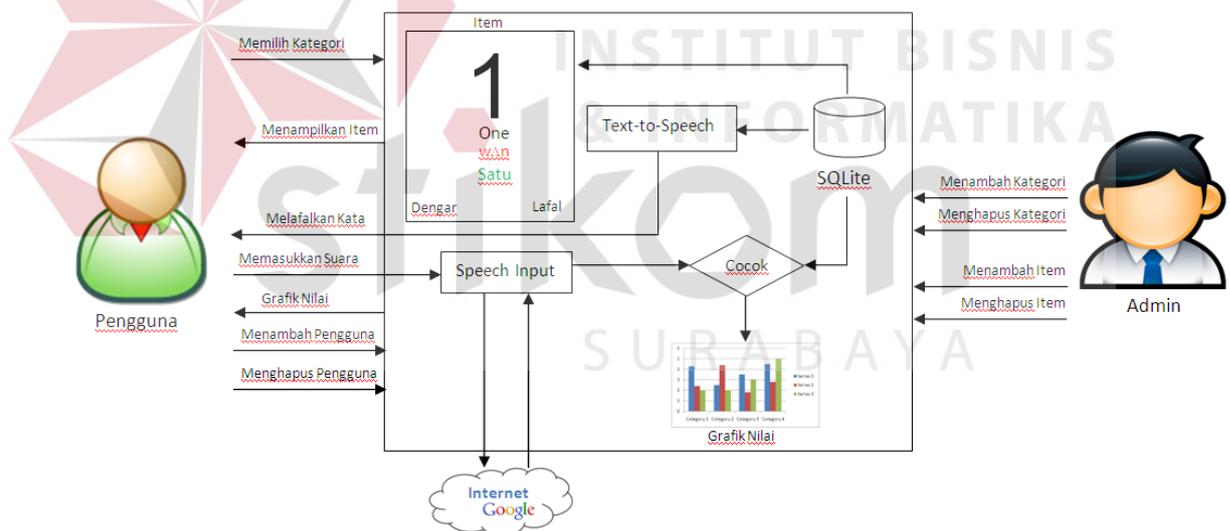


## BAB III

### PERANCANGAN SISTEM

#### 3.1. Gambaran Umum Sistem

Aplikasi *mobile* yang akan dikembangkan ini nantinya dapat membantu para pemula agar dapat belajar melafalkan bahasa Inggris dasar secara mandiri. Dalam pembuatan Aplikasi *Mobile* Belajar Melafalkan Bahasa Inggris dibutuhkan fitur *Text-to-Speech* dan *Speech Input* yang dapat melafalkan serta mencocokkan suara pengguna dengan sebuah kata bahasa Inggris yang benar pada *database*. Dalam hal ini, fitur *Text-to-Speech* dan *Speech Input* telah tersedia pada sistem operasi Android.



Gambar 3.1 Gambaran Umum Sistem

Pada gambar 3.1, pengguna terlebih dahulu memilih kategori yang telah tersedia, kemudian akan muncul gambar, kata dalam bahasa Inggris, kata cara pengucapan dan kata dalam bahasa Indonesia. Pengguna menggunakan fungsi “Dengar” untuk mendengarkan pelafalan kata dalam bahasa Inggris dari *database* (mengaktifkan fitur *Text-to-Speech*). *Text-to-Speech* yang aktif akan

mengubah teks dari *database* menjadi suara yang akan didengar pengguna. Setelah itu, pengguna menggunakan fungsi “Lafal” agar dapat melakukan pelafalan yang akan dicocokkan dengan kata bahasa Inggris yang tersimpan pada *database* (mengaktifkan fitur *Speech input*). *Speech input* yang aktif akan mengubah suara dari pengguna menjadi teks, dimana membutuhkan koneksi *internet* agar dapat merespon terhadap `RecognizerIntent` dan melakukan *streaming* pada *server* google.

Pencocokan yang dilakukan akan menghasilkan sebuah nilai yang akan disimpan dan ditampilkan dalam bentuk grafik. Selain itu, pengguna juga dapat menambah serta menghapus pengguna lain dan hanya seorang admin yang diperbolehkan untuk menambah, mengubah serta menghapus kategori dan *item* baru.

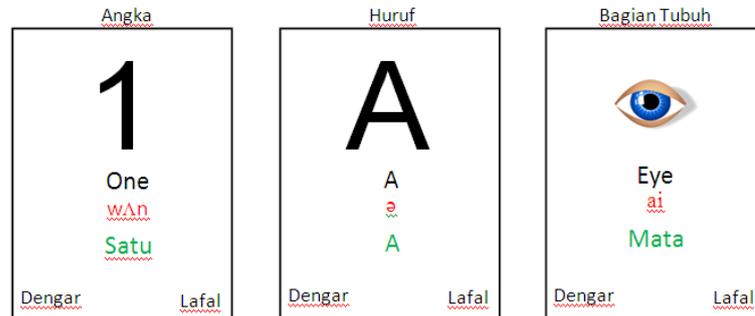
#### **3.1.1. Kategori**

Pada awal penggunaan aplikasi ini, seorang pengguna setelah membuat pengguna baru, terlebih dahulu memilih kategori. Kategori standar yang telah disediakan, yaitu angka, huruf, dan bagian tubuh. Pada aplikasi ini semua pengguna tidak dapat menambah, menghapus dan mengubah kategori. Kategori yang dipilih akan menampilkan beberapa *item* standar. Pada gambar 3.2, menunjukkan salah satu contoh *item* dari kategori yang ada.

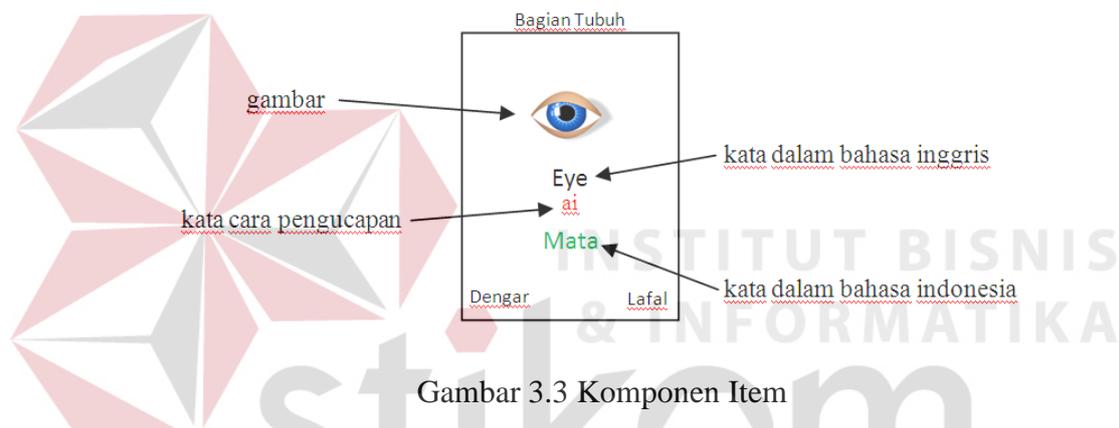
#### **3.1.2. Item**

Pada aplikasi ini *item* akan muncul setelah memilih kategori. *Item* yang akan muncul terdiri dari beberapa komponen yaitu gambar, kata dalam bahasa Inggris, kata cara pengucapan dan kata dalam bahasa Indonesia. Pada aplikasi ini

semua pengguna tidak dapat menambahkan, menghapus dan mengubah *item*. Pada gambar 3.3 menjelaskan tentang komponen *item*.



Gambar 3.2 Contoh Item pada Kategori Standar



Gambar 3.3 Komponen Item

### 3.1.3. Melafalkan Kata

Pada aplikasi ini *item* akan muncul dan pengguna dapat menggunakan fungsi “Dengar” agar dapat mendengar pelafalan kata bahasa Inggris yang tersimpan pada *database*. Proses pelafalan dapat dilakukan karena menggunakan paket/fitur Android yang telah tersedia yaitu *Text-toSpeech* (TTS) yang didukung oleh mesin TTS yaitu Pico TTS. Untuk mengimplementasikan *Text-to-Speech*, aplikasi perlu mengetahui bahasa mana yang akan digunakan untuk berbicara sebagai sumber daya bahasa, dalam kasus ini aplikasi menggunakan bahasa Inggris dimana bahasa ini telah didukung. Jadi suara dan kamus khusus sebagai sumber daya bahasa yang diperlukan diambil sebelum aplikasi TTS dapat

memulai untuk berbicara. Apabila terjadi penyimpangan terhadap pelafalan, TTS API memungkinkan aplikasi untuk melakukan *query platform* untuk ketersediaan file bahasa serta dapat melakukan *download* dan instalasi.

Sekarang *Text-to-Speech* dimisalkan dengan benar diinisialisasi dan dikonfigurasi, aplikasi dapat memulai untuk berbicara, dengan menggunakan *method speak()*, contohnya:

```
String kt_bhs_ing = "one";
mTts.speak(kt_bhs_ing, TextToSpeech.QUEUE_FLUSH, null);
```

Ketika pelayanan TTS pada aplikasi telah selesai digunakan, maka dapat dihentikan dengan memanggil `mTts.shutdown()`, dalam *activity method onDestroy()*.

### 3.1.4. Memasukkan Suara

Pada aplikasi ini setelah *item* muncul dan pengguna mendengarkan kata dalam bahasa Inggris, maka pengguna dapat menggunakan fungsi “Lafal” agar dapat melafalkan kata bahasa Inggris sesuai contoh pelafalan yang telah dilakukan. Proses memasukkan suara pengguna dapat dilakukan karena menggunakan paket/fitur Android yang telah tersedia yaitu *Speech Input* yang didukung oleh *server google* dimana membutuhkan koneksi *internet* agar dapat merespon `RecognizerIntent`. Untuk mengimplementasikan *Speech Input*, aplikasi memverifikasi bahwa perangkat target mampu mengenali *speech input*:

```
// Check to see if a recognition activity is present
PackageManager pm = getPackageManager();
List activities = pm.queryIntentActivities(
    new Intent(RecognizerIntent.ACTION_RECOGNIZE_SPEECH), 0);
if (activities.size() != 0) {
    speakButton.setOnClickListener(this);
} else {
    speakButton.setEnabled(false);
    speakButton.setText("Pengenalan suara tidak aktif");
}
```

Aplikasi kemudian menggunakan `startActivityForResult()` untuk memberitahukan bahwa sedang meminta *voice recognition*, serta sebuah parameter tambahan yang menentukan salah satu dari dua model bahasa. Aplikasi *voice recognition* yang menangani hasil dari proses *voice input*, kemudian melewati pengenalan *string* kembali ke aplikasi dengan memanggil `onActivityResult()`.

```

/**
 * Fire an intent to start the speech recognition activity.
 */
private void startVoiceRecognitionActivity() {
    Intent intent = new
    Intent(RecognizerIntent.ACTION_RECOGNIZE_SPEECH);
    intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE_MODEL,
        RecognizerIntent.LANGUAGE_MODEL_FREE_FORM);
    intent.putExtra(RecognizerIntent.EXTRA_PROMPT, "Speech
    recognition demo");
    startActivityForResult(intent, VOICE_RECOGNITION_REQUEST_CODE);
}

/**
 * Handle the results from the recognition activity.
 */
protected void onActivityResult(int requestCode, int resultCode,
    Intent data) {
    if (requestCode == VOICE_RECOGNITION_REQUEST_CODE &&
    resultCode == RESULT_OK) {
        // Fill the list view with the strings the recognizer thought
        it could have heard
        ArrayList<String> matches = data.getStringArrayListExtra(
            RecognizerIntent.EXTRA_RESULTS);
        mList.setAdapter(new ArrayAdapter<String>(this,
            android.R.layout.simple_list_item_1, matches));
    } super.onActivityResult(requestCode, resultCode, data);
}

```

### 3.1.5. Grafik Nilai

Grafik nilai menggunakan total nilai benar yang disertai tanggal perolehan pada penggunaan aplikasi, agar dapat dilihat statistik dari pengguna berdasarkan tanggal, apakah terdapat peningkatan atau penurunan. Terdapat dua grafik nilai, yaitu grafik nilai pengguna dan grafik nilai seluruh pengguna. Grafik nilai pengguna memperlihatkan perolehan nilai pengguna tiap kategori.

Grafik nilai seluruh pengguna memperlihatkan perolehan nilai seluruh pengguna tiap kategori.

### 3.1.6. Menambahkan dan Menghapus Pengguna

Pada aplikasi ini pengguna terlebih dahulu diharuskan membuat pengguna baru apabila belum terdaftar, agar dapat menggunakan aplikasi ini. Pengguna juga dapat menghapus serta mengubah data pengguna apabila menginginkannya. Pada aplikasi ini, semua pengguna diperbolehkan menambahkan pengguna baru atau menghapus pengguna yang sudah ada.

### 3.1.7. Admin

Seorang admin bertugas untuk menambah, mengubah dan menghapus kategori serta *item* baru kecuali kategori dan *item* standar. Admin melakukan tugasnya secara manual, tidak melalui sebuah *server* atau terkoneksi jaringan *internet*. Pada aplikasi ini, seorang admin diberikan sebuah *password* standar dan nantinya akan diminta untuk diubah, setelah itu admin dapat melakukan penambahan kategori dan *item* baru.

## 3.2. Perancangan Sistem

Perancangan sistem disini akan memberikan gambaran tentang sistem yang akan dibuat dengan menggunakan desain sistem yang berorientasi objek yaitu UML, serta digambarkan dengan menggunakan Star UML. Adapun desain yang dibuat antara lain:

- a. *Use Case Diagram*
- b. *Flow of Eventt*
- c. *Activity Diagram*

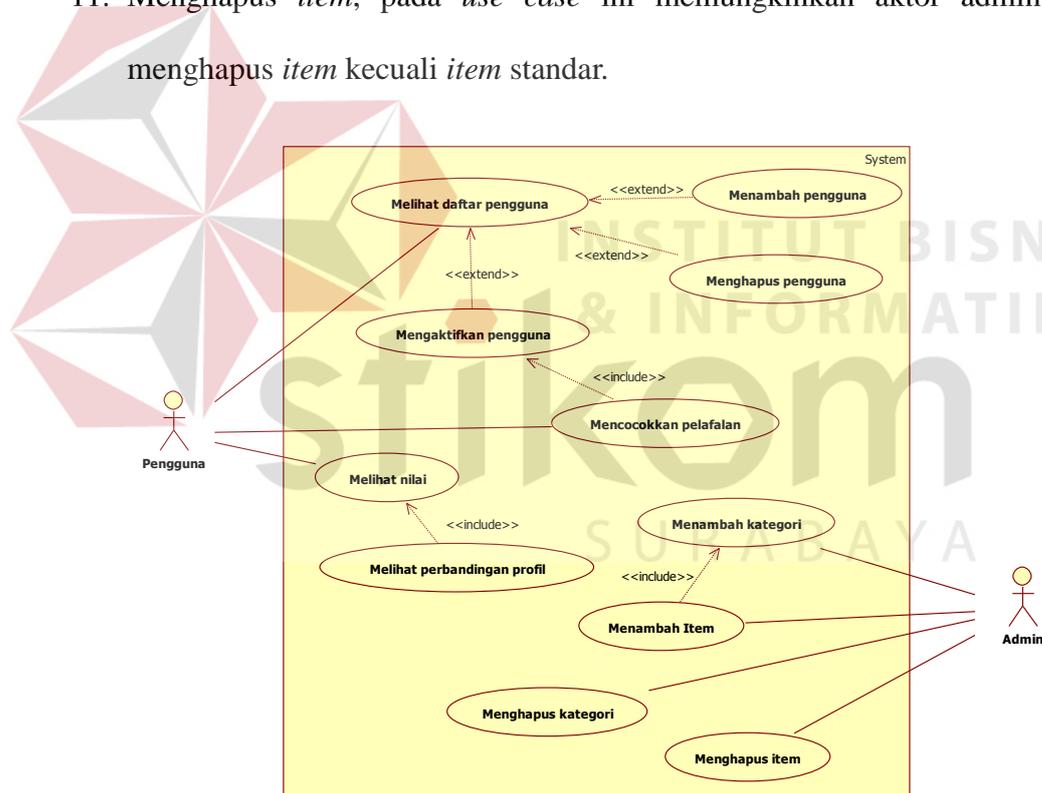
- d. *Sequence* Diagram
- e. *Class* Diagram
- f. *Component* Diagram
- g. *Deployment* Diagram

### 3.2.1. Use Case Diagram Aplikasi Mobile Melafalkan Bahasa Inggris

Kemampuan sistem untuk dapat berinteraksi dengan pengguna dan admin dapat digambarkan dalam *use case* diagram pada gambar 3.4. Pada gambar tersebut, terdapat dua aktor, yaitu pengguna dan admin. Selain terdapat dua aktor sebagai pengguna sistem, sistem ini juga terdiri dari 11 buah *use case*, yaitu:

1. Melihat daftar pengguna, pada *use case* ini memungkinkan aktor pengguna untuk melihat pengguna yang tersimpan pada *database*.
2. Menambah pengguna, pada *use case* ini memungkinkan aktor pengguna untuk menambah pengguna baru ke dalam *database*.
3. Menghapus pengguna, pada *use case* ini memungkinkan aktor pengguna untuk menghapus pengguna yang telah ada pada *database*.
4. Mengaktifkan pengguna, pada *use case* ini memungkinkan aktor pengguna untuk mengaktifkan pengguna agar dapat memulai belajar melafalkan bahasa Inggris.
5. Mencocokkan pelafalan, pada *use case* ini memungkinkan aktor pengguna untuk menggunakan fitur *Text-to-Speech* untuk mendengar pelafalan kata bahasa Inggris dan fitur *Speech Input* untuk mengenali pelafalan pengguna yang akan dicocokkan dengan kata bahasa Inggris pada *database*.
6. Melihat nilai, pada *use case* ini memungkinkan aktor pengguna untuk melihat nilai yang dihasilkan.

7. Melihat perbandingan profil, pada *use case* ini memungkinkan aktor pengguna untuk melihat nilai perbandingan antar pengguna.
8. Menambah kategori, pada *use case* ini memungkinkan aktor admin untuk menambah kategori baru.
9. Menghapus kategori, pada *use case* ini memungkinkan aktor admin untuk menghapus kategori kecuali kategori standar.
10. Menambah *item*, pada *use case* ini memungkinkan aktor admin untuk menambah *item* baru kecuali *item* standar.
11. Menghapus *item*, pada *use case* ini memungkinkan aktor admin untuk menghapus *item* kecuali *item* standar.



Gambar 3.4 Use Case Diagram Aplikasi Mobile Belajar Melafalkan

### Bahasa Inggris

*Use case* untuk menambah pengguna, menghapus pengguna, dan mengaktifkan pengguna merupakan perluasan/*extend* dari *use case* untuk melihat daftar pengguna karena pada *use case* ini belum tentu dikerjakan pada saat *use*

*case* melihat daftar pengguna dikerjakan. *Use case* untuk mencocokkan pelafalan merupakan bagian/*include* dari *use case* untuk mengaktifkan pengguna karena pada *use case* ini akan dilakukan jika *use case* mengaktifkan pengguna dikerjakan. *Use case* untuk melihat perbandingan profil merupakan bagian/*include* dari *use case* untuk melihat nilai karena pada *use case* ini akan dilakukan jika *use case* melihat nilai dikerjakan. *Use case* untuk menambah *item* merupakan bagian/*include* dari *use case* untuk menambah kategori karena pada *use case* ini akan dilakukan jika *use case* menambah kategori dikerjakan.

### **3.2.2. Flow of Event Aplikasi Mobile Melafalkan Bahasa Inggris**

Dari *use case* yang ada, dibutuhkan *flow of event* untuk menjelaskan spesifikasi proses/aliran kejadian yang terjadi pada tiap *use case*, serta untuk mendokumentasikan aliran logika dalam *use case*. *Flow of event* meliputi diskripsi singkat, kondisi awal, aliran kejadian utama, aliran kejadian alternative, kondisi akhir. *Flow of event* yang akan dibuat adalah *flow of event* untuk *use case* melihat daftar pengguna, *use case* menambah pengguna, *use case* menghapus pengguna, *use case* mengaktifkan pengguna, *use case* mencocokkan pelafalan, *use case* melihat nilai, *use case* melihat perbandingan profil, *use case* menambah kategori, *use case* menghapus kategori, *use case* menambah *item*, dan *use case* menghapus *item*.

#### **A. Flow of Event untuk Use Case Melihat Daftar Pengguna**

*Flow of event* untuk *use case* melihat daftar pengguna dapat dilihat pada tabel 3.1.

Tabel 3.1 Flow of Event Melihat daftar pengguna

Diskripsi	<i>Use case</i> melihat daftar pengguna memungkinkan pengguna untuk melihat pengguna yang telah terdaftar	
Kondisi Awal	-	
Kondisi Akhir	Pengguna berhasil menampilkan daftar pengguna	
Aliran Kejadian Utama		Aksi pemakai
	1	<i>Use case</i> memulai, ketika pengguna memilih pilihan melakukan proses melihat daftar pengguna
		Respon sistem
		Sistem menampilkan daftar pengguna yang telah terdaftar

### B. Flow of Event untuk Use Case Menambah Pengguna

*Flow of event* untuk *use case* menambah pengguna dapat dilihat pada tabel 3.2.

Tabel 3.2 Flow of Event Menambah pengguna

Diskripsi	<i>Use case</i> menambah pengguna memungkinkan pengguna untuk menambah pengguna baru	
Kondisi Awal	Pengguna berada pada daftar pengguna atau pengaturan pengguna	
Kondisi Akhir	Pengguna berhasil menambahkan pengguna baru	
Aliran Kejadian Utama		Aksi pemakai
	1	<i>Use case</i> memulai, ketika pengguna memilih pilihan melakukan proses menambah pengguna
	2	Pengguna memasukkan nama, lalu dikonfirmasi
		Respon sistem
		Sistem meminta nama pengguna
		Sistem menampilkan konfirmasi pesan “berhasil menambahkan Pengguna”

### C. Flow of Event untuk Use Case Mengaktifkan Pengguna

*Flow of event* untuk *use case* mengaktifkan pengguna dapat dilihat pada tabel 3.3.

Tabel 3.3 Flow of Event Mengaktifkan pengguna

Diskripsi	<i>Use case</i> mengaktifkan pengguna memungkinkan pengguna untuk mengaktifkan pengguna yang telah terdaftar
Kondisi Awal	Pengguna berada pada daftar pengguna

Kondisi Akhir	Pegguna berhasil mengaktifkan pegguna	
	Aksi pemakai	Respon sistem
Aliran Kejadian Utama	1 <i>Use case</i> memulai, ketika pegguna memilih pilihan melakukan proses mengaktifkan pegguna dengan memilih pegguna yang akan diaktifkan	Sistem mengaktifkan pegguna agar dapat melakukan proses selanjutnya

#### D. Flow of Event untuk Use Case Menghapus Pegguna

*Flow of event* untuk *use case* menghapus pegguna dapat dilihat pada tabel 3.4.

Tabel 3.4 Flow of Event Menghapus pegguna

Diskripsi	<i>Use case</i> menghapus pegguna memungkinkan pegguna untuk menghapus pegguna yang telah terdaftar	
Kondisi Awal	Pegguna telah terdaftar dan berada pada daftar pegguna	
Kondisi Akhir	Pegguna berhasil menghapus pegguna	
	Aksi pemakai	Respon sistem
Aliran Kejadian Utama	1 <i>Use case</i> memulai, ketika pegguna memilih pilihan melakukan proses menghapus pegguna dengan memilih pegguna yang akan dihapus	Sistem menampilkan pesan “apakah pegguna akan dihapus?”
	2 Pegguna melakukan konfirmasi	Sistem memperbarui daftar pegguna

#### E. Flow of Event untuk Use Case Mencocokkan Pelafalan

*Flow of event* untuk *use case* mencocokkan pelafalan dapat dilihat pada Tabel 3.5.

Tabel 3.5 Flow of Event Mencocokkan Pelafalan

Diskripsi	<i>Use case</i> mencocokkan pelafalan memungkinkan pegguna untuk mendengar dan mencocokkan pelafalan bahasa Inggris
-----------	---

Kondisi Awal	Pengguna telah aktif		
Kondisi Akhir	Pengguna berhasil mendengar dan mencocokkan pelafalan bahasa Inggris		
Aliran Kejadian Utama		Aksi pemakai	Respon sistem
	1	<i>Use case</i> memulai, ketika pengguna memilih pilihan melakukan proses mencocokkan pelafalan	Sistem menampilkan kategori : angka, huruf dan bagian tubuh
	2	Pengguna memilih salah satu kategori	Sistem menampilkan <i>item</i> : gambar, kata dalam bahasa Inggris, kata cara pengucapan, dan kata dalam bahasa Indonesia.
	3	Pengguna menggunakan fungsi “dengar” untuk mendengarkan teks dalam bahasa Inggris	Sistem melafalkan teks kata dalam bahasa Inggris menggunakan <i>Text-to-Speech</i>
	4	Pengguna menggunakan fungsi “lafal” untuk mengkonfirmasi pelafalan	Sistem menambahkan nilai benar dan menampilkan <i>item</i> selanjutnya A1: pelafalan salah
	5	Pengguna mengkonfirmasi pelafalan hingga <i>item</i> terakhir	Sistem menyimpan dan menampilkan nilai akhir serta tanggal sekarang
Aliran Alternatif		Respon Sistem	Aksi Pemakai
	A1	Sistem memberi konfirmasi pesan “Salah” dan menambahkan tanda salah	Kembali ke aliran utama, langkah 4

#### F. Flow of Event untuk Use Case Melihat Nilai

*Flow of event* untuk *use case* melihat nilai dapat dilihat pada tabel 3.6.

Tabel 3.6 Flow of Event Melihat Nilai

Diskripsi	<i>Use case</i> melihat nilai memungkinkan pengguna untuk melihat nilai yang diperoleh pengguna
Kondisi Awal	Pengguna telah aktif
Kondisi Akhir	Pengguna berhasil melihat nilai yang diperoleh

Aliran Kejadian Utama	Aksi pemakai		Respon sistem
	1	<i>Use case</i> memulai, ketika pengguna memilih melakukan melihat nilai	memulai, pengguna pilihan proses

### G. Flow of Event untuk Use Case Melihat Perbandingan Profil

*Flow of event* untuk *use case* melihat perbandingan profil dapat dilihat pada tabel 3.7.

Tabel 3.7 Flow of Event Melihat Perbandingan Profil

Diskripsi	<i>Use case</i> melihat perbandingan profil memungkinkan pengguna untuk melihat perbandingan nilai pengguna yang lain		
Kondisi Awal	Pengguna telah aktif		
Kondisi Akhir	Pengguna berhasil melihat perbandingan nilai pengguna yang lain		
Aliran Kejadian Utama	Aksi pemakai		Respon sistem
	1	<i>Use case</i> memulai, ketika pengguna memilih melakukan melihat perbandingan <i>profile</i>	Sistem menampilkan nilai rata-rata yang dilakukan per kategori berdasarkan pengguna terdaftar

### H. Flow of Event untuk Use Case Menambah Kategori

*Flow of event* untuk *use case* menambah kategori dapat dilihat pada tabel 3.8.

Tabel 3.8 Flow of Event Menambah Kategori

Diskripsi	<i>Use case</i> menambah kategori memungkinkan admin untuk menambah kategori baru		
Kondisi Awal	Admin telah <i>login</i> dan berada pada pengaturan kategori		
Kondisi Akhir	Admin berhasil menambah kategori baru		
Aliran Kejadian Utama	Aksi pemakai		Respon sistem
	1	<i>Use case</i> memulai, ketika admin memilih melakukan proses menambah kategori	Sistem meminta masukkan nama kategori

Aliran Kejadian Utama		Aksi pemakai	Respon sistem
	2	Admin memasukkan nama kategori, lalu dikonfirmasi	Sistem menampilkan konfirmasi pesan “berhasil menambahkan kategori”

### I. Flow of Event untuk Use Case Menghapus Kategori

*Flow of event* untuk *use case* menghapus kategori dapat dilihat pada tabel 3.9.

Tabel 3.9 Flow of Event Menghapus Kategori

Diskripsi	<i>Use case</i> menghapus kategori memungkinkan admin menghapus kategori yang ada		
Kondisi Awal	Admin telah <i>login</i> dan berada pada pengaturan kategori		
Kondisi Akhir	Admin berhasil menghapus kategori		
Aliran Kejadian Utama		Aksi pemakai	Respon sistem
	1	<i>Use case</i> memulai, ketika admin memilih pilihan melakukan proses menghapus kategori dengan memilih kategori yang akan dihapus	Sistem menampilkan pesan “apakah kategori akan dihapus?”
	2	Admin melakukan konfirmasi	Sistem memperbarui daftar kategori

### J. Flow of Event untuk Use Case Menambah Item

*Flow of event* untuk *use case* menambah *item* dapat dilihat pada tabel 3.10.

Tabel 3.10 Flow of Event Menambah Item

Diskripsi	<i>Use case</i> menambah <i>item</i> memungkinkan admin untuk menambah <i>item</i> baru		
Kondisi Awal	Admin telah <i>login</i> dan berada pada pengaturan <i>item</i> serta telah tersedia kategori baru		
Kondisi Akhir	Admin berhasil menambah <i>item</i> baru		
Aliran Kejadian Utama		Aksi pemakai	Respon sistem
	1	<i>Use case</i> memulai, ketika admin memilih pilihan melakukan proses menambah <i>item</i>	Sistem menampilkan daftar kategori

Aliran Kejadian Utama		Aksi pemakai	Respon sistem
	2	Admin memilih kategori yang akan ditambahkan <i>item</i>	Sistem meminta masukkan <i>item</i> : gambar, kata dalam bahasa Inggris, kata cara pengucapan dan kata dalam bahasa indonesia
3	Admin memasukkan <i>item</i> : gambar, kata bahasa Inggris, dan kata ucap, lalu dikonfirmasi	Sistem menampilkan konfirmasi pesan “berhasil menambahkan <i>item</i> ”	

### K. Flow of Event untuk Use Case Menghapus Item

*Flow of event* untuk *use case* menghapus *item* dapat dilihat pada tabel 3.11.

Tabel 3.11 Flow of Event Menghapus Item

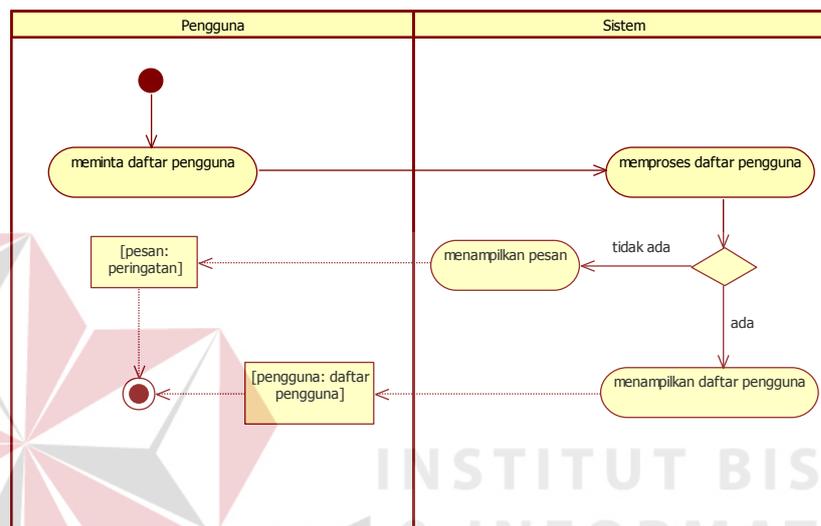
Diskripsi	<i>Use case</i> menghapus <i>item</i> memungkinkan admin untuk menghapus <i>item</i> baru yang telah ditambahkan		
Kondisi Awal	Admin telah <i>login</i> dan berada pada pengaturan <i>item</i>		
Kondisi Akhir	Admin berhasil menghapus <i>item</i>		
Aliran Kejadian Utama		Aksi pemakai	Respon sistem
	1	<i>Use case</i> memulai, ketika admin memilih pilihan melakukan proses menghapus <i>item</i>	Sistem menampilkan daftar <i>item</i>
	2	Admin memilih <i>item</i> yang akan dihapus	Sistem menampilkan pesan “apakah <i>item</i> akan dihapus?”
	3	Admin melakukan konfirmasi	Sistem memperbarui daftar <i>item</i>

### 3.2.3. Activity Diagram Aplikasi Mobile Melafalkan Bahasa Inggris

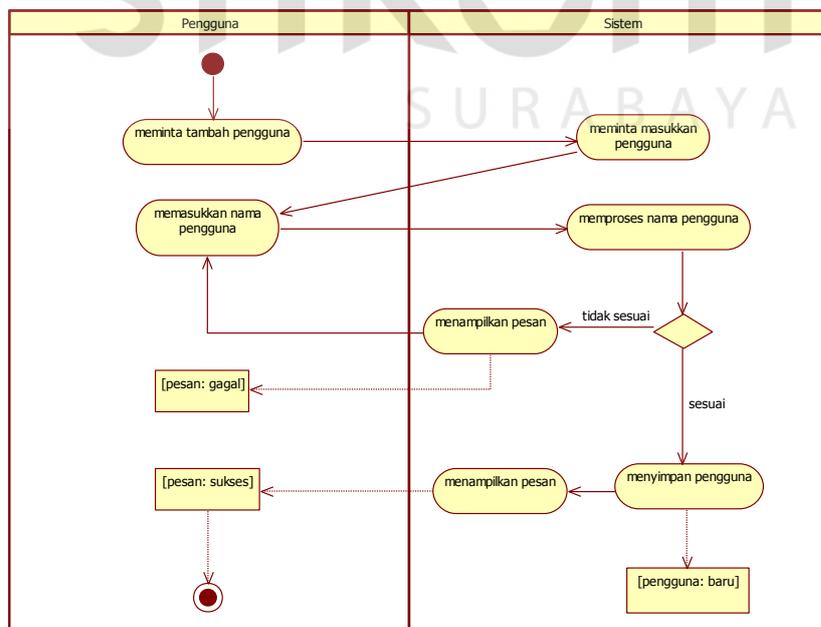
Dari *use case* yang ada, dibutuhkan *activity diagram* untuk menjelaskan proses/aliran proses yang terjadi pada tiap *use case*. *Activity diagram* akan dijelaskan secara lengkap di bawah ini.

### A. Activity Diagram untuk Use Case Melihat daftar pengguna

Proses pada gambar 3.5 dimulai dari pengguna ingin melihat daftar pengguna yang akan diproses oleh sistem untuk menampilkan daftar pengguna. Jika pengguna tidak ada, maka pengguna diminta untuk menambahkan pengguna baru agar dapat menggunakan aplikasi.



Gambar 3.5 Activity Diagram Melihat Daftar Pengguna



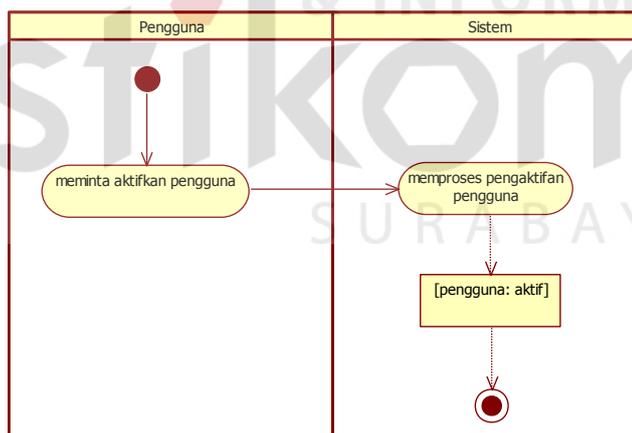
Gambar 3.6 Activity Diagram Menambah Pengguna

### B. Activity Diagram untuk Use Case Menambah pengguna

Proses pada gambar 3.6 dimulai dari pengguna ingin menambahkan pengguna baru, maka sistem meminta masukkan berupa nama pengguna. Setelah itu, pengguna memasukkan nama pengguna yang akan dikonfirmasi oleh sistem. Apabila nama pengguna sesuai dengan ketentuan (tidak boleh kosong dan kurang dari 3 huruf), maka sistem akan menyimpan pengguna baru dan memberikan pesan "berhasil menambahkan pengguna". Jika tidak sesuai, maka pengguna diberi pesan kesalahan dan diminta mengulang isian nama pengguna hingga benar.

### C. Activity Diagram untuk Use Case Mengaktifkan pengguna

Proses pada gambar 3.7 menjelaskan pengguna ingin mengaktifkan pengguna agar dapat melanjutkan ke proses berikutnya.

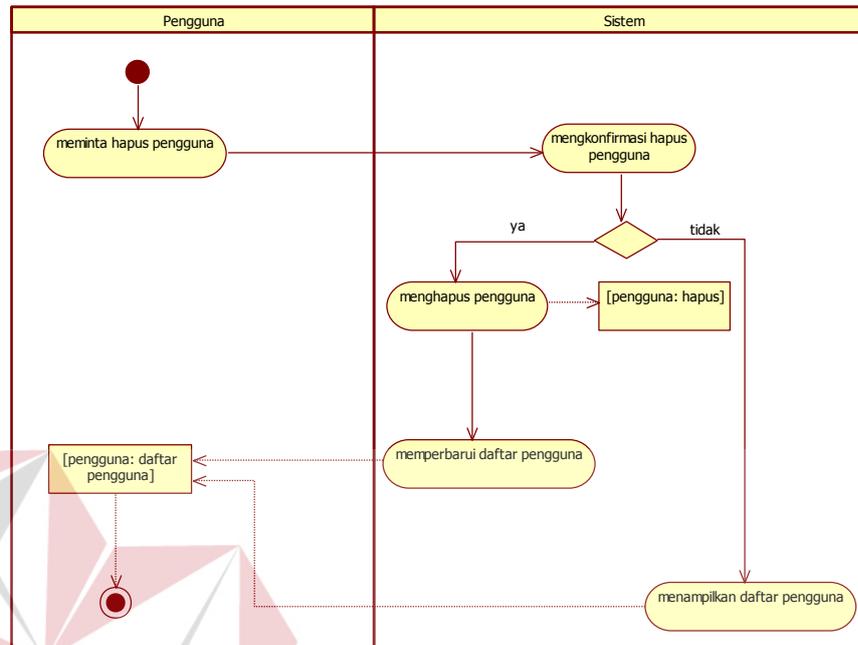


Gambar 3.7 Activity Diagram Mengaktifkan Pengguna

### D. Activity Diagram untuk Use Case Menghapus pengguna

Proses pada gambar 3.8 dimulai dari pengguna ingin menghapus pengguna, maka sistem mengkonfirmasi "apakah pengguna akan dihapus?". Setelah itu, pengguna mengesahkan penghapusan, maka sistem menghapus

pengguna tersebut dan memperbarui daftar pengguna. Jika pengguna membatalkan penghapusan, maka akan kembali pada daftar pengguna.



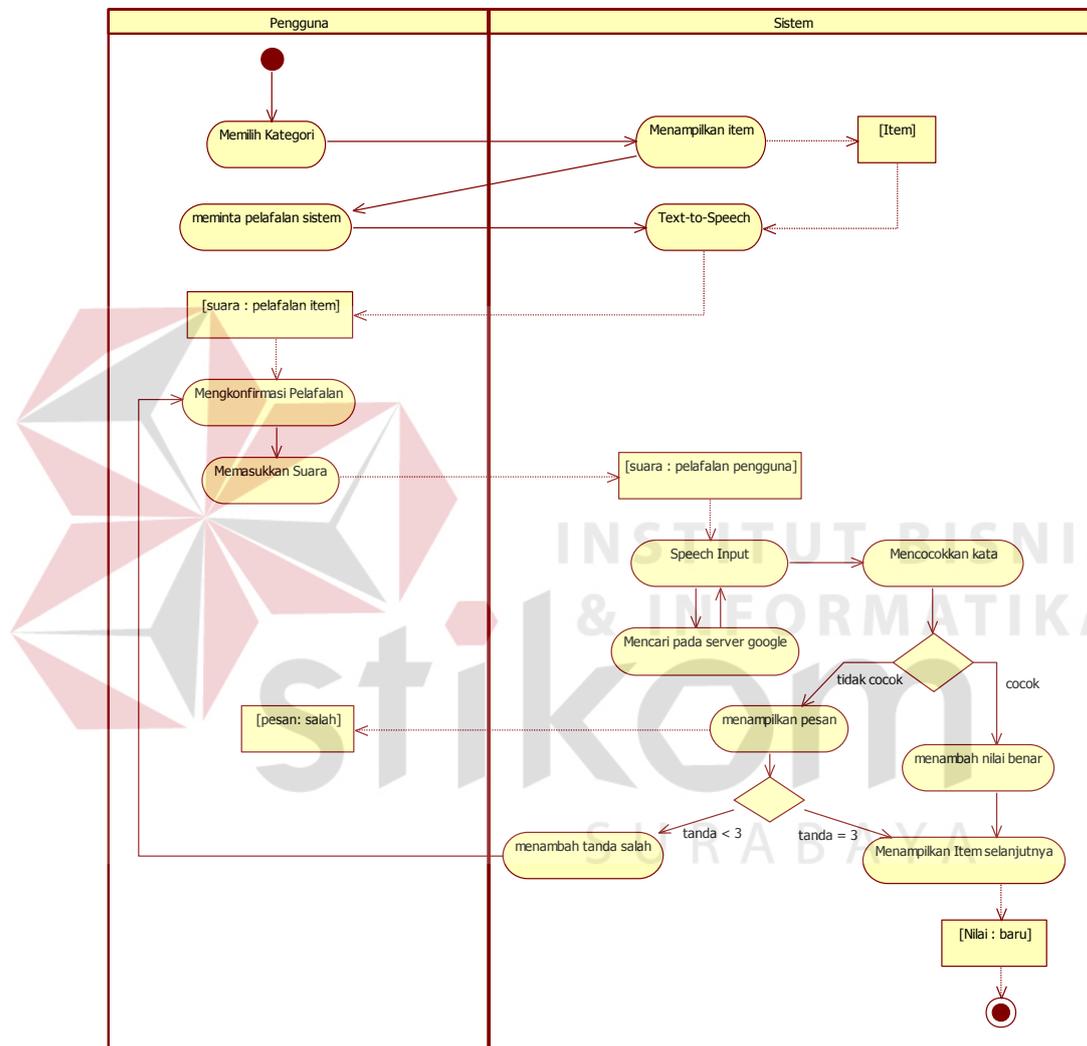
Gambar 3.8 Activity Diagram Menghapus Pengguna

### E. Activity Diagram untuk Use Case Mencocokkan pelafalan

Proses pada gambar 3.9 dimulai dari pengguna memilih kategori setelah itu akan ditampilkan *item* : gambar, kata dalam bahasa Inggris, kata ucap dan kata dalam bahasa Indonesia serta fungsi “Dengar” agar dapat mendengar suara pelafalan dari kata bahasa Inggris yang diproses dari *Text-to-Speech*. Pengguna melanjutkan dengan memasukkan suara (melafalkan sesuai contoh pelafalan) yang akan diproses oleh *Speech input* untuk diubah ke teks, setelah itu dicocokkan.

Pada saat proses pencocokkan selesai, apabila pesan “Salah” tampil maka akan ditambahkan tanda salah dan dikembalikan ke *item* tersebut untuk melakukan pengulangan lafal. Apabila telah tiga kali salah maka akan dilanjutkan

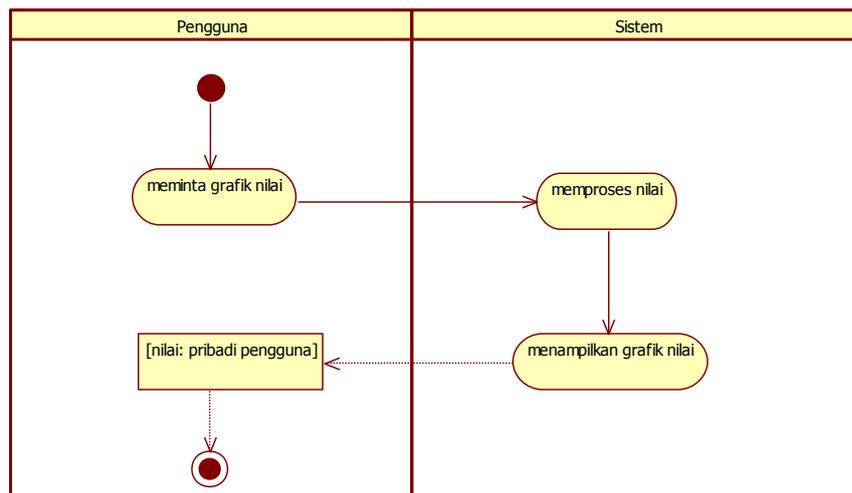
pada item selanjutnya. Apabila benar melafalkan maka akan dikalkulasi nilai benar, lalu ditampilkan *item* selanjutnya. Apabila telah menyelesaikan pembelajaran maka hasil akhir berupa nilai keseluruhan yang ditambahkan tanggal akan disimpan kedalam database.



Gambar 3.9 Activity Diagram Mencocokkan Pelafalan

#### F. Activity Diagram untuk Use Case Melihat nilai

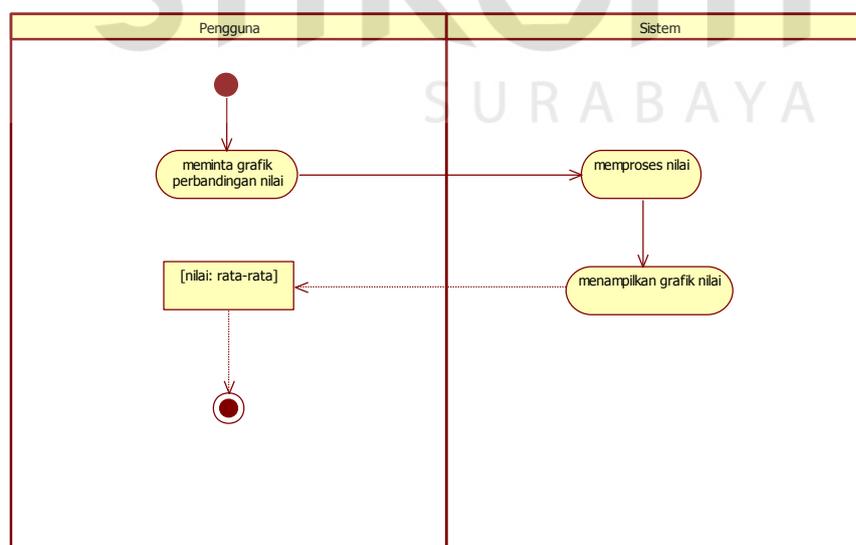
Proses pada gambar 3.10 dimulai dari pengguna ingin melihat grafik nilai yang akan diproses oleh sistem untuk menampilkan grafik nilai yang dilakukan per kategori berdasarkan tanggal.



Gambar 3.10 Activity Diagram Melihat Nilai

### G. Activity Diagram untuk Use Case Melihat perbandingan profil

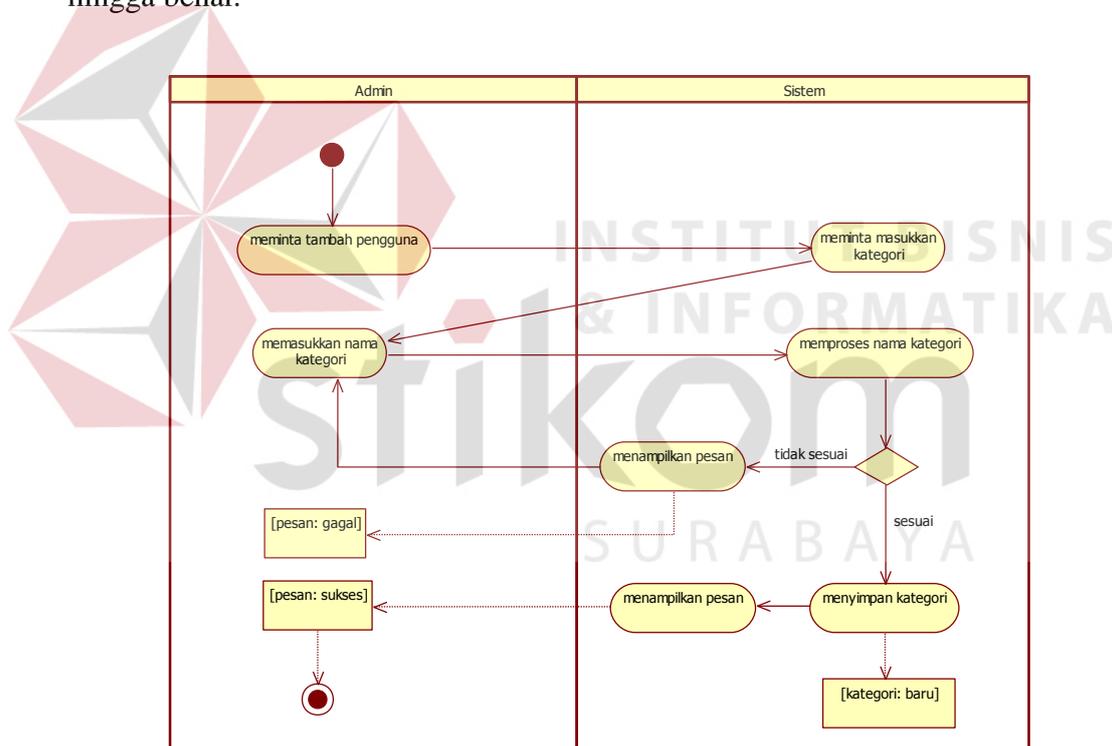
Proses pada gambar 3.11 dimulai dari pengguna ingin melihat grafik perbandingan nilai yang akan diproses oleh sistem untuk menampilkan grafik nilai yang dilakukan per kategori berdasarkan daftar pengguna. Grafik nilai yang ditampilkan adalah nilai rata-rata antar pengguna.



Gambar 3.11 Activity Diagram Melihat Perbandingan Profil

## H. Activity Diagram untuk Use Case Menambah kategori

Proses pada gambar 3.12 dimulai dari admin ingin menambahkan kategori baru, maka sistem meminta memasukkan berupa nama kategori. Setelah itu, admin memasukkan nama kategori yang akan dikonfirmasi oleh sistem. Apabila nama kategori sesuai dengan ketentuan (tidak boleh kosong dan kurang dari 3 huruf), maka sistem akan menyimpan kategori baru dan memberikan pesan "berhasil menambahkan kategori". Jika tidak sesuai, maka admin diberi pesan kesalahan dan diminta mengulang isian nama kategori hingga benar.

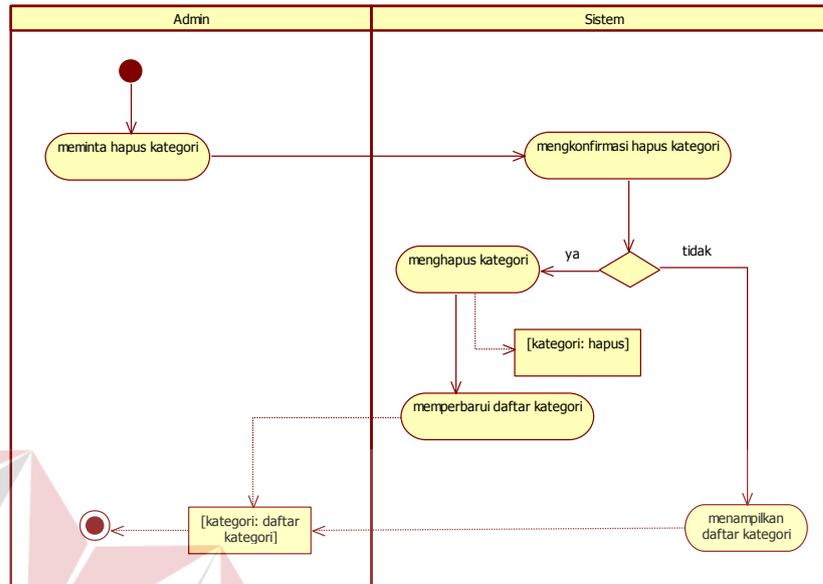


Gambar 3.12 Activity Diagram Menambah Kategori

## I. Activity Diagram untuk Use Case Menghapus kategori

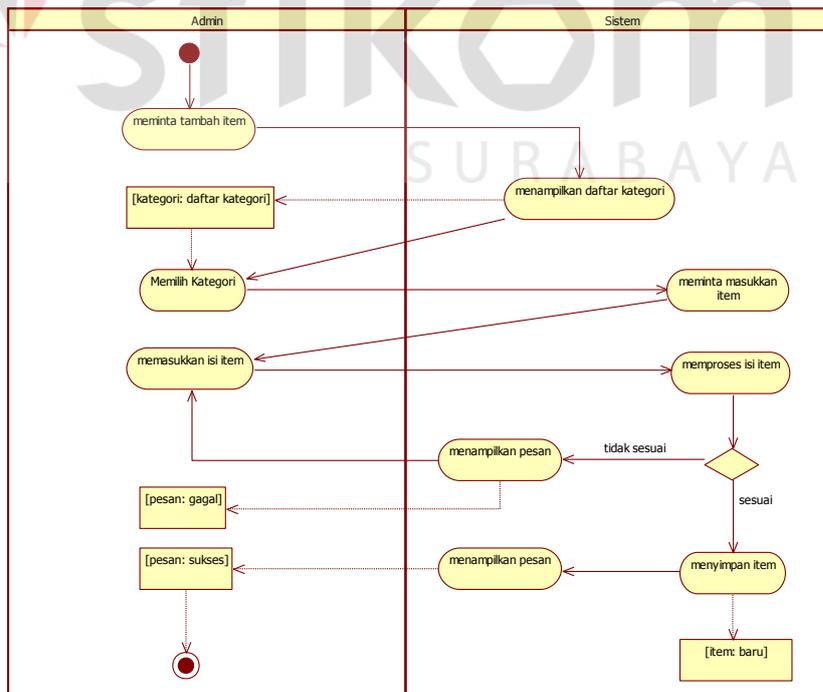
Proses pada gambar 3.13 dimulai dari admin ingin menghapus kategori, maka sistem mengkonfirmasi "apakah kategori akan dihapus?". Setelah itu, admin mengesahkan penghapusan, maka sistem menghapus kategori tersebut dan

memperbarui daftar kategori. Jika admin membatalkan penghapusan, maka akan kembali pada daftar kategori.



Gambar 3.13 Activity Diagram Menghapus Kategori

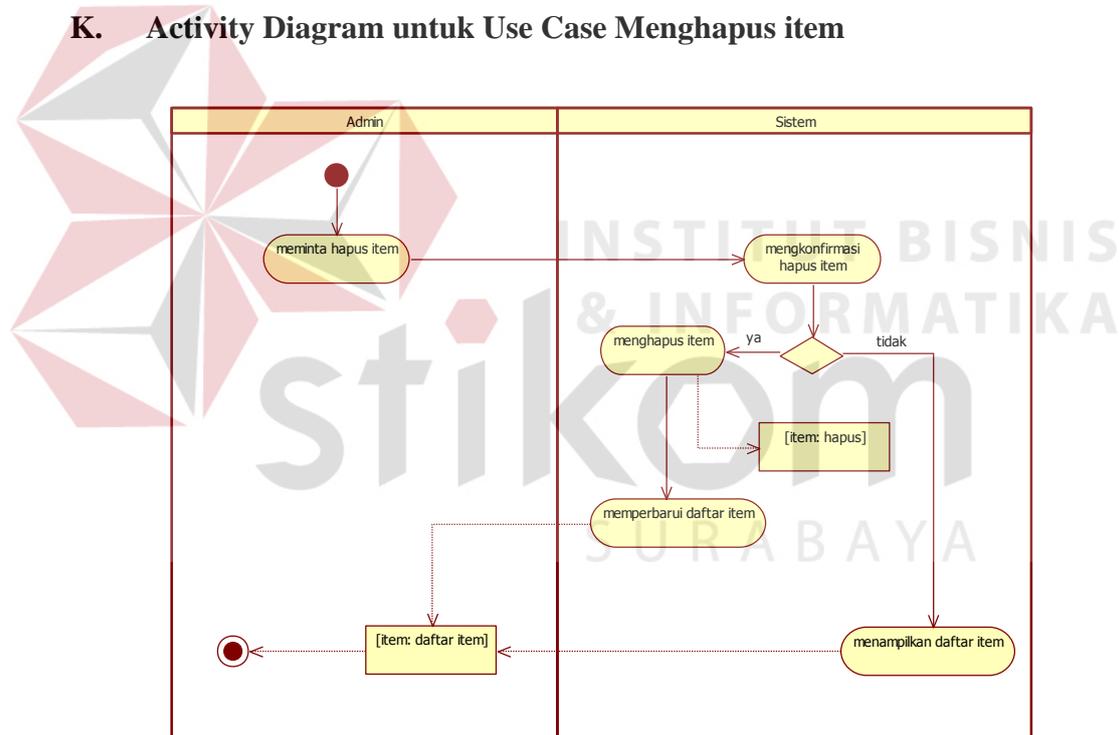
#### J. Activity Diagram untuk Use Case Menambah item



Gambar 3.14 Activity Diagram Menambah Item

Proses pada gambar 3.14 dimulai dari admin ingin menambahkan *item* baru, maka sistem meminta memilih kategori terlebih dulu. Setelah itu, admin diminta memasukkan isi *item* (kata dalam bahasa Inggris, kata cara pengucapan, kata dalam bahasa Indonesia dan gambar) yang akan dikonfirmasi oleh sistem. Apabila isi *item* sesuai dengan ketentuan (tidak boleh kosong), maka sistem akan menyimpan *item* baru dan memberikan pesan "berhasil menambahkan *item*". Jika tidak sesuai, maka admin diberi pesan kesalahan dan diminta mengulang isian *item* hingga benar.

### K. Activity Diagram untuk Use Case Menghapus item



Gambar 3.15 Activity Diagram Menghapus Item

Proses pada gambar 3.15 dimulai dari admin ingin menghapus *item*, maka sistem mengkonfirmasi "apakah *item* akan dihapus?". Setelah itu, admin mengesahkan penghapusan, maka sistem menghapus *item* tersebut dan

memperbarui daftar *item*. Jika admin membatalkan penghapusan, maka akan kembali pada daftar *item*.

#### 3.2.4. Sequence Diagram Aplikasi Mobile Melafalkan Bahasa Inggris

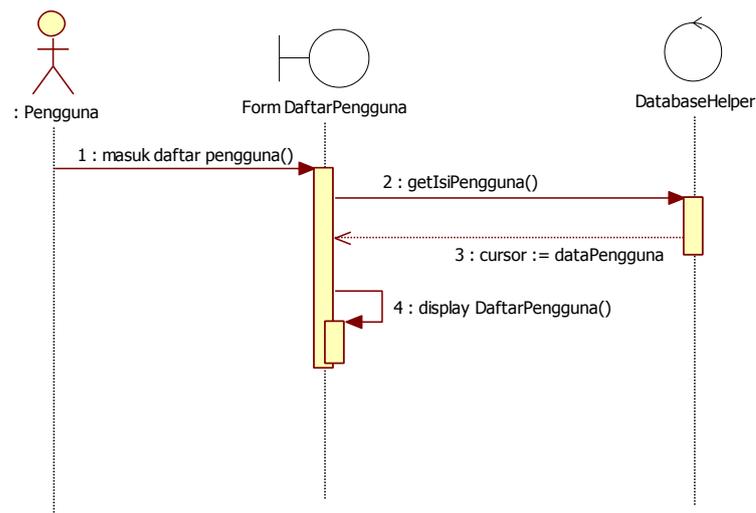
Untuk menjelaskan tiap *use case* berdasarkan pada *use case diagram* pada gambar 3.4, maka dibutuhkan *sequence diagram* yang menggambarkan jalannya suatu proses yang melibatkan objek dalam aplikasi ini. *Sequence diagram* yang akan dibuat sesuai dengan *use case* yang ada pada gambar 3.4 yaitu *sequence* melihat daftar pengguna, menambah pengguna, mengaktifkan pengguna, menghapus pengguna, mencocokkan pelafalan, melihat nilai, melihat perbandingan profil, menambah kategori, menghapus kategori, menambah *item*, menghapus *item*.

##### A. Sequence Diagram untuk Use Case Melihat Daftar Pengguna

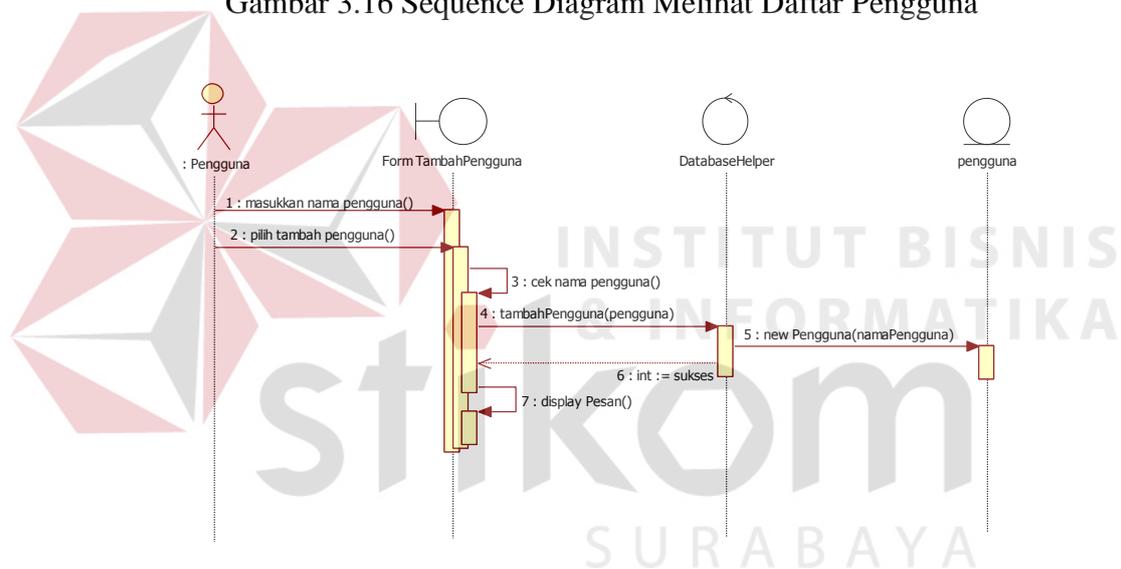
Pada gambar 3.16 dimulai dari pengguna masuk pada *form* daftar pengguna untuk dapat melihat daftar pengguna. Sistem akan mengambil nama-nama pengguna yang terdaftar pada kontrol *databasehelper*, kemudian akan ditampilkan pada *form* daftar pengguna.

##### B. Sequence Diagram untuk Use Case Menambah Pengguna

Pada gambar 3.17 dimulai dari pengguna memasukkan nama pengguna dan memilih tambah pengguna baru, setelah itu nama pengguna akan divalidasi oleh sistem dan jika benar akan dimasukkan ke dalam *database* melalui kontrol *databasehelper*, dimana pada *entity* pengguna dibuat pengguna baru. *Tab* tambah pengguna akan menampilkan pesan setelah melakukan proses tambah pengguna.



Gambar 3.16 Sequence Diagram Melihat Daftar Pengguna



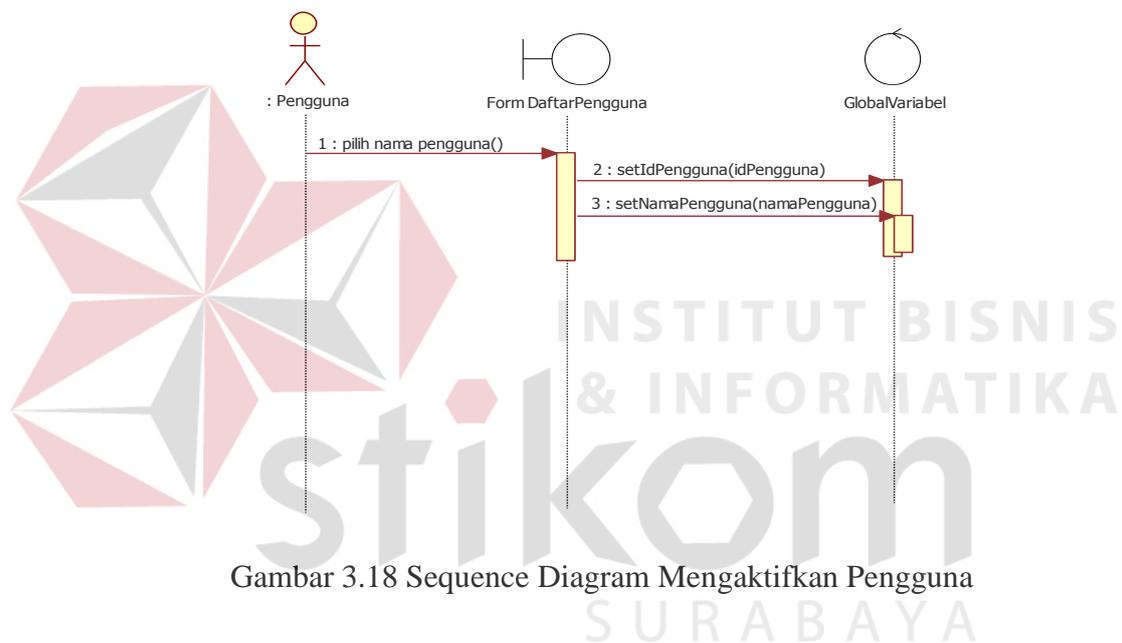
Gambar 3.17 Sequence Diagram Menambah Pengguna

### C. Sequence Diagram untuk Use Case Mengaktifkan Pengguna

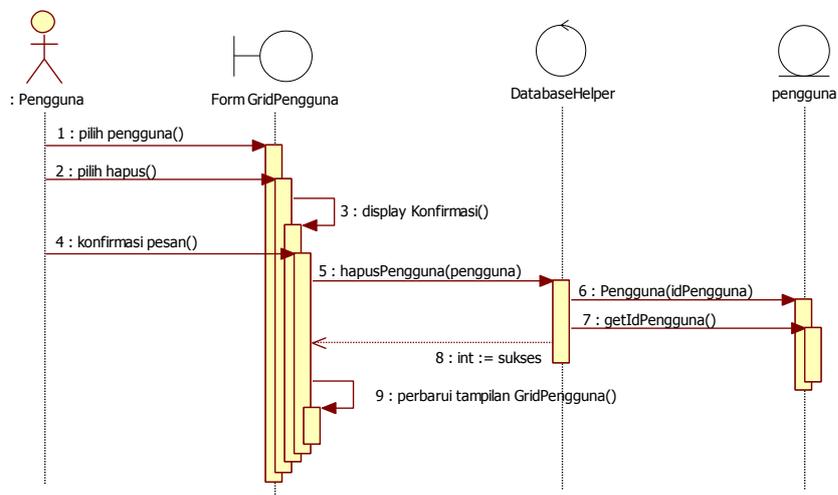
Pada gambar 3.18 dimulai dari pengguna memilih nama pengguna pada *form* daftar pengguna untuk mengaktifkan pengguna agar dapat melanjutkan ke proses selanjutnya. Pada *form* daftar pengguna sistem akan mengatur id pengguna dan nama pengguna pada kontrol *globalvariabel* agar dapat diakses oleh proses lain sebagai tanda aktifnya pengguna.

#### D. Sequence Diagram untuk Use Case Menghapus Pengguna

Pada gambar 3.19 dimulai dari pengguna memilih pengguna yang akan dihapus terlebih dulu, setelah itu sistem menampilkan pesan yang akan dikonfirmasi oleh pengguna dan jika dikonfirmasi untuk dihapus maka sistem akan menghapus pengguna dari *database* melalui kontrol *databasehelper*, dimana pada *entity* pengguna diambil id pengguna. *Tab grid* pengguna akan memperbarui tampilan *grid* pengguna setelah melakukan proses hapus pengguna.



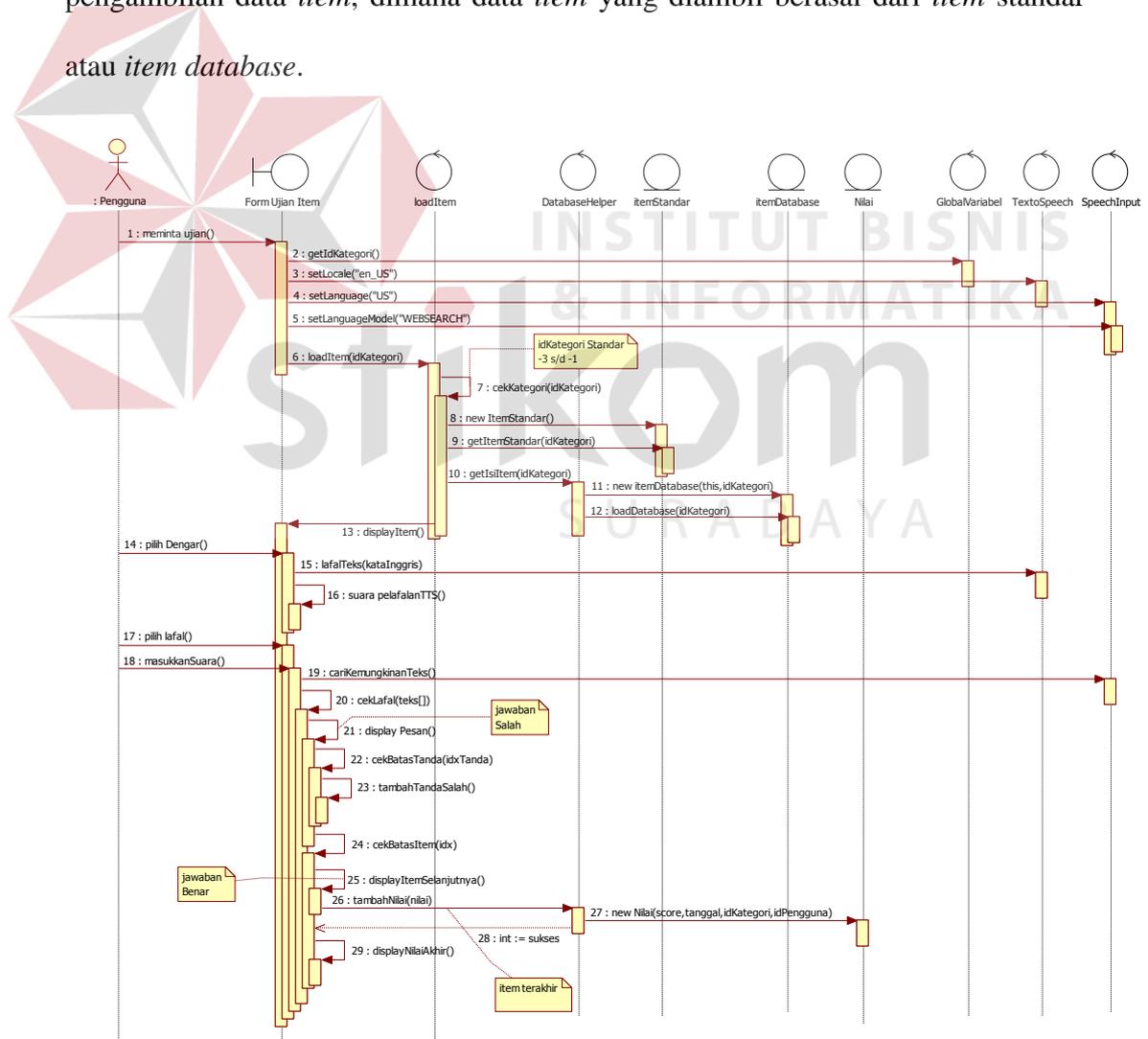
Gambar 3.18 Sequence Diagram Mengaktifkan Pengguna



Gambar 3.19 Sequence Diagram Menghapus Pengguna

### E. Sequence Diagram untuk Use Case Mencocokkan Pelafalan

Pada gambar 3.20 dimulai dari pengguna meminta ujian *item* pada *form ujian item*, dimana id kategori diambil dari *globalvariabel* agar dapat menyesuaikan dengan *item* yang akan ditampilkan. *locale* pada kontrol TTS diatur untuk dapat melafalkan kata dalam bahasa Inggris, *language* serta *language model* pada kontrol *speech input* diatur untuk dapat mengubah suara pengguna kedalam bentuk teks yang akan disesuaikan dengan bahasa yang telah diatur. Pada kontrol *loaditem* id kategori akan dicek terlebih dahulu untuk menyesuaikan pengambilan data *item*, dimana data *item* yang diambil berasal dari *item standar* atau *item database*.



Gambar 3.20 Sequence Diagram Mencocokkan Pelafalan

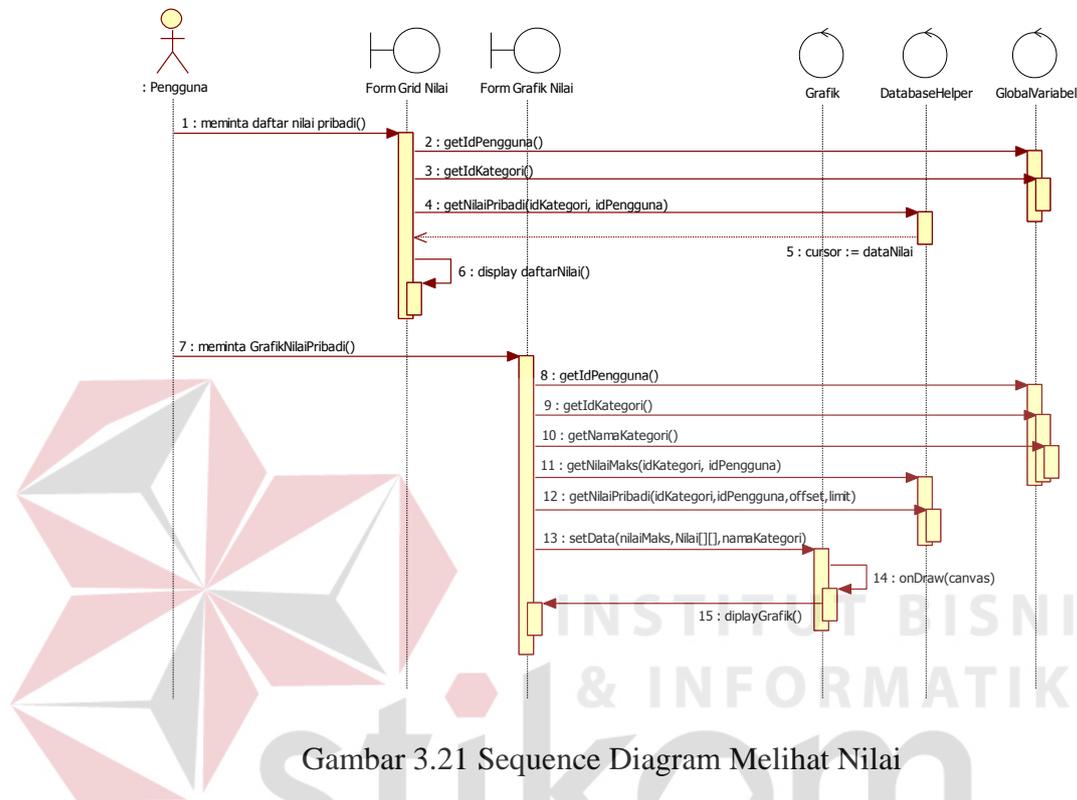
Kontrol *loaditem* akan memproses *item* standar apabila kategori standar dipilih, dimana pada *entity item* standar dibuat *item* standar baru berdasarkan id kategori, setelah itu diambil isi *item* standar tersebut. Kontrol *loaditem* akan memproses *item database* apabila kategori pada *database* dipilih, dan kontrol *databasehelper* mengambil isi *item* berdasarkan id kategori, dimana pada *entity item database* dibuat *item database* baru berdasarkan id kategori, dan diproses untuk menampilkannya. Pengguna memilih fungsi dengar agar kontrol TTS dapat melafalkan kata Inggris dari *item* yang tampil.

Pengguna memilih fungsi lafal dan kontrol *speech input* mencari kemungkinan teks, lalu divalidasi dengan kata Inggris dari *item* yang tampil. Setelah itu bila tidak terdapat kesamaan saat melakukan validasi akan tampil pesan salah serta kemungkinan teks yang dihasilkan kontrol *speech input* dan akan menambah tanda kesalahan. Apabila telah melakukan kesalahan sebanyak tiga kali maka akan ditampilkan *item* selanjutnya. Apabila terdapat kesamaan saat melakukan validasi maka akan menambah nilai benar serta menampilkan *item* selanjutnya, dan bila *item* terakhir telah muncul maka akan menyimpan total nilai serta menampilkan hasil akhir nilai dalam pesan.

#### **F. Sequence Diagram untuk Use Case Melihat Nilai**

Pada gambar 3.21 dimulai dari pengguna meminta daftar nilai pribadi pada *form grid* nilai. Sistem akan mengambil id pengguna dan id kategori pada kontrol *globalvariable* untuk digunakan mengambil nilai pribadi pada kontrol *databasehelper* setelah itu menampilkan dalam bentuk daftar nilai pribadi. Pengguna meminta grafik nilai pribadi pada *form* grafik nilai agar dapat melihat tampilan nilai dalam bentuk grafik. Pada *form* grafik nilai, sistem akan mengambil

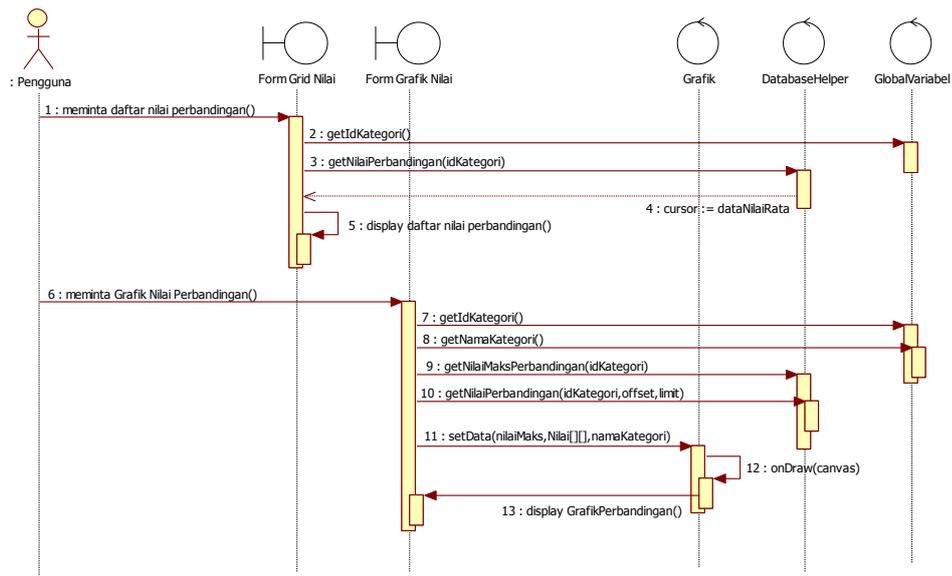
id pengguna, id kategori, dan nama kategori agar dapat digunakan pada kontrol grafik untuk mengolah bentuk grafik pada canvas, setelah itu ditampilkan nilai pribadi dalam bentuk grafik.



Gambar 3.21 Sequence Diagram Melihat Nilai

### G. Sequence Diagram untuk Use Case Melihat Perbandingan Profil

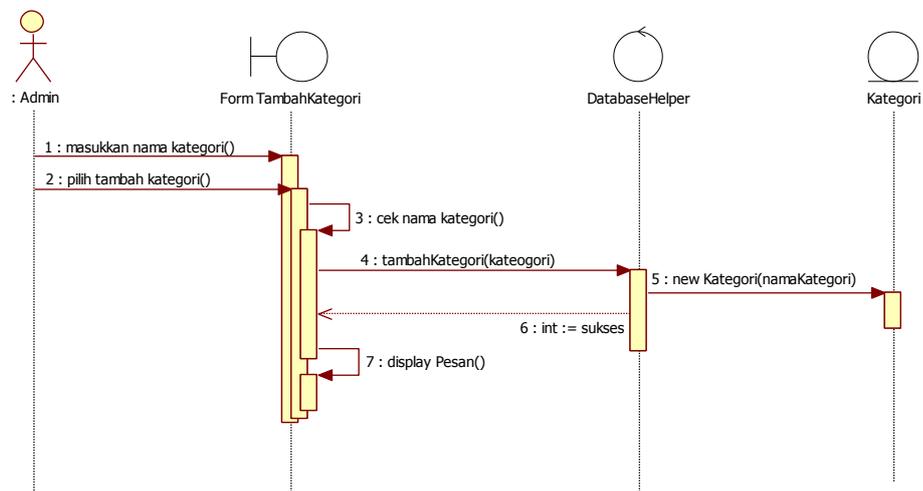
Pada gambar 3.22 dimulai dari pengguna meminta daftar nilai perbandingan pada *form grid* nilai. Sistem akan mengambil id kategori pada kontrol *globalvariable* untuk digunakan mengambil nilai perbandingan pada kontrol *databasehelper* setelah itu menampilkan dalam bentuk daftar nilai rata-rata antar pengguna. Pengguna meminta grafik nilai perbandingan pada *form* grafik nilai agar dapat melihat tampilan nilai dalam bentuk grafik. Pada *form* grafik nilai, sistem akan mengambil id kategori, dan nama kategori agar dapat digunakan pada kontrol grafik untuk mengolah bentuk grafik pada canvas, setelah itu ditampilkan nilai rata-rata antar pengguna dalam bentuk grafik.



Gambar 3.22 Sequence Diagram Melihat Perbandingan Profil

### H. Sequence Diagram untuk Use Case Menambah Kategori

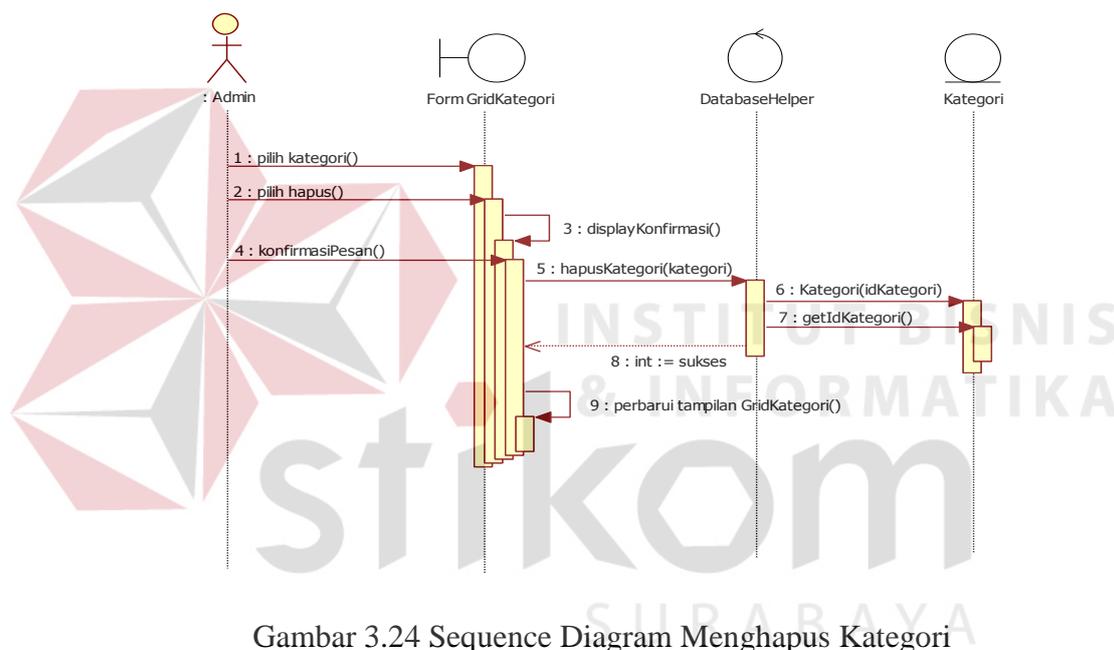
Pada gambar 3.23 dimulai dari admin memasukkan nama kategori dan memilih tambah kategori baru, setelah itu nama kategori akan divalidasi oleh sistem dan jika benar akan dimasukkan ke dalam *database* melalui kontrol *databasehelper*, dimana pada *entity* kategori dibuat kategori baru. *Tab* tambah kategori akan menampilkan pesan setelah melakukan proses tambah kategori.



Gambar 3.23 Sequence Diagram Menambah Kategori

### I. Sequence Diagram untuk Use Case Menghapus Kategori

Pada gambar 3.24 dimulai dari admin memilih kategori yang akan dihapus terlebih dulu, setelah itu sistem menampilkan pesan yang akan dikonfirmasi oleh admin dan jika dikonfirmasi untuk dihapus maka sistem akan menghapus kategori dari *database* melalui kontrol *databasehelper*, dimana pada *entity* kategori diambil id kategori. *Tab grid* kategori akan memperbarui tampilan *grid* kategori setelah melakukan proses hapus kategori.

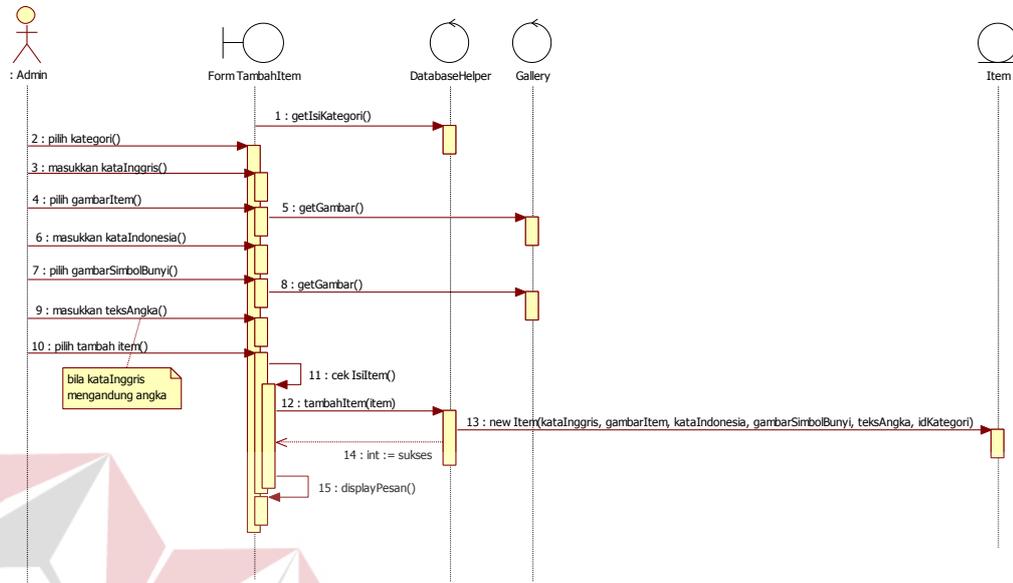


Gambar 3.24 Sequence Diagram Menghapus Kategori

### J. Sequence Diagram untuk Use Case Menambah Item

Pada gambar 3.25 dimulai dari admin memilih kategori yang telah ada pada *database* yang diambil melalui kontrol *databasehelper*, setelah itu memasukkan kata Inggris, kata Indonesia, teks angka, serta gambar *item* dan gambar simbol bunyi atau kata ucap yang diambil melalui kontrol *gallery*. Admin memilih tambah *item* baru, setelah itu isi *item* yang telah dimasukkan akan divalidasi oleh sistem dan jika benar akan dimasukkan ke dalam *database* melalui

kontrol *databasehelper*, dimana pada *entity item* dibuat *item* baru. *Tab* tambah *item* akan menampilkan pesan setelah melakukan proses tambah *item*.



Gambar 3.25 Sequence Diagram Menambah Item

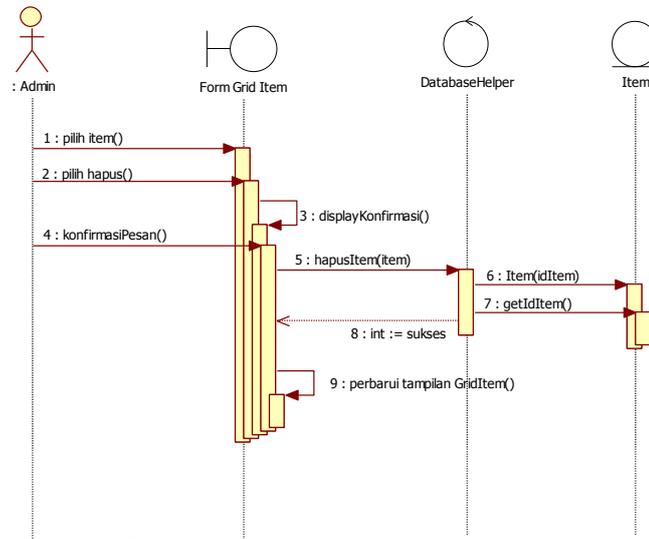
### K. Sequence Diagram untuk Use Case Menghapus Item

Pada gambar 3.26 dimulai dari admin memilih *item* yang akan dihapus terlebih dulu, setelah itu sistem menampilkan pesan yang akan dikonfirmasi oleh admin dan jika dikonfirmasi untuk dihapus maka sistem akan menghapus *item* dari *database* melalui kontrol *databasehelper*, dimana pada *entity item* diambil *id item*. *Tab grid item* akan memperbarui tampilan *grid item* setelah melakukan proses hapus *item*.

#### 3.2.5. Class Diagram Aplikasi Mobile Melafalkan Bahasa Inggris

Berdasarkan perencanaan sistem pada *use case diagram*, dibutuhkan *class-class* untuk membangun dan mendukung jalannya aplikasi. Hubungan antar *class* tersebut dapat digambarkan dalam sebuah *class diagram*. *Class diagram* dari sistem yang dibangun tidak ditampilkan secara keseluruhan, melainkan

ditampilkan secara terpisah sesuai relasi-relasi yang ada untuk mempermudah pembacaan.



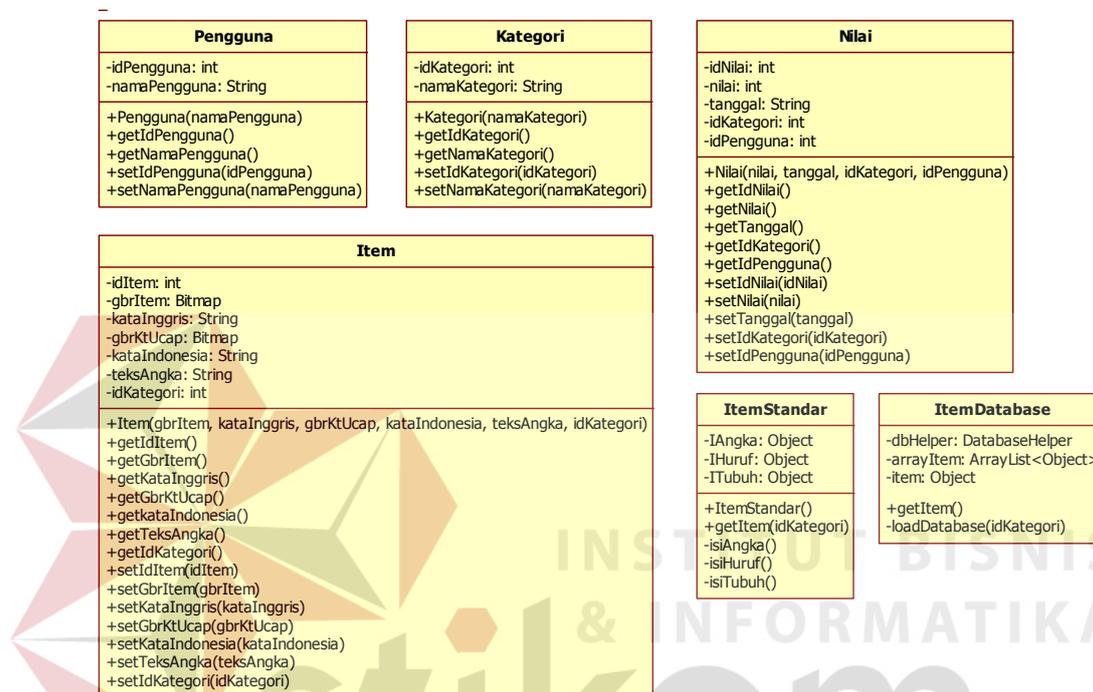
Gambar 3.26 Sequence Diagram Menghapus Item

*Class* yang dibuat untuk aplikasi ini menggunakan pemodelan UML dengan konsep pemodelan *Model-View-Controller* (MVC). *Class Model* merupakan *class* yang akan menangani segala sesuatu yang berhubungan dengan entitas. *Class view* merupakan *class* yang akan menangani segala sesuatu yang berhubungan dengan tampilan *User Interface*. *Class Controller* merupakan *class* yang akan menangani segala proses seperti proses pencarian data dan segala proses yang berhubungan dengan penyimpanan data pada *sqlite3*. *Class* yang dapat digunakan dalam *class* diagram pada aplikasi belajar melafalkan bahasa Inggris yaitu

#### A. Class Model

Pada gambar 3.27, menunjukkan *class* model yang akan digunakan dalam aplikasi belajar melafalkan bahasa Inggris. Pada gambar tersebut terdapat *class* Pengguna yang digunakan untuk menangani data pengguna, *class* Kategori

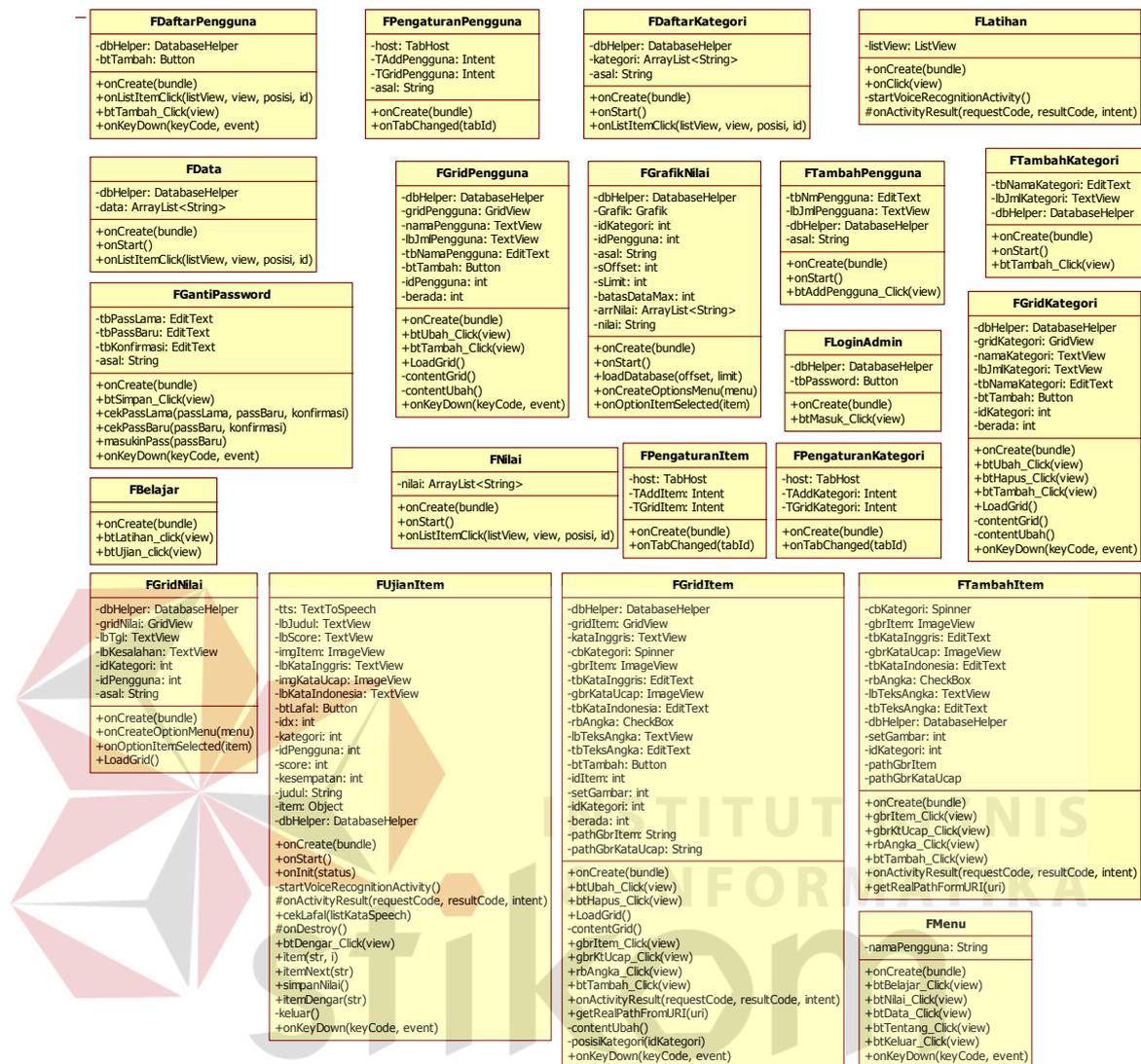
yang digunakan untuk menangani data kategori, *class* Nilai yang digunakan untuk menangani data nilai, *class* item yang digunakan untuk menangani data *item*, *class* ItemStandar yang digunakan untuk menangani data *item* standar dan *class* ItemDatabase yang digunakan untuk menangani data *item* database.



Gambar 3.27 Class Model

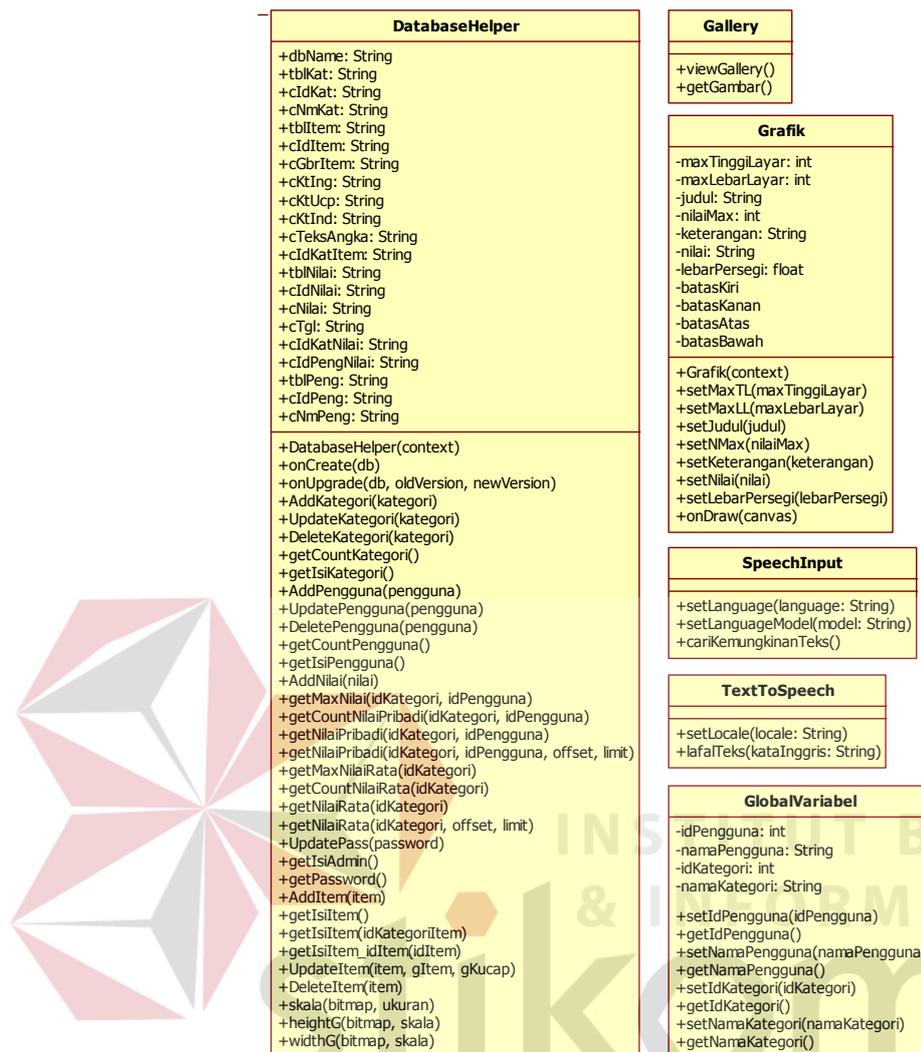
## B. Class View

Pada gambar 3.28, menunjukkan *class view* yang akan digunakan dalam aplikasi belajar melafalkan bahasa Inggris. Pada gambar tersebut terdapat *class* FDaftarPengguna yang digunakan untuk menangani fungsi pada *form* daftar pengguna, *class* FMenu yang digunakan untuk menangani fungsi pada *form* menu, *class* FPengaturanPengguna yang digunakan untuk menangani fungsi pada *form* pengaturan pengguna, *class* FTambahPengguna yang digunakan untuk menangani fungsi pada *tab* tambah pengguna, *class* FGridPengguna yang digunakan untuk menangani fungsi pada *tab* pengguna.



Gambar 3.28 Class View

*Class* FBelajar yang digunakan untuk menangani fungsi pada *form* belajar, *class* FLatihan yang digunakan untuk menangani fungsi pada *form* latihan, *class* FDaftarKategori yang digunakan untuk menangani fungsi pada *form* daftar kategori, *class* FUjianItem yang digunakan untuk menangani fungsi pada *form* ujian item, *class* FNilai yang digunakan untuk menangani fungsi pada *form* nilai, *class* FGridNilai yang digunakan untuk menangani fungsi pada *form* daftar nilai, *class* FGridNilai yang digunakan untuk menangani fungsi pada *form* grafik nilai.



Gambar 3.29 Class Controller

*Class* FData yang digunakan untuk menangani fungsi pada *form* daftar data, *class* FLoginAdmin yang digunakan untuk menangani fungsi pada *form* login admin, *class* FGantiPassword yang digunakan untuk menangani fungsi pada *form* ganti *password*, *class* FPengaturanKategori yang digunakan untuk menangani fungsi pada *form* pengaturan kategori, *class* FTambahKategori yang digunakan untuk menangani fungsi pada *tab* tambah kategori, *class* FGridKategori yang digunakan untuk menangani fungsi pada *tab* kategori, *class* FPengaturanItem yang digunakan untuk menangani fungsi pada *form*



#### D. Relasi Class Diagram Tampilan

Pada gambar 3.30, menampilkan relasi antar *class* model yang menjelaskan alur dari *class* diagram tampilan pada aplikasi belajar melafalkan bahasa Inggris.

#### E. Relasi Class Diagram Proses dan Entitas

Pada gambar 3.31, menunjukkan *class* diagram melihat daftar pengguna, dimana *class* FDaftarPengguna berelasi dengan *class* DatabaseHelper agar dapat menampilkan pengguna yang tersimpan pada *database*.



Gambar 3.31 Class Diagram Melihat Daftar Pengguna

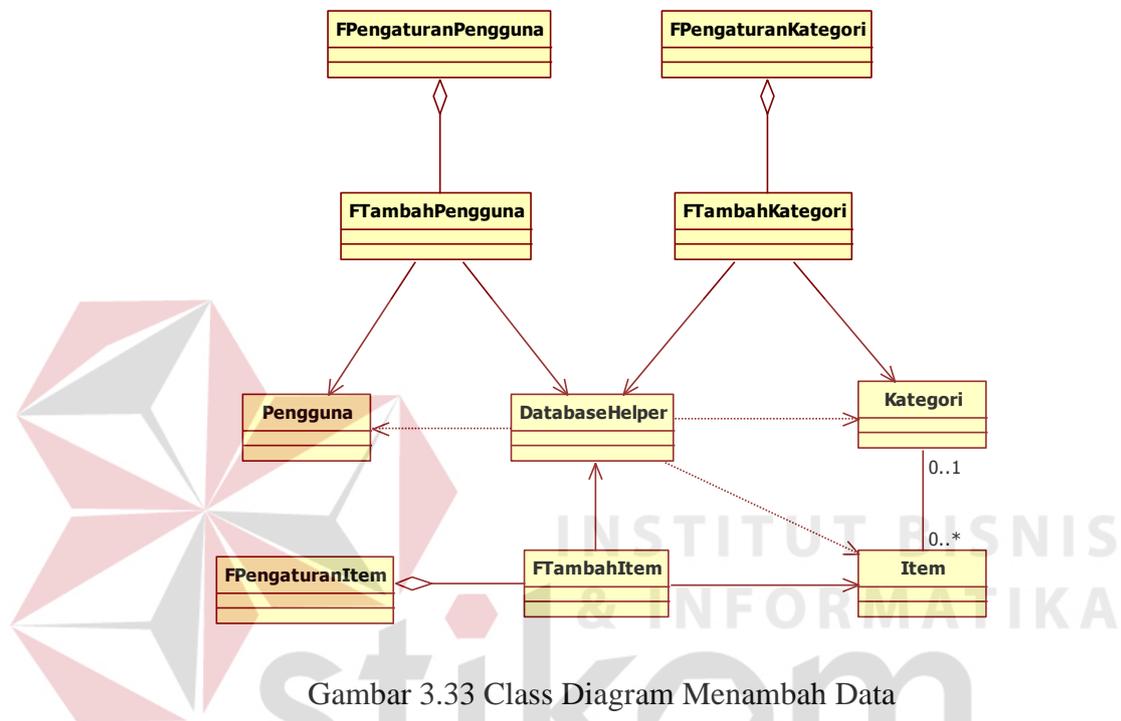
Pada gambar 3.32, menunjukkan *class* diagram mengaktifkan pengguna, dimana *class* FDaftarPengguna berelasi dengan *class* GlobalVariabel agar dapat mengatur id pengguna dan nama pengguna pada *class* GlobalVariabel sebagai penanda pengguna yang aktif saat ini.



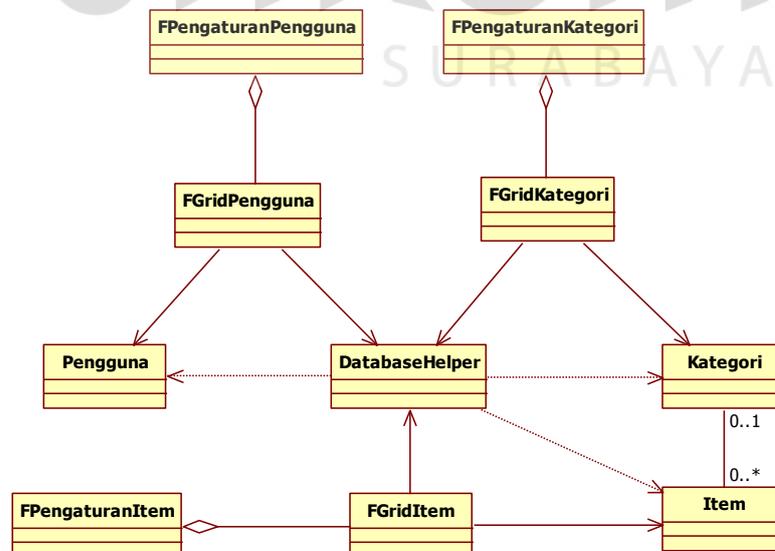
Gambar 3.32 Class Diagram Mengaktifkan Pengguna

Pada gambar 3.33, menunjukkan *class* diagram menambah data, dimana *class* FPengaturanPengguna berelasi dengan *class* FTambahPengguna yang berarti *class* FTambahPengguna adalah bagian dari *class* FPengaturanPengguna, agar pada aplikasi, *class* FTambahPengguna dapat dijadikan *tab* tambah pengguna pada *form* pengaturan pengguna. Relasi antara *class* FTambahPengguna dengan *class* Pengguna dimaksudkan agar isian yang dibutuhkan untuk membuat

pengguna baru dapat ditampung terlebih dulu. Relasi antara *class* FTambahPengguna dengan *class* DatabaseHelper dan *class* Pengguna dimaksudkan agar hasil tampilan pada *class* Pengguna, dapat digunakan *class* DatabaseHelper untuk menyimpan kedalam *database*.

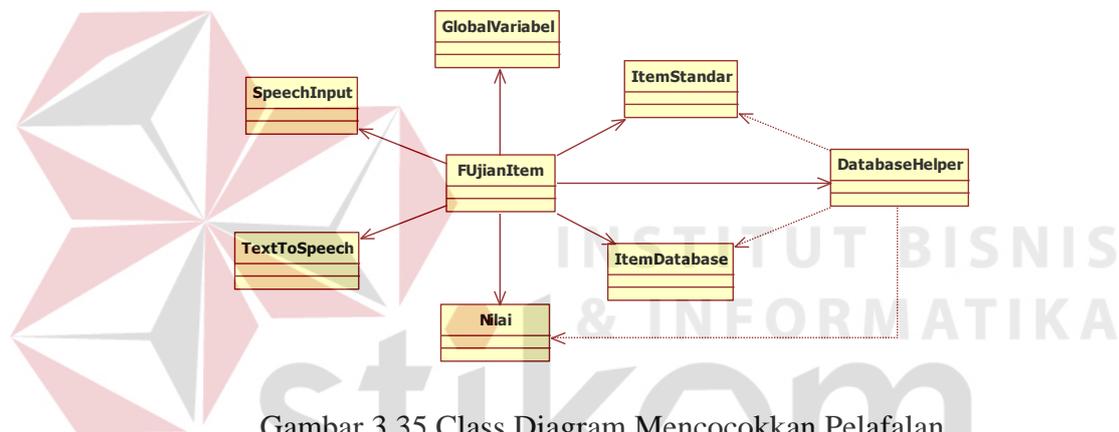


Gambar 3.33 Class Diagram Menambah Data



Gambar 3.34 Class Diagram Menghapus Data

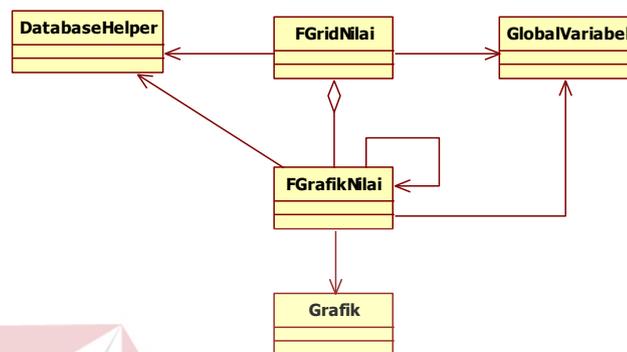
Relasi antara *class* FPengarutanKategori, *class* FTambahKategori, *class* Kategori dan *class* DatabaseHelper sama halnya dengan menangani fungsi tambah pengguna, begitu juga untuk menangani fungsi tambah item yang ditangani oleh relasi antara *class* FPengaturanItem, *class* FTambahItem, *class* Item dan *class* DatabaseHelper. Relasi antara *class* Kategori dengan *class* Item dilakukan untuk menunjukkan hubungan antara *item* dan kategori, dimana kategori boleh tidak memiliki *item* dan sebuah *item* maksimal memiliki satu kategori.



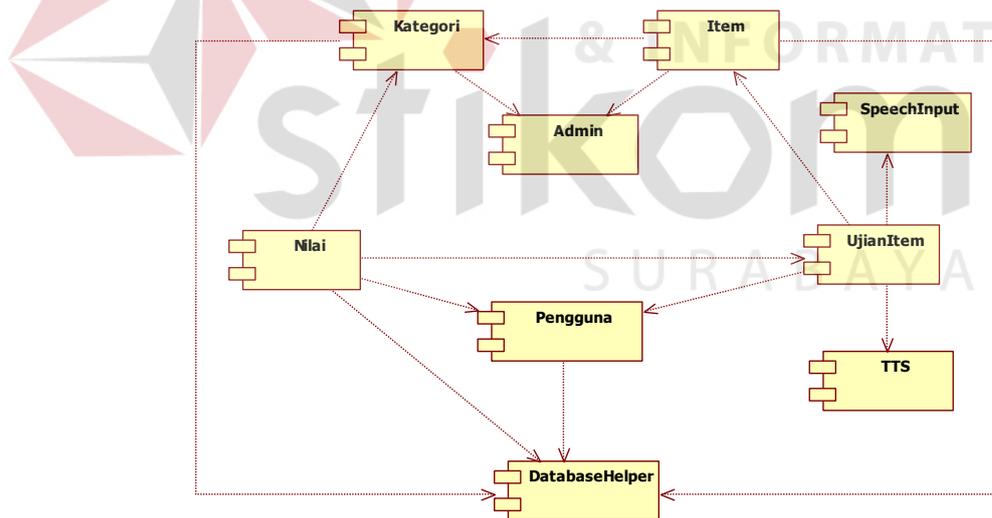
Gambar 3.35 Class Diagram Mencocokkan Pelafalan

Pada gambar 3.34, menunjukkan *class* diagram menghapus data, dimana *class* FPengaturanPengguna berelasi dengan *class* FGridPengguna yang berarti *class* FGridPengguna adalah bagian dari *class* FPengaturanPengguna, agar pada aplikasi, *class* FGridPengguna dapat dijadikan *tab* pengguna pada *form* pengaturan pengguna. Relasi antara *class* FGridPengguna dengan *class* Pengguna dimaksudkan agar isian yang dibutuhkan untuk menghapus pengguna dapat ditampilkan terlebih dulu. Relasi antara *class* FGridPengguna dengan *class* DatabaseHelper dan *class* Pengguna dimaksudkan agar hasil tampilan pada *class* Pengguna, dapat digunakan *class* DatabaseHelper untuk menghapus data pengguna pada *database*.

Relasi antara *class* FPengaturanKategori, *class* FGridKategori, *class* Kategori dan *class* DatabaseHelper sama halnya dengan menangani fungsi hapus pengguna, begitu juga untuk menangani fungsi hapus item yang ditangani oleh relasi antara *class* FPengaturanItem, *class* FGridItem, *class* Item dan *class* DatabaseHelper.



Gambar 3.36 Class Diagram Melihat Nilai

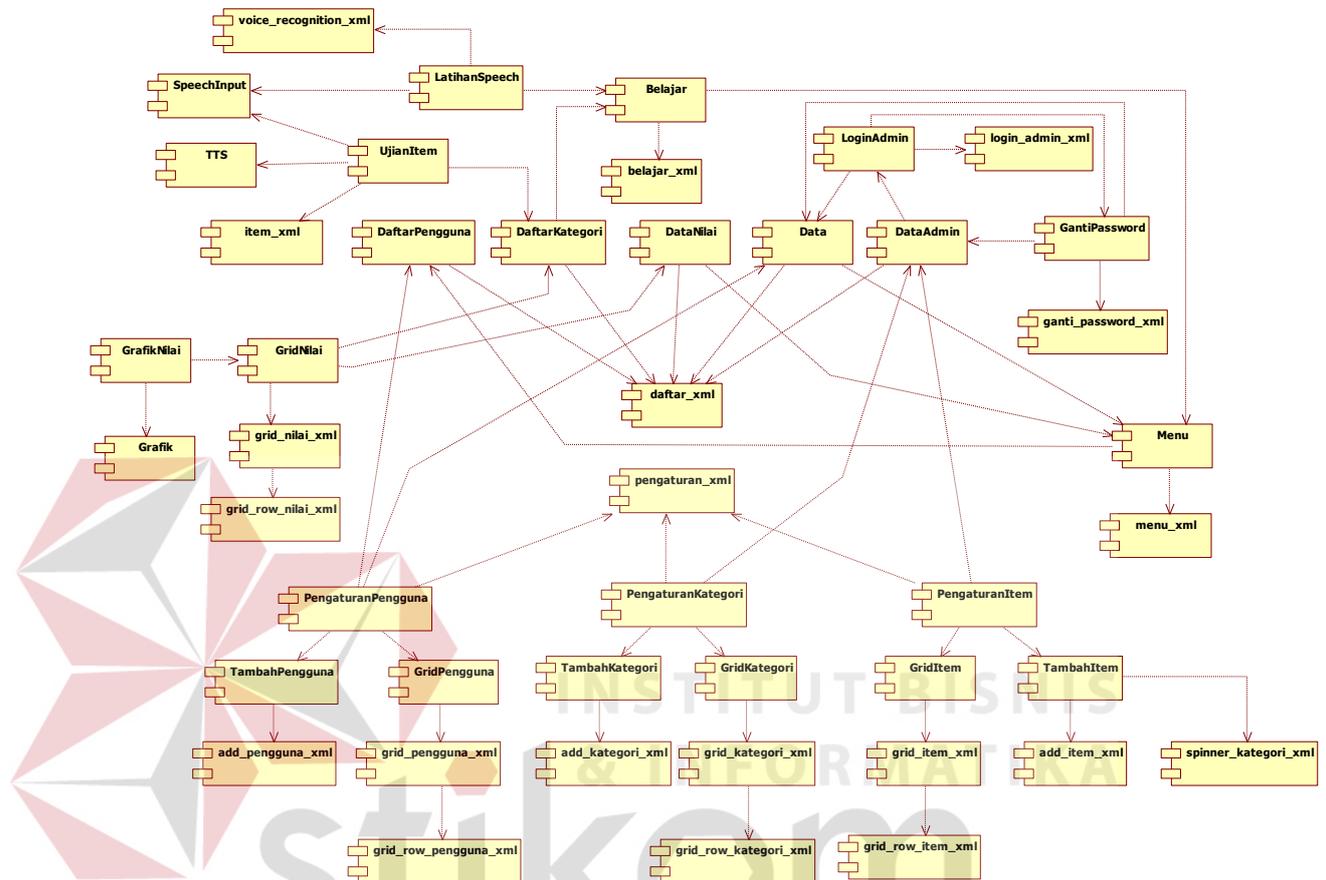


Gambar 3.37 Komponen Diagram Aplikasi Belajar Melafalkan

Bahasa Inggris

Pada gambar 3.35, menunjukkan *class* diagram mencocokkan pelafalan, dimana *class* FUjianItem berelasi dengan *class* GlobalVariabel, *class* ItemStandar, *class* ItemDatabase, *class* Nilai, *class* TextToSpeech, *class* SpeechInput, dan

*class* DatabaseHelper agar dapat melakukan fungsi mencocokkan pelafalan dan menyimpan nilai akhir kedalam *database*.



Gambar 3.38 Detail Komponen Diagram Aplikasi Belajar Melafalkan Bahasa Inggris

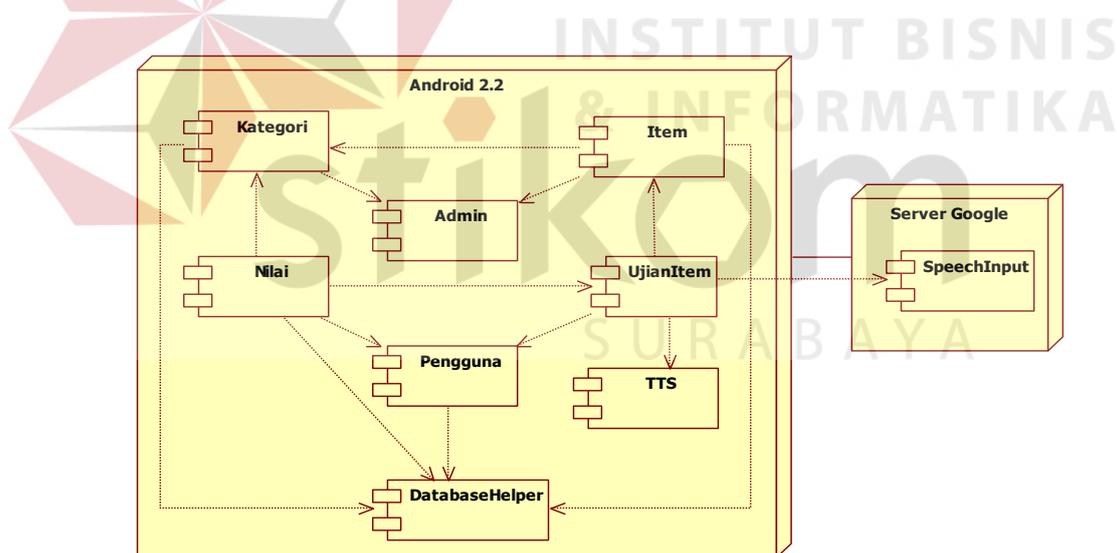
Pada gambar 3.36, menunjukkan *class* diagram melihat nilai, dimana *class* FGratik berelasi dengan *class* DatabaseHelper dan *class* GlobalVariabel agar dapat menggunakan data nilai dan ditampilkan pada daftar nilai akhir. Relasi antara *class* FGridNilai dengan *class* FGratikNilai dimaksudkan agar *class* FGratikNilai menjadi bagian dari *class* FGridNilai. Relasi antara *class* FGratikNilai dengan *class* Grafik, *class* DatabaseHelper dan *class* GlobalVariabel agar dapat menggunakan data nilai dan ditampilkan kedalam bentuk grafik. Relasi *class* FGratikNilai kepada dirinya sendiri dimaksudkan

untuk menangani fungsi melihat grafik nilai tiap halaman dikarenakan tiap halaman grafik hanya menampilkan beberapa data saja.

### 3.2.6. Component Diagram Aplikasi Mobile Melafalkan Bahasa Inggris

Pada gambar 3.37, menunjukkan komponen diagram aplikasi belajar melafalkan bahasa Inggris yang berguna untuk mempermudah pengembang untuk mengetahui urutan *compiler* dan *runtime*.

Pada gambar 3.38, menunjukkan detail komponen diagram aplikasi belajar melafalkan bahasa Inggris, dimana terdapat komponen yang dapat digunakan oleh lebih dari satu komponen, seperti komponen `daftar_xml`, yang digunakan oleh komponen `daftar pengguna`, `komponen daftar kategori`, `komponen data nilai`, `komponen data` dan `komponen data admin`.



Gambar 3.39 Deployment Diagram Aplikasi Belajar Melafalkan Bahasa Inggris

### 3.2.7. Deployment Diagram Aplikasi Mobile Melafalkan Bahasa Inggris

Pada gambar 3.39, menunjukkan *deployment* diagram aplikasi belajar melafalkan bahasa Inggris yang berguna untuk menampilkan rancangan fisik jaringan, dimana untuk mengetahui dimana tempat komponen diletakkan.

### 3.3. Perancangan User Interface

Pembuatan tampilan sangat diperlukan agar pengguna dapat berinteraksi dengan sistem, sehingga dibutuhkan perancangan secara detil mengenai tampilan aplikasi berdasarkan informasi yang ditampilkan pada layar *handphone*. Tampilan yang akan dibuat adalah tampilan daftar pengguna, pengaturan pengguna, menu utama, belajar, latihan, daftar kategori, *item*, nilai, daftar nilai pribadi, grafik nilai pribadi, daftar nilai perbandingan, grafik nilai perbandingan, menu data, ganti *password*, *login* admin, menu admin, pengaturan kategori, dan pengaturan *item*.

#### 3.3.1. Tampilan Daftar Pengguna



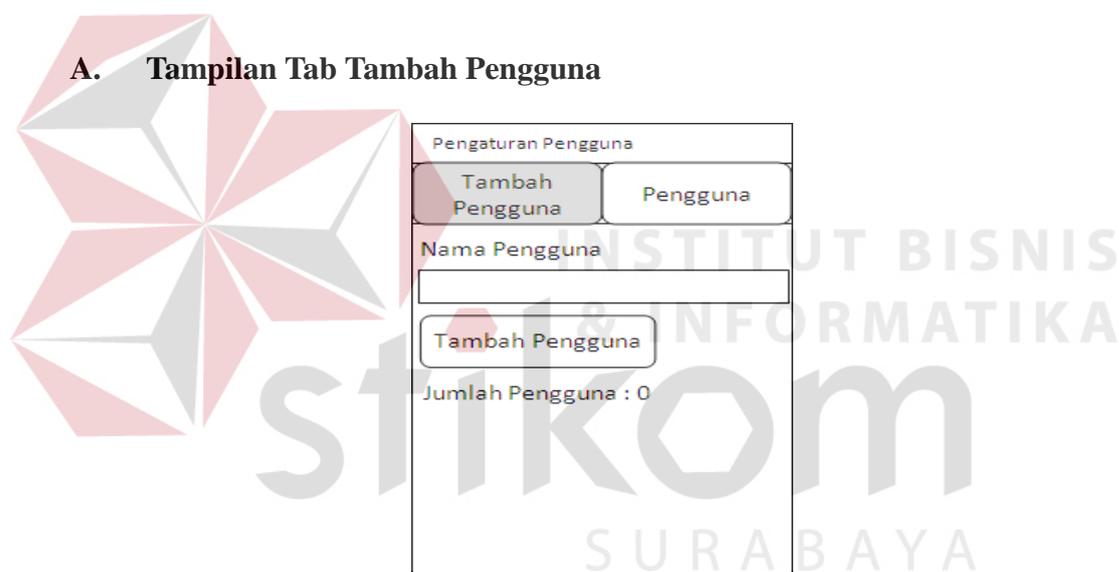
Gambar 3.40 Rancangan Tampilan Daftar Pengguna

Tampilan daftar pengguna akan tampil pertama kali pada saat aplikasi dijalankan. Terdiri dari *list view* nama pengguna, dan tombol tambah pengguna. *List view* nama pengguna menampilkan nama pengguna yang telah terdaftar dan tersimpan pada *database*. Untuk menggunakan aplikasi pengguna diharuskan mempunyai nama pengguna sendiri, apabila belum mempunyai nama pengguna maka pengguna diharuskan menambah pengguna baru dengan menekan tombol

tambah pengguna, dimana akan dialihkan ke tampilan pengaturan pengguna. Rancangan tampilan daftar pengguna dapat dilihat pada gambar 3.40.

### 3.3.2. Tampilan Pegaturan Pengguna

Tampilan pengaturan pengguna akan tampil setelah menekan tombol tambah pada daftar pengguna atau melalui menu data untuk mengatur pengguna. Terdiri dari dua buah *tab* yaitu *tab* tambah pengguna yang berfungsi untuk menambah pengguna baru dan *tab* daftar pengguna yang berfungsi untuk melihat daftar pengguna yang telah terdaftar.

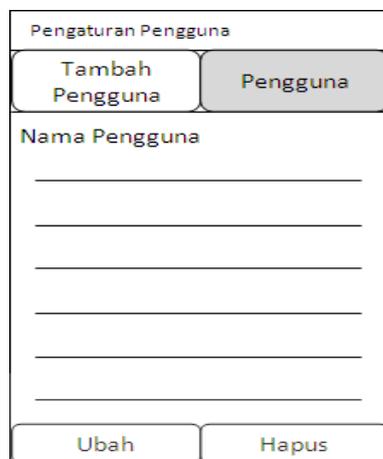


Gambar 3.41 Rancangan Tampilan Tab Tambah Pengguna

Tampilan *tab* tambah pengguna akan tampil sebagai *tab* pertama pada pengaturan pengguna, yang berfungsi untuk menambah pengguna baru. Terdiri dari *edit text* nama pengguna yang berfungsi untuk menerima masukkan pengguna berupa nama pengguna, *text view* jumlah pengguna yang berfungsi untuk menampilkan jumlah pengguna yang telah terdaftar dan tombol tambah pengguna untuk menyimpan nama pengguna kedalam *database*, dimana akan divalidasi terlebih dahulu untuk menjaga agar data yang ada dalam *database*

tetap valid. Rancangan tampilan *tab* tambah pengguna dapat dilihat pada gambar 3.41.

### B. Tampilan Tab Daftar Pengguna



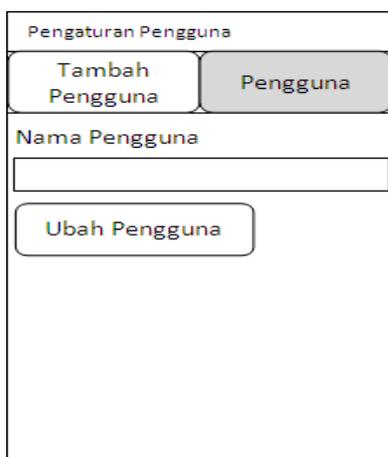
Gambar 3.42 Rancangan Tampilan Tab Daftar Pengguna

Tampilan *tab* daftar pengguna akan tampil sebagai *tab* kedua pada pengaturan pengguna, dimana dapat diakses dengan menekan *tab* tersebut atau akan tampil setelah pengguna menambahkan pengguna baru. Terdiri dari *grid view* pengguna yang berfungsi untuk menampilkan nama pengguna yang tersimpan pada *database*, tombol ubah untuk mengalihkan ke *tab* ubah pengguna untuk mengubah isi data pengguna dan tombol hapus untuk menghapus data pengguna yang telah dipilih pada *grid view* pengguna. Rancangan tampilan *tab* daftar pengguna dapat dilihat pada gambar 3.42.

### C. Tampilan Tab Ubah Pengguna

Tampilan *tab* ubah pengguna akan tampil setelah menekan tombol ubah pada *tab* daftar pengguna. Terdiri dari *edit text* nama pengguna yang berfungsi untuk mengubah isian nama pengguna, dan tombol ubah pengguna yang berfungsi untuk mengubah data pengguna pada *database*, dimana sebelumnya divalidasi

terlebih dahulu. Rancangan tampilan *tab* ubah pengguna dapat dilihat pada gambar 3.43.



The image shows a wireframe for a user management screen titled 'Pengaturan Pengguna'. At the top, there are two buttons: 'Tambah Pengguna' and 'Pengguna'. Below these is a text input field labeled 'Nama Pengguna'. Underneath the input field is a button labeled 'Ubah Pengguna'.

Gambar 3.43 Rancangan Tampilan Tab Ubah Pengguna

### 3.3.3. Tampilan Menu Utama



The image shows a wireframe for a main menu screen titled 'Menu'. Below the title, the word 'MENU' is centered. There are six buttons stacked vertically: 'Belajar', 'Nilai', 'Data', 'Cara Menggunakan', 'Tentang', and 'Keluar'.

Gambar 3.44 Rancangan Tampilan Menu Utama

Tampilan menu utama akan tampil setelah pengguna memilih nama pengguna yang telah terdaftar atau tersimpan pada *database*. Terdiri dari tombol ‘belajar’ yang berfungsi untuk mengalihkan pada tampilan menu belajar, ‘nilai’ yang berfungsi untuk mengalihkan pada tampilan menu nilai, ‘data’ yang berfungsi untuk mengalihkan pada tampilan menu data, ‘cara menggunakan’ yang berfungsi untuk mengalihkan pada tampilan cara menggunakan, ‘tentang’ yang

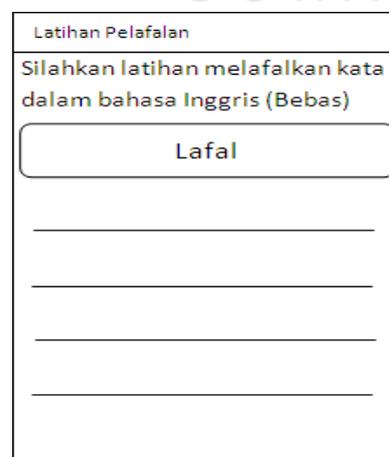
berfungsi untuk mengalihkan pada tampilan tentang, dan ‘keluar’ yang berfungsi untuk mengakhiri aplikasi. Rancangan tampilan menu utama dapat dilihat pada gambar 3.44.

### 3.3.4. Tampilan Menu Belajar



Gambar 3.45 Rancangan Tampilan Menu Belajar

Tampilan menu belajar terdiri dari dua buah tombol yaitu tombol latihan yang berfungsi untuk mengalihkan pada tampilan latihan, dan tombol ujian yang berfungsi untuk mengalihkan pada daftar kategori. Rancangan tampilan menu belajar dapat dilihat pada gambar 3.45.



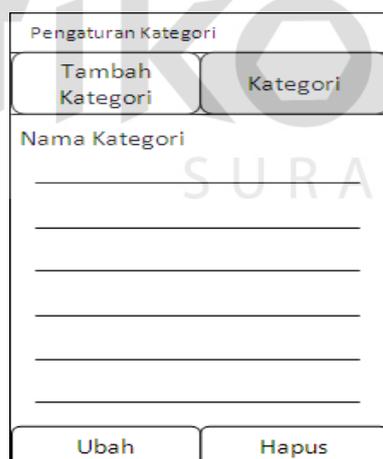
Gambar 3.46 Rancangan Tampilan Latihan

### 3.3.5. Tampilan Latihan

Tampilan latihan difungsikan untuk melatih pelafalan pengguna secara bebas, dimana tidak ada batasan harus mengucapkan kata tertentu. Terdiri dari tombol lafal yang berfungsi untuk mengaktifkan permintaan suara pengguna dan *list view* hasil yang berfungsi untuk menampilkan hasil identifikasi sistem terhadap pelafalan pengguna. Rancangan tampilan latihan dapat dilihat pada gambar 3.46.

### 3.3.6. Tampilan Daftar Kategori

Tampilan daftar kategori akan menampilkan kategori standar dan kategori yang tersimpan didalam *database* pada *list view* kategori. Kategori yang dipilih akan berpengaruh kepada *item* yang akan ditampilkan pada tampilan *item* atau berpengaruh terhadap nilai pada tampilan daftar nilai. Rancangan tampilan daftar kategori dapat dilihat pada gambar 3.47.



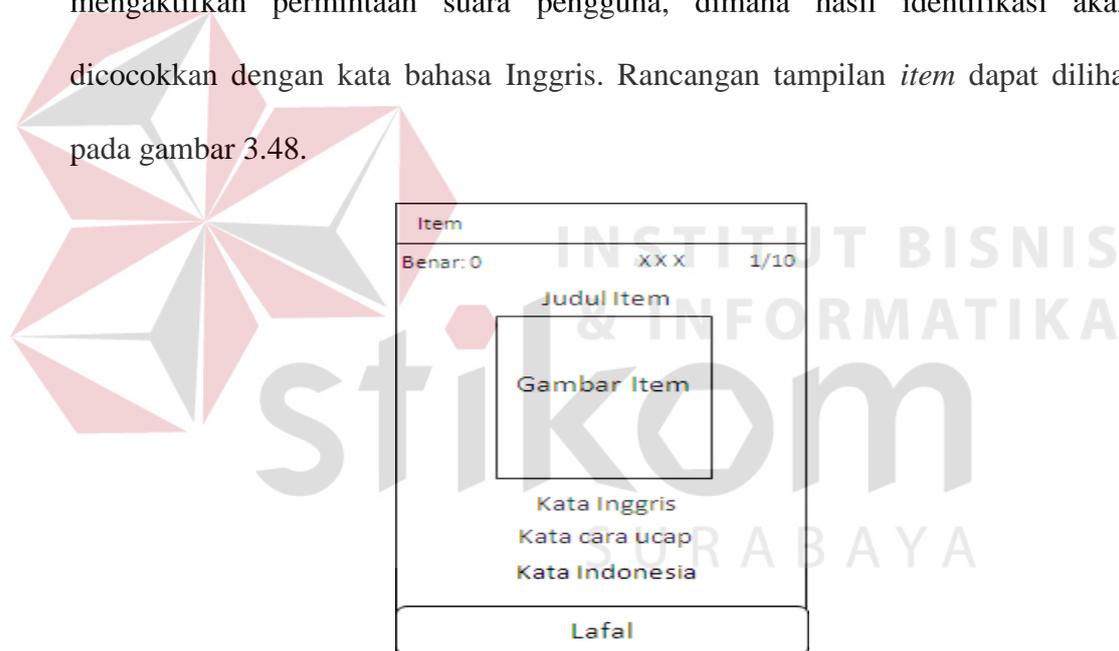
Pengaturan Kategori	
Tambah Kategori	Kategori
Nama Kategori	
_____	
_____	
_____	
_____	
_____	
Ubah	Hapus

Gambar 3.47 Rancangan Tampilan Daftar Kategori

### 3.3.7. Tampilan Item

Tampilan *item* akan menampilkan *item* berdasarkan kategori yang telah dipilih sebelumnya. Terdiri dari *text view* nilai yang berfungsi untuk menampilkan

banyak jawaban benar, *text view* salah yang berfungsi untuk menampilkan tanda salah, *text view* soal yang berfungsi untuk menampilkan *item* ke- dari seluruh *item*, *image view* gambar *item* yang berfungsi untuk menampilkan gambar *item* serta jika ditekan akan melafalkan kata bahasa Inggris, *text view* kata bahasa Inggris yang berfungsi untuk menampilkan kata bahasa Inggris, *image view* kata ucap yang berfungsi untuk menampilkan simbol bunyi atau cara pengucapan kata bahasa Inggris, *text view* kata bahasa Indonesia yang berfungsi untuk menampilkan kata bahasa Indonesia, dan tombol lafal yang berfungsi untuk mengaktifkan permintaan suara pengguna, dimana hasil identifikasi akan dicocokkan dengan kata bahasa Inggris. Rancangan tampilan *item* dapat dilihat pada gambar 3.48.



Gambar 3.48 Rancangan Tampilan Item

### 3.3.8. Tampilan Menu Nilai

Tampilan menu nilai menampilkan dua pilihan yaitu pribadi yang berfungsi untuk melihat nilai pribadi pengguna dan perbandingan yang berfungsi untuk melihat nilai perbandingan rata-rata antar pengguna, dimana harus memilih

kategori terlebih dahulu. Rancangan tampilan menu nilai dapat dilihat pada gambar 3.49.

Nilai
Pribadi
Perbandingan

Gambar 3.49 Rancangan Tampilan Menu Nilai

### 3.3.9. Tampilan Daftar Nilai Pribadi

Data Nilai	
Tanggal	Kesalahan
Ubah	Hapus

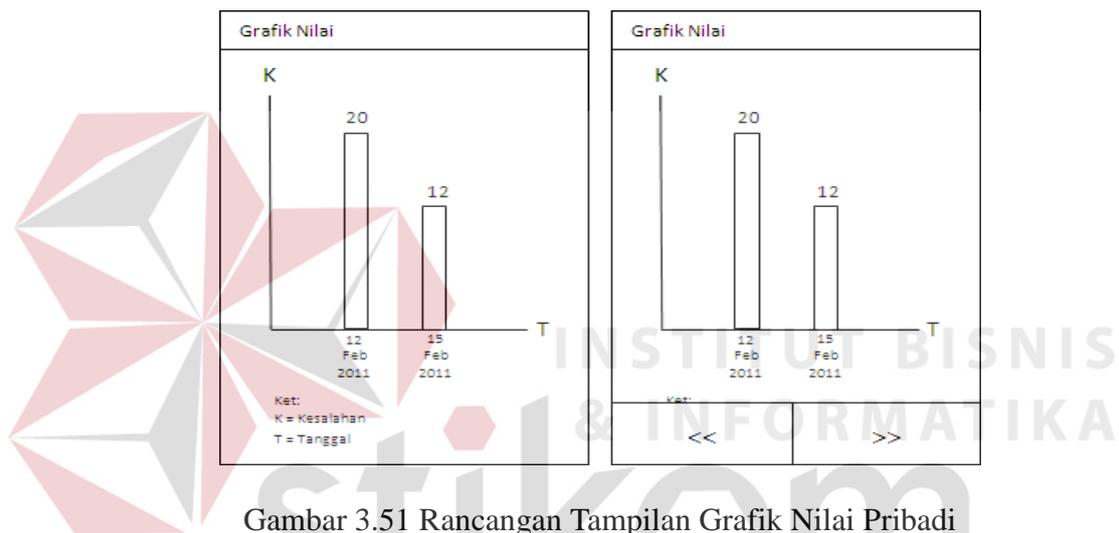
Data Nilai	
Tanggal	Kesalahan
Grafik Nilai	

Gambar 3.50 Rancangan Tampilan Daftar Nilai Pribadi

Tampilan daftar nilai pribadi menampilkan nilai pribadi pengguna berdasarkan kategori yang dipilih. Terdiri dari *grid view* nilai yang berfungsi untuk menampilkan tanggal pelaksanaan ujian dan nilai berdasarkan kategori dan *menu option* grafik yang berfungsi untuk mengalihkan pada tampilan grafik nilai pribadi. Rancangan tampilan daftar nilai pribadi dapat dilihat pada gambar 3.50.

### 3.3.10. Tampilan Grafik Nilai Pribadi

Tampilan grafik nilai pribadi menampilkan nilai kedalam bentuk grafik. Terdiri dari *canvas* grafik yang menampilkan nilai serta tanggal ujian kedalam bentuk grafik, *menu option* sebelum yang berfungsi untuk menampilkan grafik nilai sebelumnya dan *menu option* sesudah yang berfungsi untuk menampilkan grafik nilai sesudahnya. Rancangan tampilan grafik nilai pribadi dapat dilihat pada gambar 3.51.



Gambar 3.51 Rancangan Tampilan Grafik Nilai Pribadi

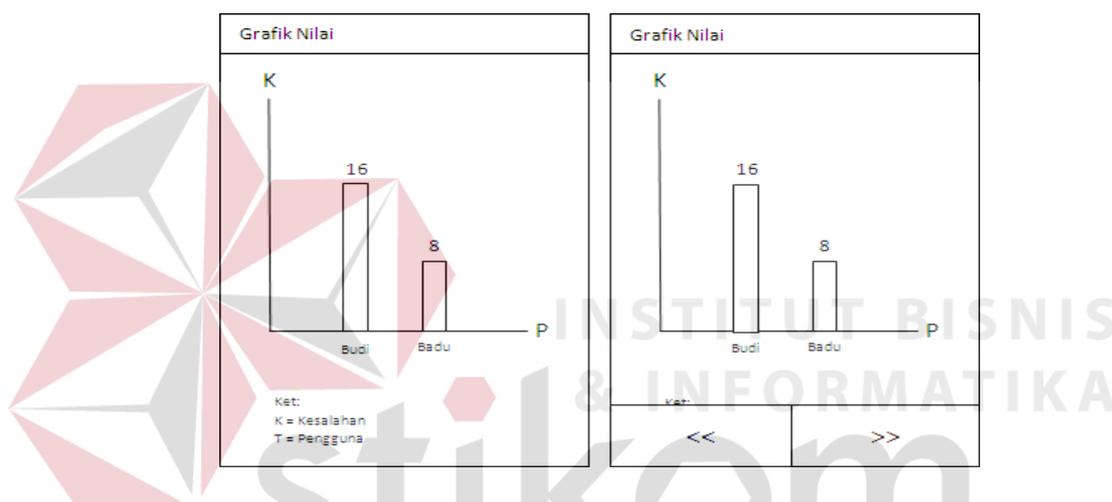
### 3.3.11. Tampilan Daftar Nilai Perbandingan

Data Nilai		Data Nilai	
Nama Pengguna	Kesalahan rata-rata	Nama Pengguna	Kesalahan rata-rata
_____	_____	_____	_____
_____	_____	_____	_____
_____	_____	_____	_____
_____	_____	_____	_____
_____	_____	_____	_____
_____	_____	_____	_____
_____	_____	_____	_____
_____	_____	_____	_____
Ubah	Hapus	Grafik Nilai	

Gambar 3.52 Rancangan Tampilan Daftar Nilai Perbandingan

Tampilan daftar nilai perbandingan menampilkan nilai perbandingan rata-rata antar pengguna berdasarkan kategori yang dipilih. Terdiri dari *grid view* nilai yang berfungsi untuk menampilkan nama pengguna dan nilai rata-rata berdasarkan kategori dan *menu option* grafik yang berfungsi untuk mengalihkan pada tampilan grafik nilai perbandingan. Rancangan dapat dilihat pada gambar 3.52.

### 3.3.12. Tampilan Grafik Nilai Perbandingan



Gambar 3.53 Rancangan Tampilan Grafik Nilai Perbandingan

Tampilan grafik nilai perbandingan menampilkan nilai rata-rata antar pengguna kedalam bentuk grafik. Terdiri dari *canvas* grafik yang menampilkan nilai rata-rata serta nama pengguna kedalam bentuk grafik, *menu option* sebelum yang berfungsi untuk menampilkan grafik nilai rata-rata sebelumnya dan *menu option* sesudah yang berfungsi untuk menampilkan grafik nilai rata-rata sesudahnya. Rancangan dapat dilihat pada gambar 3.53.

### 3.3.13. Tampilan Menu Data

Tampilan menu data menampilkan dua pilihan yaitu pengguna yang berfungsi untuk mengalihkan pada tampilan pengaturan pengguna dan admin

yang berfungsi untuk mengakses pengaturan admin, dimana akan diminta untuk memasukkan *password* admin terlebih dahulu. Rancangan tampilan menu data dapat dilihat pada gambar 3.54.

Data
Pengguna
Admin

Gambar 3.54 Rancangan Tampilan Menu Data

### 3.3.14. Tampilan Pengaturan Kategori

Tampilan pengaturan kategori terdiri dari dua buah *tab* yaitu *tab* tambah kategori yang berfungsi untuk menambah kategori baru dan *tab* daftar kategori yang berfungsi untuk melihat daftar kategori yang telah tersimpan.

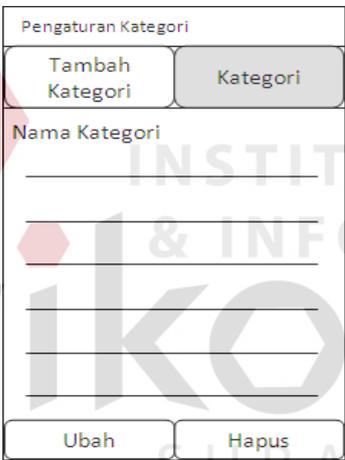
#### A. Tampilan Tab Tambah Kategori

Pengaturan Kategori	
Tambah Kategori	Kategori
Nama Kategori	
<input type="text"/>	
Tambah Kategori	
Jumlah Kategori : 0	

Gambar 3.55 Rancangan Tampilan Tab Tambah Kategori

Tampilan *tab* tambah kategori akan tampil sebagai *tab* pertama pada pengaturan kategori, yang berfungsi untuk menambah kategori baru. Terdiri dari *edit text* nama kategori yang berfungsi untuk menerima isian berupa nama kategori, *text view* jumlah kategori yang berfungsi untuk menampilkan jumlah kategori yang telah tersimpan dan tombol tambah kategori untuk menyimpan nama kategori kedalam *database*, dimana akan divalidasi terlebih dahulu untuk menjaga agar data yang ada dalam *database* tetap valid. Rancangan tampilan *tab* tambah kategori dapat dilihat pada gambar 3.55.

### B. Tampilan Tab Daftar Kategori



The image shows a mobile application interface for category management. It features a large red and white geometric logo on the left. The main content area is titled 'Pengaturan Kategori' and contains a 'Tambah Kategori' button, a 'Kategori' text input field, and a 'Nama Kategori' section with four horizontal lines for text entry. At the bottom, there are 'Ubah' and 'Hapus' buttons. A faint watermark for 'stikom SURABAYA' is visible in the background.

Gambar 3.56 Rancangan Tampilan Tab Daftar Kategori

Tampilan *tab* daftar kategori akan tampil sebagai *tab* kedua pada pengaturan kategori, dimana dapat diakses dengan menekan *tab* tersebut atau akan tampil setelah admin menambahkan kategori baru. Terdiri dari *grid view* kategori yang berfungsi untuk menampilkan nama kategori yang tersimpan pada *database*, tombol ubah untuk mengalihkan ke *tab* ubah kategori untuk mengubah isi data kategori dan tombol hapus untuk menghapus data

kategori yang telah dipilih pada *grid view* kategori. Rancangan tampilan *tab* daftar kategori dapat dilihat pada gambar 3.56.

### C. Tampilan Tab Ubah Kategori

Gambar 3.57 Rancangan Tampilan Tab Ubah Kategori

Tampilan *tab* ubah kategori akan tampil setelah menekan tombol ubah pada *tab* daftar kategori. Terdiri dari *edit text* nama kategori yang berfungsi untuk mengubah isian nama kategori, dan tombol ubah kategori yang berfungsi untuk mengubah data kategori pada *database*, dimana sebelumnya divalidasi terlebih dahulu. Rancangan tampilan *tab* ubah kategori dapat dilihat pada gambar 3.57.

### 3.3.15. Tampilan Pengaturan Item

Gambar 3.58 Tampilan Tab Tambah Item

Tampilan pengaturan *item* terdiri dari dua buah *tab* yaitu *tab* tambah *item* yang berfungsi untuk menambah *item* baru dan *tab* daftar *item* yang berfungsi untuk melihat daftar *item* yang telah tersimpan

#### A. Tampilan Tab Tambah Item

Tampilan *tab* tambah *item* akan tampil sebagai *tab* pertama pada pengaturan *item*, yang berfungsi untuk menambah *item* baru. Terdiri dari *spinner* kategori yang berfungsi untuk menampilkan kategori yang telah ditambahkan kedalam *database*, *image view* gambar *item* yang berfungsi untuk menampilkan gambar *item* yang dipilih pada *gallery*, *edit text* kata dalam bahasa Inggris yang berfungsi untuk menerima masukkan berupa kata dalam bahasa Inggris, *image view* gambar kata ucap yang berfungsi untuk menampilkan gambar kata ucap yang dipilih pada *gallery*, *checkbox* terdapat angka dalam kata yang berfungsi untuk menampilkan *text view* dan *edit text* teks angka Inggris yang berfungsi untuk menerima masukkan berupa kombinasi teks dan angka, *edit text* teks angka Inggris yang berfungsi untuk menerima masukkan teks dan angka dalam bahasa Inggris serta tombol tambah *item* untuk menyimpan *item* kedalam *database*, dimana akan divalidasi terlebih dahulu untuk menjaga agar data yang ada dalam *database* tetap valid. Rancangan tampilan *tab* tambah *item* dapat dilihat pada gambar 3.58.

#### B. Tampilan Tab Daftar Item

Tampilan *tab* daftar *item* akan tampil sebagai *tab* kedua pada pengaturan *item*, dimana dapat diakses dengan menekan *tab* tersebut atau akan tampil setelah admin menambahkan *item* baru. Terdiri dari *grid view item* yang berfungsi untuk



gambar *item*, *edit text* kata bahasa Inggris yang berfungsi untuk mengubah isian kata bahasa Inggris, *image view* gambar kata ucap yang berfungsi untuk mengubah gambar kata ucap, *check box* terdapat angka dalam kata yang berfungsi untuk menampilkan *text view* dan *edit text* teks angka Inggris, *edit text* teks angka Inggris yang berfungsi untuk mengubah isian teks dan angka dalam bahasa Inggris serta tombol ubah *item* yang berfungsi untuk mengubah data *item* pada *database*, dimana sebelumnya divalidasi terlebih dahulu. Rancangan tampilan *tab* ubah *item* dapat dilihat pada gambar 3.60.

### 3.4. Desain Uji Coba

Desain uji coba bertujuan untuk memastikan bahwa aplikasi telah dibuat dengan benar sesuai dengan kebutuhan atau tujuan yang diharapkan. Kekurangan atau kelemahan sistem pada tahap ini akan dievaluasi sebelum diimplementasikan secara nyata. Uji coba yang akan dilakukan antara lain:

1. Uji coba fungsi aplikasi
2. Uji coba aplikasi kepada pengguna

#### 3.4.1. Uji Coba Fungsi Aplikasi

Pengujian ini, dilakukan untuk mengetahui apakah fungsi-fungsi yang ada pada aplikasi berjalan dengan baik atau tidak. Adapun desain uji coba fungsi-fungsi yang akan diujikan adalah:

##### A. Desain Uji Coba Fungsi Menampilkan Isi Item

Pada desain uji coba ini, bertujuan untuk mengetahui apakah fungsi menampilkan isi *item* standar dan *database*, dapat menampilkan isi *item* berdasarkan kategori yang dipilih sebelumnya. Data kategori yang digunakan agar

dapat menampilkan *item* berdasar kategori dapat dilihat pada tabel 3.12. Desain uji coba menampilkan isi *item* standar dan *database* dapat dilihat pada tabel 3.13.

Tabel 3.12 Kategori

Nama Kategori	Keterangan
Angka	kategori standar
Huruf	kategori standar
Bagian Tubuh	kategori standar
Hewan	kategori <i>database</i>
Buah	kategori <i>database</i>

Tabel 3.13 Desain Uji Coba Menampilkan Isi Item Standar dan Database

Test Case ID	Tujuan	Input	Output yang diharapkan	Status
1	Menampilkan isi <i>item</i> standar (data ada)	Memilih kategori angka	Tampil isi <i>item</i> standar	
2	Menampilkan isi <i>item</i> standar (data tidak ada)	Memilih kategori huruf	Muncul pesan "Tidak Memiliki <i>item</i> "	
3	Menampilkan isi <i>item</i> yang tersimpan pada <i>database</i> (data ada)	Memilih kategori hewan	Tampil isi <i>item</i> yang tersimpan pada <i>database</i>	
4	Menampilkan isi <i>item</i> yang tersimpan pada <i>database</i> (data tidak ada)	Memilih kategori buah	Muncul pesan "Tidak Memiliki <i>item</i> "	

## B. Desain Uji Coba Fungsi Dengar

Tabel 3.14 Desain Uji Coba Fungsi Dengar / Meminta Pelafalan Sistem

Test Case ID	Tujuan	Input	Output yang diharapkan	Status
5	Meminta pelafalan sistem	Menekan gambar item	Sistem melafalkan teks	

Pada desain uji coba ini, bertujuan untuk mengetahui apakah fungsi dengar atau meminta pelafalan sistem dapat melafalkan teks kata Inggris yang dimiliki *item*. Desain uji coba dapat dilihat pada tabel 3.14.

### C. Desain Uji Coba Fungsi Lafal

Pada desain uji coba ini, bertujuan untuk mengetahui apakah fungsi lafal atau meminta pencocokkan pelafalan pengguna dapat menerima dan mengenali pelafalan pengguna serta penilaian yang akhirnya disimpan dalam *database*.

Desain uji coba dapat dilihat pada tabel 3.15.

Tabel 3.15 Desain Uji Coba Fungsi Lafal dan Penilaian

<i>Test Case ID</i>	<i>Tujuan</i>	<i>Input</i>	<i>Output yang diharapkan</i>	<i>Status</i>
6	Mengecek dukungan <i>voice recognition</i> (mendukung)	Masuk <i>form</i> ujian <i>item</i> / <i>form</i> latihan	Tombol “Lafal” aktif	
7	Mengecek dukungan <i>voice recognition</i> (tidak mendukung)	Masuk <i>form</i> ujian <i>item</i> / <i>form</i> latihan	Tombol “Lafal” tidak aktif	
8	Mencocokkan pelafalan pada <i>form</i> latihan (suara dikenali)	Menekan tombol “Lafal”, lalu pengguna melafalkan kata Inggris	Menampilkan daftar kata hasil identifikasi pada <i>list view</i>	
9	Mencocokkan pelafalan pada <i>form</i> latihan (suara tidak dikenali)	Menekan tombol “Lafal”, lalu pengguna melafalkan kata Inggris	Muncul pesan “Pelafanan Tidak dikenali”	
10	Mencocokkan pelafalan pada <i>form</i> ujian <i>item</i> (cocok)	Menekan tombol “Lafal”, lalu pengguna melafalkan kata Inggris	Menambah nilai benar, lalu lanjut <i>item</i> berikutnya.	
11	Mencocokkan pelafalan pada <i>form</i> ujian <i>item</i> (tidak cocok)	Menekan tombol “Lafal”, lalu pengguna melafalkan kata Inggris	Muncul pesan “Salah”, lalu menambah tanda salah	

<b>Test Case ID</b>	<b>Tujuan</b>	<b>Input</b>	<b>Output yang diharapkan</b>	<b>Status</b>
12	Lanjut <i>item</i> berikut bila melakukan kesalahan	Pengguna melafalkan kata Inggris dengan salah sebanyak tiga kali	Muncul pesan “Salah”, lalu lanjut <i>item</i> berikutnya	
13	Menyimpan total nilai akhir (tersimpan)	Melafalkan <i>item</i> terakhir dengan benar	Muncul pesan “Nilai dan menampilkan tombol “Grid Nilai” dan “Kategori”	
14	Menyimpan total nilai akhir (tidak tersimpan)	Melafalkan <i>item</i> terakhir dengan benar	Muncul pesan “Simpan Nilai Gagal”	

#### D. Desain Uji Coba Fungsi Pengolahan Nilai Kedalam Grafik

Pada desain uji coba ini, bertujuan untuk mengetahui apakah fungsi pengolahan nilai kedalam grafik dapat mengolah data nilai yang ada pada *database* kedalam bentuk grafik. Desain uji coba dapat dilihat pada tabel 3.16.

Tabel 3.16 Desain Uji Coba Fungsi Pengolahan Nilai kedalam Grafik

<b>Test Case ID</b>	<b>Tujuan</b>	<b>Input</b>	<b>Output yang diharapkan</b>	<b>Status</b>
15	Menampilkan nilai dalam bentuk grafik	Menekan menu <i>option</i> “Grafik Nilai”	Menampilkan <i>form</i> grafik	
16	Pindah <i>page</i> berikutnya	Menekan menu <i>option</i> “>>”	Menampilkan grafik nilai	
17	Pindah <i>page</i> berikutnya (data terakhir)	Menekan menu <i>option</i> “>>”	Muncul pesan “Data Terakhir, tidak ada data lagi”	
18	Pindah <i>page</i> sebelumnya	Menekan menu <i>option</i> “<<”	Menampilkan grafik nilai	
19	Pindah <i>page</i> sebelumnya (data awal)	Menekan menu <i>option</i> “<<”	Muncul pesan “Data Awal, tidak ada data lagi”	

### E. Desain Uji Coba Fungsi Tampilan Daftar Data

Pada desain uji coba ini, bertujuan untuk mengetahui apakah fungsi tampilan daftar data dapat menampilkan data pada *database* dalam bentuk daftar. Desain uji coba dapat dilihat pada tabel 3.17.

Tabel 3.17 Desain Uji Coba Fungsi Tampilan Daftar Data

<b>Test Case ID</b>	<b>Tujuan</b>	<b>Input</b>	<b>Output yang diharapkan</b>	<b>Status</b>
20	Menampilkan daftar pengguna pada <i>listview</i> (berhasil)	Masuk <i>form</i> daftar pengguna	Data pengguna tampil pada <i>listview</i>	
21	Menampilkan daftar pengguna pada <i>listview</i> (gagal)	Masuk <i>form</i> daftar pengguna	Muncul peringatan error	
22	Menampilkan daftar pengguna pada <i>gridview</i> (berhasil)	Masuk <i>form</i> pengaturan pengguna pada tab pengguna	Data pengguna tampil pada <i>gridview</i>	
23	Menampilkan daftar pengguna pada <i>gridview</i> (gagal)	Masuk <i>form</i> pengaturan pengguna pada tab pengguna	Muncul peringatan error	
24	Menampilkan daftar kategori pada <i>listview</i> (berhasil)	Masuk <i>form</i> daftar kategori	Data kategori tampil pada <i>listview</i>	
25	Menampilkan daftar kategori pada <i>listview</i> (gagal)	Masuk <i>form</i> daftar kategori	Muncul peringatan error	
26	Menampilkan daftar kategori pada <i>gridview</i> (berhasil)	Masuk <i>form</i> pengaturan kategori pada tab kategori	Data kategori tampil pada <i>gridview</i>	
27	Menampilkan daftar kategori pada <i>gridview</i> (gagal)	Masuk <i>form</i> pengaturan kategori pada tab kategori	Muncul peringatan error	
28	Menampilkan daftar <i>item</i> , berupa gambar, kata bahasa Inggris, kata bahasa Indonesia pada <i>gridview</i> (berhasil)	Masuk <i>form</i> pengaturan <i>item</i> pada tab <i>item</i>	Data <i>item</i> tampil pada <i>gridview</i>	

<b>Test Case ID</b>	<b>Tujuan</b>	<b>Input</b>	<b>Output yang diharapkan</b>	<b>Status</b>
29	Menampilkan daftar <i>item</i> , berupa gambar, kata bahasa Inggris, kata bahasa Indonesia pada <i>gridview</i> (gagal)	Masuk <i>form</i> pengaturan <i>item</i> pada tab <i>item</i>	Muncul peringatan error	
30	Menggunakan tombol ubah pada <i>gridview</i> (berhasil)	Memilih salah satu data, lalu menekan tombol “Ubah”	Mengalihkan pada <i>tab</i> ubah	
31	Menggunakan tombol ubah pada <i>gridview</i> (gagal)	Memilih salah satu data, lalu menekan tombol “Ubah”	Muncul peringatan untuk memilih data terlebih dulu	
32	Menggunakan tombol hapus pada <i>gridview</i> (berhasil)	Memilih salah satu data, lalu menekan tombol “Hapus”	Menghapus data	
33	Menggunakan tombol hapus pada <i>gridview</i> (gagal)	Memilih salah satu data, lalu menekan tombol “Hapus”	Muncul peringatan untuk memilih data terlebih dulu	

#### F. Desain Uji Coba Fungsi Tambah Data

Pada desain uji coba ini, bertujuan untuk mengetahui apakah fungsi tambah data dapat menambah data baru kedalam *database*. Desain uji coba dapat dilihat pada tabel 3.18.

Tabel 3.18 Desain Uji Coba Fungsi Tambah Data

<b>Test Case ID</b>	<b>Tujuan</b>	<b>Input</b>	<b>Output yang diharapkan</b>	<b>Status</b>
34	Menambah data pengguna (berhasil)	Memasukkan nama pengguna =”Irianto”, lalu tekan tombol “Tambah Pengguna”	Muncul pesan “Berhasil Menambah Pengguna”	
35	Menambah data pengguna (gagal)	Memasukkan nama pengguna =”Ir” , lalu tekan tombol “Tambah Pengguna”	Muncul peringatan “nama pengguna tidak boleh kurang dari tiga huruf”	
36	Menambah data kategori (berhasil)	Memasukkan nama kategori =”Hewan” , lalu tekan tombol “Tambah Kategori”	Muncul pesan “Berhasil Menambahkan Kategori”	

<i>Test Case ID</i>	<i>Tujuan</i>	<i>Input</i>	<i>Output yang diharapkan</i>	<i>Status</i>
37	Menambah data kategori (gagal)	Memasukkan nama kategori ="Hw" , lalu tekan tombol "Tambah Kategori"	Muncul peringatan "nama kategori tidak boleh kurang dari tiga huruf"	
38	Menampilkan gambar pada <i>image view</i>	Memilih gambar pada <i>gallery</i>	Gambar tampil pada <i>image view</i>	
39	Menambah data <i>item</i> (berhasil)	Memilih nama kategori="Hewan", lalu memasukkan kata Inggris="Fish", kata indonesia ="Ikan", gambar <i>item</i> ="gambar ikan", gambar kata ucap="gambar kata ucap ikan", lalu tekan tombol "Tambah <i>Item</i> "	Muncul pesan "Berhasil Menambah <i>Item</i> "	
40	Menambah data <i>item</i> (gagal)	Memilih nama kategori="Hewan", lalu memasukkan kata Inggris="Fish", kata indonesia ="Ikan", lalu tekan tombol "Tambah <i>Item</i> "	Muncul peringatan "Gambar <i>item</i> , Gambar kata ucap harus dipilih"	

### G. Desain Uji Coba Fungsi Ubah Data

Pada desain uji coba ini, bertujuan untuk mengetahui apakah fungsi ubah data dapat mengubah data pada *database*. Desain uji coba dapat dilihat pada tabel 3.19.

Tabel 3.19 Desain Uji Coba Fungsi Ubah Data

<i>Test Case ID</i>	<i>Tujuan</i>	<i>Input</i>	<i>Output yang diharapkan</i>	<i>Status</i>
41	Mengubah data pengguna (berhasil)	Memasukkan nama pengguna="Pratama", lalu tekan tombol "Ubah Pengguna"	Muncul pesan "Berhasil Mengubah Pengguna"	

<i>Test Case ID</i>	<b>Tujuan</b>	<i>Input</i>	<i>Output yang diharapkan</i>	<b>Status</b>
42	Mengubah data pengguna (gagal)	Memasukkan nama pengguna ="Pr", lalu tekan tombol "Ubah Pengguna"	Muncul peringatan "Nama pengguna tidak boleh kurang dari tiga huruf"	
43	Mengubah data kategori (berhasil)	Memasukkan nama kategori ="Fauna", lalu tekan tombol "Ubah Kategori"	Muncul pesan "Berhasil Mengubah Kategori"	
44	Mengubah data kategori (gagal)	Memasukkan nama kategori ="Fa", lalu tekan tombol "Ubah Kategori"	Muncul peringatan "Nama kategori tidak boleh kurang dari tiga huruf"	
45	Mengubah data <i>item</i> (berhasil)	Memasukkan kata Inggris ="Dolphin", kata indonesia = "Lumba-lumba", lalu tekan tombol "Ubah <i>Item</i> "	Muncul pesan "Berhasil Mengubah <i>Item</i> "	
46	Mengubah data <i>item</i> (gagal)	Memasukkan kata Inggris =""", kata indonesia = """, lalu tekan tombol "Ubah <i>Item</i> "	Muncul peringatan "Kata Inggris tidak boleh kosong, Kata Indonesia tidak boleh kosong"	

#### H. Desain Uji Coba Fungsi Hapus Data

Pada desain uji coba ini, bertujuan untuk mengetahui apakah fungsi hapus data dapat menghapus data pada *database*. Desain uji coba dapat dilihat pada tabel 3.20.

Tabel 3.20 Desain Uji Coba Fungsi Hapus Data

<i>Test Case ID</i>	<b>Tujuan</b>	<i>Input</i>	<i>Output yang diharapkan</i>	<b>Status</b>
47	Menghapus data pengguna (berhasil)	Memilih data pengguna, lalu menekan tombol hapus, dan mengkonfirmasi pesan hapus	Pada <i>gridview</i> tidak menampilkan data terhapus, data pengguna terhapus dan nilai ikut terhapus	

<i>Test Case ID</i>	<i>Tujuan</i>	<i>Input</i>	<i>Output yang diharapkan</i>	<i>Status</i>
48	Menghapus data pengguna (gagal)	Memilih data pengguna, lalu menekan tombol hapus, dan mengkonfirmasi pesan hapus	Muncul peringatan “Gagal menghapus data”, pada <i>gridview</i> masih menampilkan data terhapus	
49	Menghapus data kategori (berhasil)	Memilih data kategori, lalu menekan tombol hapus, dan mengkonfirmasi pesan hapus	Pada <i>gridview</i> tidak menampilkan data terhapus, data kategori terhapus dan nilai ikut terhapus	
50	Menghapus data kategori (gagal)	Memilih data kategori, lalu menekan tombol hapus, dan mengkonfirmasi pesan hapus	Muncul peringatan “Gagal menghapus data”, pada <i>gridview</i> masih menampilkan data terhapus	
51	Menghapus data <i>item</i> (berhasil)	Memilih data <i>item</i> , lalu menekan tombol hapus, dan mengkonfirmasi pesan hapus	Pada <i>gridview</i> tidak menampilkan data terhapus, data <i>item</i> terhapus	
52	Menghapus data <i>item</i> (gagal)	Memilih data <i>item</i> , lalu menekan tombol hapus, dan mengkonfirmasi pesan hapus	Muncul peringatan “Gagal menghapus data”, pada <i>gridview</i> masih menampilkan data terhapus	

### 3.4.2. Uji Coba Aplikasi Kepada Pengguna

Prosedur untuk melakukan pengujian aplikasi kepada pengguna adalah sebagai berikut:

a. Uji coba kepada ahli isi

Tujuan dari uji coba ini adalah untuk mengetahui isi dari materi yang digunakan pada aplikasi ini. Pengajar yang ditunjuk sebagai ahli adalah seorang pengajar bahasa Inggris yang mengajarkan bahasa Inggris secara *private* (bersifat pribadi).

b. Uji coba kelompok pengguna

Uji coba ini dilakukan kepada 30 orang pengguna dari beberapa kalangan, dimana hasil dari uji coba ini digunakan untuk revisi dan evaluasi aplikasi.

c. Instrumen Uji coba

Instrumen uji coba yang digunakan dalam pembuatan aplikasi adalah sebagai berikut:

1. Angket

Angket/Kuesioner diberikan kepada ahli dan pengguna untuk mendapat informasi mengenai cara penggunaan aplikasi, kesesuaian materi dan alur program sehingga dapat mempermudah dalam pengambilan data dan tujuan dari pembuatan aplikasi ini dapat tercapai.

Desain angket/kuesioner uji coba dapat dilihat pada lampiran Hal-2, dimana yang ditujukan kepada ahli mencakup poin A-I dan pengguna mencakup poin A-H.

2. Teknik Analisis Data

Untuk menganalisis apakah aplikasi tersebut layak atau tidak dari serangkaian uji coba tersebut diatas, maka akan digunakan analisis rasional dan ditentukan sendiri oleh individu yang merupakan justifikasi yang bersifat subjektif. Kesimpulan yang dihasilkan diambil dari penilaian butir tes berdasarkan skor tes individu ke domain butir yang lebih besar dimana serupa dengan butir-butir yang terdapat dalam tesnya sendiri. Penarikan kesimpulan dengan cara tersebut merupakan cara pengambilan hasil kesimpulan dari metode validasi isi/kurikuler.

Tabel 3.21 Desain Karakteristik Pengguna

No.	Nama Pengguna	Umur (Tahun)	Pekerjaan	Total Nilai Akhir

Pada proses uji coba ini, pengguna diharuskan mencoba Aplikasi Belajar Melafalkan Bahasa Inggris hingga menghasilkan total nilai akhir. Uji coba ini dilakukan dengan cara melakukan *survey* langsung kepada 30 orang pengguna dengan cara mengisi kuesioner pada lampiran Hal-2. Desain tabel karakteristik pengguna dapat dilihat pada tabel 3.21, dimana berfungsi sebagai rekapitulasi karakteristik pengguna aplikasi.

Desain tabel detail uji coba *item*, dimana pelafalan oleh pengguna akan dicocokkan pada sistem yang berfungsi sebagai rekapitulasi detail uji coba *item*. Pada tabel 3.22 dapat dilihat desain tabel detail uji coba *item*, dimana kategori yang dicoba pada pengguna adalah kategori angka yang memiliki *item* angka 1-10.

Tabel 3.22 Desain Detail Uji Coba Item Kategori Angka

Angka	Benar Melafalkan tetapi dianggap Salah	Salah	Tidak dikenali	Dikenali sebagai
1				
2				
3				
4				
5				
6				
7				
8				
9				
10				
total				

Kuesioner yang telah diisi oleh 30 orang pengguna akan dimasukkan kedalam rekapitulasi kuesioner, dimana desain tabel kuesioner yang telah diisi dapat dilihat pada tabel 3.23, yang berfungsi untuk mengetahui hasil keseluruhan

dari pendapat pengguna terhadap Aplikasi *Mobile* Belajar Melafalkan Bahasa Inggris.

Tabel 3.23 Desain Rekapitulasi Kuesioner

Pertanyaan No.	Penilaian					$\Sigma$	$\bar{x}$	Nilai Akhir
	1	2	3	4	5			
<b>Tampilan</b>								
<b>B</b>	1.							
	2.							
	3.							
	4.							
	5.							
	6.							
	7.							
<b>Navigasi</b>								
<b>C</b>	1.							
	2.							
	3.							
	4.							
	5.							
<b>Materi Pembelajaran</b>								
<b>D</b>	1.							
	2.							
	3.							
	4.							
	5.							
<b>Penilaian</b>								
<b>E</b>	1.							
	2.							
	3.							
<b>Manfaat</b>								
<b>F</b>	1.							
	2.							
	3.							
	4.							
	5.							
	6.							
<b>Minat dan Motivasi</b>								
<b>G</b>	1.							
	2.							
	3.							
<b>Lain-lain</b>								
<b>H</b>	1.							