan Hasil Uji (Data Salah)			
Data Masukkan	Hasil yang diharapkan	Pengamatan	Kesimpulan
Nama: (kosong/salah) Foto: (kosong/salah) id_kategori: ((kosong/salah) lokasi: (kosong/salah) lat_brand: (kosong/salah) long_brand: (kosong/salah)	Sistem menampilkan pesan kesalahan.	Sistem menampilkan pesan "Data gagal diubah." Data perubahan tidak disimpan.	[ ] Diterima [√] Ditolak

3. Hasil Pengujian *Blackbox* Hapus Data Brand

Berikut ini merupakan hasil pengujian Hapus Data Brand dengan metode *blackbox*.

Tabel 4. 19 Hasil Pengujian *Blackbox* Hapus Data Brand (Data Benar)

Kasus dan Hasil Uji (Data Benar)			
Data Masukkan	Hasil yang diharapkan	Pengamatan	Kesimpulan
Id: (id_brand)	Menampilkan pesan dan data brand dihapus.	Sistem menampilkan pesan "Data gagal dihapus." Data dengan id inputan terhapus dari database.	[√] Diterima [ ] Ditolak

Tabel 4. 20 Hasil pengujian *blackbox* Hapus Data Brand(Data Salah)

Kasus dan Hasil Uji (Data Salah)			
Data Masukkan	Hasil yang diharapkan	Pengamatan	Kesimpulan
Id: (id_brand)	Menampilkan pesan kesalahan.	Sistem dapat menampilkan pesan "Data gagal dihapus." Data tidak terhapus.	[ ] Diterima [√] Ditolak

4. Hasil Pengujian *Blackbox* Menambah Data Kategori

Berikut ini merupakan hasil pengujian Menambah Data Kategori dengan metode *blackbox*.

Tabel 4. 21 Hasil Pengujian *Blackbox* Menambah Data Kategori (Data Benar)

Kasus dan Hasil Uji (Data Benar)			
Data Masukkan	Hasil yang diharapkan	Pengamatan	Kesimpulan
Nama: (nama kategori)	Sistem dapat menyimpan data kategori.	Sistem menampilkan pesan "Data berhasil disimpan." Data tersimpan di database.	[√] Diterima [ ] Ditolak

Tabel 4. 22 Hasil Pengujian *Blackbox* Menambah Data Kategori (Data Salah)

Kasus dan Hasil Uji (Data Salah)			
Data Masukkan	Hasil yang diharapkan	Pengamatan	Kesimpulan
Nama: (kosong/salah)	Sistem dapat menampilkan pesan kesalahan.	Sistem menampilkan pesan "Data gagal disimpan." Data tidak disimpan di database.	[ ] Diterima [√] Ditolak

5. Hasil Pengujian *Blackbox* Mengubah Data Kategori

Berikut ini merupakan hasil pengujian melihat Mengubah Data Kategori dengan metode *blackbox*.

Tabel 4. 23 Hasil Pengujian *Blackbox* Mengubah Data Kategori(Data Benar)

Kasus dan Hasil Uji (Data Benar)			
Data Masukkan	Hasil yang diharapkan	Pengamatan	Kesimpulan
Nama: (nama kategori)	Sistem dapat menyimpan perubahan data kategori.	Sistem menampilkan pesan "Data berhasil diubah." Data perubahan tersimpan di database.	[√] Diterima [ ] Ditolak

Tabel 4. 24 Hasil Pengujian *Blackbox* Mengubah Data Kategori(Data Salah)

Kasus dan Hasil Uji (Data Salah)			
Data Masukkan	Hasil yang diharapkan	Pengamatan	Kesimpulan

Nama: (kosong/salah)	Sistem dapat menampilkan kesalahan.	Sistem menampilkan pesan “Data gagal diubah.” Data perubahan tidak tersimpan di database.	[ ] Diterima [√] Ditolak

6. Hasil Pengujian *Blackbox* Mengubah Data Kategori

Berikut ini merupakan hasil pengujian Mengubah Data Kategori dengan metode *blackbox*.

Tabel 4. 25 Hasil Pengujian *Blackbox* Mengubah Data Kategori(Data Benar)

Kasus dan Hasil Uji (Data Benar)			
Data Masukkan	Hasil yang diharapkan	Pengamatan	Kesimpulan
Id: (id_kategori)	Sistem menghapus data sesuai id inputan.	Sistem dapat menampilkan pesan “Data berhasil dihapus.” Data terhapus dari database.	[√] Diterima [ ] Ditolak

Tabel 4. 26 Hasil Pengujian *Blackbox* Mengubah Data Kategori(Data Salah)

Kasus dan Hasil Uji (Data Salah)			
Data Masukkan	Hasil yang diharapkan	Pengamatan	Kesimpulan
Id: (id_kategori)	Menampilkan pesan kesalahan.	Sistem dapat menampilkan pesan “Data gagal dihapus.” Data tidak terhapus.	[ ] Diterima [√] Ditolak

7. Hasil Pengujian *Blackbox* Mencari Brand

Berikut ini merupakan hasil pengujian Mencari Brand Pengguna dengan metode *blackbox*.

Tabel 4. 27 Hasil Pengujian *Blackbox* Mencari Brand(Data Benar)

Kasus dan Hasil Uji (Data Benar)			
Data Masukkan	Hasil yang diharapkan	Pengamatan	Kesimpulan
Nama: (nama brand)	Menampilkan informasi dari data pencarian.	Menampilkan informasi brand sesuai data inputan	[√] Diterima [ ] Ditolak

Tabel 4. 28 Hasil Pengujian *Blackbox* Mencari Brand(Data Salah)

Kasus dan Hasil Uji (Data Salah)			
Data Masukkan	Hasil yang diharapkan	Pengamatan	Kesimpulan
Nama: (salah/kosong)	Menampilkan pesan kesalahan	Sistem manmpilkan pesan "Data tidak ditemukan"	[√] Diterima [ ] Ditolak

8. Hasil Pengujian *Blackbox* Melihat Data Brand

Berikut ini merupakan hasil pengujian Melihat Data Brand dengan metode *blackbox*.

Tabel 4. 29 Hasil Pengujian *Blackbox* Melihat Data Brand(Data Benar)

Kasus dan Hasil Uji (Data Benar)			
Data Masukkan	Hasil yang diharapkan	Pengamatan	Kesimpulan
Memilih tombol lihat data brand	Menampilkan detail brand yang dipilih.	Menampilkan detail brand yang dipilih.	[√] Diterima [ ] Ditolak

Tabel 4. 30 Hasil Pengujian *Blackbox* List *Tweet* (Data Salah)

Kasus dan Hasil Uji (Data Salah)			
Data Masukkan	Hasil yang diharapkan	Pengamatan	Kesimpulan
Memilih tombol lihat data brand	Menampilkan pesan kesalahan	Menampilkan pesan "Detail brand tidak dapat ditampilkan."	[√] Diterima [ ] Ditolak

9. Hasil Pengujian *Blackbox* Melihat Store Terdekat

Berikut ini merupakan hasil pengujian Melihat Store Terdekat dengan metode *blackbox*.

Tabel 4. 31 Hasil Pengujian *Blackbox* Melihat Store Terdekat (Data Benar)

Kasus dan Hasil Uji (Data Benar)			
Data Masukkan	Hasil yang diharapkan	Pengamatan	Kesimpulan
Memilih tombol store terdekat	Menampilkan semua data store yang terdekat.	Sistem menampilkan list data store yang terdekat.	[√] Diterima [ ] Ditolak

Tabel 4. 32 Hasil Pengujian *Blackbox* Melihat Store Terdekat (Data Salah)

Kasus dan Hasil Uji (Data Salah)			
Data Masukkan	Hasil yang diharapkan	Pengamatan	Kesimpulan
Memilih tombol store terdekat	Menampilkan pesan kesalahan.	Sistem menampilkan pesan "tidak ada store terdekat"	[√] Diterima [ ] Ditolak

10. Hasil Pengujian *Blackbox* Melakukan *Take Photo by Camera*

Berikut ini merupakan hasil pengujian *Take Photo by Camera* dengan metode *blackbox*.

Tabel 4. 33 Hasil Pengujian *Blackbox Take Photo by Camera*(Data Benar)

Kasus dan Hasil Uji (Data Benar)			
Data Masukkan	Hasil yang diharapkan	Pengamatan	Kesimpulan
Memilih tombol photo.	Sistem dapat mengambil foto.	Sistem dapat mengambil foto.	[√] Diterima [ ] Ditolak

Tabel 4. 34 Hasil Pengujian *Blackbox Take Photo by Camera*(Data Salah)

Kasus dan Hasil Uji (Data Benar)			
Data Masukkan	Hasil yang diharapkan	Pengamatan	Kesimpulan
Memilih tombol photo.	Sistem tidak dapat mengambil foto.	Sistem tidak dapat mengambil foto.	[√] Diterima [ ] Ditolak

11. Hasil Pengujian *Blackbox* Melihat Webview

Berikut ini merupakan hasil pengujian Melihat Webview dengan metode *blackbox*.

Tabel 4. 35 Hasil Pengujian *Blackbox* Melihat Webview(Data Benar)

Kasus dan Hasil Uji (Data Benar)			
Data Masukkan	Hasil yang diharapkan	Pengamatan	Kesimpulan
Memilih tombol lihat webview.	Sistem menampilkan webview.	Sistem menampilkan webview.	[ ] Diterima [ ] Ditolak

Tabel 4. 36 Hasil Pengujian *Blackbox* Melihat Webview(Data Salah)

Kasus dan Hasil Uji (Data Benar)			
Data Masukkan	Hasil yang diharapkan	Pengamatan	Kesimpulan
Memilih tombol lihat webview.	Sistem tidak dapat menampilkan webview.	Sistem tidak dapat menampilkan webview.	[ ] Diterima [ ] Ditolak

12. Hasil Pengujian *Blackbox* Melihat Rute

Berikut ini merupakan hasil pengujian Melihat Rute dengan metode *blackbox*.

Tabel 4. 37 Hasil Pengujian *Blackbox* Melihat Rute (Data Benar)

Kasus dan Hasil Uji (Data Benar)			
Data Masukkan	Hasil yang diharapkan	Pengamatan	Kesimpulan
Memilih tombol lihat rute.	Sistem menampilkan rute.	Sistem menampilkan rute.	[ ] Diterima [ ] Ditolak

Tabel 4. 38 Hasil Pengujian *Blackbox* Melihat Rute (Data Salah)

Kasus dan Hasil Uji (Data Benar)			
Data Masukkan	Hasil yang diharapkan	Pengamatan	Kesimpulan
Memilih tombol	Sistem tidak dapat	Sistem tidak	[ ] Diterima

lihat rute.	menampilkan rute.	dapat menampilkan rute.	[ ] Ditolak

4.2.3.2. Hasil Pengujian Data

Melalui 30 responden yang telah mengisi kuesioner pengujian sistem, kemudian dilakukan perhitungan yang akan menentukan bagaimana reaksi pengguna terhadap aplikasi yang telah dibangun ini. Berikut ini merupakan hasil dari pengujian beta.

1. Hasil Pengujian Pertanyaan Satu

Dari hasil pertanyaan pertama diperoleh perhitungan serta skala sebagai berikut:

Tabel 4.29Aspek Pertanyaan Satu

Aspek Pertanyaan dan Hasil Pengujian					
Aspek Pertanyaan 1	Apakah aplikasi dapat memberikan rekomendasi brand yang sesuai?				
Kategori Jawaban	Sangat Setuju	Setuju	Cukup	Tidak Setuju	Sangat Tidak Setuju
Responden	9	18	3	0	0
Jumlah Sampel	30				
Perhitungan	(9x5)+(18x4)+(3x3)+(0x2)+(0x1) = 126				
Dibagi	126 / 30 = 4,2				
Hasil	4,2x100% = 4,2%				



2. Hasil Pengujian Pertanyaan Dua

Dari hasil pertanyaan kedua diperoleh perhitungan serta skala sebagai berikut:

Tabel 4.30 Aspek Pertanyaan Dua

Aspek Pertanyaan dan Hasil Pengujian					
Aspek Pertanyaan 2	Apakah aplikasi dapat memberikan informasi produk yang sesuai?				
Kategori Jawaban	Sangat Setuju	Setuju	Cukup	Tidak Setuju	Sangat Tidak Setuju
Responden	5	23	2	0	0
Jumlah Sampel	30				
Perhitungan	(5x5)+(23x4)+(2x3)+(0x2)+(0x1) = 123				
Dibagi	123 / 30 = 4,1				
Hasil	4,1x100% = 4,1%				



3. Hasil Pengujian Pertanyaan Tiga

Dari hasil pertanyaan ketiga diperoleh perhitungan serta skala sebagai berikut:

Tabel 4.31 Aspek Pertanyaan Tiga

Aspek Pertanyaan dan Hasil Pengujian					
Aspek Pertanyaan 3	Apakah aplikasi dapat menampilkan rute menuju tempa brand yang di cari?				
Kategori Jawaban	Sangat Setuju	Setuju	Cukup	Tidak Setuju	Sangat Tidak Setuju
Responden	11	15	3	1	0
Jumlah Sampel	30				
Perhitungan	(11x5)+(15x4)+(3x3)+(1x2)+(0x1) = 126				
Dibagi	126 / 30 = 4,2				
Hasil	4,2x100% = 4,2%				



4.2.3.3. Kesimpulan Hasil Pengujian Blackbox

Berdasarkan pada hasil dari pengujian *blackbox* yang telah dilakukan makan dapat ditarik kesimpulan:

1. Sistem dapat memberikan respons dengan baik ketika pengguna memasukkan data yang benar maka akan menghasilkan keluaran data yang diharapkan.
2. Sistem cukup baik dalam memberikan respons untuk data yang salah. Hal ini dapat terlihat ketika pengguna memasukkan data yang tidak tepat maka akan mengeluarkan hasil yang diharapkan berupa pesan kesalahan.

4.2.3.4. Kesimpulan Pengujian Beta

Berdasarkan pada hasil pengujian beta yang telah dilakukan melalui tahap memberikan 3 buah pertanyaan kepada 30 responden, maka dapat ditarik kesimpulan:

1. Aplikasi dapat memberikan rekomendasi brand yang sesuai
2. Aplikasi dapat memberikan informasi produk yang sesuai
3. Aplikasi dapat
4. menampilkan rute menuju rute tempat brand yang di cari