

BAB III

ANALISA DAN PERANCANGAN SOFTWARE

3.1. Analisa Kebutuhan *Software*

Aplikasi kamus Bahasa Indonesia-Jawa menggunakan metode *sequential search* merupakan media pembelajaran dalam bentuk *digital*, yang memanfaatkan teknologi informasi dan komunikasi, pada konsep kalimat tersebut memberikan manfaat ketersediaan materi saja yang bisa diakses kapanpun dan tampilan yang menarik.

Tujuan dari pembuatan aplikasi kamus Bahasa Indonesia-Jawa menggunakan metode *sequential search* ini adalah untuk mempermudah dalam penyajian informasi pembelajaran dan kemampuan untuk merealisasikan pembahasan ilmu kamus indonesia-jawa antara pelajar, mahasiswa serta masyarakat pada umumnya selain itu aplikasi kamus Bahasa Indonesia-Jawa menggunakan metode *sequential search* dapat dioperasikan kapanpun dan dimana pun. Salah satu unsur pokok yang harus diperhatikan dalam pembuatan aplikasi ini yaitu penyesuaian isi pembahasan, karena isi dari aplikasi ini diharapkan sesuai dengan apa yang diinginkan.

3.1.1. Identifikasi Masalah

Pada penulisan skripsi ini permasalahan yang akan di teliti dalam pembuatan aplikasi kamus Bahasa Indonesia-Jawa menggunakan metode *sequential search* adalah sebagai berikut :

1. Memberikan cara memahami Bahasa Jawa dengan menggunakan kamus Bahasa Indonesia-Jawa.
2. Menerapkan metode Sequential search untuk membangun aplikasi kamus Bahasa Indonesia-Jawa Berbasis Android.
3. Membangun Aplikasi kamus Bahasa Indonesia-Jawa Berbasis Android sebagai sarana untuk meningkatkan minat belajar bahasa jawa.

3.1.2 Analisa Kebutuhan Aplikasi

Dalam hal ini dilakukan analisa kebutuhan sesuai dengan kebutuhan sistem yang dibangun, analisa kebutuhan sistem terkait apa saja yang dibutuhkan agar sesuai dengan tujuan yang diharapkan. Selain menganalisa kebutuhan sistem, pada analisa ini juga akan menganalisa kebutuhan perangkat lunak dan perangkat keras dalam pembuatan aplikasi kamus Bahasa Indonesia-Jawa menggunakan metode *sequential search* berbasis android.

Berikut adalah analisa kebutuhan sistem, perangkat lunak dan perangkat keras dalam pembuatan aplikasi kamus Bahasa Indonesia-Jawa menggunakan metode *sequential search* berbasis android ini, adapun analisa sebagai berikut:

- **Analisa Kebutuhan Sistem**

Proses untuk mendapatkan pengetahuan kebutuhan dapat dilakukan dengan berbagai macam cara, yakni pengetahuan dari buku, jurnal ilmiah, internet. Dari Sumber pengetahuan tersebut dikumpulkan dan kemudian direpresentasikan kedalam program aplikasi.

- Analisa Perangkat Lunak

- a. Eclipse

Eclipse *Software* yang digunakan untuk membuat aplikasi android, membuat *user interface* dengan menggunakan *xml* sebagai *layout* kemudian memasukkan perintah untuk memanggil *user interface* yang telah dibuat dengan *program* dalam bahasa pemrograman *java*.

- b. *Android Development Tools* (ADT)

Android Development Tools (ADT) adalah *plugin* untuk Eclipse yang didesain untuk pengembangan aplikasi Android. ADT memungkinkan Eclipse untuk digunakan dalam membuat aplikasi Android baru, membuat *User Interface*, menambahkan komponen berdasarkan framework API Android, *debug* aplikasi, dan pemaketan aplikasi Android.

- c. Android 4.2 *Jelly Bean*

Android 4.2 *Jelly Bean* digunakan sebagai sistem operasi dari *handphone* android yang digunakan untuk pembuatan aplikasi sistem pakar ini.

- Analisa Perangkat Keras

- a. Komputer

Komputer yang digunakan untuk membuat aplikasi android, spesifikasi komputer yang di gunakan pada pembuatan aplikasi android ini adalah:

Prosesor : Intel(R) Core(TM) i5-5200U CPU @ 2.20Ghz-2.19Ghz

Memori : 4 GB DDR3

b. *Handphone* android

Pada pembuatan aplikasi ini, *handphone* android digunakan langsung untuk uji coba aplikasi, spesifikasi *handphone* android yang digunakan adalah sebagai berikut :

Prosesor : Qualcomm Snapdragon 610 1,80 GHz

Memori : 2 GB

3.2. Desain

3.2.1. Rancangan Algoritma

Algoritma *sequential search* adalah algoritma yang digunakan untuk mencari nilai pada sebuah array atau draft nilai dengan cara memeriksa satu per satu. pencarian linear atau yang dikenal sebagai *sequential search* bekerja dengan memeriksa setiap elemen dari sebuah *list* sampai sebuah kecocokan ditemukan pencarian dapat dilakukan pada barisan yang terurut secara menaik atau menurun ataupun tidak berurut. Pencarian *sequential* dapat dilakukan dengan cara membandingkan data yang dicari (X) dengan data dalam barisan $A[1] \dots A[n]$ dengan dimulai dari kata elemen pertama pada barisan A. Jika perbandingan bernilai sama, maka pencarian dihentikan dan dinyatakan sukses. Sedangkan apabila perbandingan tidak bernilai sama maka sebagai berikut:

1. Jika tidak terurut, maka pencarian akan dilanjutkan ke data selanjutnya.
2. Jika data terurut secara menaik, maka pencarian akan hanya dilakukan ke data selanjutnya yang berbeda disebelah kanan data yang sedang dibandingkan

apabila data yang dicari (X) lebih besar dari pada data yang sedang dibandingkan sekarang.

3. Jika data berurut secara menurun, maka pencarian hanya akan dilakukan sampai data yang dicari (X) lebih kecil dari data yang dibandingkan sekarang.

Cara kerja algoritma pencarian *sequential search* dapat dituliskan sebagai berikut.

```
int SequentialSearch(int x)
{
    int i = 0;
    bool ditemukan = false;
    while ((!ditemukan) && (i < Max))
    {
        if(Data[i] == x)
            ditemukan = true;
        else
            i++;
    }
    if(ditemukan)
        return i;
    else
        return -1;
}
```

Gambar III.1

Cara kerja Algoritma

Algoritma *sequential search* berdasarkan index yang telah ditentukan adalah sebagai berikut :

```

Input : M : kamusjawa, n ,1
Output : sampean
Proses
i ← 1
ketemu ← false
while ( i ≤ n ) and (not saya) do
if M [ i ]. Kata [ indoneisa ], kata[
jawa] then
saya ← true
else
i ← i + 1
endif
endwhile
{ i > n or saya }
if saya then { 1
ditemukan }
1 ← i
else { 1
tidak ditemukan }
1 ← -1
Endif

```

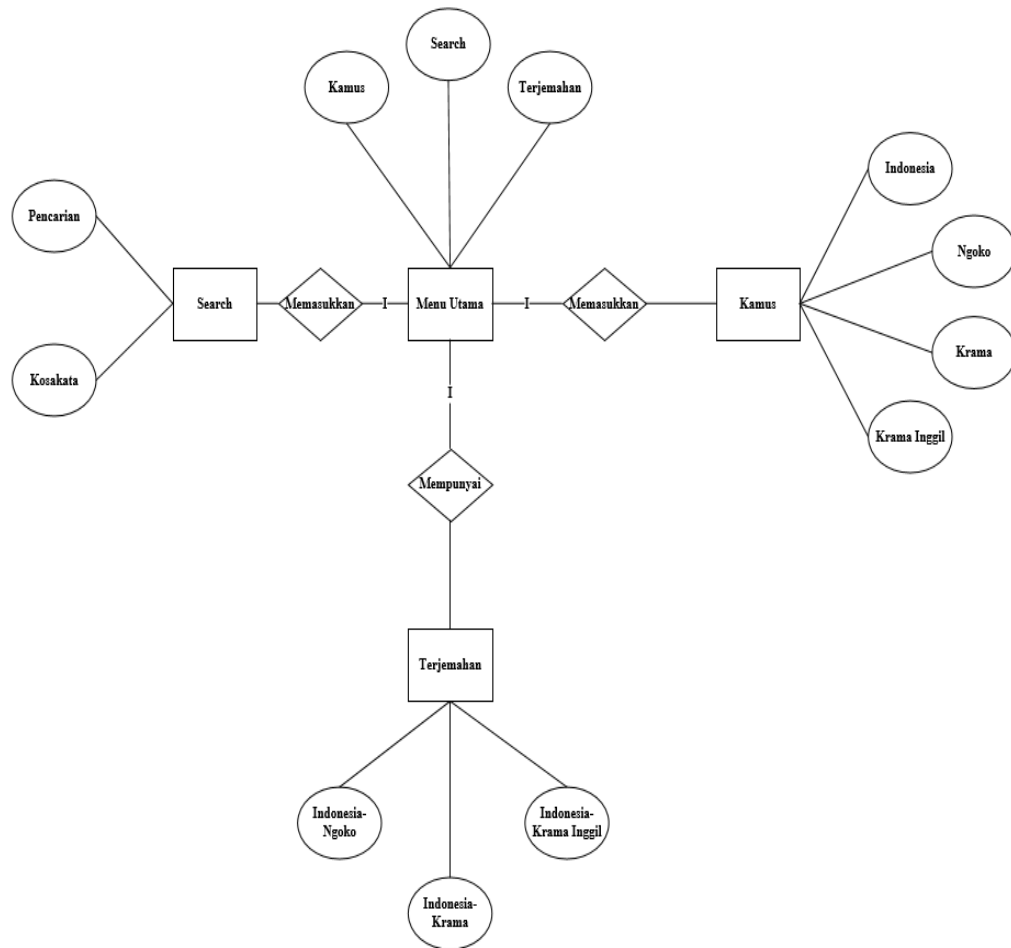
Gambar III.2

Index

3.2.2. Database

Entity Relationship Diagram (ERD)

ERD dalam aplikasi pembelajaran ilmu biologi berbasis android sebagai berikut:



Gambar III.3

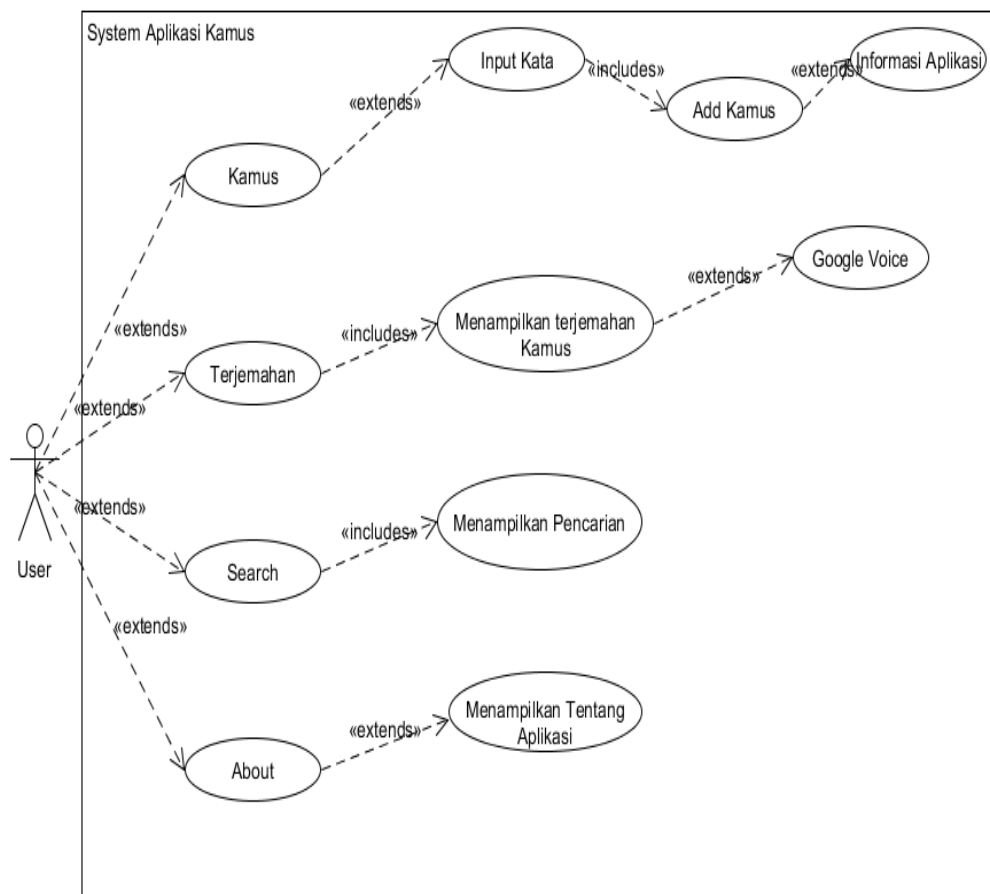
ERD aplikasi pembuatan kamus Bahasa Indonesia Bahasa Jawa.

3.2.3 Software Architecture

1. Use Case Diagram

Use Case Diagram sangat sesuai untuk menggambarkan siapa saja aktor yang terlibat dan fungsi-fungsi apa saja yang bisa dilakukan dalam sebuah aplikasi.

Aktor dan fungsi-fungsi yang dirancang aplikasi ini adalah sebagai berikut:



Gambar III.4

Use Case Diagram

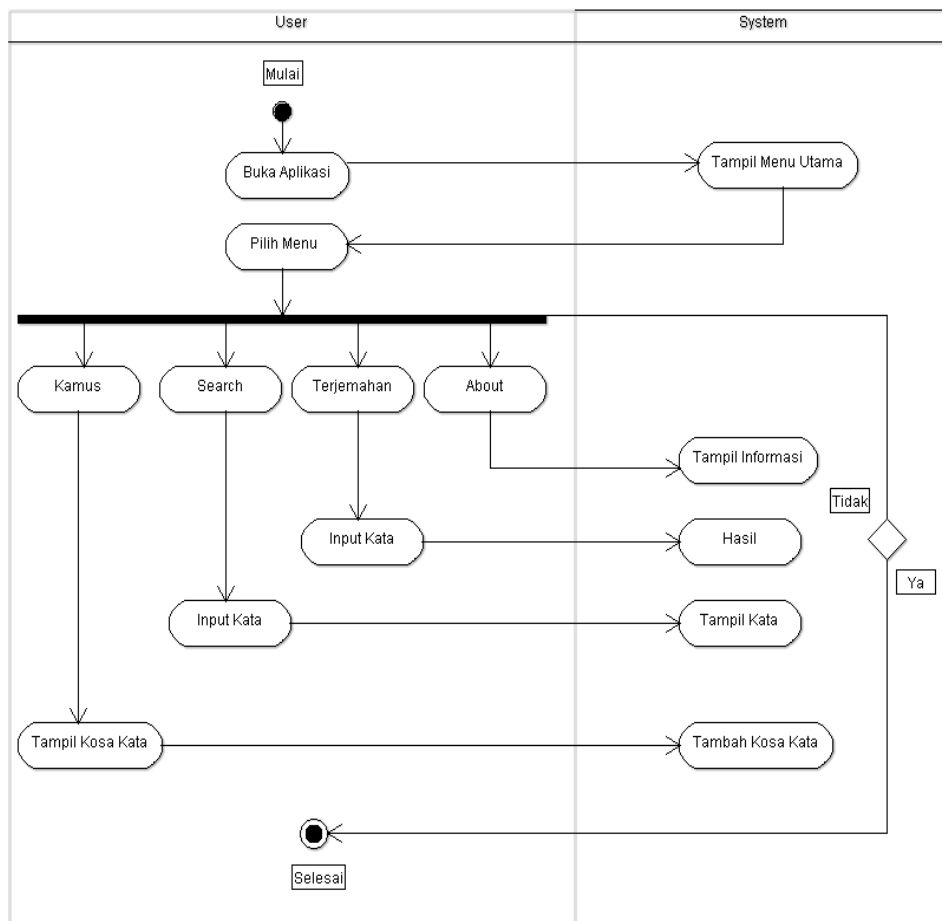
Pada gambar diatas dapat dilihat pengguna aplikasi sebagai aktor dapat melihat tampilan menu utama, Kamus, Terjemahan, *Search*, *About*.

1. Activity Diagram

Diagram ini menunjukkan langkah-langkah dalam aliran kerja, titik-titik keputusan dalam aliran kerja, siapa yang bertanggung jawab menyelesaikan masing-masing aktivitas dan objek-objek yang digunakan dalam aliran kerja.

Diagram rancangan dari aplikasi ini adalah sebagai berikut:

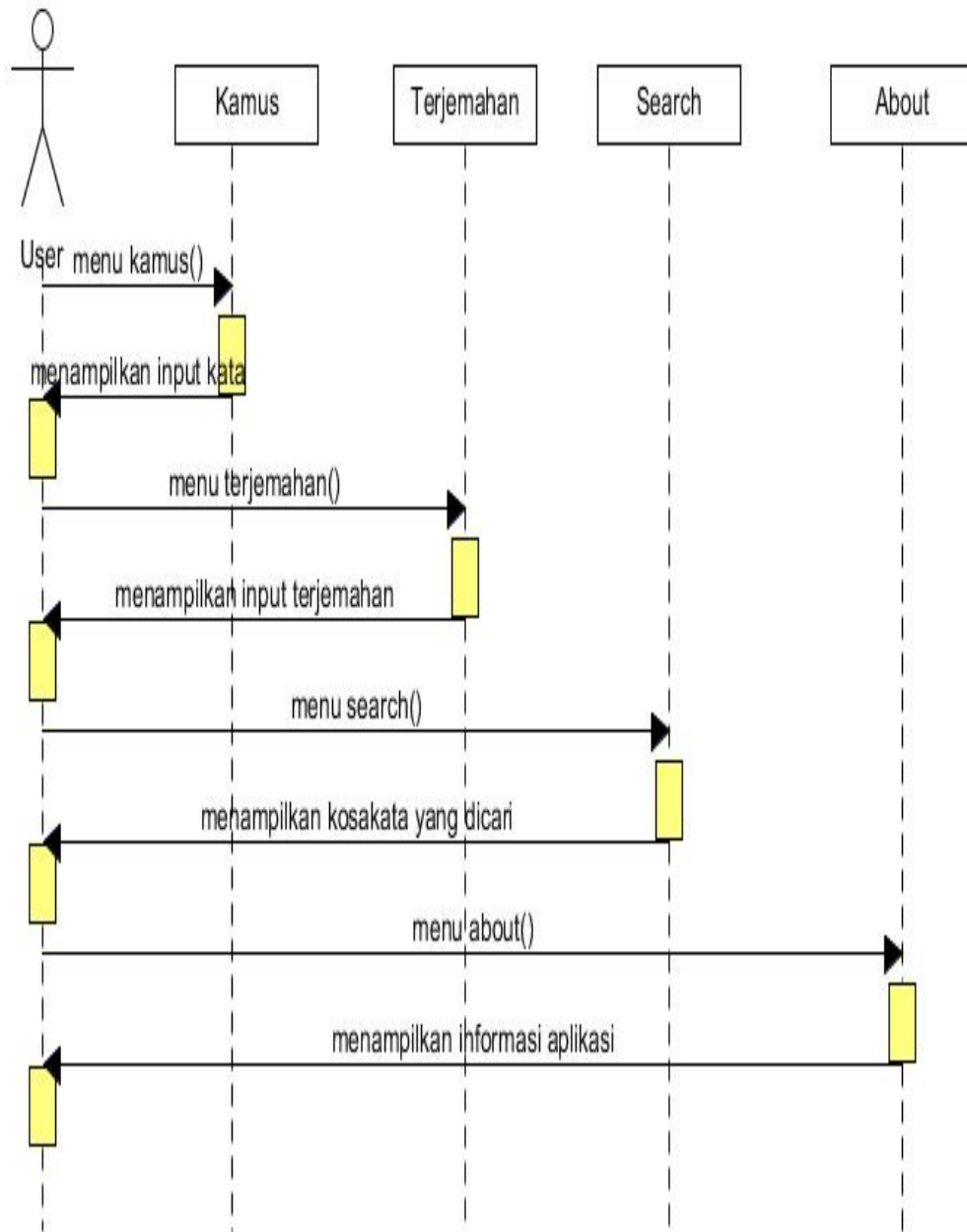
a. Activity Diagram



Gambar III.5

Activity Diagram.

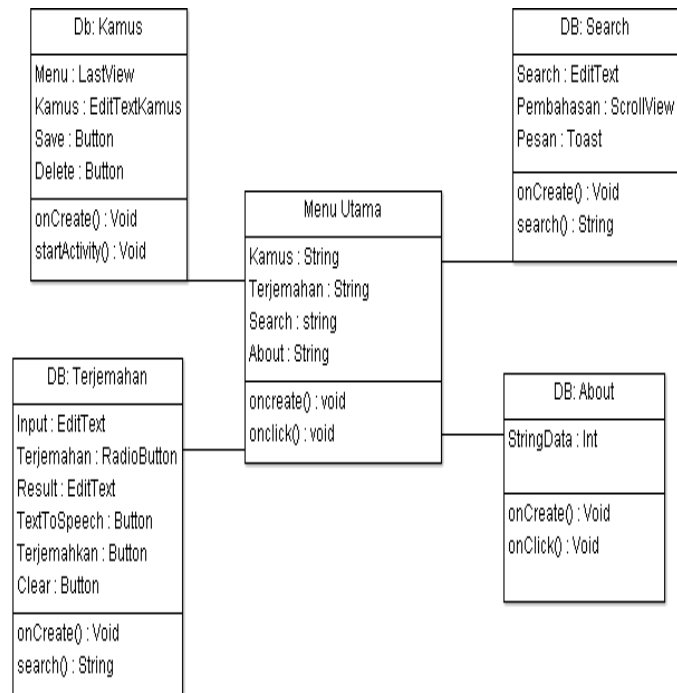
b. Activity Diagram Sequential



Gambar III.6

Gambar Activity Diagram Sequential

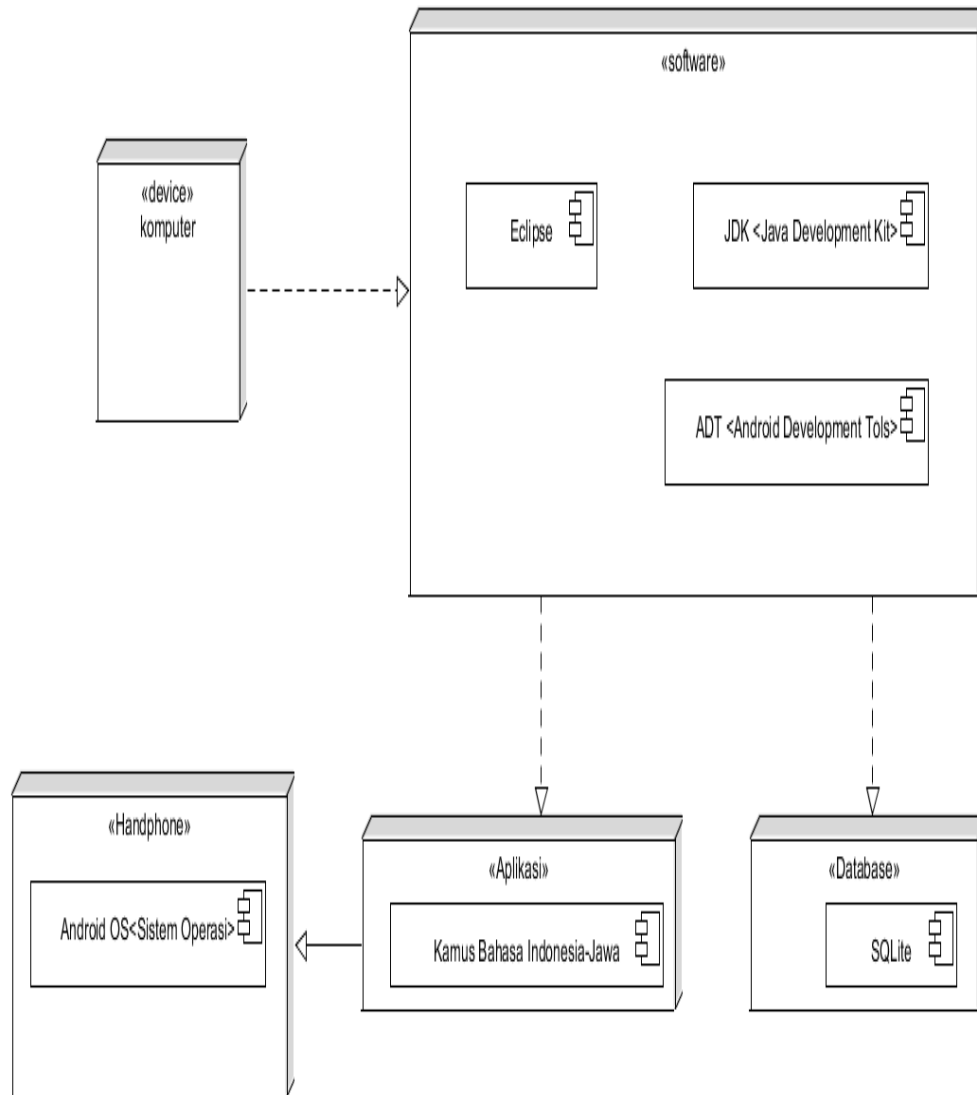
C. Class Diagram



Gambar III.7

Gambar *Class Diagram*

2. Deployment Diagram



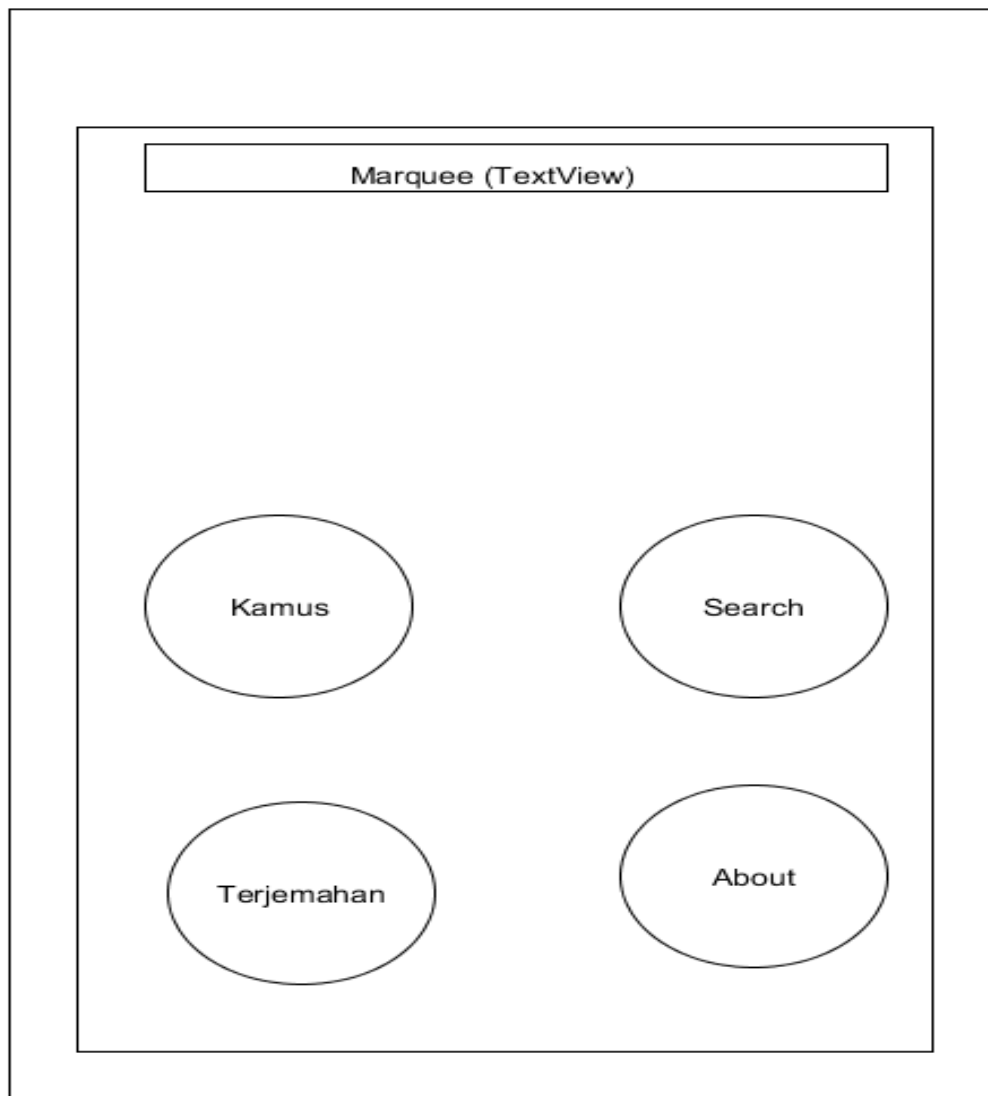
Gambar III.8

Gambar *Deployment Diagram*

3.2.4. User Interface

Dalam perancangan desain antar muka aplikasi ini, secara garis besar utama sebagai berikut:

1. Tampilan Rancangan Halaman Menu Utama



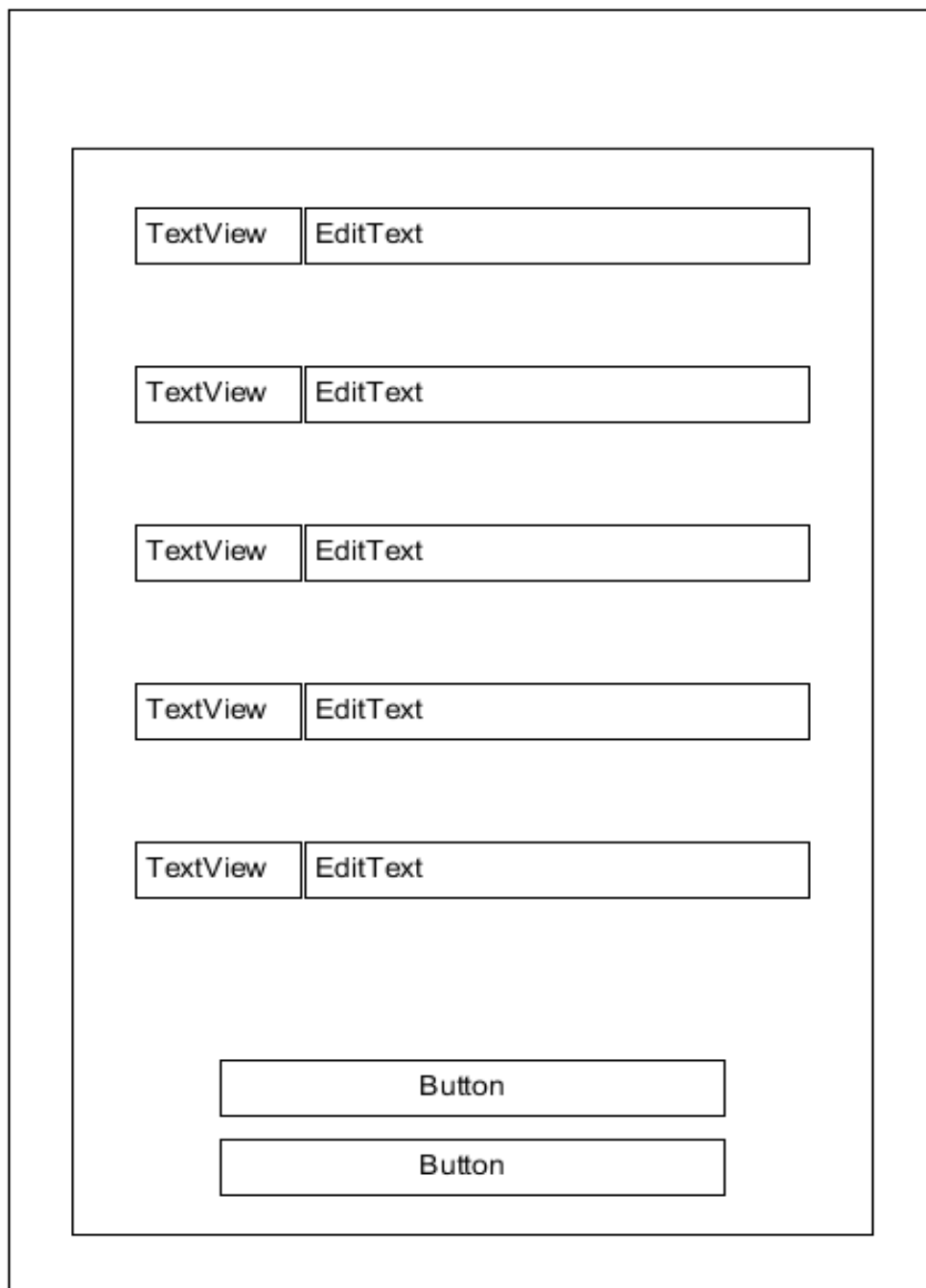
Gambar III.9

Gambar Rancangan Halaman Menu

Deskripsi dari rancangan tampilan menu utama, yaitu:

1. Teks Berjalan atau *Text Marquee* berfungsi menampilkan informasi aplikasi, tanggal dan jam yang menunjukkan waktu sekarang.
2. *Image View* Kamus digunakan untuk menambahkan atau menginput sebuah kata atau kamus Indonesia-jawa.
3. *Image View Search* yaitu untuk mencari sebuah kata yang sudah ada.
4. *Image View* Terjemahan berfungsi untuk menginput atau menampilkan kata yang dicari yang mana ketika *user* mencari sebuah kata atau kalimat yang sudah di terjemahkan.
5. *Image View About* berisikan informasi tentang aplikasi dan cara penggunaannya.

2. Tampilan Rancangan Halaman Kamus



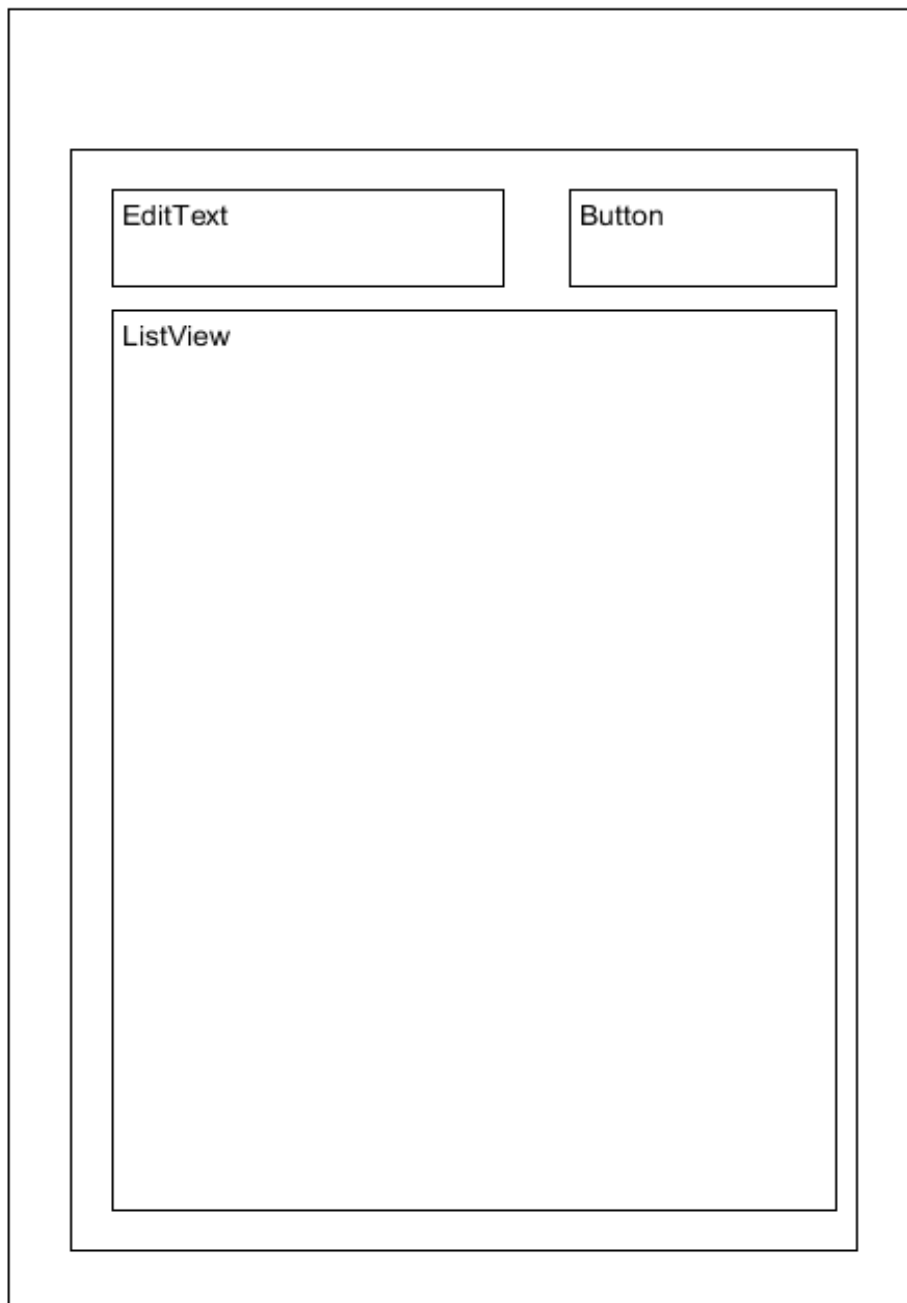
Gambar III.10

Gambar Rancangan Halaman Kamus

Deskripsi dari tampilan rancangan halaman *add* kamus di atas yaitu :

- a. *TextView* berupa isi judul kamus indonesia
- b. *EditText* yang akan ditambahkan berdasarkan bahasa
- c. *TextView* berupa jawa ngoko
- d. *EditText* yang akan ditambahkan berdasarkan bahasa
- e. *TextView* berupa krama
- f. *EditText* yang akan ditambahkan berdasarkan bahasa
- g. *TextView* berupa krama inggil
- h. *EditText* yang akan ditambahkan berdasarkan bahasa
- i. *TextView* berupa *keterangan Bahasa*
- j. *EditText* yang akan ditambahkan soal keterangan kata bahasa
- k. *Button save* digunakan untuk menyimpan
- l. *Button delete* digunakan untuk menghapus

3. Tampilan Rancangan Halaman *Search*



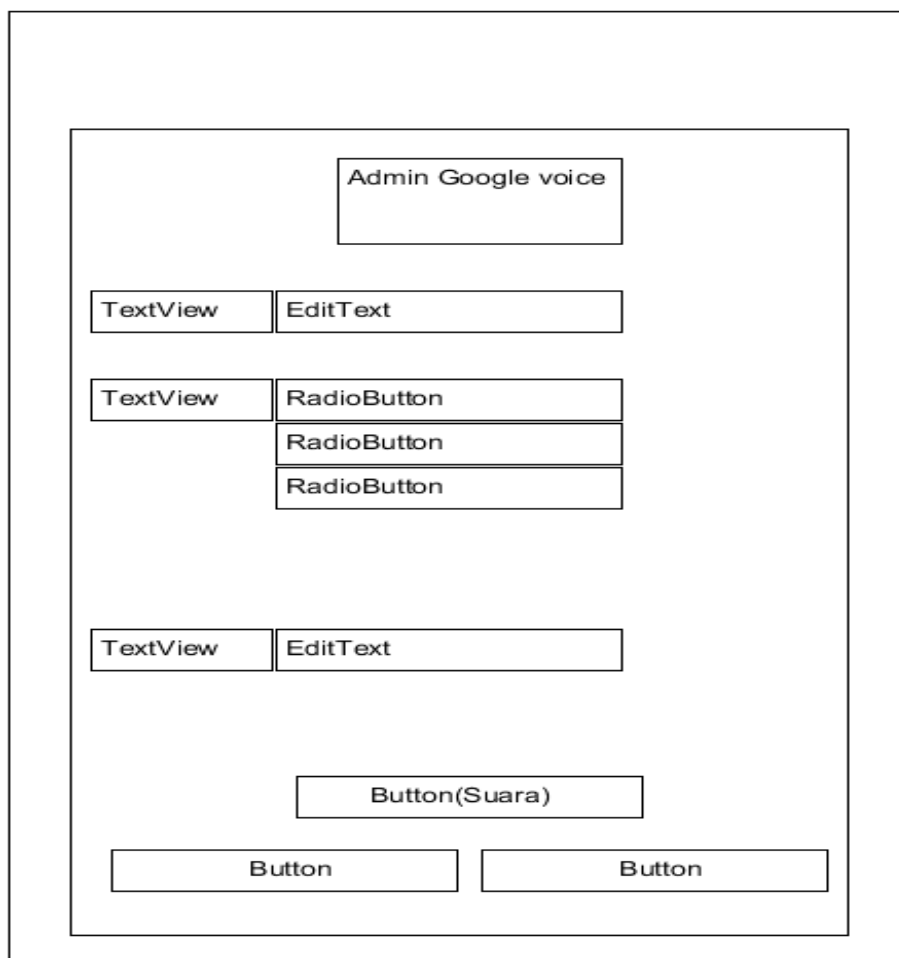
Gambar III.11

Gambar Rancangan Halaman *Search*

Deskripsi dari tampilan rancangan halaman *search* di atas yaitu :

- a. *EditText* digunakan menginput pencarian sebuah kata
- b. *Button search* untuk melakukan pencarian kamus istilah
- c. *ListView* berupa berisi data kamus Indonesia-jawa

4. Tampilan Rancangan Halaman Terjemahan



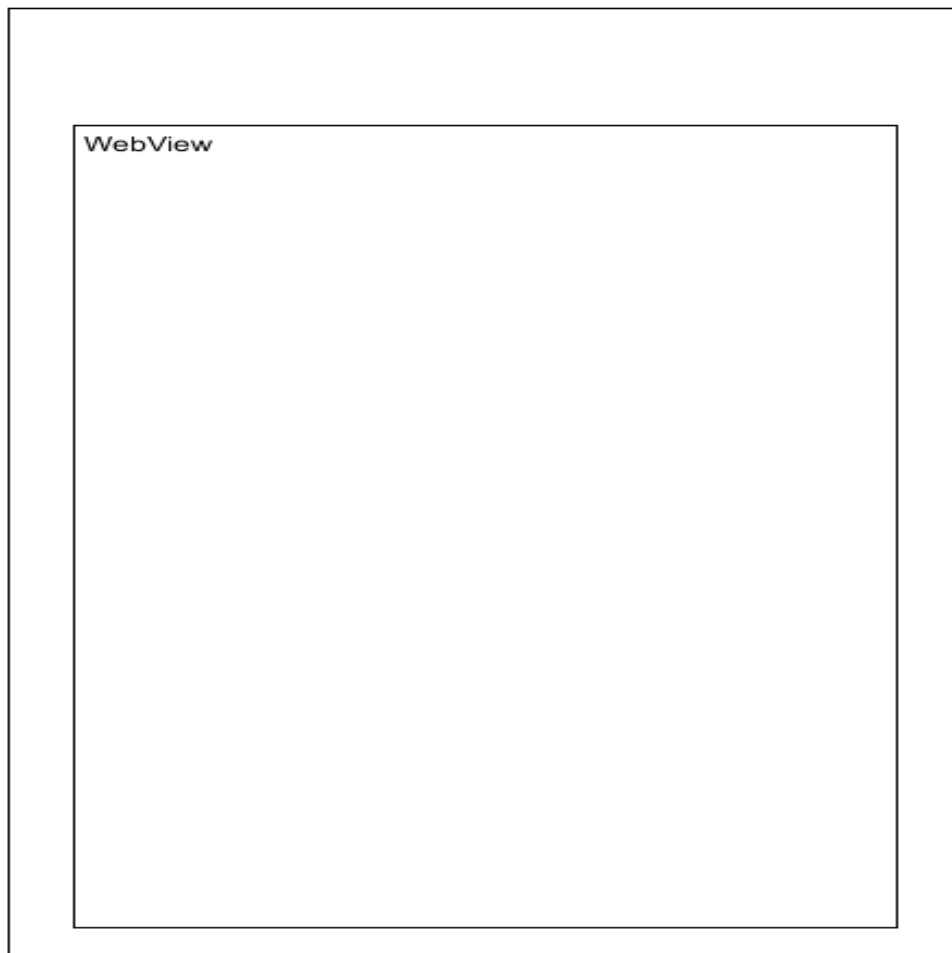
Gambar III.12

Gambar Rancangan Halaman Terjemahan

Deskripsi dari tampilan rancangan halaman terjemahan di atas yaitu :

1. Admin Google *Voice* untuk mengetahui dari masukan suara Bahasa kita yang mau di *input*
2. *TextView* berupa *input* kamus
3. *EditText* berupa kata yang kita *input* manual maupun dari google *voice* tersebut
4. *TextView* berupa Terjemahkan.
5. *RadioButton* berfungsi untuk memilih kamus yang mau kita terjemahkan
6. *TextView* berupa *result*.
7. *EditText* berupa teks yang kita cari secara otomatis.
8. *Button* (Suara) berupa suara yang kita terjemahkan melalui google *voice* ataupun manual kita ketik.
9. *Button* Terjemahkan digunakan untuk menerjemahkan dan yang mau di *translate*
10. *Button Clear* digunakan untuk menghapus

5. Tampilan Rancangan Halaman *About*



Gambar III.13

Gambar Rancangan Halaman *About*

Deskripsi dari tampilan rancangan halaman *about* di atas yaitu :

WebView berupa tampilan informasi aplikasi dan cara penggunaanya

3.3. Implementasi

Implementasi merupakan implementasi hasil dari desain yang telah dirancang sebelumnya. Berikut ini adalah implementasi dari sistem yang telah dibuat yaitu sebagai berikut:

3.3.1. Desain Halaman *Menu* Kamus Bahasa Indonesia-Bahasa Jawa



Gambar III.14

Gambar Halaman *Menu* Kamus Bahasa Indonesia-Bahasa Jawa

Desain halaman menu kamus Bahasa Indonesia-Bahasa Jawa disajikan pada gambar diatas, terdiri dari 6 bagian utama, yaitu:

1 Bar Atas

Bagian bar atas yang berada di posisi paling atas terdiri dari logo dan informasi aplikasi.

2 Teks Berjalan atau *Text Marquee*

Menampilkan informasi aplikasi, tanggal dan jam yang menunjukkan waktu sekarang.

3 *Image View* Kamus

Digunakan untuk menambahkan atau menginput sebuah kata atau kalimat istilah kamus jawa.

4 *Image View* Terjemahan

Berfungsi untuk melihat daftar sebuah kata atau istilah yang sudah di tambahkan dan juga bias menambahkan kata atau kalimat pada sebuah *toolbar*.

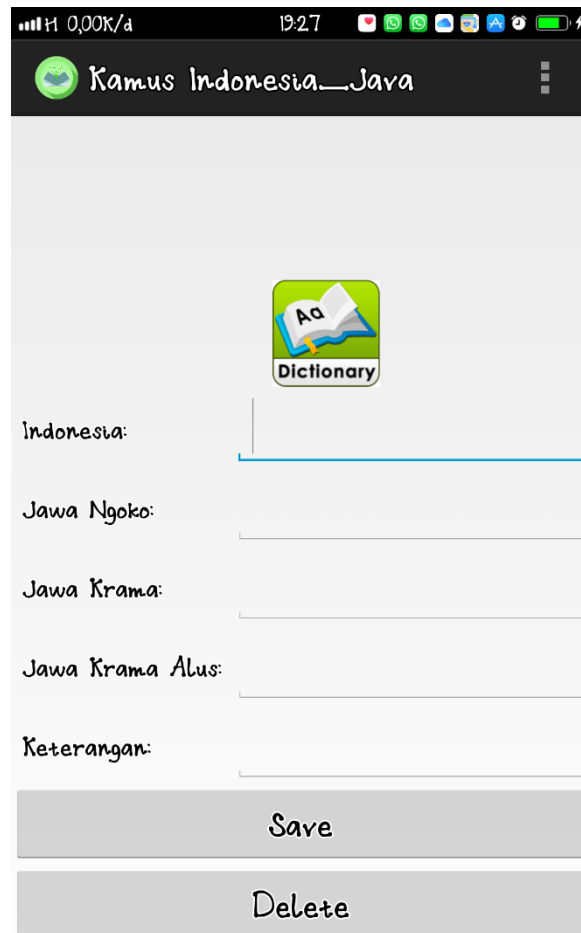
5 *Image View* Search

Digunakan untuk mencari sebuah kalimat atau kata istilah.

6 *Image View* About

Berisikan informasi tentang aplikasi dan cara penggunaanya.

3.3.2. Desain Halaman *Add* Kamus



Gambar III.15

Gambar Halaman *Add*

Desain halaman *add* disajikan dengan lima buah *Text View* untuk mengisi teks atau kalimat istilah kamus dan *Button Save* untuk menyimpan teks atau kalimat kamus dan *Button Delete* untuk menghapus halaman yang kita masukan.

3.3.3. Desain Halaman *Search*

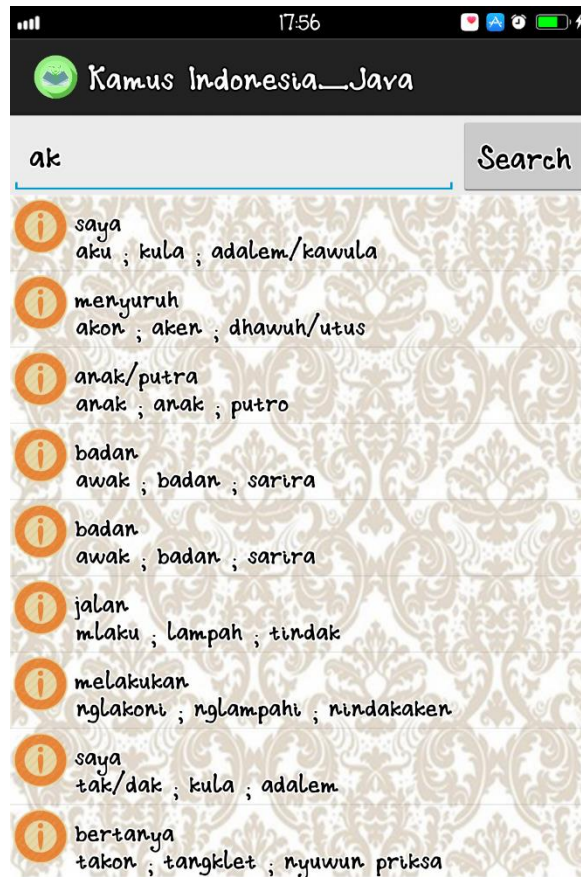


Gambar III.16

Gambar Halaman Cari

Desain Halaman *Search* berisi *Edit Text* untuk mencari sebuah teks atau kalimat kamus Indonesia-jawa dan *List View Database* kamus Indonesia-jawa.

3.3.4. Desain Halaman Hasil Search



Gambar III.17

Gambar Halaman Hasil Cari

Desain Halaman Hasil Cari menampilkan hasil dari pencarian teks atau kalimat yang dicari.

3.3.5. Desain Halaman Terjemahan

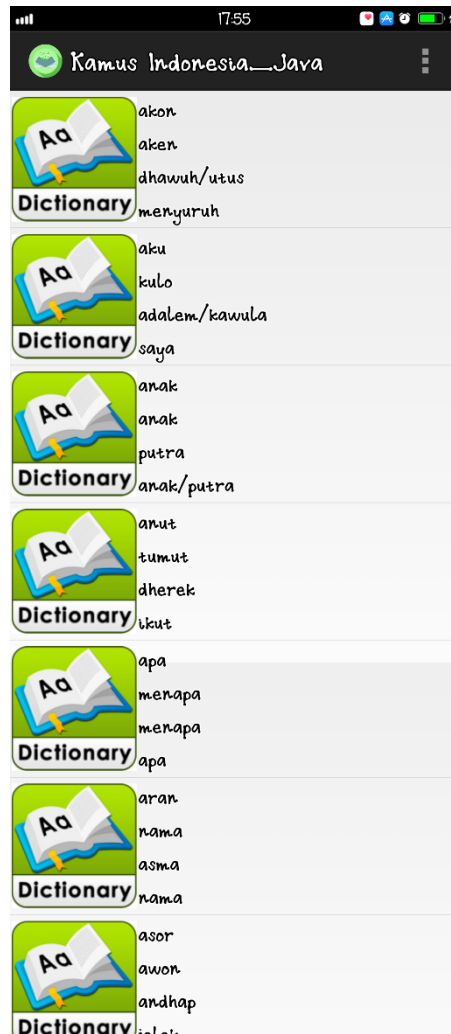


Gambar III.18

Gambar Halaman Terjemahan

Desain halaman terjemahan untuk menyimpan atau menampilkan kata yang mana ketika user mencari sebuah kata atau kalimat menekan tombol terjemahan maka akan terlihat dan menampilkan *voice* google yang di perintah oleh google.

3.3.6. Desain Halaman *List Kamus* atau *Database*

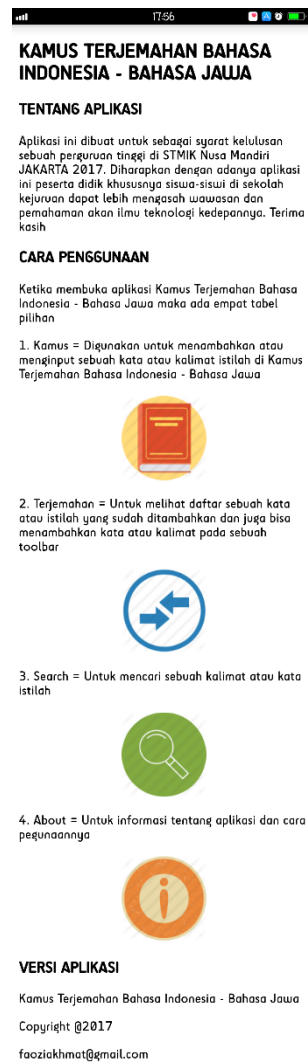


Gambar III.19

Gambar Halaman List Kamus

Desain halaman *List Kamus* untuk melihat daftar sebuah kata atau istilah yang sudah ditambahkan dan juga bisa menambahkan kata atau kalimat pada sebuah *toolbar* pojok kanan atas.

3.3.7. Desain Halaman *About*



Gambar III.20

Gambar Halaman *About*

Desain halaman *About* disajikan berupa tentang aplikasi menampilkan informasi aplikasi dan referensi kamus Indonesia-jawa, cara penggunaan berisi kosakata aplikasi.

3.3.8. Desain Halaman *Exit*



Gambar III.21

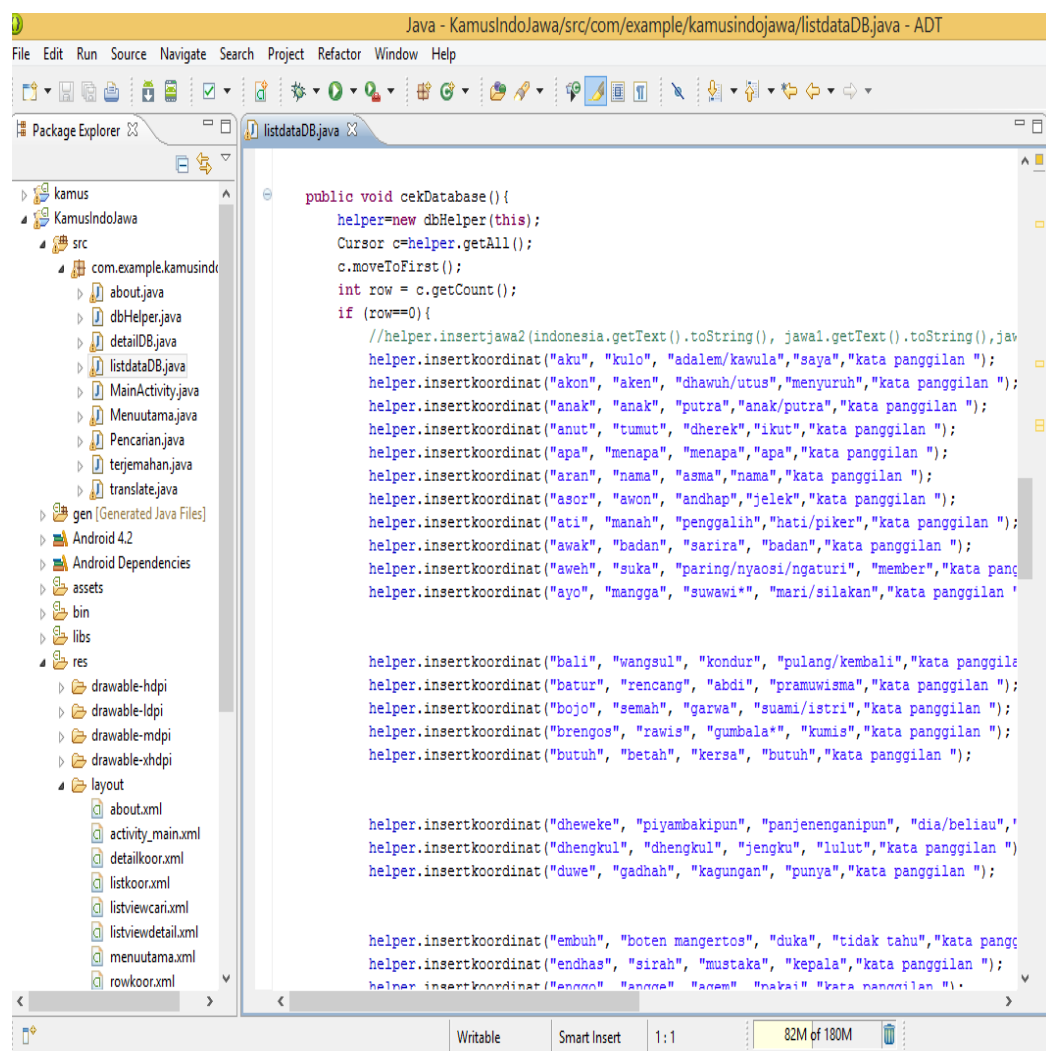
Gambar Halaman *Exit*

Desain Halaman *Exit* untuk keluar dari kamus Indonesia-Jawa

3.3.8.1. Pembuatan *database*

Seperti yang telah di jelaskan sebelumnya pada desain *database*, aplikasi kamus Bahasa Indonesia-Jawa ini menggunakan data array yang di simpan dalam *class* listdataDB.java sebagai data utama istilah kamus dan menggunakan *database* *Sqlite*

1. Berikut struktur *sourch code Database*



```

Java - KamusIndoJawa/src/com/example/kamusindojawa/listdataDB.java - ADT
File Edit Run Source Navigate Search Project Refactor Window Help
Package Explorer
kamus
  KamusIndoJawa
    src
      com.example.kamusindojawa
        about.java
        dbHelper.java
        detailDB.java
        listdataDB.java
        MainActivity.java
        Menuutama.java
        Pencarian.java
        terjemahan.java
        translate.java
      gen [Generated Java Files]
      Android 4.2
      Android Dependencies
      assets
      bin
      libs
      res
        drawable-hdpi
        drawable-ldpi
        drawable-mdpi
        drawable-xhdpi
        layout
          about.xml
          activity_main.xml
          detailkoor.xml
          listkoor.xml
          listviewcari.xml
          listviewdetail.xml
          menuutama.xml
          rowkoor.xml
listdataDB.java
public void cekDatabase(){
    helper=new dbHelper(this);
    Cursor c=helper.getAll();
    c.moveToFirst();
    int row = c.getCount();
    if (row==0){
        //helper.insertjawa2(indonesia.getText().toString(), jawa1.getText().toString(),jawa2.getText().toString());
        helper.insertkoordinat("aku", "kulo", "adalem/kawula","saya","kata panggilan ");
        helper.insertkoordinat("akon", "aken", "dhawuh/utus","menyuruh","kata panggilan ");
        helper.insertkoordinat("anak", "anak", "putra","anak/putra","kata panggilan ");
        helper.insertkoordinat("anut", "cumut", "dherek","ikut","kata panggilan ");
        helper.insertkoordinat("apa", "menapa", "menapa","apa","kata panggilan ");
        helper.insertkoordinat("aran", "nama", "asma","nama","kata panggilan ");
        helper.insertkoordinat("asor", "awon", "andhap","jelek","kata panggilan ");
        helper.insertkoordinat("ati", "manah", "penggalih","hati/piker","kata panggilan ");
        helper.insertkoordinat("awak", "badan", "sarira", "badan","kata panggilan ");
        helper.insertkoordinat("aweh", "suka", "paring/nyaosi/ngaturi", "member","kata panggilan ");
        helper.insertkoordinat("ayo", "mangga", "suwawi", "mari/silakan","kata panggilan ");

        helper.insertkoordinat("bali", "wangsul", "kondur", "pulang/kembali","kata panggilan ");
        helper.insertkoordinat("batur", "rencang", "abdi", "pramuwisma","kata panggilan ");
        helper.insertkoordinat("bojo", "semah", "garwa", "suami/istri","kata panggilan ");
        helper.insertkoordinat("bregos", "rawis", "gumbala*", "kumis","kata panggilan ");
        helper.insertkoordinat("butuh", "betah", "kersa", "butuh","kata panggilan ");

        helper.insertkoordinat("dheweke", "piyambakipun", "panjenenganipun", "dia/beliau","kata panggilan ");
        helper.insertkoordinat("dhengkul", "dhengkul", "jengku", "lulut","kata panggilan ");
        helper.insertkoordinat("duwe", "gadhadh", "kagungan", "punya","kata panggilan ");

        helper.insertkoordinat("embuh", "boten mangertos", "duka", "tidak tahu","kata panggilan ");
        helper.insertkoordinat("endhas", "sirah", "mustaka", "kepala","kata panggilan ");
        helper.insertkoordinat("enggo", "enggo", "agan", "nakai","kata panggilan ");
    }
}

```

Gambar III.22

Sourch Code Database

2. Berikut struktur *source code* dbHelper

```

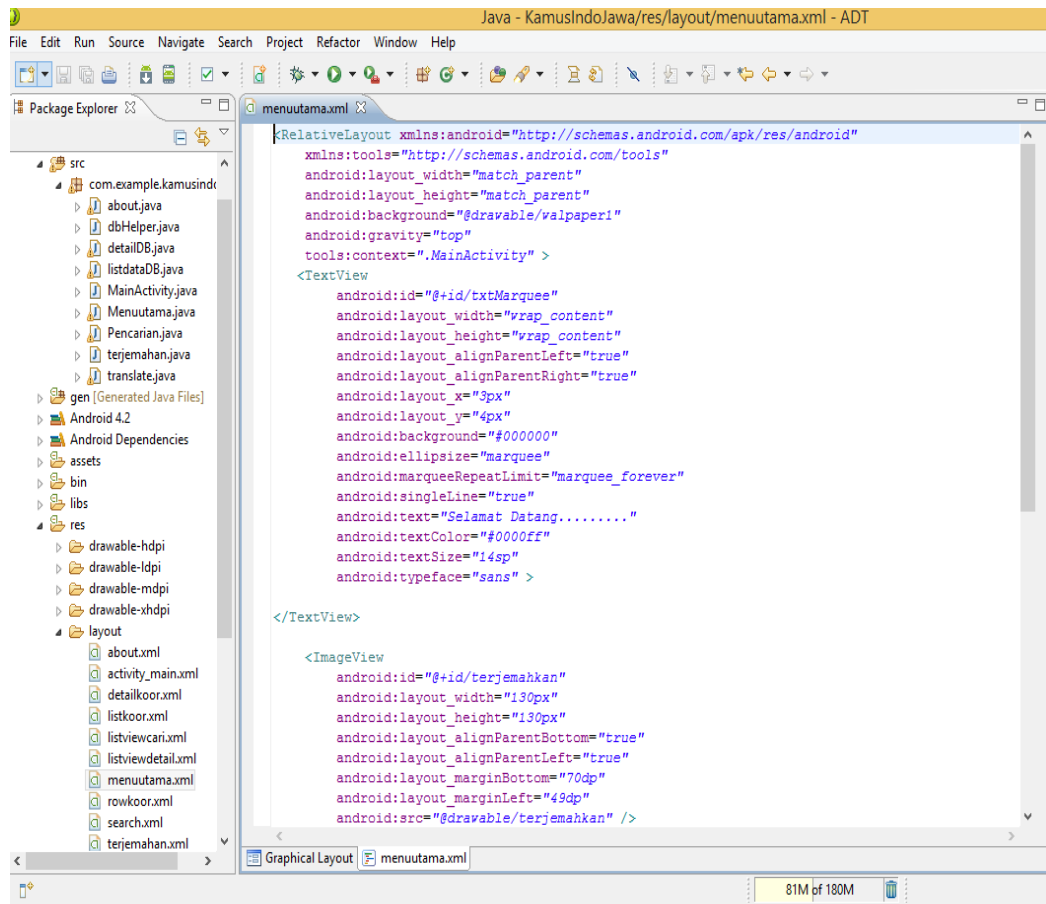
Java - KamusIndoJawa/src/com/example/kamusindojava/dbHelper.java - ADT
File Edit Run Source Navigate Search Project Refactor Window Help
Package Explorer
  kamus
  KamusIndoJawa
    src
      com.example.kamusindo
        about.java
        dbHelper.java
        detailDB.java
        listdataDB.java
        MainActivity.java
        Menuutama.java
        Pencarian.java
        terjemahan.java
        translate.java
      gen [Generated Java Files]
      Android 4.2
      Android Dependencies
      assets
      bin
      libs
      res
        drawable-hdpi
        drawable-ldpi
        drawable-mdpi
        drawable-xhdpi
        layout
          about.xml
          activity_main.xml
          detailkoor.xml
          listkoor.xml
          listviewcari.xml
          listviewdetail.xml
          menuutama.xml
          rowkoor.xml
  listdataDB.java
  dbHelper.java
@Override
public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion){}
public Cursor getAll(){
    return (getReadableDatabase().rawQuery("SELECT _id, indonesia, jawa1, jawa2, jawa3, ket
})
public Cursor getCount(){
    return (getReadableDatabase().rawQuery("SELECT COUNT(*) as row FROM "+tb_koordinat+",
})
public Cursor getBy_id(String id){
    String[] args={id};
    return (getReadableDatabase().rawQuery("SELECT _id, indonesia, jawa1, jawa2, jawa3, ket
})
//insertkoordinat
public void insertkoordinat(String indonesia, String jawa1, String jawa2, String jawa3, Str
    ContentValues cv=new ContentValues();
    cv.put("indonesia", indonesia);
    cv.put("jawa1", jawa1);
    cv.put("jawa2", jawa2);
    cv.put("jawa3", jawa3);
    cv.put("keterangan", keterangan);
    getWritableDatabase().insert(tb_koordinat,"indonesia", cv);
})
public void updatekoordinat (String id, String indonesia, String jawa1, String jawa2, Strir
    ContentValues cv=new ContentValues();
    String[] args={id};
    cv.put("indonesia", indonesia);
    cv.put("jawa1", jawa1);
    cv.put("jawa2", jawa2);
    cv.put("jawa3", jawa3);
    cv.put("keterangan", keterangan);
    getWritableDatabase().update(tb_koordinat,cv,"_id=?", args);
})
public void deletekoordinat (String id){
    String[] args={id};
    getWritableDatabase().delete(tb_koordinat,"_id=?", args);
})
Writable Smart Insert 1:1 68M of 180M

```

Gambar III.23

Source Code dbHelper

3. Berikut struktur *sourch code* menu utama.xml



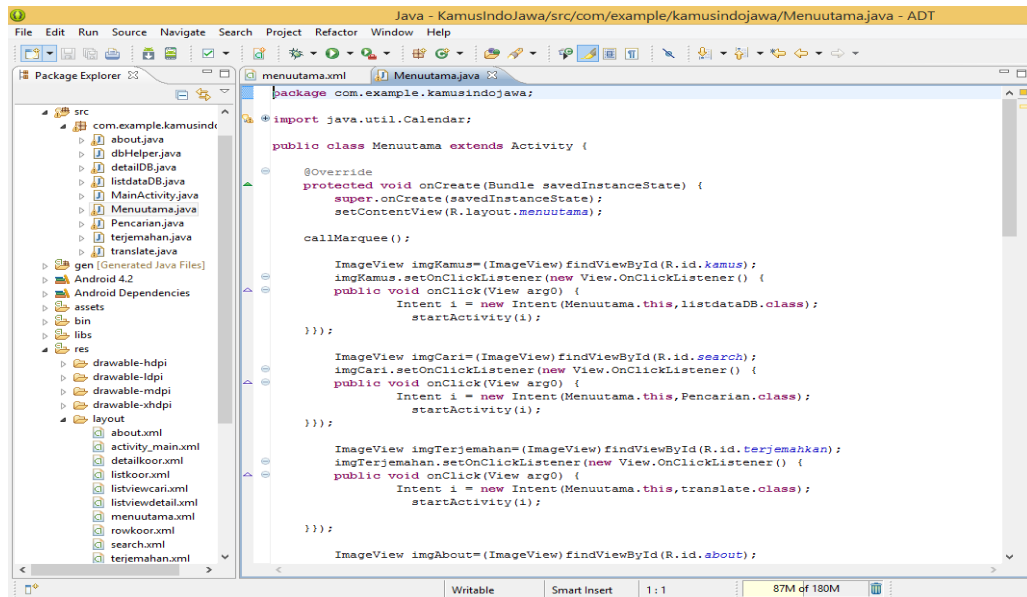
Gambar III.24

Sourch Code Menu Utama.xml

Penjelasan:

Pada *menuutama.xml* terdapat perintah *sourch code button*, yang mana urutan tersebut merupakan perintah menampilkan teks berjalan atau *Text Marquee*, memasukkan ukuran *button* pada layar, warna *button*, posisi penempatan *button* pada layar, dan id yang digunakan untuk pemanggilan atau didaftarkan di *Menuutama.java*.

4. Berikut struktur *sourch code* aplikasi yang terdapat pada *Menuutama.java*



Gambar III.25

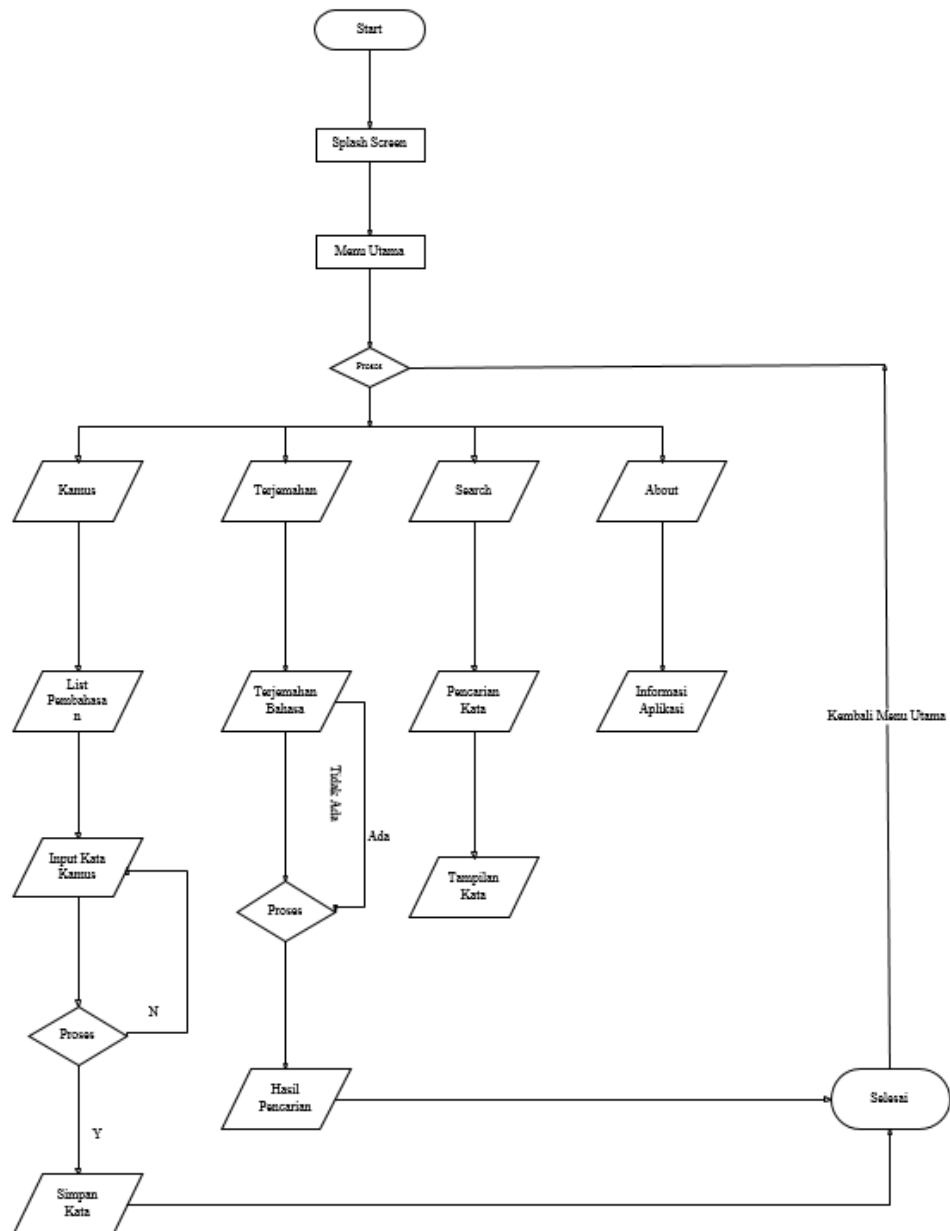
Sourch Code Menuutama.java

Penjelasan:

Pada *Menuutama.java* bisa kita lihat pada *sourch code* ini adalah perintah untuk masuk ke *activity* yang baru, maksudnya ketika dalam aplikasi kita menekan salah satu tombol *button*, maka *user* akan di suguhkan dengan tampilan lainnya seperti pada Gambar III.

2. *Flowchart*

Flowchart dari perancangan program aplikasi kamus Indoneia-Jawa adalah sebagai berikut :



Gambar III.26

Gambar *Flowchart*

3.4. Testing

Pengujian *Black Box* Aplikasi Kamus Indonesia-Jawa adalah sebagai berikut :

Tabel III.1

Pengujian *Black Box* Aplikasi Kamus Indonesia-Jawa.

No.	Rancangan Proses	Hasil Yang Diharapkan	Hasil
1.	Masuk Ketampilan Utama	Memilih button yang akan dipilih	Sesuai
2.	<i>Button Image</i> Kamus	Menampilkan kata atau kalimat jawa	Sesuai
3.	Pembahasan	Menampilkan kata-kata atau terjemahan kamus	Sesuai
4.	Input kata Kamus	Menambahkan sebuah kata atau istilah kalimat baru	Sesuai
5.	Menampilkan toast informasi	Menampilkan Informasi Aplikasi	Sesuai
6.	<i>Save</i>	Untuk menyimpan kata atau kalimat istilah	Sesuai
7.	<i>Delete</i>	Untuk membatalkan atau delete dari sebuah kata atau kalimat	Sesuai
8.	<i>Button Image</i> Kamus <i>Search</i>	Menampilkan sebuah database	Sesuai
9.	Input kata Kamus Pencarian	Mencari sebuah kata atau kalimat	Sesuai
10.	<i>Image View</i> Kamus Pencarian	Untuk menampilkan list pencarian kata	Sesuai
11.	<i>Image View</i> Kamus Terjemahan	Untuk menampilkan kata yang di input dan terjemahan	Sesuai
12.	<i>Button Image</i> <i>google voice</i>	Untuk input suara dan terjemahan	Sesuai
13.	<i>Button Image</i> <i>View List</i> Terjamahan	Menampilkan database kamus Indonesia-jawa	Sesuai
14.	<i>Button Image</i> <i>View About</i>	Menampilkan informasi tentang Aplikasi	Sesuai

3.5. *Support*

Dalam pengujian pemasangan aplikasi sistem pakar diagnosa kerusakan komputer ini menggunakan sebuah *handphone* android dengan spesifikasi perangkat keras (*hardware*) dan perangkat lunak (*software*) sebagai berikut:

Tabel III.2

Spesifikasi Perangkat Keras *Handphone* android

PERANGKAT KERAS	KETERANGAN
Prosesor	Quad Core 1.2 GHz ARM Cortex A7
RAM	1GB
Memory Internal	4 GB

Tabel III.3

Spesifikasi Perangkat Lunak *Handphone* android

PERANGKAT LUNAK	KETERANGAN
Sistem Operasi	Android <i>JellyBean</i> 4.2

Pada saat pengujian, penulis menggunakan *handphone* android langsung tanpa menggunakan *emulator*, alasan penulis langsung melakukan pengujian pada *handphone* android dan tanpa menggunakan *emulator* adalah dikarena ketika menggunakan *emulator* untuk aplikasi android, *emulator* akan memakan banyak memori pada laptop/komputer dan biasanya akan membuat hang laptop/komputer, untuk itu penulis langsung melakukan pengujian ke *handphone* android nyata.