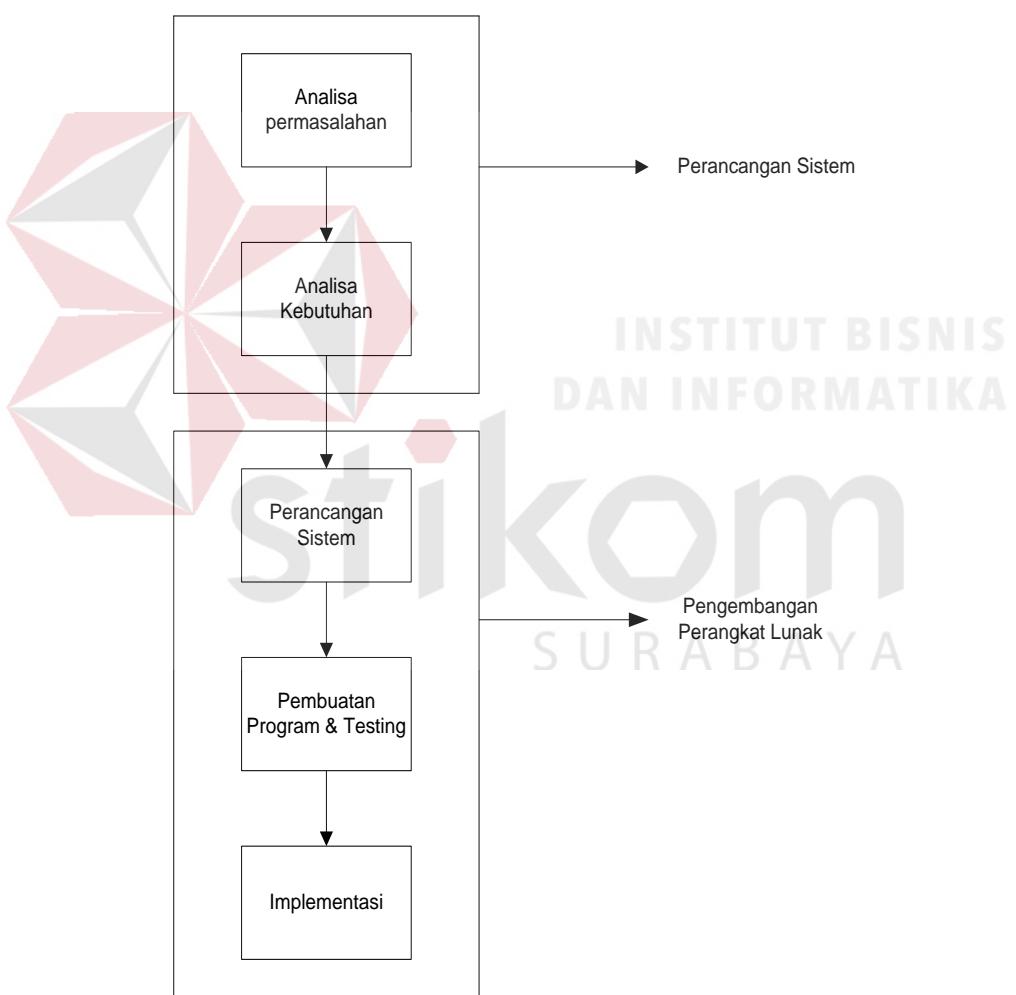


## BAB III

### ANALISA dan PERANCANGAN SISTEM

Di dalam bab III ini akan dijelaskan tentang langkah-langkah yang akan dilakukan dalam perencanaan pembuatan aplikasi kamus Korea-Indonesia, yang akan dijelaskan dalam gambar 3.1 dibawah ini :



Gambar 3.1 Diagram Alir

### 3.1 Analisa Permasalahan

Kamus adalah sejenis buku rujukan yang menerangkan makna kata-kata. Diperlukannya bantuan kamus dikarenakan kamus Korea tidak seperti kamus Indonesia yang disusun berdasarkan abjad a sampai z, pengurutan kamus Korea berdasarkan Alphabet Korea dalam penyusunannya memakai sistem sillabic. Untuk membentuk sebuah suku kata atau blok itu tersusun dari 2 atau 3 huruf dan disusun dari kiri ke kanan atau dari atas ke bawah, tergantung huruf-huruf apa yang disusun. Bahasa Korea sendiri tidak mengenal kata sandang, tidak mengenal bentuk tunggal dan bentuk jamak. Untuk menggunakan kamus terjemahan dari bahasa Korea ke Indonesia, pengguna harus terlebih dahulu menginstal keyboard huruf hangeul dan mengerti cara penulisan huruf Korea.

Kamus Korea-Indonesia yang sudah ada saat ini dalam bentuk kamus buku , kamus online di website serta juga terdapat kamus *online* dan kamus *offline* mobile. Pencarian secara manual dalam kamus buku tentu saja menghabiskan waktu dan cukup rumit, Sedangkan kelemahan dari kamus *online* website dan kamus *online mobile* adalah diperlukannya koneksi internet untuk mengakses kamus tentunya akan mengurangi kuota internet, serta jika tidak terdapat koneksi internet maka pencarian kata dalam kamus sehingga tidak dapat digunakan ketika pengguna tidak mempunyai koneksi internet dan terkadang masyarakat juga tidak mengerti bunyinya dari karakter huruf *hangul* tersebut dikarenakan tidak adanya bahasa latin atau romanisasi dari kamus *mobile online* yang sudah ada.

Berdasarkan kelemahan dari permasalahan kamus yang sudah ada maka untuk mengatasinya diperlukan sebuah kamus yang bersifat *offline*. Saat ini di

*play store* sudah ada 1(satu) kamus yang bersifat *offline*, akan tetapi masih ada kelemahannya pada kamus tersebut.

Berikut tabel 3.1 perbandingan dari kamus Deh ( kamus *offline* Indonesia-Korea) yang terdapat di *play store* android dengan aplikasi kamus yang dibuat.

Tabel 3.1 Perbandingan Kamus

Perbedaan	Kamus Korea-Indonesia	Kamus Deh
Internet	Tidak	Tidak
Cara baca	Ada	Tidak ada
Cara Pencarian	Dari 2 bahasa Indonesia dan bahasa Korea	Dari 2 bahasa Indonesia dan bahasa Korea
Hasil	1. Hasil kata yang terjemahkan langsung tepat dengan kata yang dicari	1. Semua kata yang mengandung kata yang diinputkan muncul dan tidak terurut. Hasil suatu kata bisa lebih dari 1 tetapi jika pengguna mencari kata yang didalam artinya tadi tidak muncul. 2. Jika diinputkan dengan huruf hangul tidak muncul kata yang dicari. Atau tidak bisa dilakukan pencarian dengan huruf hangul.
Fitur lain	1. Contoh percakapan 2. Huruf hangul	1. Tempat restoran Korea

Dari kelemahan yang terdapat pada kamus *offline* Indonesia-Korea yang sudah ada maka diperlukan sebuah kamus Indonesia-Korea berbasis android yang dapat digunakan tanpa koneksi internet atau *offline*.

Pembuatan kamus dijalankan secara *offline* terdapat beberapa faktor, yaitu :

1. Pada pemrosesan secara *offline* adalah pengolahan data dengan mengumpulkan data terlebih dahulu atau biasa disebut dengan *batch processing* kemudian data diproses.
2. Untuk mendapatkan kosakata terbaru harus update, update yang dimaksud adalah update pada apk. Update kamus bisa dilakukan ketika ada notifikasi update dan pengguna mendownload update terbaru pada *play store*.

Kamus ini akan digunakan pada gadget yang menggunakan operating sistem Android. Kamus digital lebih mengutamakan pada fasilitas pengolahan kata elektronis yaitu memungkinkan aplikasi pengolah kata memeriksa ejaan dari dokumen yang diketik, sehingga meminimumkan kemungkinan salah eja atau salah ketik. Algoritma pencarian yang digunakan untuk mempercepat pencarian dalam aplikasi kamus Indonesia-Korea ini menggunakan algoritma pencarian biner dikarenakan data sudah terurut.

Dari permasalahan diatas maka dilakukan analisis, analisis dilakukan terhadap kebutuhan aplikasi yang akan dibangun, agar pemanfaatan aplikasi memperoleh hasil yang optimal, maka maka penulis akan membangun aplikasi kamus Indonesia-Korea menggunakan metode *Binary Search* berbasis Android. Cara pencarian pada kamus ini, yang pertama pengguna harus memilih dalam pencarian terjemahan kedalam bahasa apa yang akan digunakan. Setelah menu kamus dipilih pengguna mengetikan huruf di form pencarian dan menekan tombol terjemahkan. Pada saat itu dilanjutkan pada proses pencarian dengan metode yaitu mencari data tengah pada *indeks* kemudian membandingkan pada kata yang

diinputkan apabila Jika data yang dicari lebih kecil dari data tengah, proses dilakukan kembali tetapi posisi akhir dianggap sama dengan posisi tengah -1, Jika data yang dicari lebih besar dari data tengah , proses dilakukan kembali tetapi posisi awal dianggap sama dengan posisi tengah +1. Proses pencarian lebih lengkap dijelaskan pada tabel Tabel 3.6 Flow Of Event Menerjemahkan Korea-Indonesia dan Tabel 3.7 Flow Of Event Menerjemahkan Indonesia-Korea.

Dengan adanya pembuatan aplikasi ini diharapkan mempermudah pengguna mendapatkan informasi terjemahan kata bahasa Korea dan indonesia yang cepat dan lengkap sedangkan untuk mendapatkan kosakata terbaru hanya perlu mengupdate kamus melalui *playstore*.

### 3.2 Analisis Kebutuhan

Berdasarkan analisa permasalahan terdapat analisa kebutuhan yang digunakan untuk mengetahui dan membandingkan antara pengetahuan yang didapatkan pada saat studi pustaka yang berkaitan dengan aplikasi kamus Korea untuk menentukan kebutuhan – kebutuhan yang harus ada dalam sistem, maka diperlukan suatu aplikasi Kamus Indonesia-Korea berbasis Android dengan metode *Binary Search*. Aplikasi kamus Indonesia-Korea mempunyai beberapa fungsi. Berikut merupakan daftar fungsionalitasnya :

- i. Dibutuhkan aplikasi yang dapat yang bisa mempercepat pencarian kosakata Korea ke Indonesia maupun sebaliknya.
- ii. Dibutuhkan aplikasi kamus yang bisa digunakan kapan saja oleh pengguna.

- iii. Dibutuhkan aplikasi kamus yang memberikan huruf baca Korea sehingga memudahkan pengguna untuk memahami dalam pencarian arti kata dari bahasa Korea.
- iv. Dibutuhkan sebuah aplikasi yang bisa memberikan contoh percakapan dalam bahasa Korea dan Indonesia untuk digunakan kapan saja dan dimana saja.

### 3.2.1 Spesifikasi Program

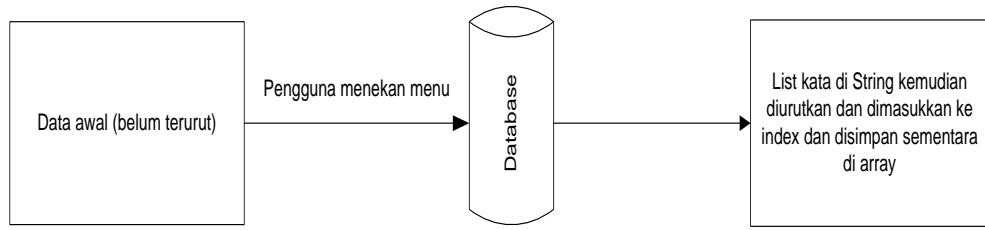
Berdasarkan analisa permasalahan dan analisa kebutuhan maka terdapat spesifikasi program yaitu :

- 1. Fitur *offline* untuk memudahkan pengguna melakukan pencarian dimana saja dan kapanpun tanpa menggunakan koneksi internet.
- 2. Fitur pencarian kata dari bahasa Korea dan bahasa Indonesia dan sebaliknya.
- 3. Terdapat contoh percakapan dalam bahasa Indonesia dan bahasa Korea.

### 3.3 Perancangan Sistem

Perancangan sistem merupakan tahap pengembangan setelah analisis sistem dilakukan. Pembuatan aplikasi didasarkan pada pembuatan basis data (database) standar sebagai tempat penyimpanan yang berisi kumpulan kata-kata bahasa Indonesia dan Bahasa Korea yang diinputkan oleh Admin kemudian diurutkan karena merupakan sebuah syarat dari metode *Binary Search* untuk melakukan pencarian.

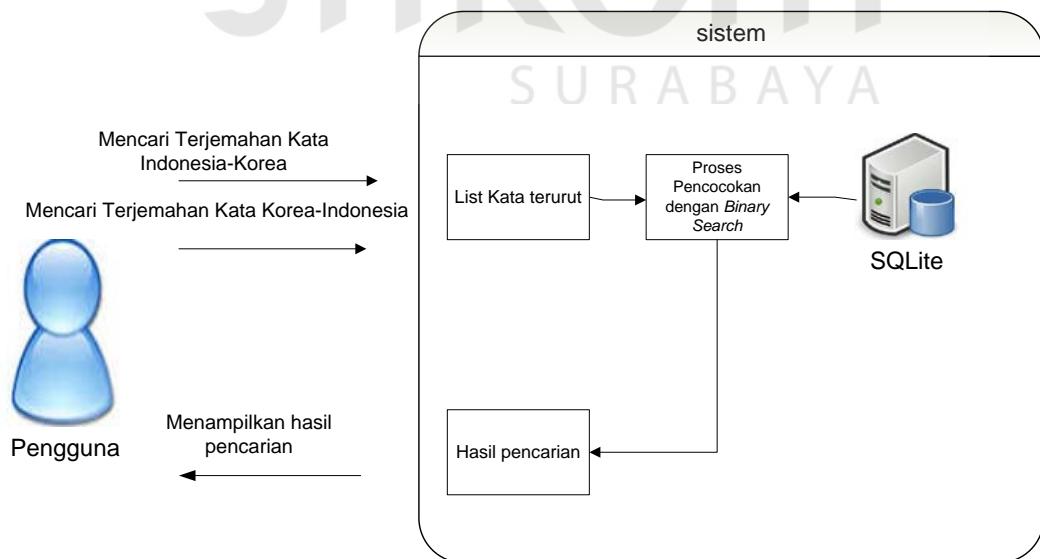
Proses penyimpanan dan pengurutan database akan dijelaskan pada gambar berikut.



Gambar 3.2 Proses pengurutan kata

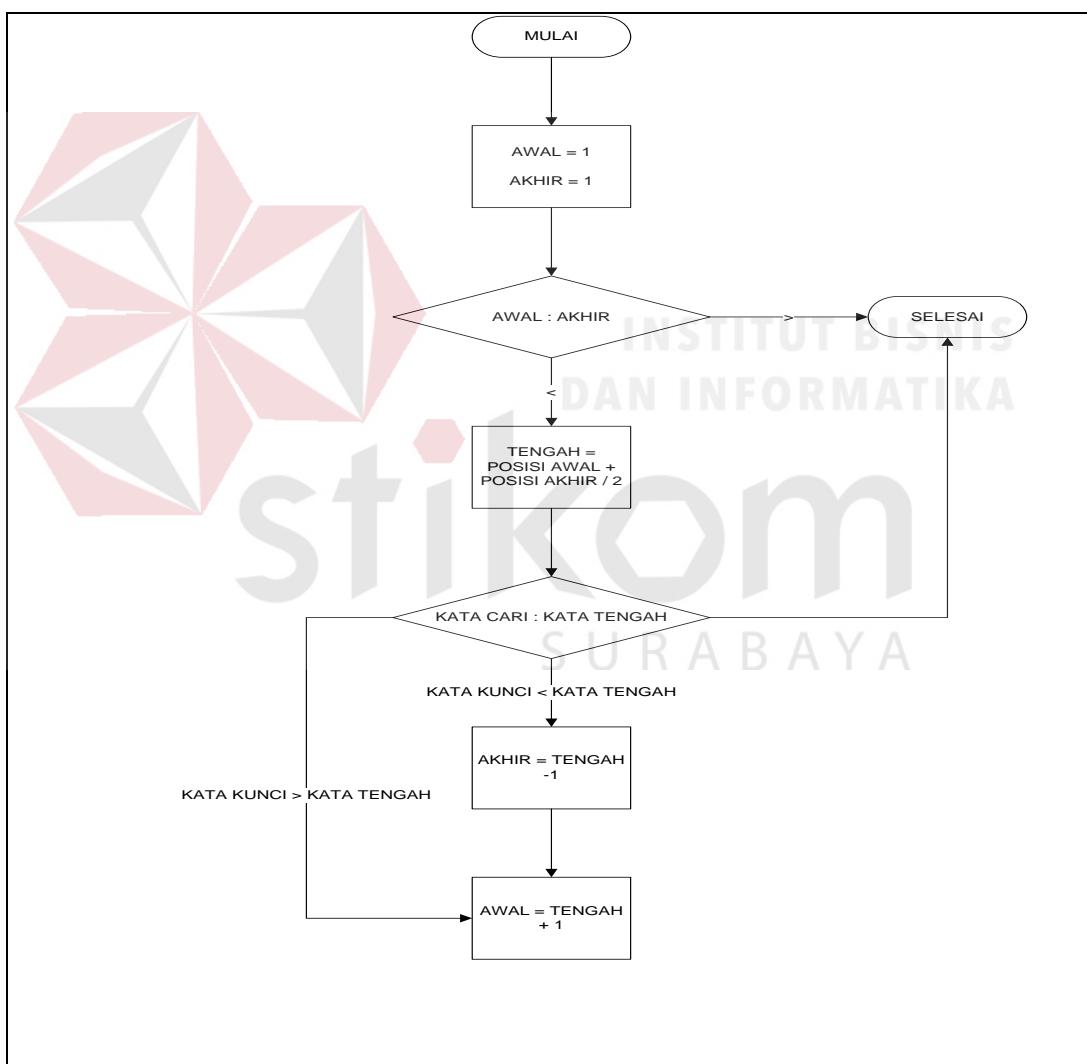
Cara kerja proses pengurutan kata dari bahasa Indonesia dan bahasa Korea yang disimpan pada file text(.txt) adalah *developer* memasukkan data di SQLite dan data yang dimasukkan belum terurut sesuai abjad Indonesia dan Huruf Hangul. Proses pengurutan terjadi ketika pengguna menekan menu kamus, data yang ada pada database di list sesuai urutan kemudian dimasukkan ke dalam *indeks* dan disimpan sementara di array.

Setelah data sudah dalam kondisi terurut selanjutnya akan dilanjutkan pada proses pencarian terjemahan yang akan dijelaskan pada design blok diagram untuk Aplikasi Kamus Indonesia – Korea dapat dilihat pada gambar 3.3.



Gambar 3.3 Blok Diagram Pencarian Terjemahan Kata

Pada proses awal pada saat pengguna membuka menu utama, pengurutan di array dijalankan. Setelah data terurut dilanjutkan dengan proses pencarian dengan metode *Binary Search* yang dijelaskan pada tabel 3.6 *flow of events* dan tabel 3.7 *flow of events*. Setelah hasil pencarian diketemukan, akan ditampilkan pada halaman terjemahan. Metode *Binary Search* akan digambarkan pada flowchart seperti dibawah ini.



Gambar 3.3 *flowchart* metode *Binary Search*

Untuk lebih jelasnya perhatikan contoh berikut. Misalnya ingin mencari kata *aku* pada sekumpulan data berikut :

Tabel 3.2 Contoh Data Pencarian

Array ke	0	1	2	3	4
data	aku	adil	abu	dua	acak
diurutkan	abu	acak	adil	aku	dua

Setelah data tersebut diurutkan maka fungsi binary sort baru mulai bekerja mencari data. Berikut cara dari binary sort mencari data tersebut misalnya data yang dicari adalah aku. Maka pencarinya dijelaskan pada tabel berikut ini :

Tabel 3.3 Contoh Data Pencarian Pertama

Tabel 3. Cukup Data Penerapan Portofolio				
0	1	2	3	4
Abu	Acak	adil	Aku	dua
Awal	tengah			akhir

Jumlah data di database ada 5,mula-mula dicari data tengah dengan rumus  $(0+5) / 2 = 2$ . Berarti data tengah adalah data ke -2 yaitu adil . Data yang dicari,yaitu aku, kerene kata aku urutennya dibawah dari kata adil maka aku > adil (data tengah).

Jika yang dicari ternyata lebih besar maka +1.

Tabel 3.4 Contoh Data Pencarian Kedua

0	1	2	3	4
Abu	Acak	adil	Aku	dua

Awal= tengah akhir

Dari pencarian pertama didapat kata yang dicari lebih besar dari kata tengah,dikarenakan pencarian belum diketemukan maka nilai tengah ditambah 1 kolom array akan didapatkan rumus  $(2+5)/2=3$ . Berarti data tengah yang baru adalah kolom array ke 3 yaitu aku. Data yang dicari dibandingkan dengan data tengah ternyata sama. Jadi data yang dicari yaitu kata aku ditemukan pada indeks ke -3.

### 3.4 Teknik Pengumpulan Data

Pengumpulan data dilakukan melalui studi pustaka terhadap berbagai sumber yang berkaitan dengan masalah yang dibahas baik bersumber dari buku, majalah, artikel atau tutorial dari internet untuk dijadikan sebuah dokumen.

Yang digunakan dalam metode penelitian tugas akhir ini adalah sebagai berikut:

#### 1. Studi Pustaka

Digunakan untuk penelitian agar penulis mendapatkan landasan teori yang berkaitan tentang objek penelitian. Metode ini bersumber dari buku-buku serta media internet yang berhubungan dengan penulisan tugas akhir yaitu contoh-contoh kamus Korea, pembuatan kamus android dan perancangan sistem. Pada studi pustaka ini ada beberapa tahapan yang dilakukan untuk memperoleh informasi yang berkaitan dengan penelitian tugas akhir ini, yaitu:

- i. Mempelajari cara membuat aplikasi android melalui media buku atau internet yang membahas tentang android sehingga menghasilkan desain sistem informasi yang dapat diimplementasikan.
- ii. Mempelajari buku-buku korea untuk mendapatkan informasi cara bacanya agar mendapatkan informasi yang valid.
- iii. Mempelajari metode binary search sebagai alat bantu pencarian di sistem yang akan dibuat.

## 2. Observasi

Observasi atau pengamatan merupakan salah satu teknik pengumpulan data/fakta yang cukup efektif untuk mempelajari suatu sistem. Observasi adalah pengamatan langsung para pembuat keputusan berikut lingkungan fisiknya dan atau pengamatan langsung suatu kegiatan yang sedang berjalan. Pada observasi kali ini dilakukan pencarian tentang cara membuat aplikasi kamus android melalui media internet dan buku, mencari informasi tentang aplikasi kamus Korea yang di terjemahkan ke bahasa Indonesia yang sudah ada di *playstore*.

### 3.5 Usecase Diagram



Gambar 3.4 Use Case Diagram User Aplikasi Kamus Korea – Indonesia

Diagram *use case* perancangan aplikasi kamus bahasa Indonesia ke bahasa Korea yang berbasis Android pada gambar menjelaskan interaksi antara use case dan aktor. Dimana aktor dapat berupa orang, peralatan , atau sistem lain yang sedang dibangun. Aktor User memberikan arah panah ke beberapa use case,

antara lain: bahasa Indonesia ke bahasa Korea, bahasa Korea ke bahasa Indonesia, huruf hangul, dan contoh percakapan.

Berikut adalah keterangan singkat use case diagram aplikasi android untuk pengguna.

Tabel 3.5 Keterangan Use Case Diagram Aplikasi Android untuk Pengguna.

Nama Use Case	Deskripsi
Menerjemahkan Korea-Indonesia	Proses untuk melihat terjemahan dari bahasa korea
Menerjemahkan Indonesia-Korea	Proses untuk melihat terjemahan dari bahasa Indonesia
Melihat Contoh Percakapan	Proses yang digunakan untuk melihat contoh percakapan dari bahasa Korea ke bahasa Indonesia
Melihat Huruf Hangul	Proses yang digunakan untuk melihat macam

### 3.5.2 *Flow Of Events*

*Flow of events* adalah sebuah cara untuk mendokumentasikan aliran logika dalam use case, yang menjelaskan secara rinci apa yang pemakai akan lakukan dan apa yang sistem lakukan, bukan bagaimana sistem bekerja.

#### A. *Flow Of Events* Menerjemahkan Korea-Indonesia

Tabel 3.6 Flow Of Event Menerjemahkan Korea-Indonesia

Deskripsi Use Case Menerjemahkan Korea - Indonesia	
Nama Use Case	:
Deskripsi Singkat	:
Aktor	:
Prasyarat (Kondisi)	: <p>Hanya bisa digunakan bila <i>smartphone</i> berbasis Android pengguna sudah mempunyai font huruf hangul</p>
Alur Utama	: <ol style="list-style-type: none"> <li>1. Pengguna memilih kamus menerjemahkan Korea-Indonesia.</li> <li>2. Aplikasi menampilkan form pencarian untuk kata dalam bahasa korea yang ingin diterjemahkan.</li> <li>3. Pengguna memasukkan kata yang dicari kemudian menekan tombol terjemahkan.</li> <li>4. Kata yang dicari pengguna akan</li> </ol>

	<p>dilakukan pencarian pada tabel. Cara mengurutkan kata pada tabel database, ketika pengguna memilih kamus di menu utama maka data di array diurutkan berdasarkan abjad.</p> <ol style="list-style-type: none"> <li>5. Pada proses pencarian di database, sesuai dengan langkah- langkah metode yang pertama diambil dari posisi awal=1 dan posisi akhir = n.</li> <li>6. Kemudian kita cari posisi data tengah dengan rumus posisi tengah = (posisi awal + posisi akhir ) div 2.</li> <li>7. Kemudian data yang dicari dibandingkan data tengah .</li> <li>8. Jika data yang dicari lebih kecil dari data tengah, proses dilakukan kembali tetapi posisi akhir dianggap sama dengan posisi tengah -1,</li> <li>9. Jika data yang dicari lebih besar dari data tengah , proses dilakukan kembali tetapi posisi awal dianggap sama dengan posisi tengah +1.</li> <li>10. Jika data yang dicari belum ditemukan mengulang kembali dari mencari posisi tengah pada data hingga data ditemukan atau tidak ditemukan.</li> <li>11. Pencarian menggunakan biner akan berakhir jika data ditemukan posisi awal lebih besar dari data akhir. Jika posisi awal sudah lebih besar dari posisi akhir berarti data yang dicari tidak ada.</li> </ol>
Alur Alternatif	:
Kondisi akhir sukses	Aplikasi menampilkan hasil pencarian yang diterjemahkan ke dalam bahasa Indonesia beserta bahasa latin dari bahasa Korea tersebut.
Kondisi akhir sukses	Aplikasi menampilkan hasil pencarian yang diterjemahkan ke dalam bahasa Indonesia beserta bahasa latin dari bahasa Korea tersebut.
Kondisi akhir gagal	Hasil pencarian tidak ada, maka informasi tidak tampil

## B. *Flow Of Events* Menerjemahkan Indonesia – Korea

Tabel 3.7 Flow Of Event Menerjemahkan Indonesia-Korea

Deskripsi <i>Use Case</i> Menerjemahkan Indonesia - Korea		
Nama <i>Use Case</i>	:	Menerjemahkan Indonesia - Korea
Deskripsi Singkat	:	<i>Use Case</i> ini mendeskripsikan proses menerjemahkan dari kata bahasa Indonesia ke Bahasa Korea
Aktor	:	Pengguna
Prasyarat (Kondisi)	:	-
Alur Utama	:	<ol style="list-style-type: none"> <li>1. Pengguna memilih kamus menerjemahkan Korea-Indonesia.</li> <li>2. Aplikasi menampilkan form pencarian untuk kata dalam bahasa korea yang ingin diterjemahkan.</li> <li>3. Pengguna memasukkan kata yang dicari kemudian menekan tombol terjemahkan.</li> <li>4. Kata yang dicari pengguna akan dilakukan pencarian pada tabel. Cara mengurutkan kata pada tabel database, ketika pengguna memilih kamus di menu utama maka data di array diurutkan berdasarkan abjad.</li> <li>5. Pada proses pencarian di database, sesuai dengan langkah- langkah metode yang pertama diambil dari posisi awal=1 dan posisi akhir = n.</li> <li>6. Kemudian kita cari posisi data tengah dengan rumus posisi tengah = (posisi awal + posisi akhir ) div 2.</li> <li>7. Kemudian data yang dicari dibandingkan data tengah .</li> <li>8. Jika data yang dicari lebih kecil dari data tengah, proses dilakukan kembali tetapi posisi akhir dianggap sama</li> </ol>
Alur Alternatif	:	-
Kondisi akhir sukses		Aplikasi menampilkan hasil pencarian yang diterjemahkan ke dalam bahasa Korea beserta bahasa latin dari bahasa Korea tersebut.

Deskripsi Use Case Menerjemahkan Indonesia - Korea	
Kondisi akhir gagal	: Hasil pencarian tidak ada, maka informasi tidak tampil.

### C. *Flow Of Events* Contoh Percakapan

Tabel 3.8 Flow Of Event Contoh Percakapan

Deskripsi Use Case Contoh Percakapan	
Nama Use Case	: Contoh Percakapan
Deskripsi Singkat	: Use Case ini mendeskripsikan contoh percakapan sehari-hari dari bahasa korea yang diterjemahkan ke bahasa Indonesia.
Aktor	: Pengguna
Prasyarat (Kondisi)	: -
Alur Utama	: 1. Pengguna memilih menu “Contoh Percakapan” 2. Aplikasi menampilkan contoh percakapan tersebut.
Alur Alternatif	: -
Kondisi akhir sukses	: Pengguna dapat melihat contoh percakapan pada aplikasi.
Kondisi akhir gagal	: -

### D. *Flow Of Events* Huruf Hangul

Tabel 3.9 Flow Of Event Huruf Hangul

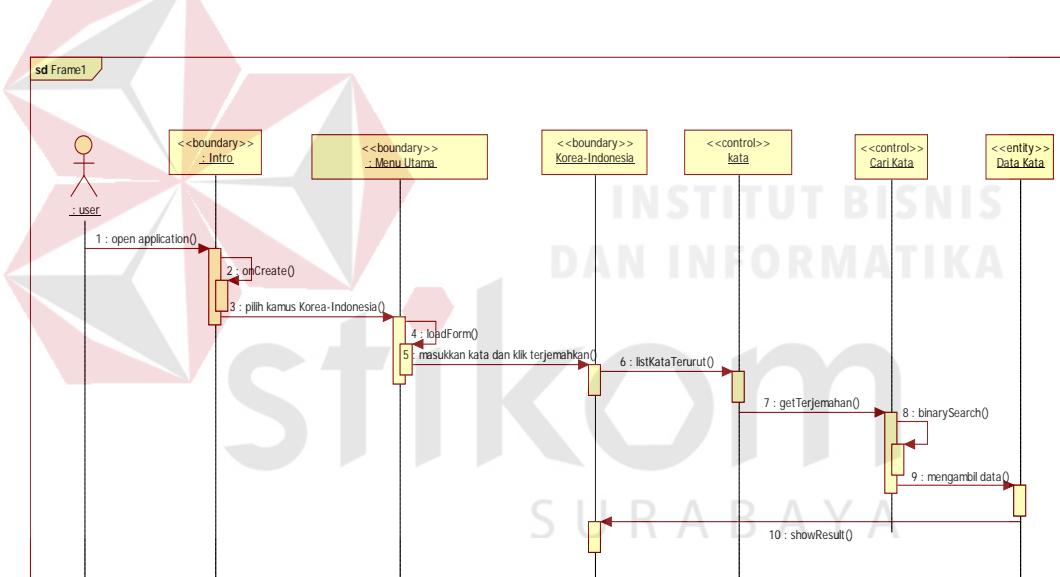
Deskripsi Use Case Huruf Hangul	
Nama Use Case	: Huruf Hangul
Deskripsi Singkat	: Use Case ini mendeskripsikan macam – macam huruf hangul dan cara menyusun huruf hangul.
Aktor	: Pengguna
Prasyarat (Kondisi)	: -
Alur Utama	: 1. Pengguna memilih menu huruf hangul 2. Aplikasi menampilkan menu huruf hangul

Deskripsi Use Case Huruf Hangul		
Alur Alternatif	:	-
Kondisi akhir sukses	:	Pengguna dapat melihat macam-macam huruf hangul pada aplikasi.
Kondisi akhir gagal	:	-

### 3.6 Sequence Diagram

*Sequence diagram* menggambarkan interaksi antar obyek di dalam dan disekitar sistem berupa *message* yang digambarkan terhadap waktu.

#### 3.6.1 Sequence Diagram Korea – Indonesia

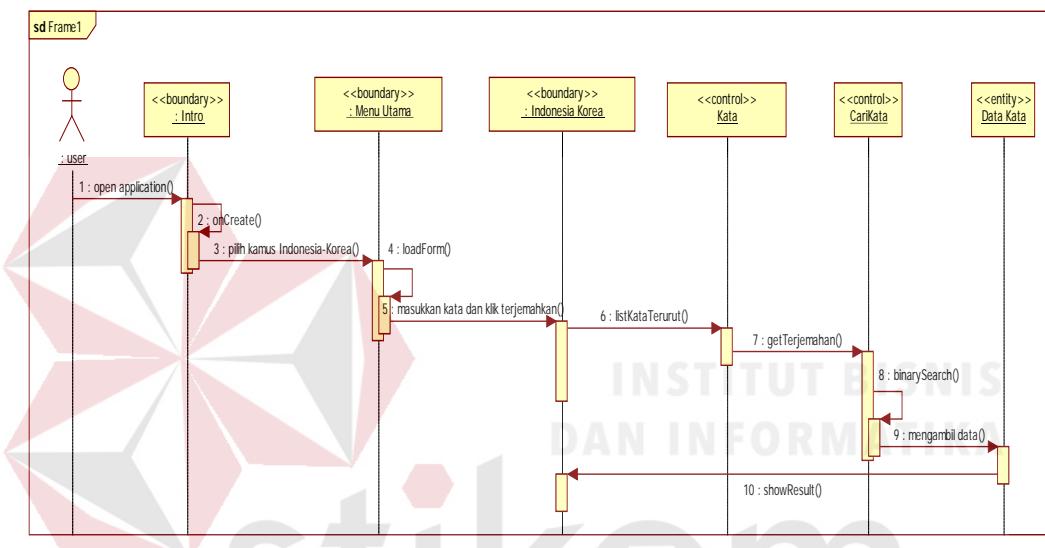


Gambar 3.5 Sequence Diagram Kamus Korea –Indonesia

Pada gambar 3.5 dijelaskan pada diagram sequencial kamus Korea-Indonesia dimulai ketika pengguna membuka aplikasi dengan fungsi `onCreate()` dijalankan kemudian akan tampil menu pilihan kamus. Pengguna dapat memilih kamus mana yang ingin digunakan. Jika pengguna memilih kamus Korea - Indonesia maka aplikasi akan menampilkan form untuk pencarian kamus. Ketika memilih menu pencarian maka kata yang ada pada tabel diurutkan dan disimpan

di array. Kemudian dilanjutkan dengan melakukan pencarian dengan metode *Binary Search* yang dijelaskan pada tabel 3.3 *flow of event* menerjemahkan Korea-Indonesia. Setelah kata yang dicari ditemukan maka hasil pencarian ditampilkan pada menu pencarian Korea-Indonesia.

### 3.6.2 Sequence Diagram Indonesia – Korea

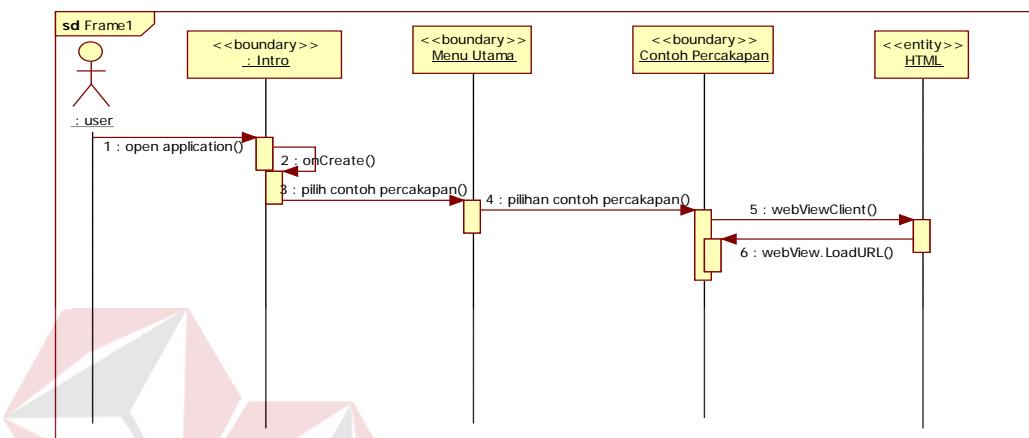


Gambar 3.6 Sequence Diagram Kamus Indonesia - Korea

Pada gambar 3.6 dijelaskan pada diagram sequencial kamus Indonesia – Korea dimulai ketika pengguna membuka aplikasi dengan fungsi `onCreate()` dijalankan kemudian akan tampil menu pilihan kamus. Pengguna dapat memilih kamus mana yang ingin digunakan. Jika pengguna memilih kamus Indonesia – Korea maka aplikasi akan menampilkan form untuk pencarian kamus. Ketika memilih menu pencarian maka kata yang ada pada tabel diurutkan dan disimpan di array. Kemudian dilanjutkan dengan melakukan pencarian dengan metode *Binary Search* yang dijelaskan pada tabel 3.6 *flow of event* menerjemahkan

Indonesia-Korea. Setelah kata yang dicari ditemukan maka hasil pencarian ditampilkan pada menu pencarian Indonesia-Korea.

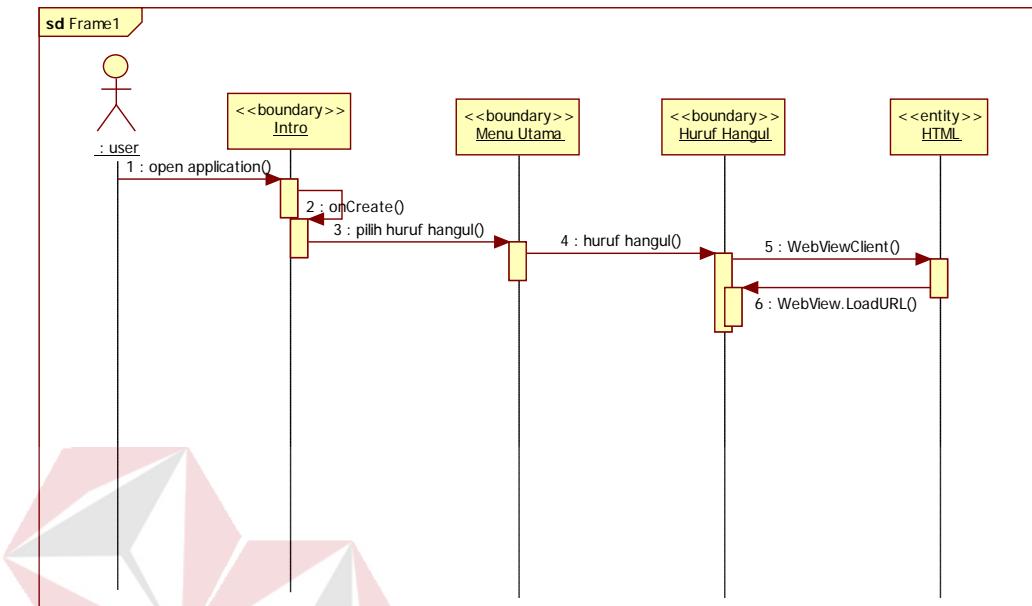
### 3.6.3 Sequence Diagram Contoh Percakapan



Gambar 3.7 Sequence Diagram Contoh Percakapan

Sequence diagram menunjukkan aliran fungsionalitas dalam *use case*. Proses dalam *use case* diawali ketika *user* membuka aplikasi akan dijalankan fungsi *onCreate()*. Kemudian *user* memilih menu contoh percakapan. Setelah memilih menu contoh percakapan fungsi *webViewClient()* dijalankan untuk mengambil data dan menampilkan ke halaman contoh percakapan dengan fungsi *webView.LoadURL()*.

### 3.6.4 Sequence Diagram Huruf Hangul



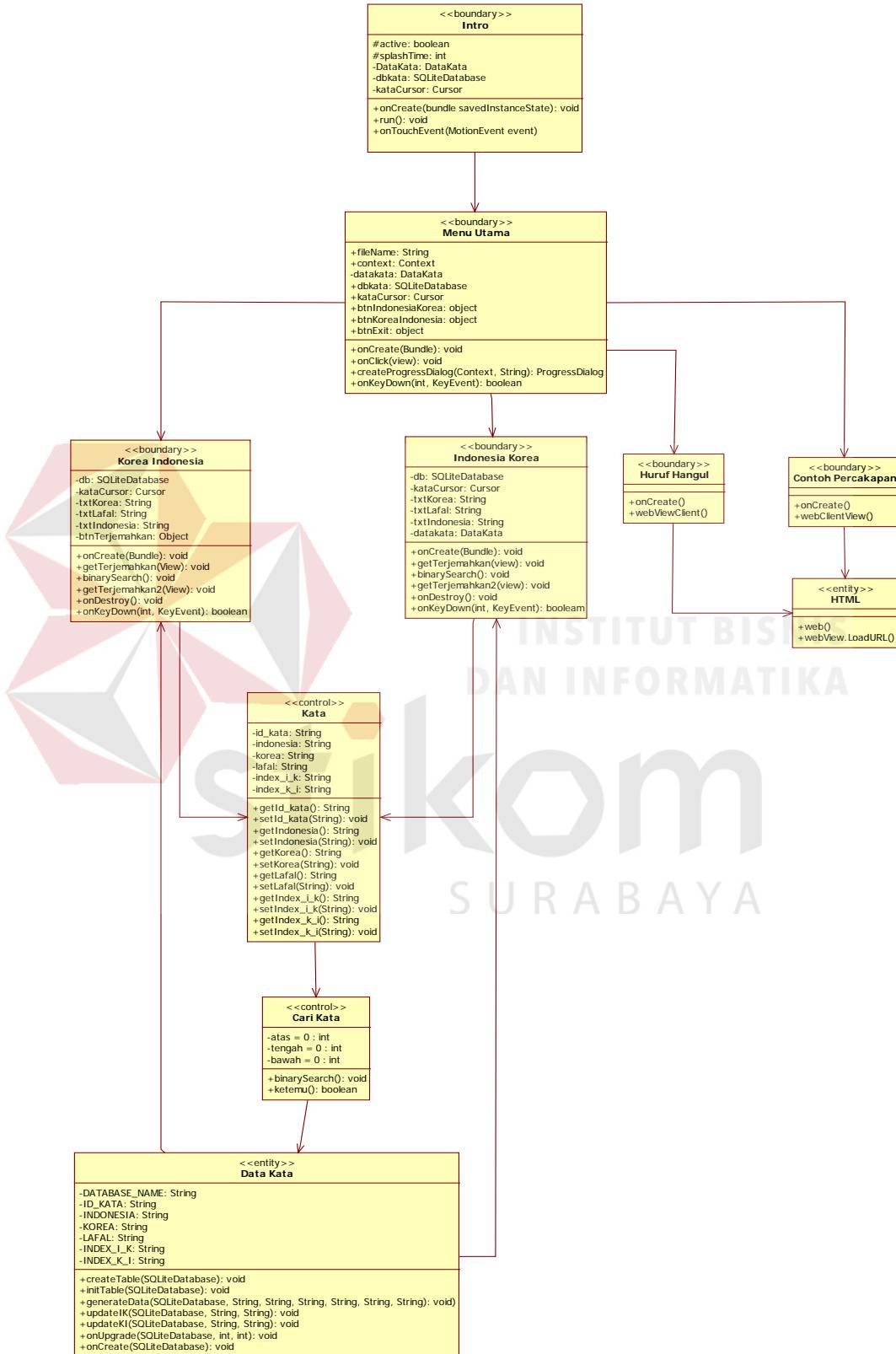
Gambar 3.8 Sequence Diagram Huruf Hangul

Sequence diagram menunjukkan aliran fungsionalitas dalam *use case*.

Proses dalam *use case* diawali ketika *user* membuka aplikasi akan dijalankan fungsi *onCreate()*. Kemudian *user* memilih menu huruf hangul. Setelah memilih menu huruf fungsi *webViewClient()* dijalankan untuk mengambil data dan menampilkan ke halaman contoh percakapan dengan *webView.LoadURL()*.

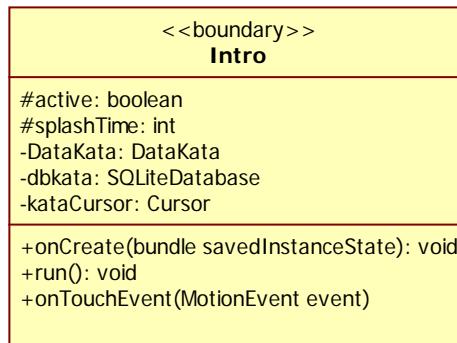
### 3.7 Class Diagram

*Class diagram* atau diagram kelas adalah alat perancangan terbaik untuk tim pengembang perangkat lunak.



Gambar 3.9 Class Diagram Aplikasi Kamus Korea-Indonesia

### 3.7.1 Class Intro Boundary



Gambar 3.10 Class Intro

*Class Intro* digunakan sebagai layar pembuka untuk menyiapkan aplikasi seperti menata database sebelum aplikasi dijalankan.

### 3.7.2 Class Menu Utama Boundary



Gambar 3.11 Class Menu Utama

*Class Menu Utama* digunakan untuk menampilkan menu – menu pada aplikasi. Di dalam *class* ini terdapat atribut dan operasi untuk memanggil *class* lain yang berhubungan.

### 3.7.3 Class Menu Korea – Indonesia Boundary



Gambar 3.12 Class Korea-Indonesia

*Class* Korea Indonesia digunakan untuk melakukan pencarian kata dari bahasa Korea dengan cara menginputkan kata dengan huruf hangul dan nantinya akan ditampilkan pada halaman tersebut.

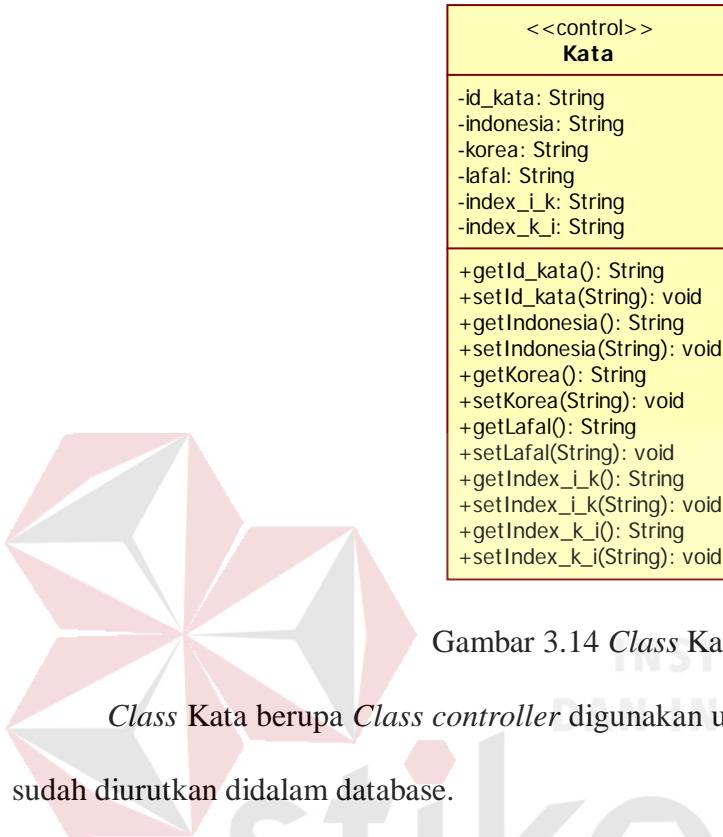
### 3.7.4 Class Menu Indonesia – Korea Boundary



Gambar 3.13 Class Indonesia Korea

*Class* Indonesia Korea digunakan untuk melakukan pencarian kata dari bahasa Indonesia dengan cara menginputkan kata dan nantinya akan ditampilkan pada halaman tersebut.

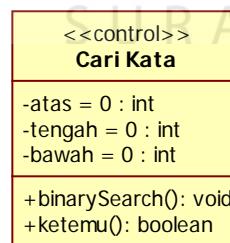
### 3.7.5 Class Kata Control



Gambar 3.14 Class Kata

*Class* Kata berupa *Class controller* digunakan untuk mengambil data yang sudah diurutkan didalam database.

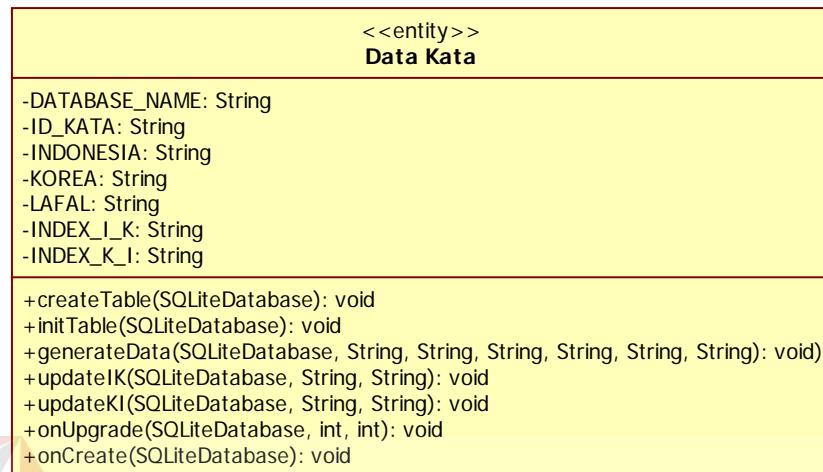
### 3.7.6 Class Cari Kata Control



Gambar 3.15 Class Cari Kata

*Class* Cari Kata termasuk *Class controller*, digunakan untuk menghubungkan kan dari file fisik ke dalam database.

### 3.7.7 Class Data Kata Entity

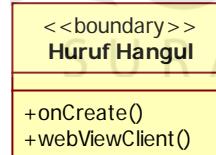


Gambar 3.17 Class Data Kata

*Class* Data Kata digunakan untuk mengontrol aplikasi dengan *database*.

*Class* ini berisi atribut – atribut yang digunakan untuk melakukan koneksi dan operasi – operasi yang digunakan untuk melakukan *query* dan *read database*.

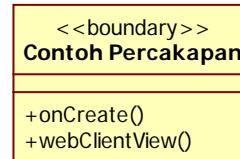
### 3.7.8 Class Menu Huruf Hangul Boundary



Gambar 3.18 Class Huruf Hangul

*Class* Huruf Hangul digunakan untuk menampilkan huruf hangul dan cara penyusunannya.

### 3.7.9 Class Menu Contoh Percakapan Boundary



Gambar 3.19 Class Contoh Percakapan

*Class* Huruf Hangul digunakan untuk menampilkan beberapa contoh percakapan.

### 3.7.10 Class HTML Entity



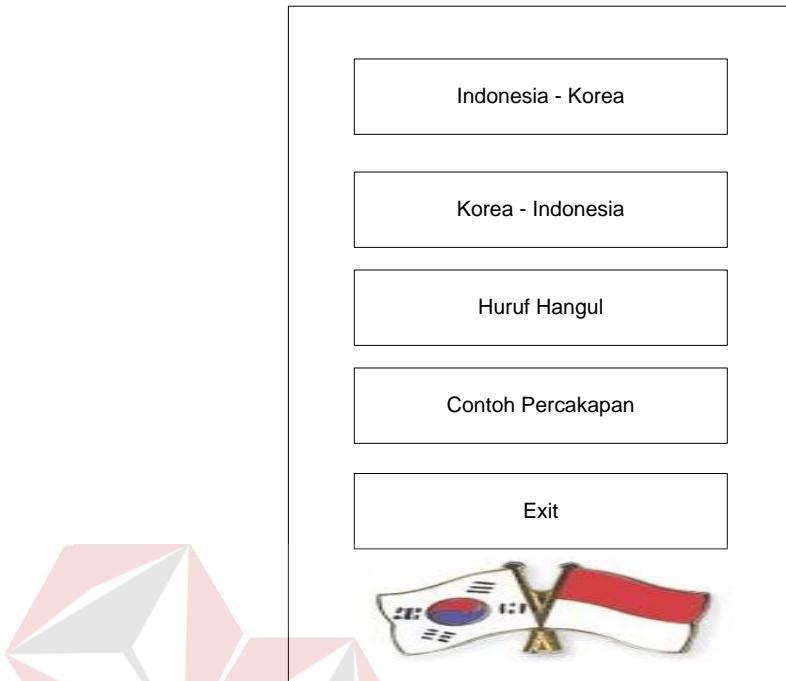
Gambar 3.20 Class HTML

*Class* HTML digunakan untuk menampilkan menu huruf hangul dan contoh percakapan via HTML.

## 3.8 Perancangan User Interface

### A. Menu Utama

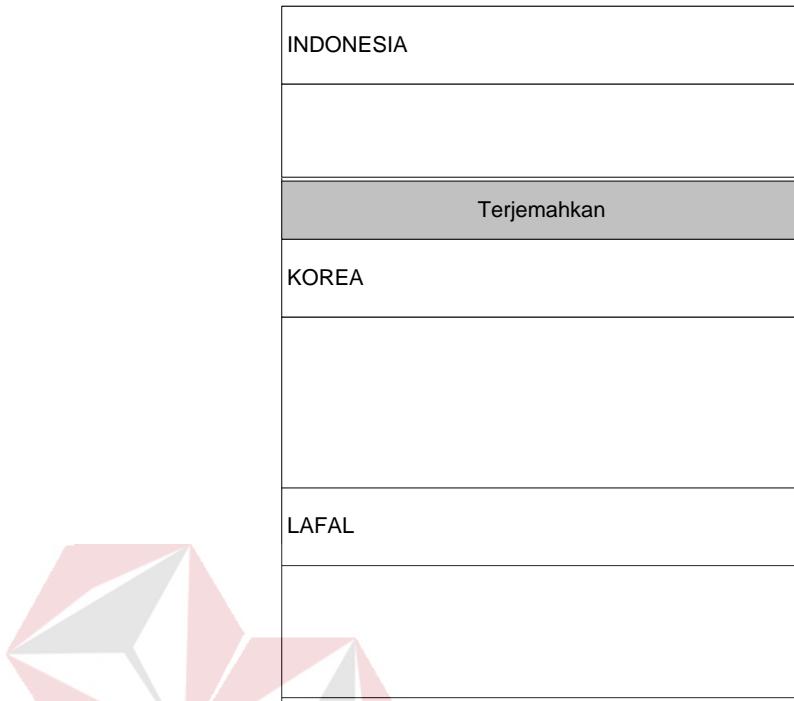
Rancangan menu utama ini berisi menu pilihan kamus. Pada tampilan menu pilihan kamus ini memiliki 3 menu pilihan, yaitu menu kamus Indonesia – Korea, menu kamus Korea – Indonesia dan contoh percakapan.



Gambar 3.21 Desain Halaman Menu Utama

#### B. Halaman Pencarian Kamus Indonesia – Korea

Rancangan menu Halaman pencarian kamus Indonesia – Korea digunakan oleh pengguna untuk mencari arti kata dari bahasa Indonesia ke bahasa Korea. Menu ini bisa dijalankan apabila pengguna sudah memilih tombol menu kamus Indonesia – Korea pada menu pilihan kamus. Tampilan menu pencarian ini terdapat kolom *text field* dan tombol terjemahkan dan menu. Pengguna memasukkan kata yang akan dicari pada kolom *text field* kemudian jika kata yang dicari ada maka muncul di kolom *list view*. Jika kata yang dicari ada maka selanjutnya pengguna menekan tombol cari untuk melihat hasil pencarian.



Gambar 3.22 Desain Halaman Menu Pencarian Indonesia – Korea

### C. Pencarian Kamus Korea – Indonesia

Rancangan menu pencarian kamus Indonesia – Korea digunakan oleh pengguna untuk mencari arti kata dari bahasa Indonesia ke bahasa Korea. Menu ini bisa dijalankan apabila pengguna sudah memilih tombol menu kamus Indonesia – Korea pada menu pilihan kamus. Tampilan menu pencarian ini terdapat kolom *text field* dan tombol – tombol yaitu cari dan menu. Pengguna memasukkan kata yang akan dicari pada kolom *text field* kemudian jika kata yang dicari ada maka muncul di kolom *list view*. Jika kata yang dicari ada maka selanjutnya pengguna menekan tombol cari untuk melihat hasil pencarian. Tombol menu digunakan apabila pengguna ingin memilih menu lain pada menu utama.

KOREA
Terjemahkan
LAFAL
INDONESIA

Gambar 3.23 Desain Halaman Menu Pencarian Korea-Indonesia

#### D. Menu Huruf Hangul

Rancangan menu huruf hangul menampilkan macam – macam huruf hangul dan contoh penyusunan huruf hangul.

Abjad Hangeul (한글)											
ㄱ	ㄲ	ㄴ	ㄸ	ㄹ	ㅁ	ㅂ	ㅃ				
기역	쌍기역	니운	디귿	쌍디귿	리을	미음	비을				
g, k	kk	n	d, t	ssang digeut	rieul	mieum	bieup				
k, g	kk	n	t, d	tt	l, r	m	b, p	pp	pp	pp	pp
[k/g]	[k*]	[n]	[t/d]	[t*]	[l/r]	[m]	[p/b]	[p*]			
ㅅ	ㅆ	ㅇ	ㅈ	ㅉ	ㅊ	ㅋ	ㅌ	ㅍ	ㅎ		
시옷	쌍시옷	이옹	지옷	쌍지옷	치읓	기읔	티읕	피읕	히읕		
s	ss	ng	j	jj	ch	k	t	p	h		
s	ss	-ng	ch, j	tch	ch'	k'	t'	p'	h		
[s]	[s*]	[ø-ŋ]	[tʃ-ʃ]	[tʃ*]	[tʃ']	[k*]	[t*]	[p*]	[h*]		
Vowels (모음/母音)											
ㅏ	ㅑ	ㅓ	ㅕ	ㅗ	ㅕ	ㅓ	ㅕ	ㅗ	ㅕ	ㅕ	ㅕ
a	ae	ya	yae	eo	e	yeo	ye	o	wa	wae	
a	ae	ya	yae	ö	e	yö	ye	o	wa	wae	
[a]	[ae]	[ja]	[yae]	[ø]	[e]	[yö]	[ye]	[o]	[wa]	[wae]	
ㅓ	ㅕ	ㅏ	ㅕ	ㅓ	ㅓ	ㅓ	ㅓ	ㅓ	ㅓ	ㅓ	ㅓ
oe	yo	u	wö	we	wi	yu	eu	ui	i		
oe	yo	u	wö	we	wi	yu	ü	üi	i		
[we]	[jo]	[u]	[wö]	[we]	[wi]	[ju]	[i]	[ü]	[i]		

#### Cara penyusunan

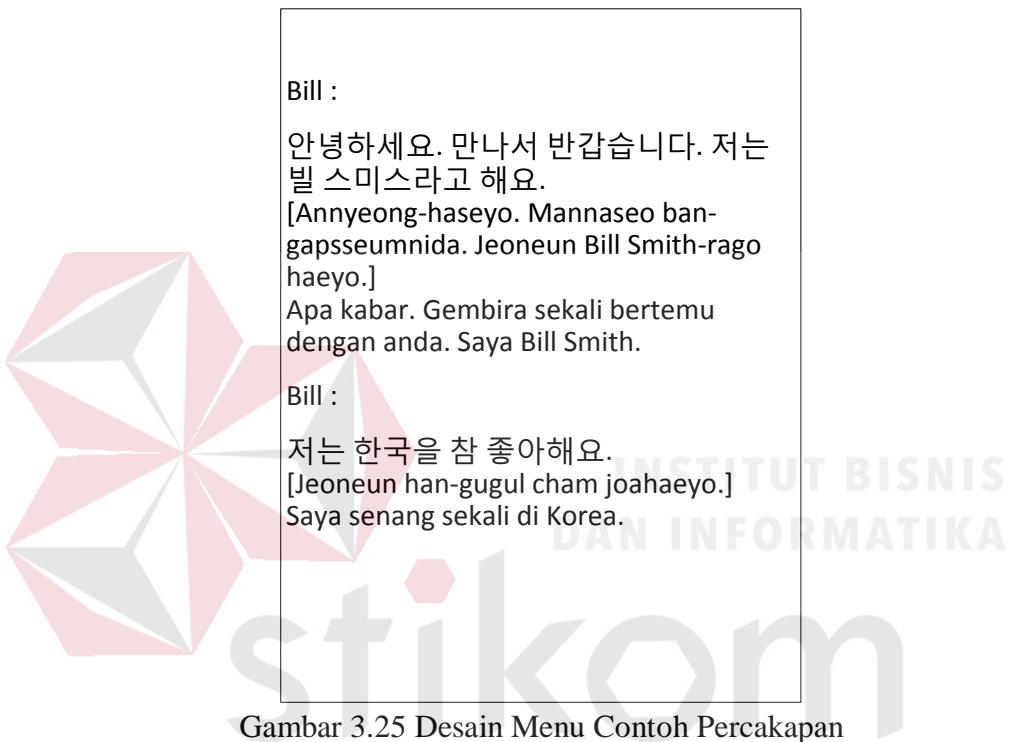
1 K	2 V
-----	-----

ㅅ + ㅓ = ㅓ

Gambar 3.24 Desain Menu Huruf Hangul

### E. Menu Contoh Percakapan

Rancang bangun menu contoh percakapan ini hanya menampilkan beberapa contoh saja percakapan dalam bahasa Korea beserta terjemahannya dalam bahasa Indonesia.



Gambar 3.25 Desain Menu Contoh Percakapan

### 3.9 Desain Uji Coba

Desain uji coba bertujuan untuk memastikan bahwa aplikasi yang telah dibuat dengan benar sesuai dengan petunjuk yang diharapkan. Dalam uji coba aplikasi ini menggunakan *white box testing* dan *black box testing*.

#### 3.9.1 Unit Testing

##### A. White Box Testing

Pengujian *white box* adalah pengujian yang didasarkan pada pengecekan terhadap detail perancangan, menggunakan struktur kontrol dari

desain program secara prosedural untuk membagi pengujian ke dalam beberapa kasus pengujian.

Tujuan penggunaan *white box* untuk menguji semua statement program. Penggunaan metode pengujian *white box* dilakukan untuk :

1. Menguji pengurutan kata pada data awal dan disimpan ke dalam array sementara.

```

ix=0;

datakata = new DataKata(Intro.this);

        dbkata = datakata.getWritableDatabase();

        kataCursor = dbkata.rawQuery("SELECT id_kata, indonesia,
korea, lafal, indeks_i_k, indeks_k_i FROM kata order by indonesia", null);

        if (kataCursor.moveToFirst()) {

            for ( ; !kataCursor.isAfterLast();
kataCursor.moveToNext()) {

                ix++;

                datakata.updateIK(dbkata,kataCursor.getString(0).toString(),Integer
.toString(ix));
            }
        }

        ix=0;

        datakata = new DataKata(Intro.this);

        dbkata = datakata.getWritableDatabase();

        kataCursor = dbkata.rawQuery("SELECT id_kata, indonesia,
korea, lafal, indeks_i_k, indeks_k_i FROM kata order by korea", null);

        if (kataCursor.moveToFirst()) {

            for ( ; !kataCursor.isAfterLast();
kataCursor.moveToNext()) {

                ix++;

                datakata.updateKI(dbkata,kataCursor.getString(0).toString(),Integer
.toString(ix));
            }
        }
    }
}

```

Gambar 3.25 Mengurutkan Kata di Array

## 2. Menguji proses pengambilan data di array

```

ix=0;

datakata = new DataKata(KoreaIndonesia.this);

db = datakata.getWritableDatabase();

kataCursor = db.rawQuery("SELECT id_kata, indonesia,
korea, lafal, indeks_i_k, indeks_k_i FROM kata order by
indeks_k_i", null);

if (kataCursor.moveToFirst()) {

    for (; !kataCursor.isAfterLast();
kataCursor.moveToNext()) {

        ix++;

        //datakata.updateIK(db,kataCursor.getString(0).toString(),In
teger.toString(ix));

        String id =
kataCursor.getString(0).toString();

        String i =
kataCursor.getString(1).toString();

        String k =
kataCursor.getString(2).toString();

        String l =
kataCursor.getString(3).toString();

        String ik =
kataCursor.getString(4).toString();

        String ki =
kataCursor.getString(5).toString();

        ListKata.add(new Kata(id,i,k,l,ik,ki));

    }

}

```

Gambar 3.26 Mengambil data di Array

## 3. Menguji pencarian dengan metode *Binary Search*

```

kataindonesia = txtIndonesia.getText().toString();

atas=1;

```

```

bawah=ix;

tengah=(atas+bawah)/2;

ketemu=false;

binarySearch();

```

Gambar 3.27 Mencari Indeks Tengah

```

binarySearch();

String result = "";
String result2 = "";

if(ketemu==false){
    result = "Terjemahan Not Found";
    result2 = "Terjemahan Not Found";
} else{
    result = ListKata.get(tengah-1).getKorea();
    result2 = ListKata.get(tengah-1).getLafal();
}
txtKorea.setText(result);
txtLafal.setText(result2);
}

```

Gambar 3.28 Membandingkan Data Tengah

```

public void binarySearch()

    while ( atas < bawah && ketemu==false) {

        Log.d("atas,tengah,bawah,ketemu",
        atas+","+tengah+","+bawah+","+ketemu);

        Log.d("kata 1 vs kata 2", kataindonesia+" vs
        "+ListKata.get(tengah-1).getIndonesia());

        if
        (kataindonesia.compareToIgnoreCase(ListKata.get(tengah-
        1).getIndonesia()) == 0) {

            ketemu=true;
        }
    }
else {
    if
    (kataindonesia.compareToIgnoreCase(ListKata.get(tengah-1).getIndonesia()) >

```

```

0) {
    atas = tengah;
    //binarySearch();

} else if
(kataindonesia.compareToIgnoreCase(ListKata.get(tengah-1).getIndonesia()) <
0) {
    bawah = tengah;
    //binarySearch();

}
tengah = ( atas + bawah) / 2;
}

if(ketemu==false && bawah-atas==1){

    if
(kataindonesia.compareToIgnoreCase(ListKata.get(bawah-1).getIndonesia()) ==
0) {
        ketemu=true;
        tengah=bawah;
    } else if
(kataindonesia.compareToIgnoreCase(ListKata.get(atas-1).getIndonesia()) ==
0) {
        ketemu=true;
        tengah=atas;
    } else{
        break;
    }
}
}

```

Gambar 3.29 Mencari Dengan Metode Binary Search

## B. Black Box Testing

Tujuan dari uji coba ini adalah untuk mengetahui apakah fungsi-fungsi dari aplikasi kamus Indonesia-Korea berbasis Android dengan menggunakan metode *Binary Search* ini telah berjalan dengan lancar. Teknik tes pada pengujian ini yaitu *functional analysis*, setiap fitur dan fungsi yang

disediakan akan diuji hasilnya sesuai dengan *test case*. Desain uji coba pada aplikasi ini adalah sebagai berikut :

### 1. Desain Uji Coba Pencarian Korea-Indonesia

Proses pencarian dilakukan dengan cara menginputkan kata kunci atau *keyword* dari bahasa Korea yang ingin dicari kemudian sistem akan melakukan pencarian dan menampilkan hasil pencarian berupa kata indonesia dan cara baca huruf hangul yang sesuai dengan kata kunci. Desain pengujian pada fungsi pencarian Korea-Indonesia dapat dilihat pada tabel 3.10.

Tabel 3.10 Desain Uji Coba Pencarian Korea-Indonesia

Test Case	Tujuan	Input	Output yang diharapkan
1	Mencari Terjemahan kata dari bahasa Korea ke bahasa Indonesia	Memasukkan kata dengan <i>Hangul</i>	Aplikasi berhasil menampilkan kata yang dicari beserta latin dan terjemahan ke dalam bahasa Indonesia
2	Menampilkan pesan jika pencarian tidak ketemu	Memasukkan kata yang salah	form menampilkan kata “Terjemahan Not Found”
3	Menampilkan pesan jika yang diinputkan bukan huruf <i>Hangul</i>	Memasukkan kata dengan alfabet	form menampilkan kata “Terjemahan Not Found”

### 2. Desain Uji Coba Pencarian Indonesia-Korea

Proses pencarian dilakukan dengan cara menginputkan kata kunci atau *keyword* dari bahasa Indonesia yang ingin dicari kemudian sistem akan

melakukan pencarian dan menampilkan hasil pencarian berupa kata Korea dan cara baca huruf hangul yang sesuai dengan kata kunci. Desain pengujian pada fungsi pencarian Indonesia-Korea dapat dilihat pada tabel 3.11.

Tabel 3.11 Desain Uji Coba Pencarian Indonesia-Korea

Test Case	Tujuan	Input	Output yang diharapkan
4	Mencari Terjemahan kata dari bahasa Indonesia ke bahasa Korea	Memasukkan kata dengan huruf alfabet	Aplikasi berhasil menampilkan kata yang dicari beserta terjemahan bahasa Korea dan latinnya
5	Menampilkan pesan jika pencarian tidak ketemu	Memasukkan kata yang salah	form menampilkan kata “Terjemahan Not Found”
6	Menampilkan pesan jika yang diinputkan bukan huruf alfabet	Memasukkan kata dengan huruf <i>Hangul</i>	form menampilkan kata “Terjemahan Not Found”

### 3. Desain Uji Coba Huruf Hangul

Proses pencarian dilakukan dengan cara memilih menu huruf hangul dan akan muncul tabel huruf hangul dan cara penyusunannya. Desain pengujian pada fungsi huruf hangul dapat dilihat pada tabel 3.12.

Tabel 3.12 Desain Uji Coba Huruf Hangul

Test Case	Tujuan	Input	Output yang diharapkan
7	Menampilkan tabel macam – macam huruf hangul dan cara penyusunan huruf hangul	Memilih menu “huruf hangul” pada menu utama	Aplikasi berhasil menampilkan tabel yang berisi huruf hangul dan tabel cara penyusunan huruf hangul

#### 4. Desain Uji Coba Contoh Percakapan

Proses pencarian dilakukan dengan cara memilih menu contoh percakapan. Desain pengujian pada fungsi contoh percakapan dapat dilihat pada tabel 3.13.

Tabel 3.13 Desain Uji Coba Contoh Percakapan

Test Case	Tujuan	Input	Output yang diharapkan
8	Menampilkan contoh-contoh percakapan	Memilih menu “contoh percakapan” pada menu utama	Aplikasi berhasil menampilkan contoh percakapan dalam bahasa Indonesia dan bahasa Korea beserta latinnya

