BAB 4
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 Pengguna

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

Tabel 4.1 Kebutuhan Perangkat Keras Pengguna

No.	Spesifikasi Perangkat Keras
1	Koneksi data seluler HSDPA atau 4G LTE
2	Dapat mengaktifkan GPS
3	Sensor Accelerometer dapat berjalan dengan baik
4	RAM 1 Gb

2. Spesifikasi Kebutuhan Perangkat Lunak Pengguna

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

Tabel 4.2 Kebutuhan Perangkat Lunak Pengguna

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

3. Spesifikasi Kebutuhan Perangkat Keras Server

Spesifikasi perangkat keras yang digunakan untuk server adalah sebagai berikut:

Tabel 4.3 Kebutuhan Perangkat Keras Server

No.	Spesifikasi Perangkat Keras
1	Intel® Xeon® Processor E3-1225v3 (3.2 GHz)
2	RAM 2 Gb
3	Minimum hard disk space 10 Gb
4	System type 64-Bit Operating System
5	Mouse, Keyboard, Monitor

4. Spesifikasi Kebutuhan Perangkat Lunak Server

Spesifikasi untuk perangkat lunak digunakan server adalah sebagai berikut:

Tabel 4.4 Kebutuhan Perangkat Lunak Server

No.	Spesifikasi Perangkat Lunak
1	1 Website
2	1 akun E-mail
3	1 User FTP
4	1 Database MySQL

4.1.2. Implementasi API Twitter

Berikut ini merupakan implementasi API Twitter yang digunakan dalam proses *login* sistem:

```
session = new SessionManager(getApplicationContext());
if (logged_in_check_is_true = !session.isLoggedIn())
{
    Intent intent = new Intent(MainActivity.this,
SigninActivity.class);
    startActivity(intent);
    finish();
}
HashMap<String, String> user = session.getUserDetails();
nama = user.get(SessionManager.KEY_NAMA);
foto = user.get(SessionManager.KEY_FOTO);
email = user.get(SessionManager.KEY_EMAIL);
PREF_KEY_OAUTH_TOKEN =
user.get(SessionManager.KEY_PREF_KEY_OAUTH_TOKEN);
PREF_KEY_OAUTH_SECRET =
user.get(SessionManager.KEY_PREF_KEY_OAUTH_SECRET);
aq.id(R.id.tvNama).text("Hello, " + nama);
aq.id(R.id.profile_image).image(foto, true, true, 200, 0);
listView = (ExpandableHeightListView)
findViewById(R.id.listView);
listView.setOnItemClickListener(this);
layout = (PullRefreshLayout)
findViewById(R.id.swipeRefreshLayout);
layout.setOnRefreshListener(new
PullRefreshLayout.OnRefreshListener() {
    @Override
```

```
    public void onRefresh() {
        // start refresh
        ambilTweet();
    }
})
```

Berikut ini merupakan implementasi API Twitter yang digunakan dalam proses mengambil *tweet* pengguna:

```
String url = AppConstans.APITWEET;
Map<String,String> params = new HashMap<>();
params.put("id",email);
APIManager.loadTwitterList(this, new OnApiListener() {
    @Override
    public void onApiStartLoad(String url) {
        layout.setRefreshing(true);
    }
    @Override
    public void onApiSuccessLoad(String url, String message,
Object object, int resultNumber) {
        layout.setRefreshing(false);
        twitterItems = (ArrayList<TwitterItem>) object;
        setList();
    }
    @Override
    public void onApiErrorLoad(String url, String message) {
        layout.setRefreshing(false);
    }
}, url, params);
```

Berikut ini merupakan implementasi API Twitter yang digunakan dalam proses *update* status twitter pengguna:

```
Log.d("Tweet Text", "> " + args[0]);
    String status = args[0];
    try {
        ConfigurationBuilder builder = new
ConfigurationBuilder();
builder.setOAuthConsumerKey(TWITTER_CONSUMER_KEY);
   builder.setOAuthConsumerSecret(TWITTER_CONSUMER_SECRET);
        AccessToken accessToken = new AccessToken(access_token,
access_token_secret);
        Twitter twitter = new
TwitterFactory(builder.build()).getInstance(accessToken);
        twitter4j.Status response =
twitter.updateStatus(status);
        Log.d("Status", "> " + response.getText());
    } catch (TwitterException e) {
        Log.d("Twitter Update Error", e.getMessage());
    }
```

4.1.3. Implementasi API Uclassify

Berikut ini merupakan implementasi API Uclassify untuk mengubah teks pada *tweet* pengguna menjadi *mood*:

```
String url =
"http://angpao.web.id/musicomender_api/mood_detection.php";
List<NameValuePair> params = new ArrayList<NameValuePair>();
params.add(new BasicNameValuePair("tweet", tweet_edit));
JSONParser jParser = new JSONParser();
JSONObject json = jParser.makeHttpRequest(
        url, "POST", params);
try {
    Happy = json.getString("happy");
    Upset = json.getString("upset");
} catch (Exception e) {
    // TODO: handle exception
    Log.e("failed", "cannot retrieve data from 1 condition");
}
```

4.1.4. Implementasi API YouTube

Berikut ini merupakan implementasi API YouTube untuk layanan streaming video ke YouTube pada aplikasi:

```
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_player);
Bundle extras = getIntent().getExtras();
if(extras!=null){
    final String id = extras.getString("id");
    YouTubePlayerView youTubePlayerView =
            (YouTubePlayerView) findViewById(R.id.player);

youTubePlayerView.initialize("AIzaSyCLv8EILoYLdrhDEUBqTuvRbb7pwx
AjcHA",
            new YouTubePlayer.OnInitializedListener() {
                @Override
                public void
onInitializationSuccess(YouTubePlayer.Provider provider,
YouTubePlayer youTubePlayer, boolean b) {
                    youTubePlayer.cueVideo(id);
                }
                @Override
                public void
onInitializationFailure(YouTubePlayer.Provider provider,
YouTubeInitializationResult youTubeInitializationResult) {
                }
            });
}
```

4.1.5. Implementasi Basis Data

Database dibangun dengan nama musicomender. Berikut ini merupakan implementasi dari pembuatan tabel dalam basis data untuk sistem yang dibangun:

1. Tabel *user*

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

```
CREATE TABLE `user`(
  `id` VARCHAR(30) NOT NULL,
  `username` VARCHAR(30) NOT NULL,
  `foto` VARCHAR(200) NOT NULL,
  PRIMARY KEY (`id`)
);
```

2. Tabel *tweet*

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

```
CREATE TABLE `tweet`(
  `id_tweet` INT(11) NOT NULL AUTO_INCREMENT,
  `id_user` VARCHAR(20) NOT NULL,
  `tweet` VARCHAR(200) NOT NULL,
  `time` DATETIME,
  PRIMARY KEY (`id_tweet`),
  CONSTRAINT `id_user` FOREIGN KEY (`id_user`) REFERENCES
`musicomender`.`user`(`id`)
);
```

3. Tabel *mood*

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

```
CREATE TABLE `mood`(
  `id` INT(11) NOT NULL AUTO_INCREMENT,
  `name` VARCHAR(20) NOT NULL,
  PRIMARY KEY (`id`)
);
```

4. Tabel *playlist*

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

```
CREATE TABLE `playlist`(
  `id` INT(11) NOT NULL AUTO_INCREMENT,
  `link_id` VARCHAR(100) NOT NULL,
  `id_mood` INT(11),
  PRIMARY KEY (`id`)
);
```

5. Tabel favorit

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

```
CREATE TABLE `favorit`(
  `id` INT(11) NOT NULL AUTO_INCREMENT,
  `id_user` VARCHAR(50) NOT NULL,
  `id_playlist` INT(11) NOT NULL,
  `id_song` VARCHAR(100) NOT NULL,
  PRIMARY KEY (`id`),
  CONSTRAINT `email` FOREIGN KEY (`id_user`) REFERENCES
`musicomender`.`user`(`id`),
  CONSTRAINT `id_playlist` FOREIGN KEY (`id_playlist`) REFERENCES
`musicomender`.`playlist`(`id`)
);
```

6. Tabel *location*

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

```
CREATE TABLE `location`(
  `id` INT(11) NOT NULL AUTO_INCREMENT,
  `min_degree` DOUBLE NOT NULL,
  `max_degree` DOUBLE NOT NULL,
  `id_mood` INT(11) NOT NULL,
  PRIMARY KEY (`id`),
  CONSTRAINT `id_mood` FOREIGN KEY (`id_mood`) REFERENCES
`musicomender`.`mood`(`id`));
```

4.1.6. 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 4.5 Implementasi Antarmuka

No.	Nama Antarmuka	Nama *Class*
1	*Activity_main.xml*	*MainActivity*
2	*Content_main.xml*	*MainActivity*
3	*Activity_mood.xml*	*MoodActivity*
4	*Activity_player.xml*	*PlayerActivity*
5	*Activity_signin.xml*	*SigninActivity*
6	*Activity_splash.xml*	*SplashActivity*

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 kepada pengguna 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. 6 Skenario Pengujian *Blackbox*

No	Kelas Pengujian	Poin Pengujian	Jenis Pengujian
1	*Login*	Validasi data *login*	*Blackbox*
2	Lagu Berdasarkan Aktivitas	Menampilkan playlist lagu berdasarkan aktivitas	*Blackbox*
3	Informasi Gerakan Pengguna	Menampilkan informasi gerakan pengguna	*Blackbox*
4	Lagu Berdasarkan Lokasi	Menampilkan playlist lagu berdasarkan lokasi pengguna	*Blackbox*

5	Informasi Lokasi Pengguna	Menampilkan informasi lokasi yang dikunjungi pengguna	*Blackbox*
6	Lagu Berdasarkan *Mood*	Menampilkan playlist lagu *mood*	*Blackbox*
7	Informasi *Mood* Pengguna	Menampilkan informasi *mood* pengguna	*Blackbox*
8	List *Tweet*	Menampilkan List *tweet* pengguna	*Blackbox*
9	Membuat *Tweet*	Mengirimkan *tweet* baru	*Blackbox*
10	Streaming YouTube	Melakukan streaming video	*Blackbox*
11	Menampilkan Lagu Favorit	Melihat list lagu favorit	*Blackbox*
12	Menyimpan Lagu Favorit	Menyimpan lagu list lagu favorit	*Blackbox*

2. Skenario Pengujian Beta

Pengujian Beta dilakukan dengan menyebarkan kuesioner kepada 30 responden. Kuesioner akan berisikan 3 pertanyaan yang kemudian akan diolah dengan menggunakan metode Skala Likert. Berikut ini adalah pertanyaan yang akan dibagikan kepada responden:

1. Apakah aplikasi dapat memberikan lagu rekomendasi yang cocok sesuai dengan *mood*?
2. Apakah aplikasi dapat memberikan lagu rekomendasi yang cocok sesuai dengan lokasi yang dikunjungi?
3. Apakah aplikasi dapat memberikan lagu rekomendasi yang cocok dengan aktivitas yang sedang dilakukan?

Pertanyaan di atas memiliki bobot jawaban dengan penilaian seperti pada tabel berikut ini:

Tabel 4. 7 Bobot Jawaban

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:

$$p = \frac{f}{N} * 100\% \quad [28]$$

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 Login*

Berikut ini merupakan hasil pengujian *Login* dengan metode *blackbox*.

Tabel 4.8 Hasil Blackbox *Login* (Data Benar)

Kasus dan Hasil Uji (Data Benar)			
Data Masukkan	Hasil yang diharapkan	Pengamatan	Kesimpulan
Username: (nama username twitter pengguna = jsdae) Password: (password akun twitter pengguna = jsdae17)	Data *login* benar dan dapat masuk ke halaman utama aplikasi. Aplikasi menampilkan list *tweet* dari twitter jsdae.	Masuk ke halaman utama aplikasi	[✓] Diterima [ ] Ditolak

Tabel 4.9 Hasil Blackbox *Login* (Data Salah)

Kasus dan Hasil Uji (Data Salah)			
Data Masukkan	Hasil yang diharapkan	Pengamatan	Kesimpulan
Username: (Kosong/salah) Password: (Kosong/salah)	Sistem menampilkan pesan kesalahan	Tampil pesan "Nama pengguna dan kata sandi yang Anda masukkan tidak cocok dengan catatan kami. Silakan periksa dan coba lagi.	[✓] Diterima [ ] Ditolak

2. Hasil Pengujian *Blackbox* Lagu Berdasarkan Aktivitas

Berikut ini merupakan hasil pengujian Lagu Berdasarkan Aktivitas dengan metode *blackbox*.

Tabel 4. 10 Hasil *Blackbox* Lagu Berdasarkan Aktivitas (Data Benar)

Kasus dan Hasil Uji (Data Benar)			
Data Masukkan	Hasil yang diharapkan	Pengamatan	Kesimpulan
Informasi gerakan pengguna: (Berjalan / Berlari)	Menampilkan playlist lagu berdasarkan aktivitas	Sistem menampilkan playlist lagu berdasarkan aktivitas	[✓] Diterima [ ] Ditolak

Tabel 4. 11 Hasil *Blackbox* Lagu Berdasarkan Aktivitas (Data Salah)

Kasus dan Hasil Uji (Data Salah)			
Data Masukkan	Hasil yang diharapkan	Pengamatan	Kesimpulan
Informasi gerakan pengguna: (Tidak dapat terdeteksi)	Sistem menampilkan pesan kesalahan "Gerakan Tidak Terdeteksi"	Sistem tidak menampilkan pesan kesalahan	[ ] Diterima [✓] Ditolak

3. Hasil Pengujian *Blackbox* Informasi Aktivitas Pengguna

Berikut ini merupakan hasil pengujian Informasi Aktivitas Pengguna dengan metode *blackbox*.

Tabel 4. 12 Hasil *Blackbox* Informasi Aktivitas Pengguna (Data Benar)

Kasus dan Hasil Uji (Data Benar)			
Data Masukkan	Hasil yang diharapkan	Pengamatan	Kesimpulan
(Nilai gerakan pengguna)	Menampilkan informasi aktivitas pengguna	Sistem menampilkan informasi gerakan pengguna	[✓] Diterima [ ] Ditolak

Tabel 4. 13 Hasil *Blackbox* Informasi Aktivitas Pengguna (Data Salah)

Kasus dan Hasil Uji (Data Salah)			
Data Masukkan	Hasil yang diharapkan	Pengamatan	Kesimpulan
(Nilai gerakan pengguna tidak terdeteksi)	Menampilkan pesan kesalahan "Gerakan tidak terdeteksi"	Sistem tidak menampilkan pesan kesalahan	[ ] Diterima [✓] Ditolak

4. Hasil Pengujian *Blackbox* Lagu Berdasarkan Lokasi

Berikut ini merupakan hasil pengujian Lagu Berdasarkan Lokasi dengan metode *blackbox*.

Tabel 4. 14 Hasil *Blackbox* Lagu Berdasarkan Lokasi (Data Benar)

Kasus dan Hasil Uji (Data Benar)			
Data Masukkan	Hasil yang diharapkan	Pengamatan	Kesimpulan
Informasi lokasi: (Nama tempat dari tempat yang dikunjungi)	Sistem menampilkan playlist lagu berdasarkan lokasi	Sistem menampilkan playlist lagu berdasarkan lokasi	[✓] Diterima [ ] Ditolak

Tabel 4. 15 Hasil *Blackbox* Lagu Berdasarkan Lokasi (Data Salah)

Kasus dan Hasil Uji (Data Salah)			
Data Masukkan	Hasil yang diharapkan	Pengamatan	Kesimpulan
Lokasi tidak terdeteksi	Sistem menampilkan pesan kesalahan "Lokasi tidak terdeteksi"	Sistem tidak menampilkan pesan kesalahan	[ ] Diterima [✓] Ditolak

5. Hasil Pengujian *Blackbox* Informasi Lokasi Pengguna

Berikut ini merupakan hasil pengujian melihat Informasi Lokasi Pengguna dengan metode *blackbox*.

Tabel 4. 16 Hasil *Blackbox* Informasi Lokasi Pengguna (Data Benar)

Kasus dan Hasil Uji (Data Benar)			
Data Masukkan	Hasil yang diharapkan	Pengamatan	Kesimpulan
Nilai *longitude* dan *latitude* lokasi	Menampilkan informasi lokasi dan cuaca yang dikunjungi	Menampilkan lokasi dan cuaca yang dikunjungi	[✓] Diterima [ ] Ditolak

Tabel 4. 17 Hasil *Blackbox* Informasi Lokasi Pengguna (Data Salah)

Kasus dan Hasil Uji (Data Salah)			
Data Masukkan	Hasil yang diharapkan	Pengamatan	Kesimpulan
GPS ponsel mati / tidak terdeteksi	Menampilkan pesan kesalahan "Lokasi tidak terdeteksi"	Tidak menampilkan pesan kesalahan	[ ] Diterima [✓] Ditolak

6. Hasil Pengujian *Blackbox* Lagu Berdasarkan *Mood*

Berikut ini merupakan hasil pengujian Lagu Berdasarkan *Mood* dengan metode *blackbox*.

Tabel 4. 18 Hasil *Blackbox* Lagu Berdasarkan *Mood* (Data Benar)

Kasus dan Hasil Uji (Data Benar)			
Data Masukkan	Hasil yang diharapkan	Pengamatan	Kesimpulan
Informasi *mood*: "Happy"	Menampilkan playlist lagu senang	Menampilkan playlist lagu senang	[✓] Diterima [ ] Ditolak
Informasi *mood*: "Upset"	Menampilkan playlist lagu sedih	Menampilkan playlist lagu sedih	[✓] Diterima [ ] Ditolak

7. Hasil Pengujian *Blackbox* Informasi *Mood* Pengguna

Berikut ini merupakan hasil pengujian Informasi *Mood* Pengguna dengan metode *blackbox*.

Tabel 4. 19 Hasil *Blackbox* Informasi *Mood* Pengguna (Data Benar)

Kasus dan Hasil Uji (Data Benar)			
Data Masukkan	Hasil yang diharapkan	Pengamatan	Kesimpulan
(*Tweet* senang dengan bahasa inggris)	Menampilkan informasi *mood*: "Happy"	Menampilkan informasi *mood*: "Happy"	[✓] Diterima [ ] Ditolak
(*Tweet* sedih dengan bahasa inggris)	Menampilkan informasi *mood*: "Upset"	Menampilkan informasi *mood*: "Upset"	[✓] Diterima [ ] Ditolak

Tabel 4. 20 Hasil *Blackbox* Informasi *Mood* Pengguna (Data Salah)

Kasus dan Hasil Uji (Data Salah)			
Data Masukkan	Hasil yang diharapkan	Pengamatan	Kesimpulan
Tweet senang dengan bahasa Indonesia	Menampilkan informasi *mood*: "Happy"	Menampilkan informasi *mood*: "Happy"	[✓] Diterima [ ] Ditolak
Tweet sedih dengan bahasa Indonesia	Menampilkan informasi *mood*: "Upset"	Menampilkan informasi *mood*: "Upset"	[✓] Diterima [ ] Ditolak

8. Hasil Pengujian *Blackbox* List *Tweet*

Berikut ini merupakan hasil pengujian List *Tweet* dengan metode *blackbox*.

Tabel 4. 21 Hasil *Blackbox* List *Tweet* (Data Benar)

Kasus dan Hasil Uji (Data Benar)			
Data Masukkan	Hasil yang diharapkan	Pengamatan	Kesimpulan
Data login: (Username dan password)	Menampilkan list *tweet* pengguna	Menampilkan list *tweet* pengguna	[✓] Diterima [ ] Ditolak

Tabel 4. 22 Hasil *Blackbox* List *Tweet* (Data Salah)

Kasus dan Hasil Uji (Data Salah)			
Data Masukkan	Hasil yang diharapkan	Pengamatan	Kesimpulan
Data login salah atau kosong	Menampilkan pesan kesalahan	Tampil pesan "Nama pengguna dan kata sandi yang Anda masukkan tidak cocok dengan catatan kami. Silakan periksa dan coba lagi.	[✓] Diterima [ ] Ditolak

9. Hasil Pengujian *Blackbox* Membuat *Tweet*

Berikut ini merupakan hasil pengujian Membuat *Tweet* dengan metode *blackbox*.

Tabel 4. 23 Hasil *Blackbox* Membuat *Tweet* (Data Benar)

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

(*Tweet* dengan karakter ≥ 1 atau ≤ 280)	*Tweet* masuk ke list *tweet* pengguna di halaman utama	*Tweet* masuk ke list *tweet*	[✓] Diterima [ ] Ditolak

Tabel 4. 24 Hasil *Blackbox* Membuat *Tweet* (Data Salah)

Kasus dan Hasil Uji (Data Salah)			
Data Masukkan	Hasil yang diharapkan	Pengamatan	Kesimpulan
(*Tweet* dengan karakter < 1 atau > 280)	*Tweet* tidak dapat diproses	Button *tweet* bekerja, pengguna dapat mengirim *tweet* tetapi *tweet* tidak masuk ke akun twitter pengguna	[✓] Diterima [ ] Ditolak

10. Hasil Pengujian *Blackbox* Streaming Video

Berikut ini merupakan hasil pengujian Streaming Video dengan metode *blackbox*.

Tabel 4. 25 Hasil *Blackbox* Streaming Video (Data Benar)

Kasus dan Hasil Uji (Data Benar)			
Data Masukkan	Hasil yang diharapkan	Pengamatan	Kesimpulan
Data lagu yang ingin dimainkan dari playlist lagu: (Nama penyanyi dan judul lagu)	Menampilkan fasilitas streaming video ke YouTube	Menampilkan fasilitas streaming video ke YouTube	[✓] Diterima [ ] Ditolak

11. Hasil Pengujian *Blackbox* Menambahkan Lagu Favorit

Berikut ini merupakan hasil pengujian Menambahkan Lagu Favorit dengan metode *blackbox*.

Tabel 4. 26 Hasil *Blackbox* Menambahkan Lagu Favorit (Data Benar)

Kasus dan Hasil Uji (Data Benar)			
Data Masukkan	Hasil yang diharapkan	Pengamatan	Kesimpulan
Data lagu yang ingin ditambahkan ke daftar lagu favorit: (Nama penyanyi, judul lagu)	Lagu berhasil ditambahkan ke list lagu favorit	Lagu ditambahkan ke list lagu favorit	[✓] Diterima [ ] Ditolak

12. Hasil Pengujian *Blackbox* List Lagu Favorit

Berikut ini merupakan hasil pengujian List Lagu Favorit dengan metode *blackbox*.

Tabel 4. 27 Hasil *Blackbox* List Lagu Favorit (Data Benar)

Kasus dan Hasil Uji (Data Benar)			
Data Masukkan	Hasil yang diharapkan	Pengamatan	Kesimpulan
-	Menampilkan playlist lagu favorit	Menampilkan playlist lagu favorit	[✓] Diterima

			[ ] Ditolak

4.2.3.2. Hasil Pengujian Beta

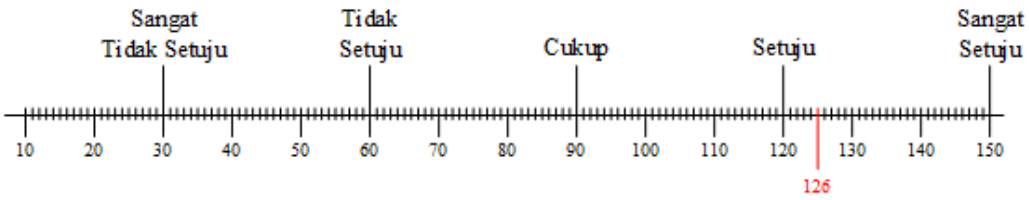
Melalui 30 responden yang telah mengisi kuesioner pengujian sistem, kemudian dilakukan perhitungan yang akan menentukan bagaimana reaksi pengguna terhadap aplikasi yang telah dibangun ini. Berikut ini merupakan hasil dari pengujian beta.

1. Hasil Pengujian Pertanyaan Satu

Dari hasil pertanyaan pertama diperoleh perhitungan serta skala sebagai berikut:

Tabel 4. 28 Hasil Pertanyaan Satu

Hasil Pertanyaan					
Aspek Pertanyaan 1	Apakah aplikasi dapat memberikan lagu rekomendasi yang sesuai dengan *mood*?				
Kategori Jawaban	Sangat Setuju	Setuju	Cukup	Tidak Setuju	Sangat Tidak Setuju
Skor	5	4	3	2	1
Frekuensi Jawaban	9	18	3	0	0
Jumlah Sampel	30				
Skor x Frekuensi Jawaban	45	72	9	0	0
Total Skor	126				
Persentase	$\frac{126}{30 \times 5} \times 100\% = 84\%$				
Hasil	Sangat Setuju				



Gambar 4.1 Hasil Pengujian Satu

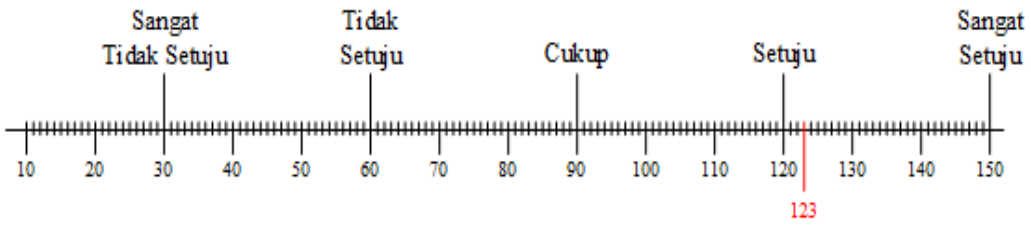
Dari gambar di atas, dapat dilihat bahwa terdapat total skor sebanyak 126 dengan persentase 84% responden yang mengaku aplikasi yang telah dibuat pada penelitian ini Sangat Setuju bahwa aplikasi dapat memberikan rekomendasi lagu berdasarkan *mood* pengguna.

2. Hasil Pengujian Pertanyaan Dua

Dari hasil pertanyaan kedua diperoleh perhitungan serta skala sebagai berikut:

Tabel 4. 29 Aspek Pertanyaan Dua

Hasil Pertanyaan					
Aspek Pertanyaan 2	Apakah aplikasi dapat memberikan lagu rekomendasi yang sesuai dengan lokasi yang dikunjungi?				
Kategori Jawaban	Sangat Setuju	Setuju	Cukup	Tidak Setuju	Sangat Tidak Setuju
Skor	5	4	3	2	1
Frekuensi Jawaban	5	23	2	0	0
Jumlah Sampel	30				
Skor x Frekuensi Jawaban	25	92	6	0	0
Total Skor	123				
Persentase	$\frac{123}{30 \times 5} \times 100\% = 82\%$				
Hasil	Sangat Setuju				



Gambar 4.2 Hasil Pengujian Dua

Dari gambar di atas, dapat dilihat bahwa terdapat total skor sebanyak 123 dengan persentase 82% responden yang mengaku aplikasi yang telah dibuat pada penelitian ini Sangat Setuju bahwa aplikasi dapat memberikan rekomendasi lagu berdasarkan lokasi yang dikunjungi oleh pengguna.

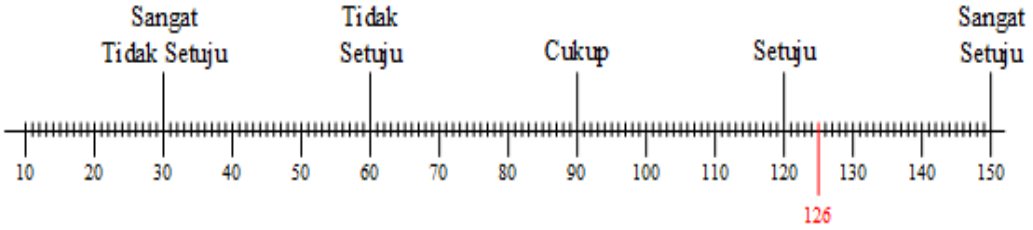
3. Hasil Pengujian Pertanyaan Tiga

Dari hasil pertanyaan ketiga diperoleh perhitungan serta skala sebagai berikut:

Tabel 4. 30 Aspek Pertanyaan Tiga

Hasil Pertanyaan					
Aspek Pertanyaan 3	Apakah aplikasi dapat memberikan lagu rekomendasi yang sesuai dengan aktivitas yang sedang dilakukan?				
Kategori Jawaban	Sangat Setuju	Setuju	Cukup	Tidak Setuju	Sangat Tidak Setuju

Skor	5	4	3	2	1
Frekuensi Jawaban	11	15	3	1	0
Jumlah Sampel	30				
Skor x Frekuensi Jawaban	55	60	9	2	0
Total Skor	126				
Persentase	$\frac{126}{30 \times 5} \times 100\% = 84\%$				
Hasil	Sangat Setuju				



Gambar 4.3 Hasil Pengujian Tiga

Dari gambar di atas, dapat dilihat bahwa terdapat total skor sebanyak 126 dengan persentase 84% responden yang mengaku aplikasi yang telah dibuat pada penelitian ini Sangat Setuju bahwa aplikasi dapat memberikan rekomendasi lagu berdasarkan aktivitas yang dilakukan pengguna.

4.2.4. Kesimpulan 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.

4.2.5. 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. Sebanyak 84% responden mengaku sangat setuju bahwa aplikasi ini dapat memberikan lagu sesuai dengan *mood* yang sedang dirasakan.
2. Sebanyak 82% responden mengaku sangat setuju bahwa aplikasi ini dapat memberikan lagu sesuai dengan lokasi yang dikunjungi.
3. Sebanyak 84% responden mengaku sangat setuju bahwa aplikasi ini dapat memberikan lagu sesuai dengan aktivitas yang sedang dikerjakan.