BAB 3
ANALISIS DAN PERANCANGAN SISTEM

3.1 Analisis Sistem

Analisis sistem dapat didefinisikan sebagai penguraian dari suatu sistem yang utuh ke dalam bagian-bagian komponennya dengan maksud untuk mengidentifikasikan dan mengevaluasikan permasalah-permasalahan, kesempatan-kesempatan, hambatan-hambatan yang terjadi dan kebutuhan-kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan-perbaikan. Analisis dapat juga diartikan sebagai penelitian atas sistem yang telah ada dengan tujuan untuk merancang sistem yang baru atau diperbarui. Dalam proses pembuatan suatu sistem mutlak dilakukan penelitian dan penganalisaan tentang sistem yang akan dibangun.

3.1.1 Analisis Masalah

Dengan beragam daerah dan negara yang dikunjungi *traveler*, terdapat beberapa kendala saat *traveling* salah satunya adalah perbedaan bahasa. Penggunaan buku kamus bahasa masih menjadi salah satu pilihan dalam menerjemahkan bahasa. Namun dalam penggunaannya sebagian *traveler* memiliki kendala yaitu ketika akan menerjemahkan bahasa asing yang sulit untuk ditulis dan diucapkan seperti bahasa jepang, korea, china, rusia dan lain sebagainya akan mempersulit *traveler* dalam menerjemahkan bahasa. Kendala lain pada saat *traveling* yaitu tidak semua tempat yang dikunjungi *traveler* terdapat koneksi internet seperti daerah atau tempat terpencil, hal tersebut mengakibatkan terbatasnya penggunaan aplikasi yang dapat digunakan, khususnya aplikasi yang membutuhkan koneksi internet dalam pengoperasiannya. Selain itu mengingat kegiatan *traveling* yang memiliki mobilitas tinggi dibutuhkan pula sebuah aplikasi yang dapat diakses kapanpun dan dimanapun.

Untuk mengatasi kendala-kendala tersebut, pertama dibutuhkan sebuah aplikasi penterjemah bahasa yang menerapkan teknologi *speech to text* dan *text to*

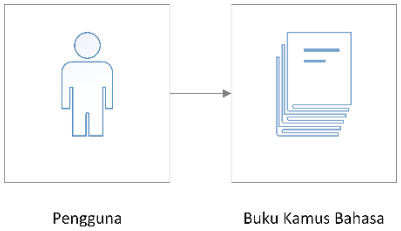
speech yang dapat mengkonversi ucapan menjadi teks dan mengkonversi teks menjadi ucapan guna membantu *traveler* dalam menerjemahkan bahasa asing yang sulit untuk ditulis dan diucapkan. Dengan memanfaatkan API (*Application Programming Interface*) yang disediakan oleh Google dan Microsoft pada aplikasi yang akan dibangun diharapkan dapat menangani kendala tersebut, selain media teks yang digunakan dalam menerjemahkan bahasa, proses penterjemahan dapat dilakaukan melalui suara (ucapan). Kedua, dibutuhkan suatu aplikasi yang dapat digunakan dalam kondisi terhubung internet (*online*) maupun tidak terhubung internet (*offline*). Dalam hal ini, untuk penggunaan secara *offline* pada aplikasi akan dibangun sebuah fitur *Phrasebook* yang berisi *files* percakapan umum yang sering digunakan pada saat *traveling* yang terdiri dari beberapa kategori. Ketiga, dengan dibangunnya sebuah aplikasi berbasis *mobile*, maka aplikasi dapat digunakan kapanpun dan dimanapun.

Berdasarkan uraian yang dikemukakan, maka terdapat masalah yaitu bagaimana menerapkan teknologi *speech to text* dan *text to speech* pada aplikasi penterjemah bahasa dengan memanfaatkan API (*Application Programming Interface*) yang disediakan *Google* dan *Microsoft* guna membantu *traveler* dalam menerjemahkan bahasa asing yang sulit untuk ditulis dan diucapkan, bagaimana membuat fitur *Phrasebook* yang berisi *files* percakapan umum yang sering digunakan pada saat *traveling* yang terdiri dari beberapa kategori yang dapat digunakan secara *offline*, serta bagaimana membuat sebuah aplikasi berbasis *mobile* yang dapat diakses kapanpun dan dimanapun.

3.1.2 Analisis Prosedur yang Sedang Berjalan

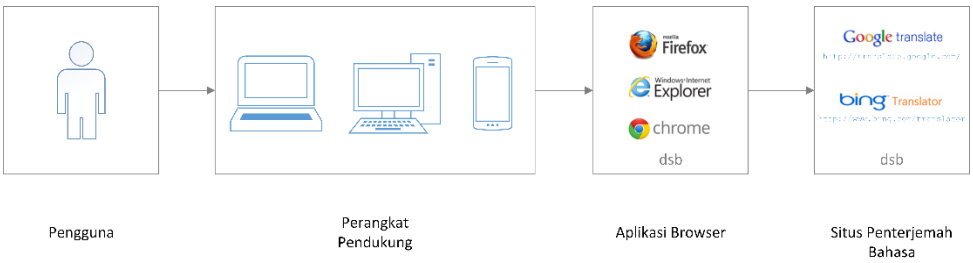
Bahasa merupakan sebuah alat komunikasi yang penting bagi manusia. Di dunia ini banyak sekali keanekaragaman bahasa namun tidak mudah untuk dapat memahami seluruh bahasa yang ada di dunia ini. Terdapat beberapa cara/prosedur yang dapat digunakan untuk menerjemahkan bahasa diantaranya penggunaan buku kamus bahasa, penggunaan situs yang menyediakan fasilitas penterjemahan bahasa, dan penggunaan aplikasi penterjemahan bahasa yang tersedia pada perangkat *mobile*.

1. Penggunaan buku kamus bahasa masih menjadi salah satu media dalam menangani perbedaan bahasa. Dalam proses penggunaannya, pengguna diharuskan mencari satu persatu kata/kalimat dari sebuah percakapan yang hendak di terjemahkan yang pada umumnya tersusun berdasarkan abjad. Prosedur ini membutuhkan waktu yang relatif lama dikarenakan dalam pencariannya dilakukan secara manual. Selain itu, pengguna akan disulitkan dengan lawan bicara yang memiliki bahasa yang sulit untuk ditulis dan diucapkan seperti bahasa china, korea, jepang, rusia dan lain sebagainya.



Gambar 3.1 Alur Penggunaan Buku Kamus Bahasa

2. Penggunaan situs penterjemah bahasa seperti *Google Translator* ataupun *Bing Translator* yang tersedia pada http://translate.google.com/ dan http://www.bing.com/translator sangat familiar dalam menerjemahkan bahasa. Dalam penggunaannya, cara/prosedur ini relatif lebih cepat dalam menerjemahkan bahasa dibanding menggunakan buku kamus bahasa. Untuk dapat menerjemahkan bahasa menggunakan *Google Translator* ataupun *Bing Translator*, pengguna diharuskan memiliki koneksi internet dalam pengoperasiannya.



Gambar 3.2 Alur Penggunaan Situs Penterjemah Bahasa

3. Saat ini tersedia aplikasi *mobile* yang menyediakan fasilitas penterjemahan bahasa. Dalam penggunaannya, cara/prosedur ini lebih praktis karena

pengguna tidak terlebih dahulu harus membuka aplikasi *browser* untuk dapat menerjemahkan bahasa.



Gambar 3.3 Alur Penggunaan Aplikasi Penterjemah Bahasa

Conversation Translator merupakan contoh aplikasi penterjemah bahasa yang digunakan untuk berkomunikasi dengan orang asing. Aplikasi ini dapat digunakan untuk *travel, vacations or business trips*. Aplikasi yang tersedia di *Google Play* ini menawarkan beberapa fitur yaitu, dapat menerjemahkan kata atau frasa, menerjemahkan melalui suara (*speech to text*), dan mendengarkan hasil terjemahan (*text to speech*). Aplikasi ini membutuhkan koneksi internet dalam pengoperasiannya. Berikut beberapa antarmuka aplikasi *Conversation Translator:*



Gambar 3.4 Antarmuka Aplikasi *Conversation Translator*

Dalam penggunaan *Google Translator* ataupun *Bing Translator* dan aplikasi *Conversation Translator* terdapat kelemahan yaitu dibutuhkan koneksi internet dalam pengaksesannya. Jika melihat aktivitas *traveling* yang memiliki mobilitas tinggi, tidak semua tempat yang dikunjungi terdapat koneksi internet seperti

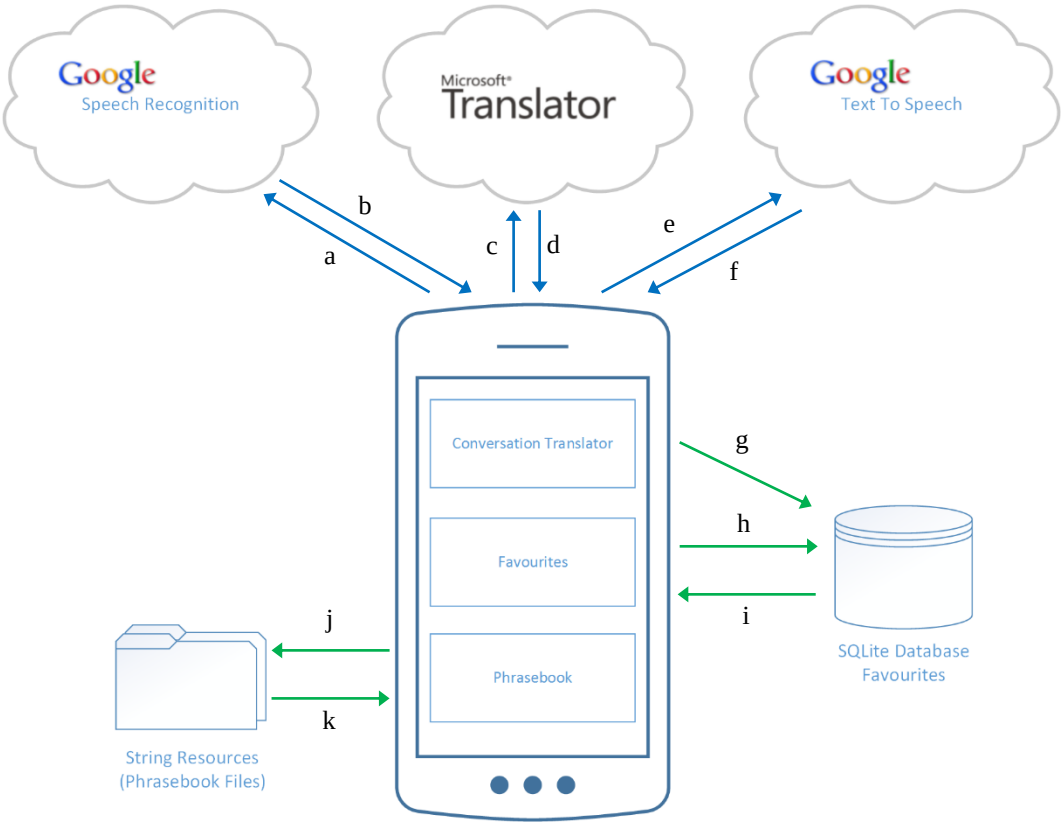
daerah atau tempat terpencil, hal tersebut mengakibatkan terbatasnya penggunaan aplikasi yang dapat digunakan, khususnya aplikasi yang membutuhkan koneksi internet dalam pengoperasiannya.

3.1.3 Analisis Sistem Yang Dibangun

Sistem yang akan dibangun adalah aplikasi *Traveling Conversation Assistance* berbasis *mobile* yang dapat dioperasikan menggunakan *platform* Android. Dalam pembangunannya, aplikasi ini mengimplementasi teknologi *speech to text* dan *text to speech* yang bertujuan untuk mempermudah pengguna (*traveler*) dalam berkomunikasi dengan lawan bicara yang memiliki bahasa yang sulit untuk ditulis dan diucapkan, seperti bahasa jepang, korea, china, rusia dan lain sebagainya. Untuk menunjang kegiatan *traveler* yang memiliki mobilitas tinggi, aplikasi yang akan dibangun dapat digunakan dalam kondisi *online* maupun *offline*.

Dalam penggunaan *online*, pengguna dapat menjalankan fitur penterjemahan yang didukung oleh *Microsoft Translator*. Bahasa dan hasil terjemahan bahasa dapat disimpan kedalam fitur *Favourites* yang nantinya dapat diakses secara *offline*. Selain media teks yang digunakan dalam proses penterjemahan, pengguna dapat menerjemahkan percakapan melalui suara yang didukung oleh *Google Speech Recognition*, selain itu dengan dukungan *Google Text To Speech* pengguna dapat melafalkan hasil terjemahan kedalam bentuk suara.

Dalam penggunaan *offline*, pengguna dapat menjalankan fitur *Phrasebook* yang berisi percakapan (kata/kalimat) umum yang sering digunakan pada saat *traveling* yang terdiri dari beberapa kategori yaitu *greetings*, *general conversation*, *number*, *directions and places*, *transportation*, *eating out*, dan *emergency*. Bahasa yang didukung pada fitur *Phrasebook* yaitu *Indonesian*, *English*, *Spain*, *France*, *China*, *Italy*, dan *Germany*. Gambar 3.5 menunjukan gambaran umum sistem yang akan dibangun.



Gambar 3.5 Alur Kerja Sistem Yang Akan Dibangun

Berikut adalah skenario alur kerja sistem yang akan dibangun:

1. *Conversation Translator*



Gambar 3.6 Pilih Bahasa Asal dan Tujuan

Mula-mula pengguna memilih bahasa asal dan bahasa tujuan (contoh: *from Indonesian to English*)

a. Saat pengguna menjalankan fungsi *Speech To Text* sistem akan menampilkan dialog *Voice Input* sesuai dengan bahasa asal yang telah dipilih sebelumnya (*Indonesian*). Setelah dialog *Voice Input* tampil, saat itulah pengguna mulai berbicara. Suara tersebut akan dikirim ke

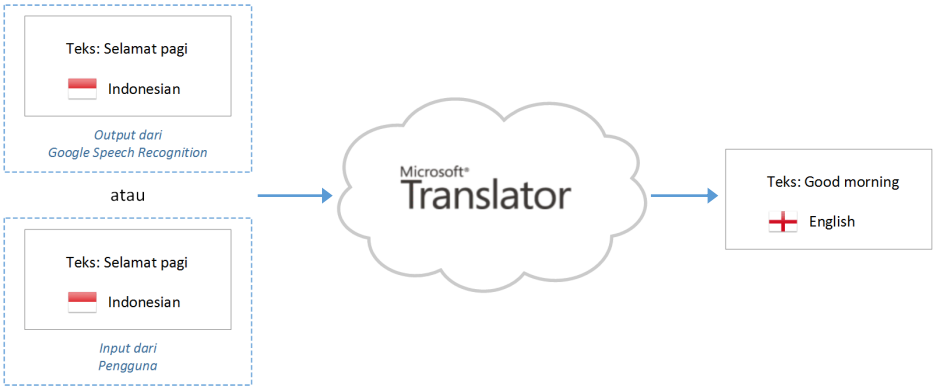
server Google Speech Recognition secara *realtime* untuk di konversi menjadi teks



Gambar 3.7 Konversi Suara ke Teks

b. *Google Speech Recognition* mengkonversi suara kedalam bentuk teks dengan cara membandingkan data suara yang diterima dengan *database* yang dimiliki *Google*. Setelah dilakukan konversi, *Google Speech Recognition* mengirim teks hasil konversi dalam bahasa *Indonesian* (bahasa asal) ke *device*

c. Ketika pengguna menjalankan fungsi *Translate*, teks yang diterima *device* dari *Google Speech Recognition* atau teks yang di-*input*-kan langsung oleh pengguna pada aplikasi (berbahasa *Indonesian*) kemudian dikirim ke *server Microsoft Translator* beserta parameter bahasa asal dan bahasa tujuan yang telah dipilih sebelumnya (*from Indonesian to English*) untuk diterjemahkan



Gambar 3.8 Terjemahkan Bahasa Asal ke Bahasa Tujuan

d. *Microsoft Translator* akan memproses/menerjemahkan teks sesuai dengan bahasa tujuan (*English*). Setelah dilakukan penterjemahan, *Microsoft Translator* mengirim teks hasil terjemahan dalam bahasa *English* (bahasa tujuan) ke *device*

e. Ketika pengguna menjalankan fungsi *Text To Speech*, teks hasil terjemahan yang diterima *device* dari *Microsoft Translator* kemudian dikirim ke *server Google Text To Speech* beserta paramenter bahasa tujuan yang telah dipilih sebelumnya (*English*) untuk dikonversi menjadi suara/ucapan



Gambar 3.9 Konversi Teks ke Suara

f. *Google Text To Speech* akan memproses/mengkonversi teks menjadi suara/ucapan sesuai dengan bahasa tujuan (*English*). Setelah dilakukan konversi, *Google Text To Speech* memutar/menjalankan suara (ucapan) dalam bahasa *English* (bahasa tujuan) secara *realtime* ke *device*, selain itu *device* akan membuat *file* suara dari percakapan hasil terjemahan (*English*) dalam format .mp3 yang disimpan kedalam *SD card*

g. Ketika pengguna menjalankan fungsi *Add To Favourites*, sistem akan menyimpan bahasa asal, bahasa tujuan, teks percakapan asal, teks percakapan hasil terjemahan, dan alamat/*path* suara percakapan hasil terjemahan kedalam *SQLite Database* sebagai *Favourites* yang nantinya dapat diakses secara *offline*. Namun ketika pengguna menjalankan fungsi selain *Add To Favourites* pada fitur *Conversation Translator*, maka sistem akan menghapus file suara yang ada di *SD card* yang telah dibuat sebelumnya

2. *Favourites*

h. Ketika pengguna menjalankan fungsi *Favourites*, sistem akan membaca seluruh data yang berada pada *database*

i. Sistem akan merespon dengan menampilkan data dalam bentuk *ListView*. Data tersebut meliputi teks asal, bahasa asal, teks hasil

terjemahan, bahasa tujuan. Ketika pengguna memilih salah satu *ListView* sistem akan memutar/menjalankan *file* suara (.mp3) dari teks hasil terjemahan

3. *Phrasebook*

   Ketika pengguna menjalankan fungsi *Phrasebook*, sistem akan memeriksa *Language Setting* yang berada pada fitur *Phrasebook*, jika pengguna belum mengatur bahasa maka sistem akan menampilkan dialog *Language Setting*. Pengguna diminta untuk memilih bahasa asal dan bahasa tujuan (contoh: *from English to Frence*), setelah pengguna memilih bahasa asal dan bahasa tujuan sistem akan menyimpan pengaturan bahasa dan akan menampilkan data *Phrasebook* yang dibagi menjadi beberapa kategori yaitu *greetings*, *general conversation*, *number*, *directions and places*, *transportation*, *eating out*, dan *emergency*

   j. Ketika pengguna memilih salah satu kategori dalam *Phrasebook* (contoh: *Greetings*) sistem akan memeriksa bahasa asal dan bahasa tujuan kemudian akan membaca kategori yang telah dipilih sebelumnya (*Greetings*)

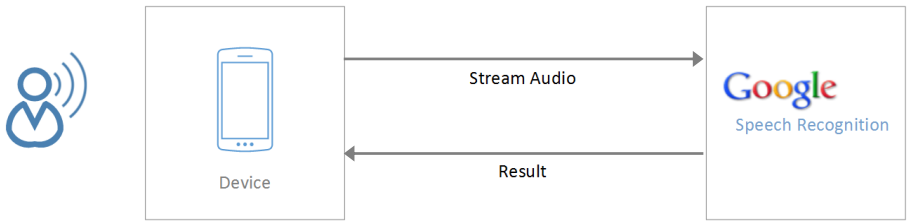
   k. Sistem akan merespon dengan menampilkan *phrase* (kata/percakapan) sesuai kategori yang telah dipilih sebelumnya (*Greetings*) dalam bahasa *English* berberntuk *ListView*. Ketika pengguna memilih salah satu *phrase* (kata/percakapan) sistem akan menampilkan *phrase* berbentuk *toast* dan menjalankan file suara (.mp3) dalam bahasa *Frence*.

3.1.3.1 Analisis *Google Speech Recognition API*

Penggunaan *Google Speech Recognition API* memungkinkan *developer* untuk mengkonversi ucapan kedalam teks. Layanan ini dapat diproses secara *offline* dan *online*. Namun pada pemrosesan secara *offline*, bahasa yang didukung terbatas pada bahasa yang terdapat pada masing-masing *device*/*smartphone*, selain itu dibeberapa versi Android fitur ini tidak dapat dioperasikan. Berbeda dengan pemrosesan secara *online* yang memungkinkan dukungan bahasa yang lengkap

dan dapat dioperasikan hampir di semua versi Android. Namun dalam penggunaannya layanan ini memerlukan koneksi internet karena pengenalan suara (*speech recognition*) terjadi di *server* Google.

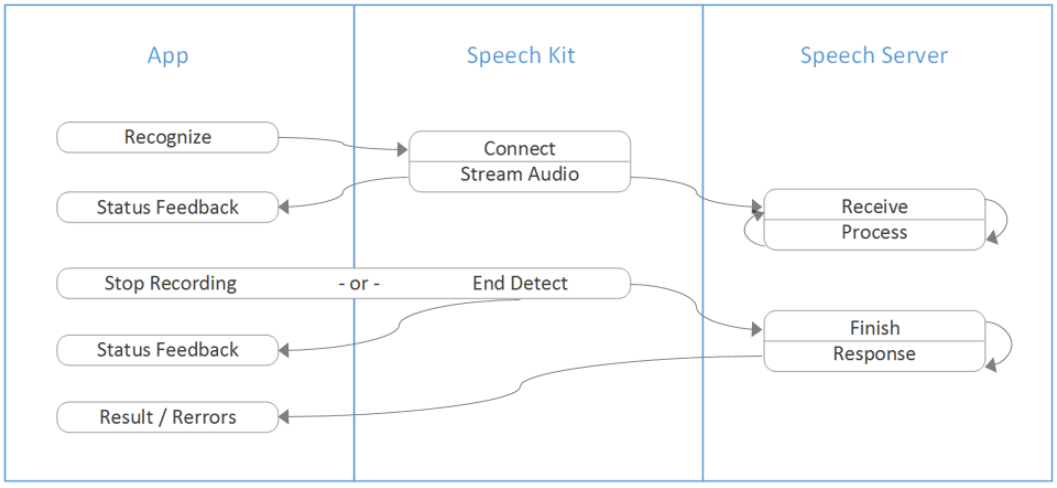
Layanan *Google Speech Recognition* digunakan pada aplikasi ini pada fitur *Conversation Translator* dimana untuk menginputkan pesan berupa percakapan yang hendak di terjemahkan melalui suara.



Gambar 3.10 Alur Penggunaan *Google Speech Recognition*

Input pada proses ini merupakan suara pengguna dan *output* yang dihasilkan pada proses ini berupa teks yang nantinya akan digunakan pada tahapan berikutnya yaitu penterjemahan menggunakan layanan *Microsoft Translator*.

Terdapat beberapa tahapan yang dilakaukan pada proses *speech recognition.* Tahapan tersebut dapat dilhat pada Gambar 3.11.



Gambar 3.11 *Speech Recognition Process*

Pada level *application* ada komponen utama yang tersedia untuk *developer* yaitu *recognizer*.

Pada level *speech kit* ada beberapa proses yang terkoordinasi, yaitu:

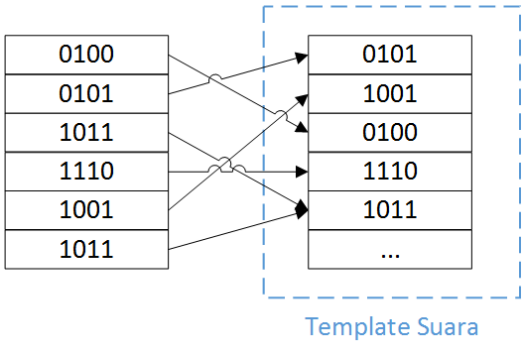
1. *Library* sepenuhnya mengelola sistem audio untuk *recording*
2. Komponen *networking* mengelola koneksi ke *server*

3. *End detection* menentukan bila pengguna telah berhenti berbicara dan secara otomatis menghentikan rekaman.

Sistem *server* bertanggung jawab untuk sebagian besar pekerjaan dalam siklus pemrosesan suara (*speech processing*). Proses pengenalan ucapan (*recognition*) dilakukan pada *server*.

Dalam pemrosesannya untuk dapat mengkonversi suara ke dalam bentuk teks terdapat beberapa tahapan, diantaranya:

1. Suara pengguna berupa sinyal analog dikonversi oleh *device* kedalam bentuk diskrit yang kemuadian di ubah bentuk kedalam bentuk *biner* dan secara *realtime* dikirim ke *server* untuk dilakukan proses konversi
2. Setelah pengguna berhenti berbicara (*stop recording or end detect*) / *server* menerima seluruh data percakapan dalam bentuk digital / *biner*, kemudian *server* akan melakukan proses konversi. Dalam proses konversi terdapat beberapa tahapan, diantaranya:
   a) Data *biner* yang diterima oleh *server* akan dibandingkan dengan *template* data suara

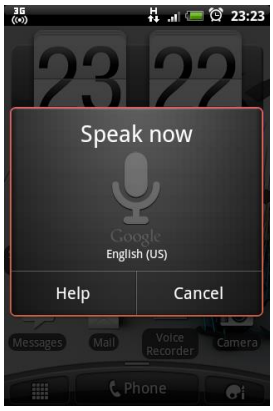


Gambar 3.12 Ilustrasi Perbandingan Data *Biner* dengan *Template*

   b) Tiap *data biner* yang telah dibandingkan dengan *template* suara, kemudian disatukan kembali dan dianalisis secara keseluruhan, kemudian akan dicocokan dari segi tata bahasa dan apakah data yang diucapkan sesuai dengan kata yang tersedia pada *template* data pada *server*
3. Setelah proses konversi selesai, *server* akan mengirim hasil konversi berupa teks (*string*) ke *device*.

Untuk dapat menggunakan fitur *Google Speech Recognition API* pada program diperlukan beberapa langkah, diantaranya mengimpor *class* `android.speech.RecognizerIntent;`

Pada dasarnya inisialisasi *Intent* dari *class RecognizerIntent* berfungsi menampilkan kotak dialog "*Speak now*" untuk mengenali suara masukan (Gambar 3.13). Selanjutnya dengan memanggil *method* `startActivityForResult(i, REQUEST_OK)`, *system* akan mengirimkan secara *realtime* hasil *capture* suara dan menunggu respon dari *server*. Termasuk di dalamnya pada *method* ini terdapat proses ADC (*Analog To Digital Converter*). Jika respon "REQUEST_OK" sistem akan menjalankan *method* `getStringArrayListExtra()` untuk mendapatkan hasil konversi berupa *string* dalam bentuk *ArrayList*.



Gambar 3.13 Dialog *Speech Input*



Gambar 3.14 Dialog Proses *Speech Input*

Pada saat inisialisasi *Intent* dari *class RecognizerIntent* terdapat nilai *extra* yang dibutuhkan yaitu EXTRA_LANGUAGE_MODEL. *Constant Value* yang dapat digunakan yaitu:

1. LANGUAGE_MODEL_FREE_FORM: Model bahasa yang digunakan berdasarkan pengenalan suara-bentuk bebas
2. LANGUAGE_MODEL_WEB_SEARCH: Model bahasa yang digunakan berdasarkan istilah pencarian web.

Adapun *extra* yang bersifat opsional diantaranya:

1. EXTRA_PROMPT: Teks tambahan untuk menunjukan informasi bahasa yang digunakan kepada pengguna ketika mereka berbicara
2. EXTRA_LANGUAGE: Opsional *tag* bahasa misalnya "en-US". *Tag* ini menginformasikan *recognizer* untuk melakukan pengenalan suara dalam bahasa yang berbeda dari yang ditetapkan dalam `getDefault()`.

Pada aplikasi yang akan dibangun nilai *extra* yang digunakan adalah LANGUAGE_MODEL_FREE_FORM, sedangkan nilai *extra* opsional yang digunakan pada parameter EXTRA_PROMPT dan EXTRA_LANGUAGE didapat dari pemilihan bahasa asal. Contoh: Ketika pengguna memilih bahasa asal *Indonesian* maka nilai parameter EXTRA_PROMPT adalah string dari *Indonesian* dan nilai parameter EXTRA_LANGUAGE adalah *id*. Ketika inisialisasi *Intent* pada class maka sistem akan melakukan pengenalan suara dalam bahasa *Indonesian*.



Gambar 3.15 Parameter Ekstra Dialog *Speech Input*

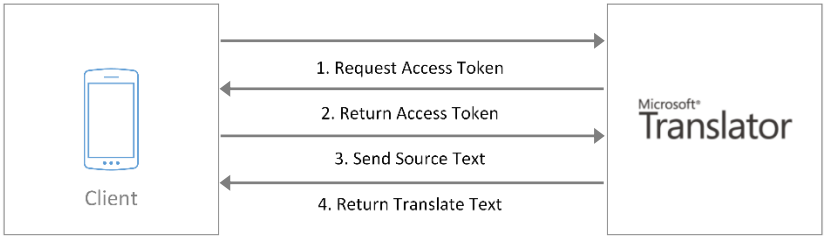
3.1.3.2 Analisis *Microsoft Translator API*

Layanan *Microsoft Translator* diterapkan pada aplikasi ini pada fitur *Conversation Translator*. Layanan *Microsoft Translator* digunakan untuk menerjemahkan suatu bahasa ke bahasa lainnya yang telah ditentukan. Terdapat dua macam *input* pada proses ini, yaitu:

1. Hasil konversi dari suara ke teks pada proses sebelumnya yang menggunakan layanan *Google Speech Recognition*, atau
2. Teks yang di-*input*-kan langsung oleh pengguna.

Sedangkan *output* yang dihasilkan pada proses ini berupa teks yang telah diterjemahkan yang nantinya akan digunakan pada tahapan berikutnya yaitu konversi teks ke suara menggunakan layanan *Google Text To Speech*.

Langkah-langkah menerjemahkan pesan menggunakan *Microsoft Translator API* dapat dilihat pada Gambar 3.16 [11].



Gambar 3.16 Alur Penggunaan *Microsoft Translator API*

1. *Client* memulai dengan meminta *access token* ke *server*, kemudian *server* akan merespon dengan memberikan *access token* ke *Client*.
2. Respon untuk permintaan akses token berisi akses token yang dapat digunakan untuk mengakses *Microsoft Translator API*.
3. Setelah *client* mendapatkan akses token, *client* dapat memulai menerjemahkan pesan teks dari suatu bahasa ke bahasa lain.
4. Jika *request* sukses, maka *server* akan memberikan sebuah nilai *string* yang mewakili teks yang diterjemahkan. Namun ketika *request* gagal, *server* akan mengirim pesan kegagalan. (kegagalan *request* dapat disebabkan oleh terputusnya koneksi internet pada saat proses konversi)

3.1.3.3 Analisis *Google Text To Speech API*

Penggunaan *Google Text To Speech API* memungkinkan *developer* untuk mengkonversi teks menjadi suara. Layanan ini dapat diproses secara *offline* dan *online*. Pada pemrosesan secara *offline*, bahasa yang didukung terbatas pada bahasa yang terdapat pada masing-masing *device*/*smartphone*. Berbeda dengan pemrosesan secara *online* yang memungkinkan dukungan bahasa yang lengkap.

Namun dalam penggunaannya pengguna diharuskan memiliki koneksi internet untuk dapat mengakses *database* pada *server* Google secara *online*.

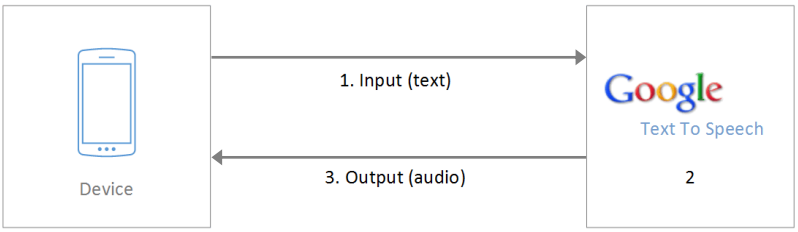
Layanan *Google Text To Speech* digunakan pada aplikasi ini pada fitur *Conversation Translator* dimana *input* pada proses ini merupakan teks percakapan yang telah diterjemahkan sebelumnya pada proses penterjemahan menggunakan layanan *Microsoft Translator* dan *output* yang dihasilkan pada proses ini berupa suara (ucapan).

Prosedur yang digunakan untuk mengkonversi teks menjadi ucapan pada aplikasi yang akan dibangun yaitu dengan menggunakan HTTP GET (REST) *request*: http://translate.google.com/translate_tts?ie=UTF-8&tl=en&q=text dengan keterangan sebagai berikut:

tl : merupakan bahasa yang digunakan, misalnya, *ja* untuk bahasa *Japanese* atau *en* untuk bahasa *English*

q : merupakan *query string* (teks) yang akan dikonversi menjadi suara (ucapan)

Dalam penggunaan URL diatas untuk mendapatkan suara dengan bahasa yang berbeda, hanya mengganti *en* menjadi kode bahasa yang diinginkan. Misalnya pada *de* untuk bahasa *German* atau *ja* untuk bahasa *Japanese*. Alur penggunaan *Google Text To Speech API* untuk dapat mengkonversi teks menjadi suara (ucapan) dapat dilihat pada Gambar 3.17.



Gambar 3.17 Alur Penggunaan *Google Text To Speech*

1. Input teks dan bahasa dikirim *device* ke *Google TTS* melalui *http request*
2. *Google TTS* mengkonversi tulisan yang dikirimkan melalui *http request* sebelumnya kedalam bentuk suara (*audio*)
3. Setelah proses konversi selesai *Google TTS* akan menjalankan suara ke *device* secara *realtime*

Input teks pada *Google TTS* harus memperhatikan tanda baca dan juga jumlah karakternya. Titik dan juga koma akan mempengaruhi artikulasi. Setelah titik “.”

waktu jeda akan lebih lama karena titik merupakan akhir dari sebuah kalimat. Tanda baca "!?.," juga menunjukkan akhir potongan pada kalimat yang mengandung unsur seru dan juga tanya, namun waktu jeda tidak terlalu lama.

Dari 41 bahasa yang terdapat pada *library microsoft-translator-java-api-0.6.2-jar-with-dependencies.jar* hanya 31 bahasa yang didukungan oleh *Google* dalam proses *Text To Speech*. Bahasa yang didukung oleh *Google* dapat dilihat pada Tabel 3.1.

Tabel 3.1 Dukungan Bahasa *Google Text To Speech*

No	Nama Bahasa	*Google Speak Code*	*Google Speak Code* (asumsi)
1	*Arabic*	ar	
2	*Bulgarian*	-	ru
3	*Catalan*	ca	
4	*Chinese Simplified*	zh-CN	
5	*Chinese Traditional*	zh-TW	
6	*Czech*	cs	
7	*Danish*	da	
8	*Dutch*	nl	
9	*English*	en	
10	*Estonian*	-	nl
11	*Finnish*	fi	
12	*French*	fr	
13	*German*	de	
14	*Greek*	el	
15	*Haitian Creole*	ht	
16	*Hebrew*	-	iw
17	*Hindi*	hi	
18	*Hmong Daw*	-	nl
19	*Hungarian*	hu	
20	*Indonesian*	id	
21	*Italian*	it	
22	*Japanese*	ja	
23	*Korean*	ko	
24	*Latvian*	lv	
25	*Lithuanian*	-	nl
26	*Malay*	-	id

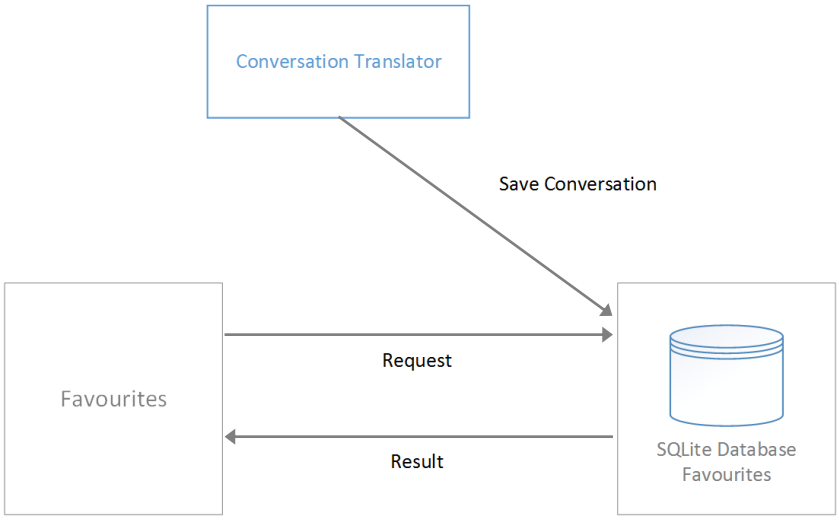
27	*Norwegian*	no	
28	*Persian*	-	ar
29	*Polish*	pl	
30	*Portuguese*	pt	
31	*Romanian*	ro	
32	*Russian*	ru	
33	*Slovak*	sk	
34	*Slovenian*	-	nl
35	*Spanish*	es	
36	*Swedish*	sv	
37	*Thai*	th	
38	*Turkish*	tr	
39	*Ukrainian*	-	ru
40	*Urdu*	-	ar
41	*Vietnamese*	vi	

Dari 10 bahasa yang saat ini tidak didukung oleh *Google*, penulis mengasumsikan untuk menggunakan bahasa yang memiliki penulisan dan pelafalan serupa dengan bahasa yang didukung oleh *Google*. Contoh: untuk bahasa *Bulgarian* dan *Ukrainian* menggunakan bahasa *Russian* (ru).

3.1.3.4 Analisis *SQLite Database Favourites*

Pada fitur *Favourites* memungkinkan pengguna untuk mengakses kembali terjemahan percakapan yang telah dilakukan sebelumnya pada fitur *Conversation Translator* secara *offline*.

Output pada fitur *Conversation Translator* berupa percakapan awal, percakapan hasil terjemahan, dan suara hasil terjemahan akan menjadi *input* pada fitur *Favourites*. Pesan percakapan tersebut akan di-*input*-kan ke dalam *SQLite Database Favourites* yang nantinya akan digunakan pada fitur *Favourites*.



Gambar 3.18 Alur Penggunaan *SQLite Databases Favourites*

1. *Save Conversation*

   Ketika pengguna menjalankan fungsi *Add To Favourites* terdapat 5 parameter yang disimpan kedalam *database* yaitu bahasa asal, bahasa tujuan, teks percakapan asal, teks percakapan hasil terjemahan, dan alamat/*path* audio dari percakapan hasil terjemahan. Mula-mula sistem akan membuka koneksi *database*, lalu memproses *query insert*, setelah proses *insert* selesai kemudian sistem akan menutup koneksi *database*.

2. *Request*

   Ketika pengguna menjalankan fungsi *Favourites* sistem akan membaca seluruh data yang berada pada *database*.

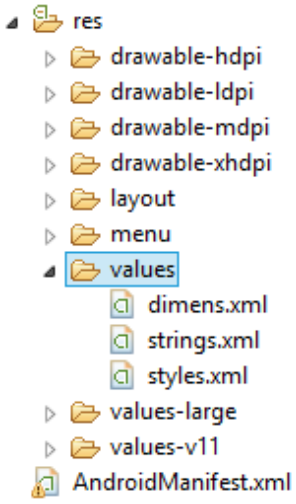
3. *Result*

   Sistem akan merespon dengan memberikan data sesuai apa yang telah di-*request* sebelumnya dalam bentuk *ArrayList* dan menampilkannya dalam bentuk *ListView*. Data tersebut meliputi teks percakapan asal, bahasa asal, teks percakapan hasil terjemahan, dan bahasa tujuan. Ketika pengguna memilih salah satu *ListView* sistem akan membaca alamat/*path* audio yang berada pada *database* yang kemudian menjalankan *file* audio berformat .mp3 (percakapan hasil terjemahan) yang berada pada *SD card*.

3.1.3.5 Analisis *Phrasebook Files*

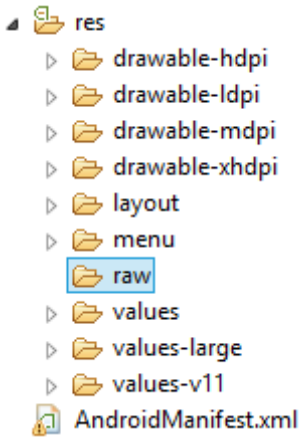
Phrasebook files merupakan *file*s yang berisi percakapan umum yang sering digunakan pada saat *traveling* yang terdiri dari tujuh kategori yaitu *greetings*, *general conversation*, *number*, *directions and places*, *transportation*, *eating out* dan *emergency*. Bahasa yang didukung pada fitur *Phrasebook* yaitu *Indonesian*, *English*, *Spain*, *France*, *China*, *Italy*, dan *Germany*.

Files pada fitur *Phrasebook* terdiri dari dua jenis yaitu *files* .xml yang berisi percakapan dan *files* suara dengan format .mp3 sebagai media pendukung pada fitur *Phrasebook.* *Files* .xml disimpan pada folder *res/values* dan *files* suara disimpan di folder *res/raw.*

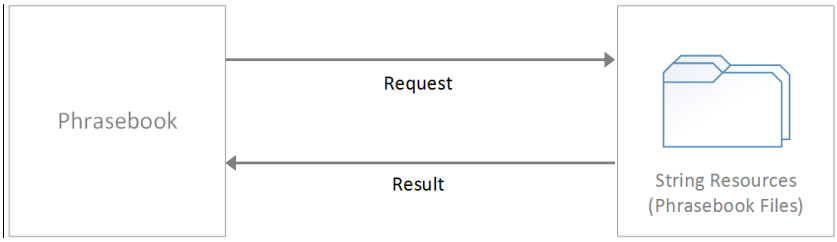


Gambar 3.19 Folder *Values* pada *Package Explorer*

Ketika membuat *project* android baru pada *eclipse* secara default pada folder (*res/values/*) terdapat tiga *file* .xml yaitu *dimens.xml*, *string.xml*, dan *style.xml*. *File* percakapan dapat digabung di salah satu *file* yang sudah ada atau dapat dibuat *file* .xml baru yang berisi percakapan. Untuk penyimpanan *files* suara dibuat sebuah folder pada *res/* dengan nama *raw*.



Gambar 3.20 Folder *Raw* pada *Package Explorer*



Gambar 3.21 Alur Penggunaan *Phrasebook Files*

Pada fitur *Phrasebook*, *files* percakapan sudah ditentukan dan di-*input*-kan sebelumnya, yang artinya pengguna tidak dapat menambahkan atau menghapus data percakapan yang sudah ada. Pengaksesan/penggunaan fitur ini dapat dilakukan secara *offline*.

Ketika pengguna menjalankan fungsi *Phrasebook*, sistem akan memeriksa *Language Setting* yang berada pada fitur *Phrasebook*, jika pengguna belum mengatur bahasa maka sistem akan menampilkan dialog *Language Setting*. Pengguna diminta untuk memilih bahasa asal dan bahasa tujuan, setelah pengguna memilih bahasa asal dan bahasa tujuan sistem akan menyimpan pengaturan bahasa dan akan menampilkan data *Phrasebook* yang dibagi menjadi beberapa kategori yaitu *greetings*, *general conversation*, *number*, *directions and places*, *transportation*, *eating out*, dan *emergency*.

Karena data pengaturan bahasa yang disimpan hanya berupa *string* dan sedikit maka pengaturan bahasa disimpan kedalam *SharedPreferences*. *SharedPreferences* memiliki fungsi yang serupa dengan *cookies* pada *web*

browser. Penyimpanan data pada *SharedPreferences* memiliki ukuran yang lebih kecil bila dibandingkan dengan penyimpanan kedalam *database* atau *file*.

1. *Request*

   Ketika pengguna memilih salah satu kategori dalam Phrasebook (contoh: Greetings) sistem akan memeriksa bahasa asal dan bahasa tujuan kemudian akan membaca kategori yang telah dipilih sebelumnya (Greetings)

2. *Result*

   Sistem akan merespon dengan menampilkan *phrase* (kata/percakapan asal) sesuai apa yang telah di-*request* sebelumnya dalam bentuk *String Array* dan menampilkannya dalam bentuk *ListView*. Ketika pengguna memilih salah satu *phrase* (kata/percakapan asal) sistem akan menampilkan *toast* berupa *string* dan menjalankan *file* audio berformat .mp3 dari kata/percakapan tujuan sesuai dengan urutan data *array*.

3.1.4 Analisis Kebutuhan Non Fungsional

Analisis dan kebutuhan non-fungsional meliputi analisis dan kebutuhan perangkat keras serta analisis dan kebutuhan perangkat lunak. Adapun kebutuhan non-fungsional untuk menjalankan aplikasi ini meliputi kebutuhan perangkat keras, kebutuhan perangkat lunak dan pengguna sistem yang akan menggunakan aplikasi. Analisis kebutuhan non-fungsional bertujuan agar aplikasi yang dibangun dapat digunakan sesuai dengan kebutuhan pengguna aplikasi dalam mencari informasi yang dibutuhkan.

3.1.4.1 Analisis Kebutuhan Perangkat Keras

Spesifikasi perangkat keras yang digunakan dalam pembangunan aplikasi ini adalah:

1. Perangkat keras PC:
   a) *Processor* Intel Core i3 1.33 GHz
   b) RAM yang digunakan 4 GB
   c) *Hardisk* 320 GB
   d) Koneksi internet.

2. Perangkat keras *Smartphone*:
    a) *Processor* 600 MHz
    b) RAM 512 Mb
    c) Media penyimpanan internal 512 MB
    d) Media penyimpanan eksternal 8 GB
    e) Koneksi internet.

Spesifikasi minimum perangkat keras *Smartphone* yang dibutuhkan untuk dapat menjalankan aplikasi ini adalah:

a) *Processor* 600 MHz
b) RAM 256 Mb
c) Media penyimpanan internal *free space* 15 MB
d) Media penyimpanan eksternal 1 GB
e) Koneksi internet.

3.1.4.2 Analisis Kebutuhan Perangkat Lunak

Spesifikasi perangkat lunak yang digunakan dalam pembangunan aplikasi ini adalah:

1. Perangkat lunak PC:
    a) Sistem Operasi yang digunakan adalah Windows 8.1 64-bit
    b) ADT (*Android Development Tools*) *Bundle* v22.6.2-1085508
    c) Java Development Kit v7.0.510.13
2. Perangkat lunak *Smartphone* yang digunakan adalah sistem operasi Android versi 2.3 *Gingerbread.*

Spesifikasi minimum perangkat lunak *Smartphone* yang dibutuhkan untuk dapat menjalankan aplikasi ini adalah sistem operasi Android versi 2.3 *Gingerbread.*

3.1.4.3 Analisis Pengguna

Analisis pengguna sistem dimaksudkan untuk mengetahui siapa saja pengguna yang terlibat dalam menjalankan sistem. Pengguna yaitu publik khususnya *traveler* yang dapat mengoperasikan *smartphone* khususnya *platform* android.

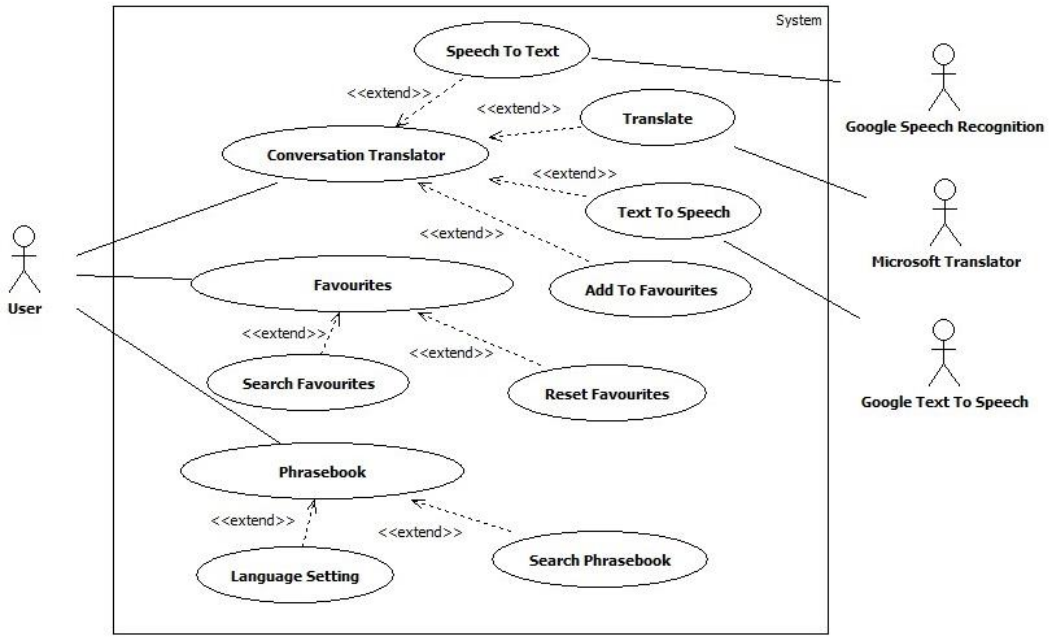
Aplikasi ini tidak hanya dapat digunakan oleh *traveler* melainkan dapat digunakan oleh semua kalangan sebagai media pembelajaran bahasa asing. Untuk dapat menjalankan seluruh fitur yang terdapat pada aplikasi, pengguna diharuskan memiliki koneksi internet.

3.1.5 Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional menggambarkan proses kegiatan yang akan diterapkan dalam sebuah sistem dan menjelaskan kebutuhan yang diperlukan sistem agar sistem dapat berjalan dengan baik. Analisis yang dilakukan dimodelkan dengan menggunakan UML (*Unified Modeling Language*). Tahap-tahap pemodelan dalam analisis tersebut antara lain *Use Case Diagram*, *Activity Diagram*, *Sequence Diagram*, dan *Class Diagram*.

3.1.5.1 *Use Case Diagram*

Use Case Diagram mendeskripsikan sebuah interaksi antar satu atau lebih aktor dengan sistem yang akan dibuat. Secara kasar *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. *Use Case* terdiri dari tiga bagian yaitu identifikasi aktor, identifikasi *use case* dan skenario *use case*. Analisis *Use Case Diagram* pada sistem yang akan bibangun dapat dilihat pada Gambar 3.22.



Gambar 3.22 *Use Case Diagram*

1. Identifikasi Aktor

   Aktor merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem yang akan dibangun. Berikut adalah aktor-aktor yang terdapat pada sistem yang akan dibangun.

Tabel 3.2 Identifikasi Aktor

No	Aktor	Deskripsi
A-01	*User*	Merupakan aktor dari aplikasi yang dibangun
A-02	*Google Speech Recognition*	Merupakan aktor yang berperan mengkonversi suara menjadi teks
A-03	*Microsoft Translator*	Merupakan aktor yang berperan menerjemahkan teks percakapan ke bahasa yang diinginkan
A-04	*Google Text To Speech*	Merupakan aktor yang berperan mengkonversi teks menjadi suara (ucapan)

2. Identifikasi *Use Case*

   Use case merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor. Berikut adalah *use case* yang terdapat pada sistem yang akan dibangun.

Tabel 3.3 Identifikasi *Use Case*

No	*Use Case*	Deskripsi
UC-01	*Conversation Translator*	Fungsionalitas untuk menampilkan terjemahan percakapan
UC-02	*Speech To Text*	Fungsionalitas untuk menkonversi percakapan dalam bentuk suara (ucapan) ke dalam bentuk teks
UC-03	*Translate*	Fungsionalitas untuk menerjemahkan teks percakapan ke bahasa yang diinginkan
UC-04	*Text To Speech*	Fungsionalitas untuk menkonversi teks percakapan ke dalam bentuk suara (ucapan)
UC-05	*Add To Favourites*	Fungsionalitas untuk menyimpan percakapan asal dan percakapan yang telah diterjemahkan kedalam *Favourites*
UC-06	*Favourites*	Fungsionalitas untuk menampilkan percakapan asal dan percakapan hasil terjemahan
UC-07	*Search Favourites*	Fungsionalitas untuk mencari percakapan asal dan percakapan hasil terjemahan yang berada di dalam *Favourites*
UC-08	*Reset Favourites*	Fungsionalitas untuk menghapus seluruh percakapan asal dan percakapan hasil terjemahan (reset) yang berada pada *Favourites*
UC-09	*Phrasebook*	Fungsionalitas untuk menampilkan percakapan (kata/kalimat) umum yang sering digunakan pada saat *traveling*
UC-10	*Search Phrasebook*	Fungsionalitas untuk mencari percakapan (kata/kalimat) umum yang sering digunakan pada saat *traveling* yang berada di dalam *Phrasebook*
UC-11	*Language Setting*	Fungsionalitas untuk mengatur bahasa asal dan bahasa tujuan yang berada di dalam *Phrasebook*

3. Skenario *Use Case*

   Skenario *Use Case* mendeskripsikan urutan langkah-langkah dalam proses bisnis, baik yang dilakukan aktor terhadap sistem maupun yang dilakukan oleh sistem terhadap aktor.

a) Skenario *Use Case Conversation Translator*

Tabel 3.4 Skenario *Use Case Conversation Translator*

Identifikasi	
Nama *Use Case*	*Conversation Translator*
Aktor	*User*
Tujuan	Menampilkan *Conversation Translator*
Kondisi Awal	Menampilkan halaman utama
Skenario Utama	
Aksi Aktor	**Reaksi Sistem**
1. Menekan *button* "*Conversation Translator*"	
	2. Mengecek koneksi internet
	3. Menampilkan halaman *Conversation Translator*
Skenario Alternatif	
Aksi Aktor	**Reaksi Sistem**
	1. Mengecek koneksi internet
	2. Menampilkan dialog "*You must be connected to the Internet to use this feature*."
3. Menekan *button* "*Ok*"	
	4. Menampilkan halaman utama
Kondisi Akhir	Menampilkan halaman *Conversation Translator*

b) Skenario *Use Case Speech To Text*

Tabel 3.5 Skenario *Use Case Speech To Text*

Identifikasi	
Nama *Use Case*	*Speech To Text*
Aktor	*Google Speech Recognition*
Tujuan	Mengkonversi suara percakapan ke dalam bentuk teks
Kondisi Awal	Menampilkan halaman *Conversation Translator*
Skenario Utama	
Aksi Aktor	**Reaksi Sistem**
1. Menekan *button* "*Voice*"	
	2. Mengecek koneksi internet
	3. Menyambungkan *device* ke *Google Speech Recognition*

	4. Menampilkan notifikasi *Speak now*
5. Berbicara	
	6. Menampilkan notifikasi pemrosesan
	7. Menampilkan hasil konversi percakapan dalam bentuk teks
Skenario Alternatif	
Aksi Aktor	**Reaksi Sistem**
	1. Mengecek koneksi internet
	2. Menampilkan dialog "*You must be connected to the Internet to use this function.*"
3. Menekan *button* "*Ok*"	
	4. Menampilkan halaman *Conversation Translator*
Kondisi Akhir	Suara percakapan telah dikonversi kedalam bentuk teks

c) Skenario *Use Case Translate*

Tabel 3.6 Skenario *Use Case Translate*

Identifikasi	
Nama *Use Case*	*Translate*
Aktor	*Microsoft Translator*
Tujuan	Menerjemahkan percakapan
Kondisi Awal	Pesan percakapan sudah terdapat pada *form input*
Skenario Utama	
Aksi Aktor	**Reaksi Sistem**
1. Menekan *button* "*Translate*"	
	2. Mengecek koneksi internet
	3. Menampilkan dialog "*Loading...*"
	4. Mengirim teks percakapan ke *Microsoft Translator* untuk diterjemahkan
	5. Menampilkan terjemahan percakapan
Skenario Alternatif	
Aksi Aktor	**Reaksi Sistem**

	1. Mengecek koneksi internet
	2. Menampilkan dialog "*You must be connected to the Internet to use this function.*"
3. Menekan *button* "*Ok*"	
	4. Menampilkan halaman *Conversation Translator*
Kondisi Akhir	Pesan percakapan telah diterjemahkan

d) Skenario *Use Case Text To Speech*

Tabel 3.7 Skenario *Use Case Text To Speech*

Identifikasi	
Nama *Use Case*	*Text To Speech*
Aktor	*Google Text To Speech*
Tujuan	Mengkonversi pesan percakapan yang sudah diterjemahkan ke dalam bentuk suara (ucapan)
Kondisi Awal	Pesan percakapan sudah terdapat pada *form output*
Skenario Utama	
Aksi Aktor	**Reaksi Sistem**
1. Menekan *button* "*Speak*"	
	2. Mengecek koneksi internet
	3. Menampilkan dialog "*Loading...*"
	4. Mengirim teks percakapan ke *Google Text To Speech* untuk dikonversi menjadi suara (ucapan)
	5. Menampilkan/menjalankan audio hasil terjemahan
	6. Membuat *file* suara dalam format .mp3 dan disimpan kedalam *SD card*
Skenario Alternatif	
Aksi Aktor	**Reaksi Sistem**
	1. Mengecek koneksi internet
	2. Menampilkan dialog "*You must be connected to the Internet to use this function.*"
3. Menekan *button* "*Ok*"	
	4. Menampilkan halaman *Conversation Translator*

Kondisi Akhir	Teks percakapan telah dikonversi kedalam bentuk suara (ucapan) dan disimpan kedalam *SD card* dengan format .mp3

e) Skenario *Use Case Add To Favourites*

Tabel 3.8 Skenario *Use Case Add To Favourites*

Identifikasi	
Nama *Use Case*	*Add To Favourites*
Aktor	*User*
Tujuan	Menambahkan percakapan ke *Favourites*
Kondisi Awal	Menampilkan halaman *Conversation Translator*
Skenario Utama	
Aksi Aktor	**Reaksi Sistem**
1. Menekan *button* "*Add to Favourites*"	
	2. Mengecek percakapan yang akan ditambahkan
	3. Menyimpan percakapan ke *database*
	4. Menampilkan *toast* "*New sentence inserted.*"
Skenario Alternatif	
Aksi Aktor	**Reaksi Sistem**
	1. Mengecek percakapan yang akan ditambahkan
	2. Tidak menyimpan percakapan ke *database*
	3. Menampilkan dialog "*You need to translate first before can save it.*"
4. Menekan *button* "*Ok*"	
	5. Menampilkan halaman *Conversation Translator*
Kondisi Akhir	Menyimpan percakapan kedalam *database*

f) Skenario *Use Case Favourites*

Tabel 3.9 Skenario *Use Case Favourites*

Identifikasi	
Nama *Use Case*	*Favourites*
Aktor	*User*

Tujuan	Menampilkan *Favourites*
Kondisi Awal	Menampilkan halaman utama
Skenario Utama	
Aksi Aktor	**Reaksi Sistem**
1. Menekan *button* "*Favourites*"	
	2. Cek *database Favourites*
	3. Menampilkan halaman dan data *Favourites*
Skenario Alternatif	
Aksi Aktor	**Reaksi Sistem**
	1. Cek *database Favourites*
	2. Menampilkan halaman *Favourites*
	3. Menampilkan *toast* "*You don't have any favorites.*"
Kondisi Akhir	Menampilkan halaman *Favourites*

g) Skenario *Use Case Search Favourites*

Tabel 3.10 Skenario *Use Case Search Favourites*

Identifikasi	
Nama *Use Case*	*Search Favourites*
Aktor	*User*
Tujuan	Mencari percakapan pada fitur *Favourites*
Kondisi Awal	Menampilkan halaman *Favourites Search*
Skenario Utama	
Aksi Aktor	**Reaksi Sistem**
1. Memasukan *keyword* pencarian	
2. Menekan *button*/*symbol* "*Search*"	
	3. Menampilkan dialog "*Loading...*"
	4. Mencari data *Favourites* dalam *database*
	5. Menampilkan data *Favourites* yang dicari
6. Menekan hasil pencarian	
	7. Menampilkan detail hasil pencarian

Skenario Alternatif	
Aksi Aktor	**Reaksi Sistem**
	1. Mencari data *Favourites* dalam *database*
	2. Tidak menampilkan data *Favourites* yang dicari
Kondisi Akhir	Menampilkan atau tidak menampilkan data *Favourites* yang dicari

h) Skenario *Use Case Reset Favourites*

Tabel 3.11 Skenario *Use Case Reset Favourites*

Identifikasi	
Nama *Use Case*	*Reset Favourites*
Aktor	*User*
Tujuan	Menghapus seluruh data *Favourites*
Kondisi Awal	Menampilkan halaman *Favourites*
Skenario Utama	
Aksi Aktor	**Reaksi Sistem**
1. Menekan *button* "*Reset*"	
	2. Menampilkan dialog "*Are you sure to reset the favourites?*"
3. Menekan *button* "*Ok*"	
	4. Menghapus seluruh data *Favourites* dari *database*
	5. Menampilkan halaman *Favourites* dan menampilkan *toast* "*Your favourites has been reset.*"
Skenario Alternatif	
Aksi Aktor	**Reaksi Sistem**
	1. Menampilkan dialog "*Are you sure to reset the favourites?*"
2. Menekan *button* "*Cancel*"	
	3. Menampilkan halaman *Favourites*
Kondisi Akhir	Seluruh data *Favourites* telah terhapus

i) Skenario *Use Case Phrasebook*

Tabel 3.12 Skenario *Use Case Phrasebook*

Identifikasi	
Nama *Use Case*	*Phrasebook*
Aktor	*User*
Tujuan	Menampilkan *Phrasebook*
Kondisi Awal	Menampilkan halaman utama
Skenario Utama	
Aksi Aktor	**Reaksi Sistem**
1. Menekan *button* "*Phrasebook*"	
	2. Mengecek *default language*
	3. Menampilkan *Language Setting*
4. Memilih bahasa asal dan bahasa tujuan	
5. Menekan *button* "*Save*"	
	6. Menyimpan pengaturan bahasa
	7. Menampilkan halaman *Phrasebook*
Skenario Alternatif	
Aksi Aktor	**Reaksi Sistem**
	1. Mengecek *default language*
	2. Menampilkan halaman *Phrasebook*
Kondisi Akhir	Menampilkan halaman *Phrasebook*

j) Skenario *Use Case Search Phrasebook*

Tabel 3.13 Skenario *Use Case Search Phrasebook*

Identifikasi	
Nama *Use Case*	*Search Phrasebook*
Aktor	*User*
Tujuan	Mencari percakapan pada fitur *Phrasebook*
Kondisi Awal	Menampilkan halaman *Phrasebook Search*
Skenario Utama	
Aksi Aktor	**Reaksi Sistem**
1. Memasukan *keyword* pencarian	
2. Menekan *button*/*symbol*	

"*Search*"	
	3. Menampilkan dialog "*Loading...*"
	4. Mencari data *Phrasebook* dalam *Phrasebook Files*
	5. Menampilkan data *Phrasebook* yang dicari
6. Menekan hasil pencarian	
	7. Menampilkan detail hasil pencarian
Skenario Alternatif	
Aksi Aktor	**Reaksi Sistem**
	1. Mencari data *Phrasebook* dalam *Phrasebook Files*
	2. Tidak menampilkan data *Phrasebook* yang dicari
Kondisi Akhir	Menampilkan atau tidak menampilkan data *Phrasebook* yang dicari

k) Skenario *Use Case Language Setting*

Tabel 3.14 Skenario *Use Case Language Setting*

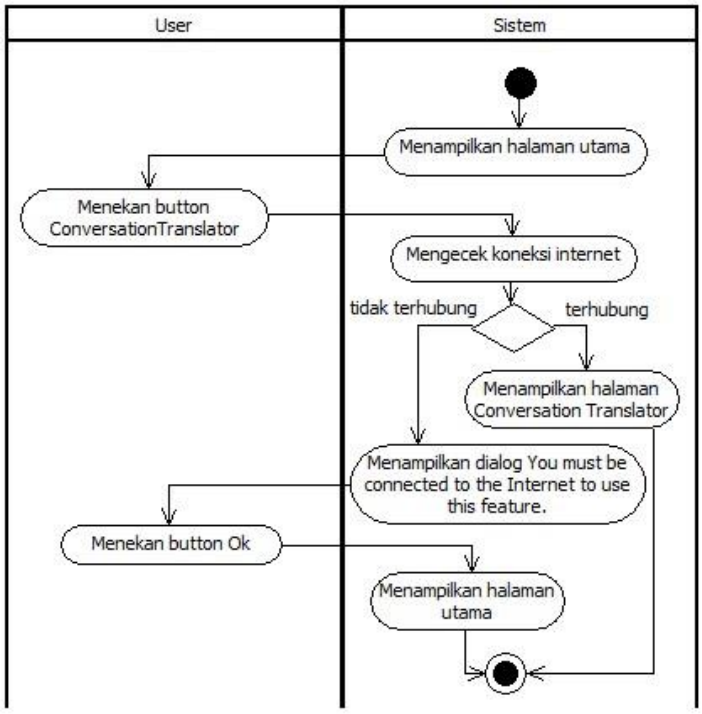
Identifikasi	
Nama *Use Case*	*Language Setting*
Aktor	*User*
Tujuan	Mengatur bahasa pada fitur *Phrasebook*
Kondisi Awal	Menampilkan halaman *Phrasebook*
Skenario Utama	
Aksi Aktor	**Reaksi Sistem**
1. Menekan *button*/*symbol* "*Setting*"	
	2. Menampilkan *Language Setting*
3. Memilih bahasa asal dan bahasa tujuan	
4. Menekan *button* "*Save*"	
	5. Menyimpan pengaturan bahasa
	6. Menampilkan halaman *Phrasebook*
Skenario Alternatif	
Aksi Aktor	**Reaksi Sistem**

Kondisi Akhir	Pengaturan bahasa telah disimpan dan menampilkan halaman *Phrasebook*

3.1.5.2 *Activity Diagram*

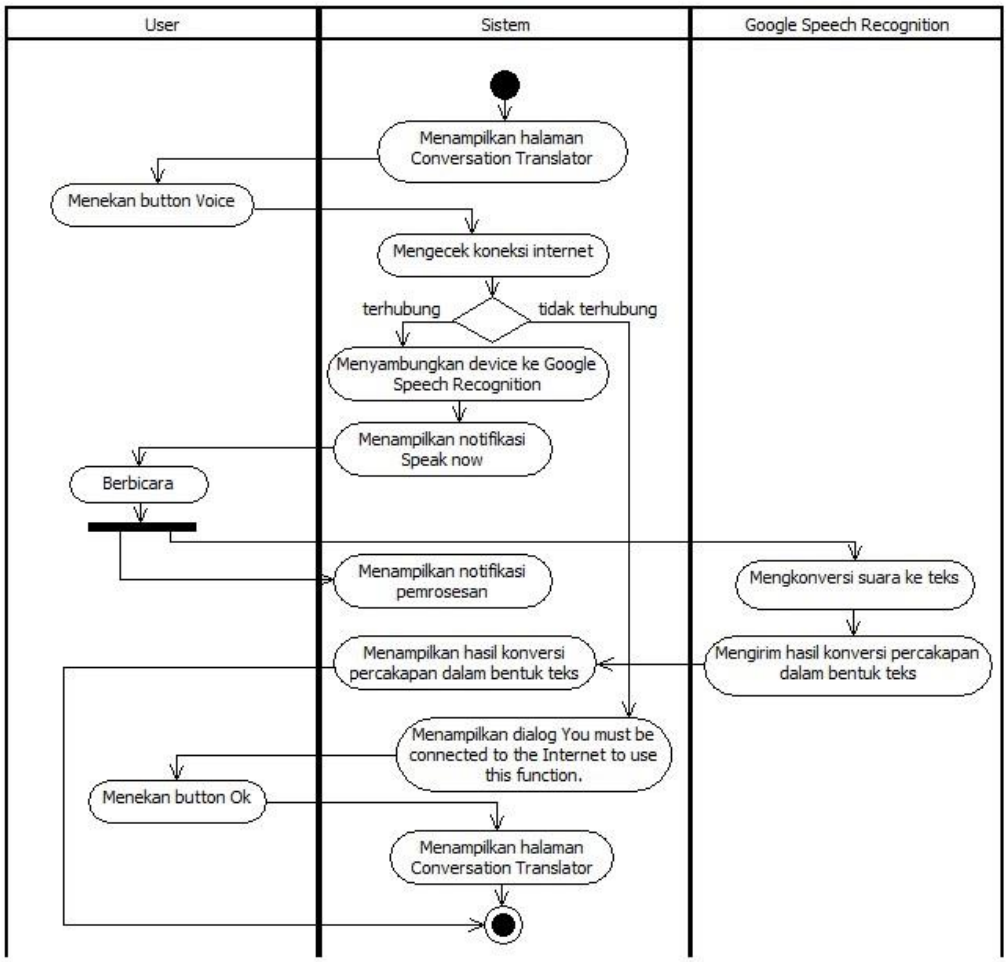
Activity Diagram, menggambarkan aliran kerja (*workflow*) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Penggambaran *activity diagram* memiliki kemiripan dengan *flowchart diagram*.

1. *Activity Diagram Conversation Translator*



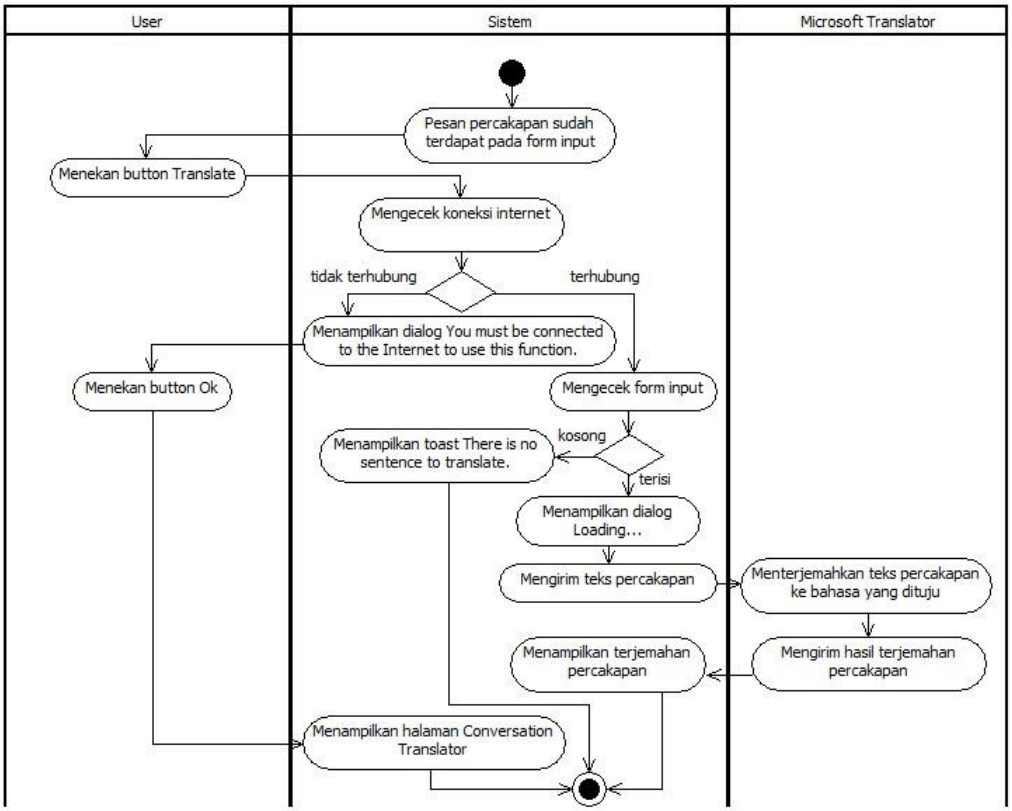
Gambar 3.23 *Activity Diagram Conversation Translator*

2. *Activity Diagram Speech To Text*



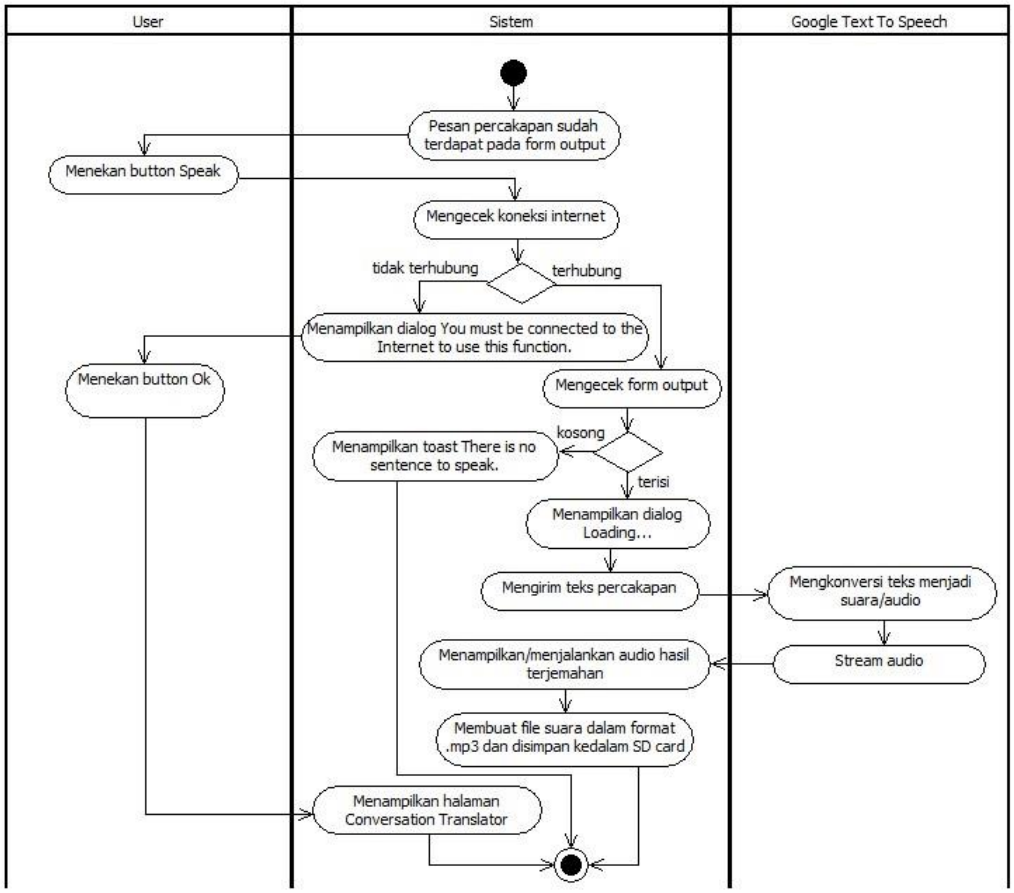
Gambar 3.24 *Activity Diagram Speech To Text*

3. *Activity Diagram Translate*



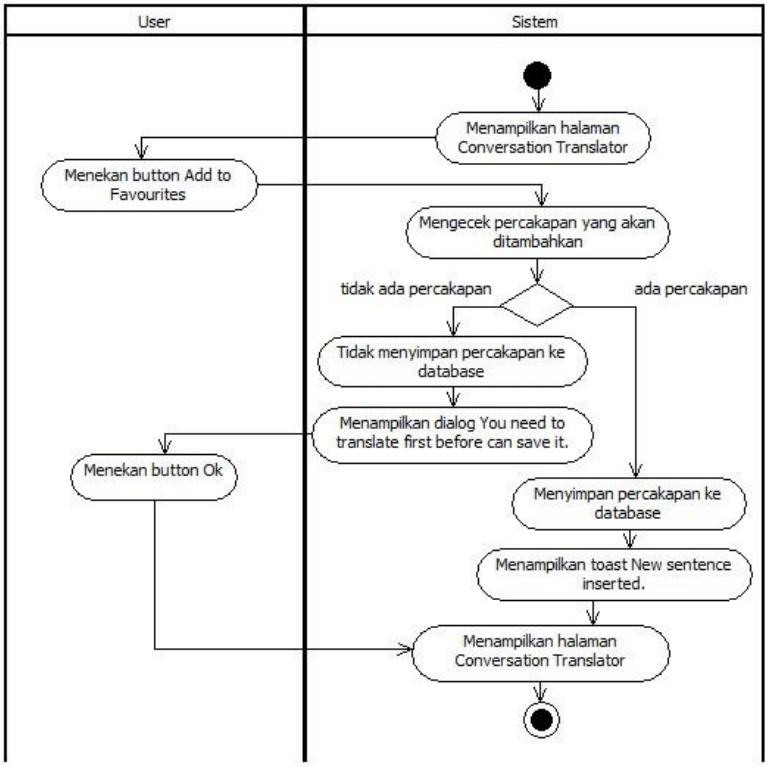
Gambar 3.25 *Activity Diagram Translate*

4. *Activity Diagram Text To Speech*



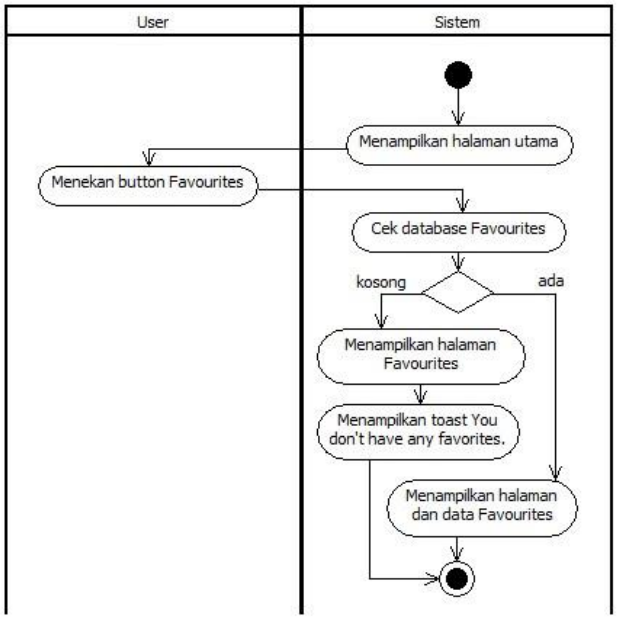
Gambar 3.26 *Activity Diagram Text To Speech*

5. *Activity Diagram Add To Favourites*



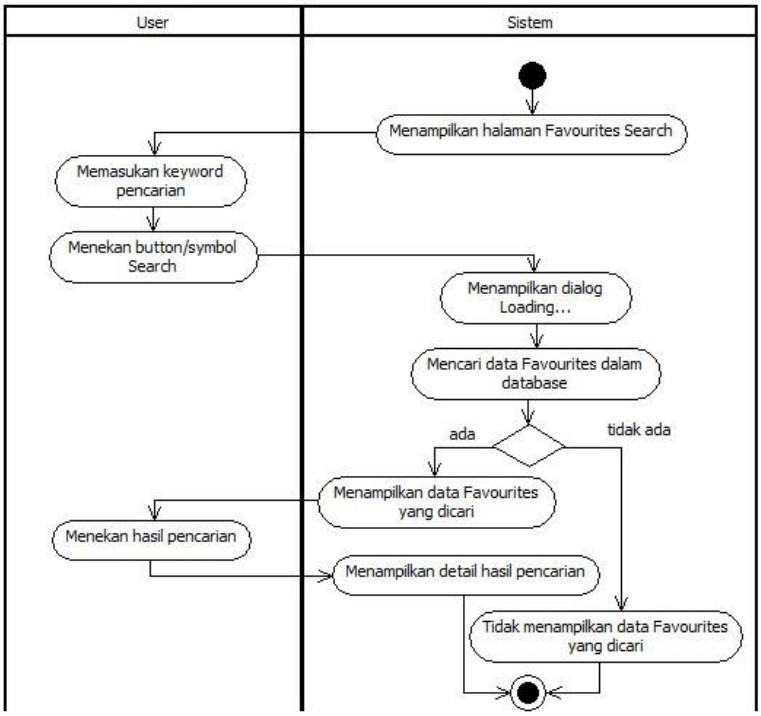
Gambar 3.27 *Activity Diagram Add To Favourites*

6. *Activity Diagram Favourites*



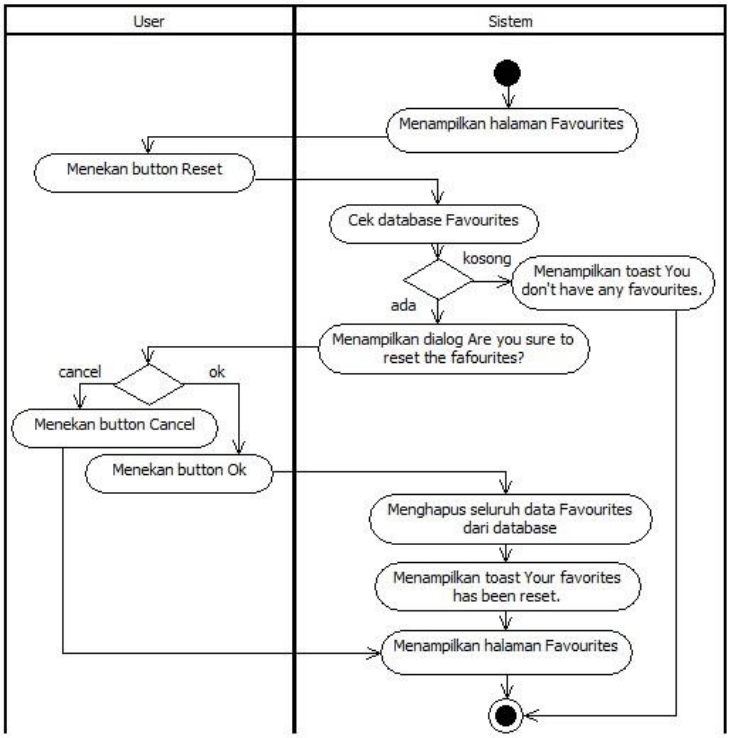
Gambar 3.28 *Activity Diagram Favourites*

7. *Activity Diagram Search Favourites*



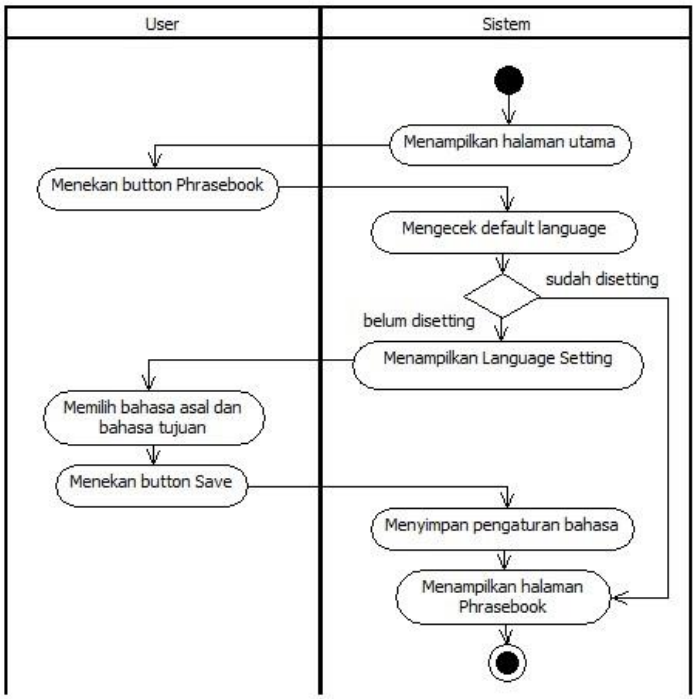
Gambar 3.29 *Activity Diagram Search Favourites*

8. *Activity Diagram Reset Favourites*



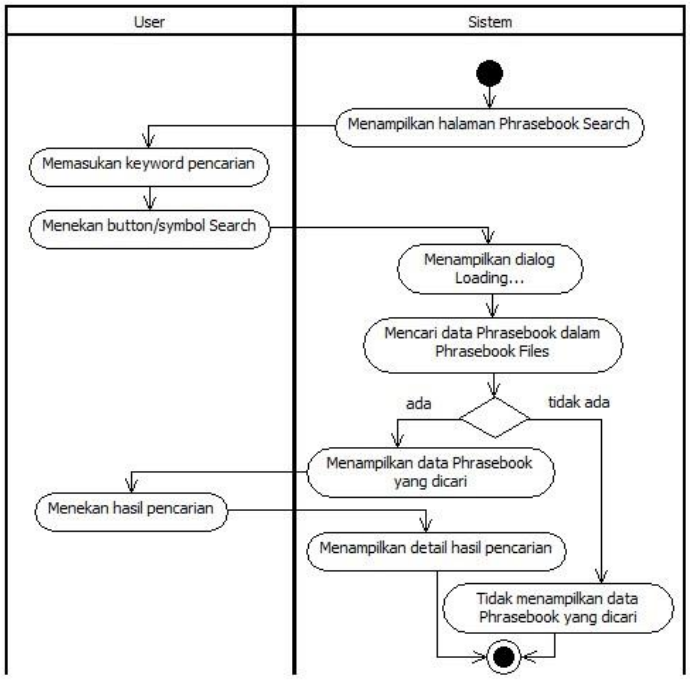
Gambar 3.30 *Activity Diagram Reset Favourites*

9. *Activity Diagram Phrasebook*



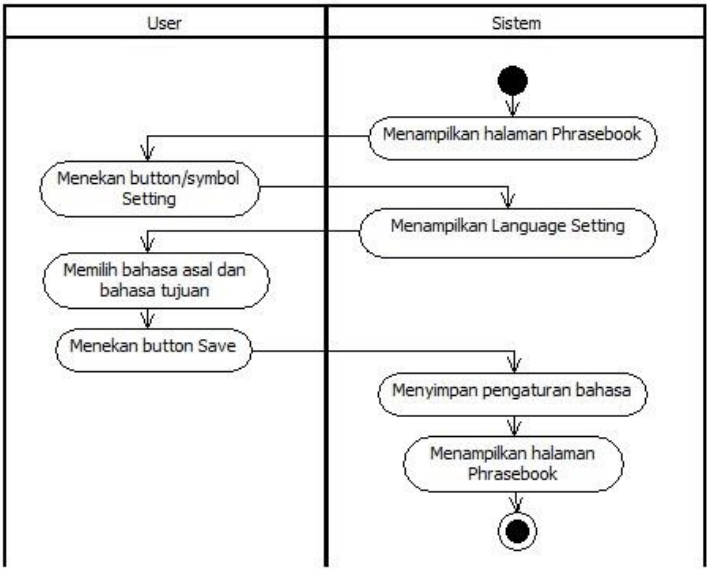
Gambar 3.31 *Activity Diagram Phrasebook*

10. *Activity Diagram Search Phrasebook*



Gambar 3.32 *Activity Diagram Search Phrasebook*

11. *Activity Diagram Language Setting*

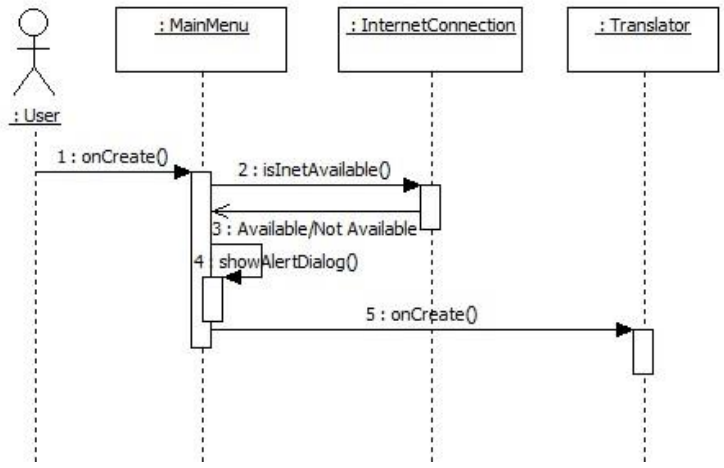


Gambar 3.33 *Activity Diagram Language Setting*

3.1.5.3 *Sequence Diagram*

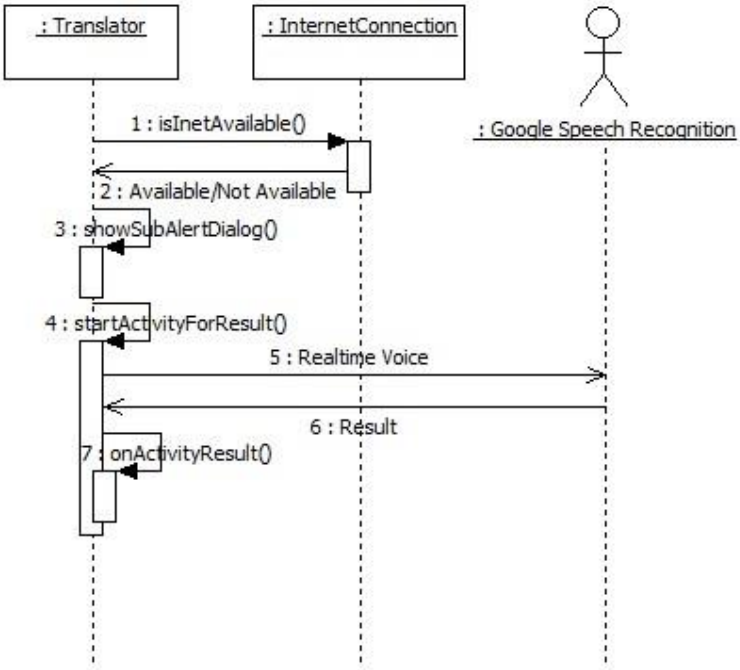
Sequence Diagram, menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek.

1. *Sequence Diagram Conversation Translator*



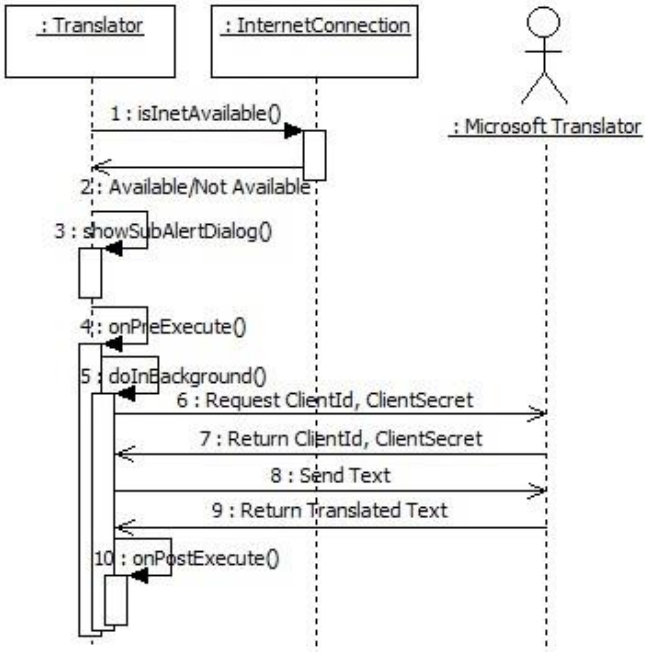
Gambar 3.34 *Sequence Diagram Conversation Translator*

2. *Sequence Diagram Speech To Text*



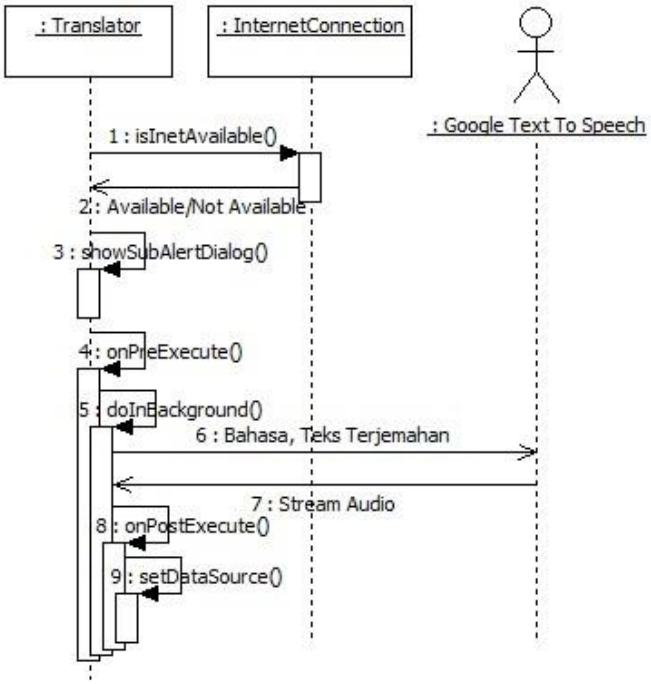
Gambar 3.35 *Sequence Diagram Speech To Text*

3. *Sequence Diagram Translate*



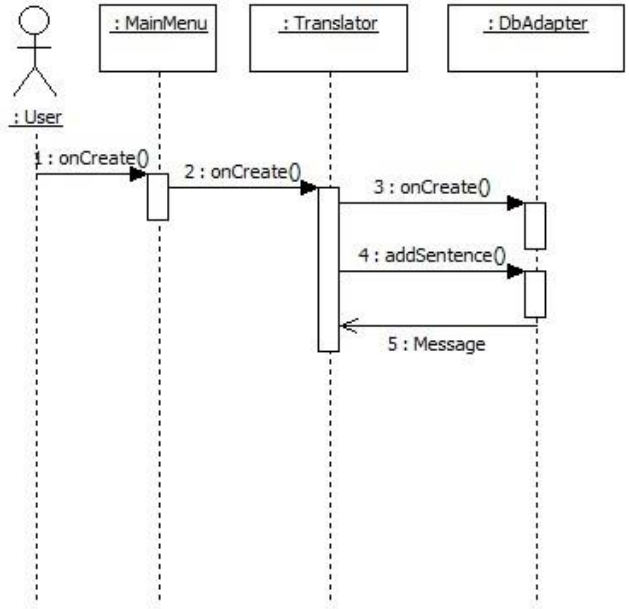
Gambar 3.36 *Sequence Diagram Translate*

4. *Sequence Diagram Text To Speech*



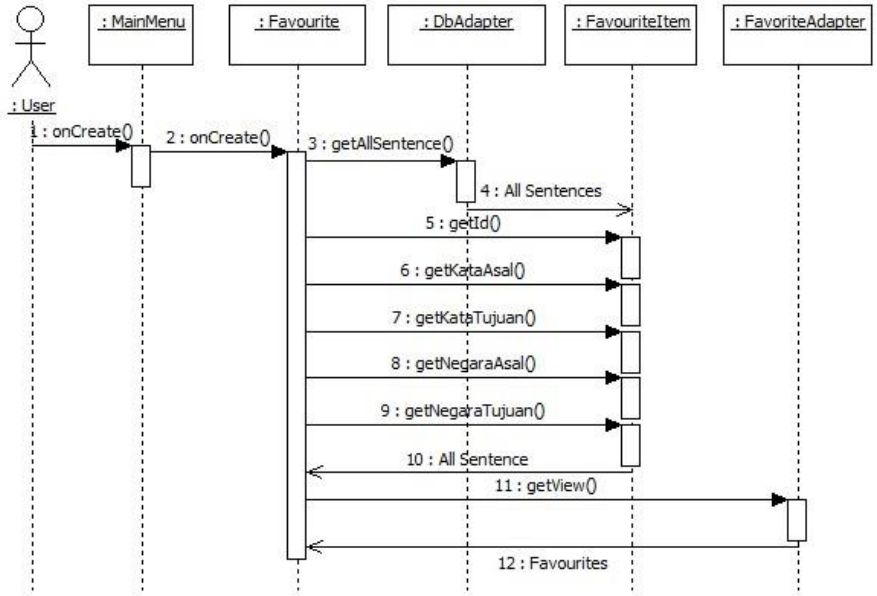
Gambar 3.37 *Sequence Diagram Text To Speech*

5. *Sequence Diagram Add To Favourites*



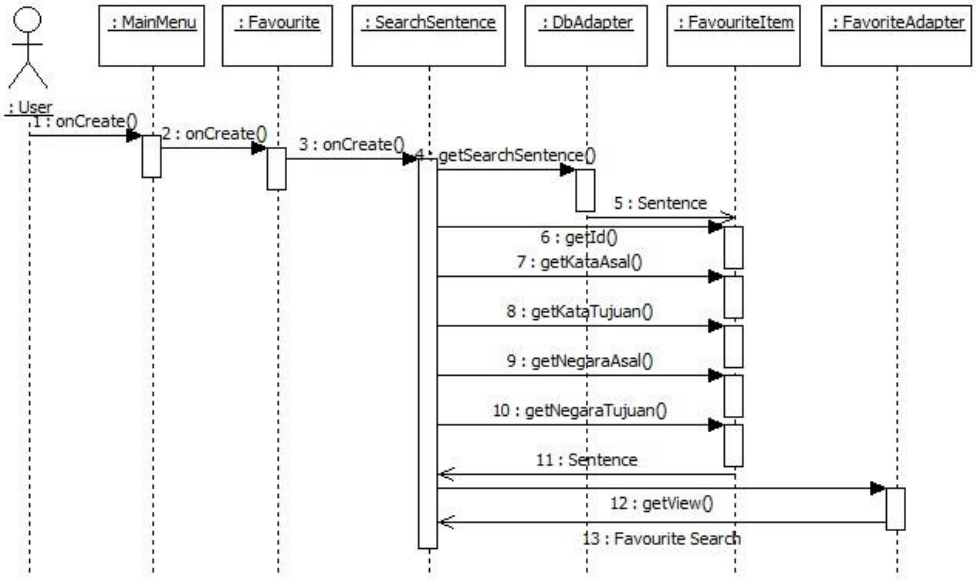
Gambar 3.38 *Sequence Diagram Add To Favourites*

6. *Sequence Diagram Favourites*



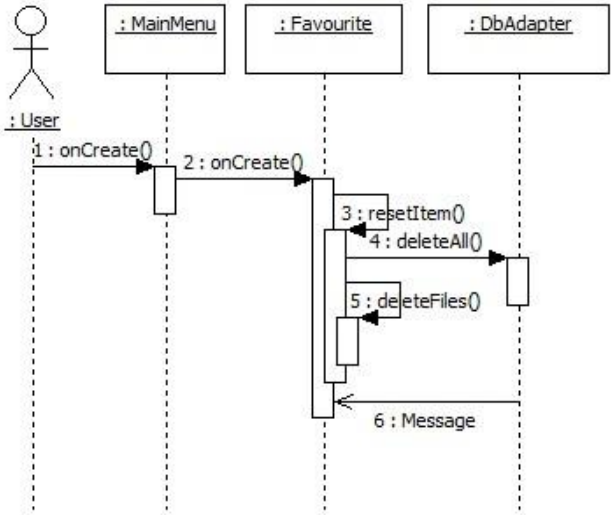
Gambar 3.39 *Sequence Diagram Favourites*

7. *Sequence Diagram Search Favourites*



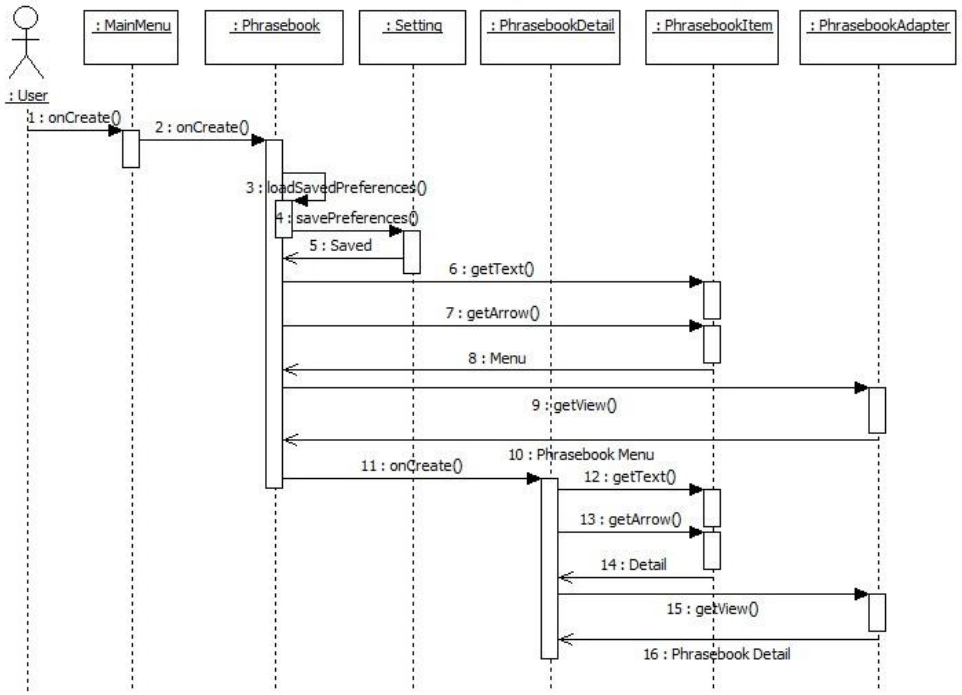
Gambar 3.40 *Sequence Diagram Search Favourites*

8. *Sequence Diagram Reset Favourites*



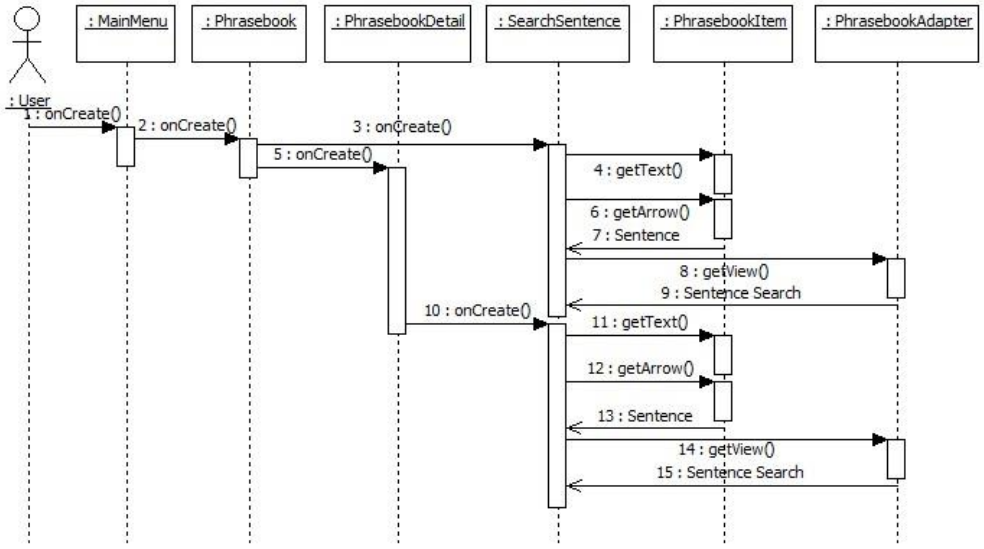
Gambar 3.41 *Sequence Diagram Reset Favourites*

9. *Sequence Diagram Phrasebook*



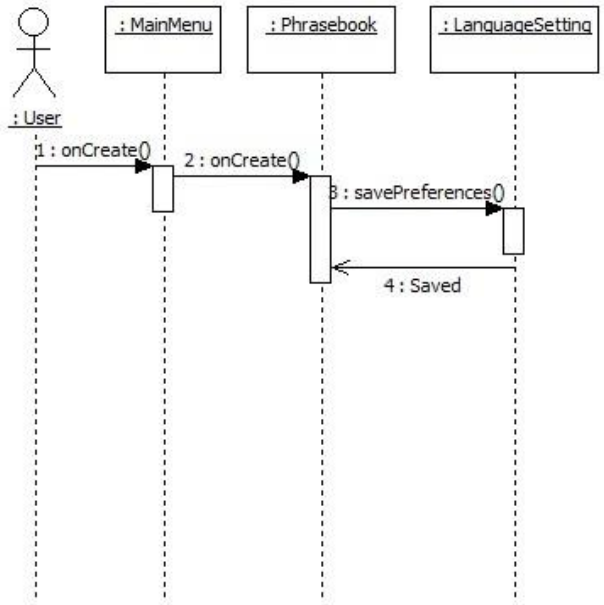
Gambar 3.42 *Sequence Diagram Phrasebook*

10. *Sequence Diagram Search Phrasebook*



Gambar 3.43 *Sequence Diagram Search Phrasebook*

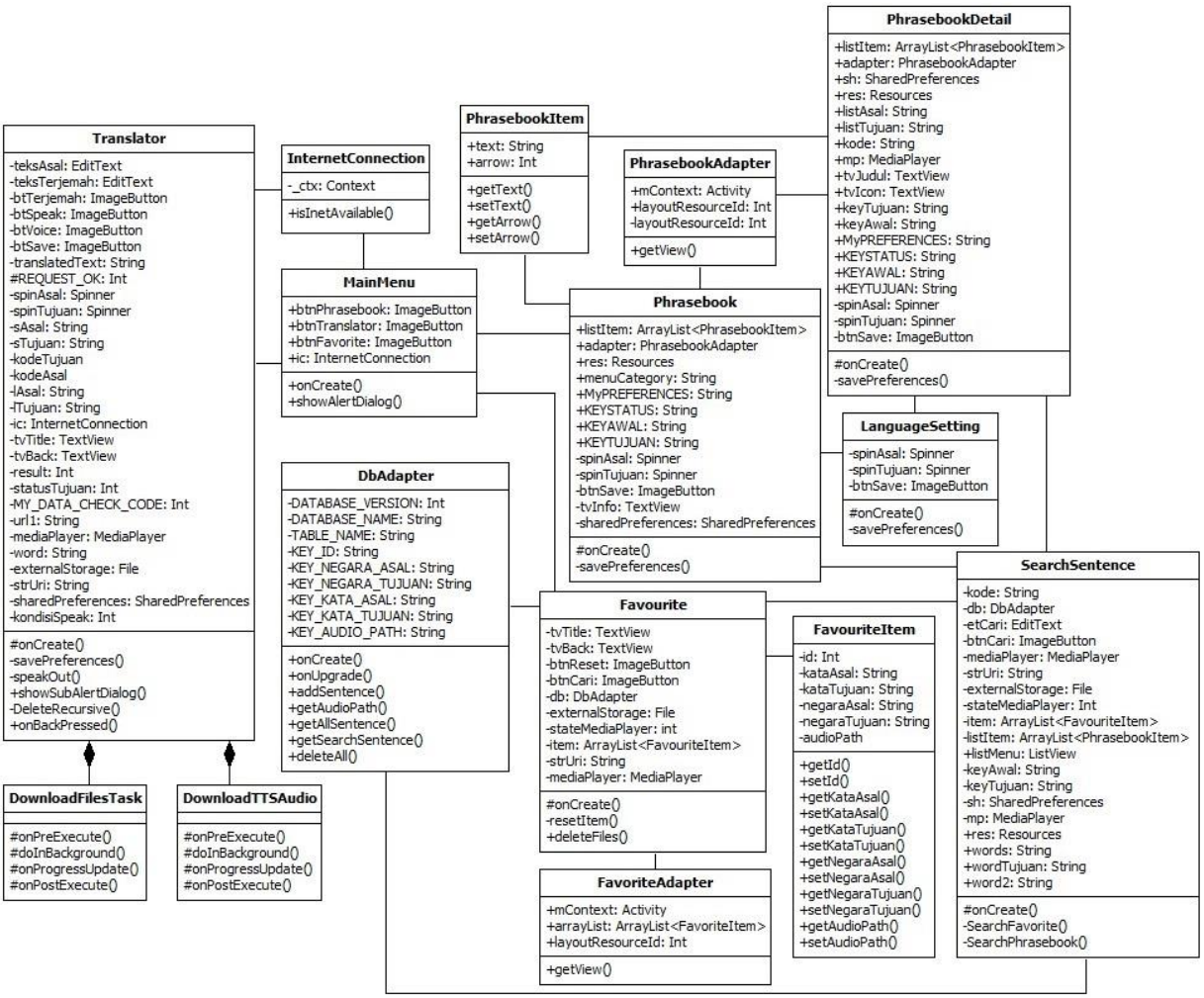
11. *Sequence Diagram Language Setting*



Gambar 3.44 *Sequence Diagram Language Setting*

3.1.5.4 *Class Diagram*

Class diagram menggambarkan struktur dan hubungan antar objek-objek yang ada pada sistem. Struktur ini meliputi atribut-atribut dan metode-metode yang ada pada masing-masing *class*. Hubungannya dari masing-masing *class* yang ada dapat dilihat pada Gambar 3.45.



Gambar 3.45 *Class Diagram*

Tabel 3.15 Keterangan *Class Diagram*

No	Nama *Class*	Jenis *Class*	Keterangan
1	MainMenu	*Boundary*	Merupakan *class* yang digunakan untuk menampilkan halaman utama
2	InternetConnection	*Control*	Merupakan *class* yang digunakan untuk mengecek ketersediaan koneksi internet
3	LanguageSetting	*Control*	Merupakan *class* yang berisi *method*

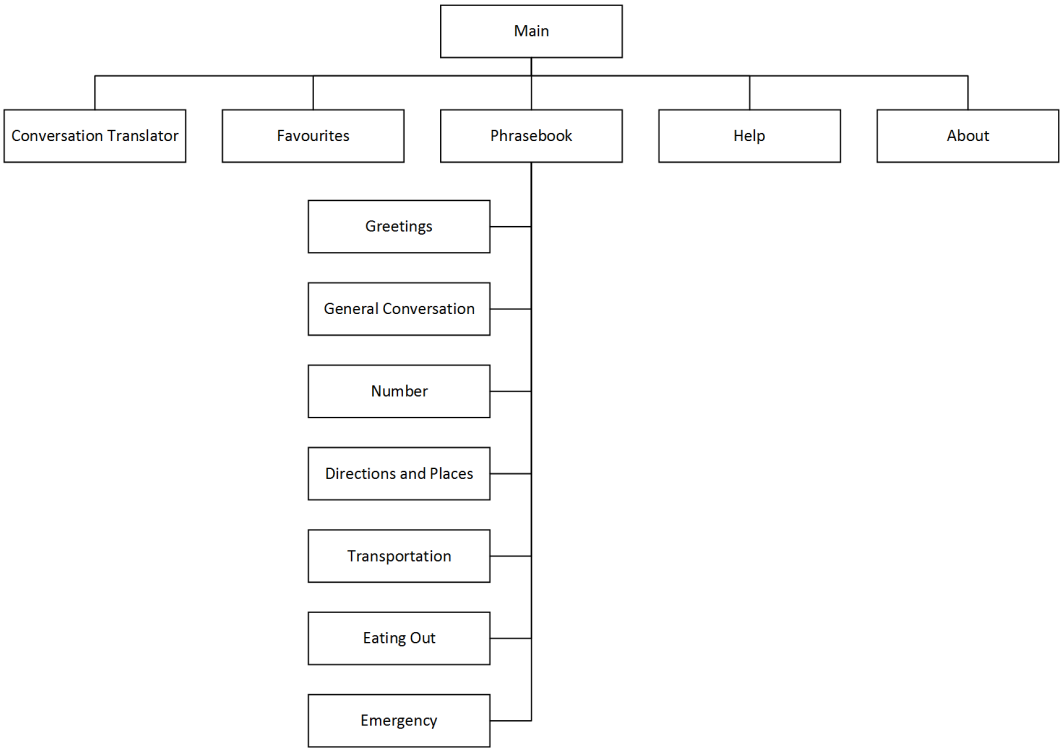
			untuk mengontrol pengaturan bahasa pada fitur *Phrasebook*
4	DbAdapter	*Control*	Merupakan *class* yang digunakan untuk meng-*handle* masalah SQLite beserta fungsi-fungsi yang dibutuhkan seperti struktur dan skema *database*
5	Translator	*Control*	Merupakan *class* yang berisi *method-method* untuk menangani proses penterjemahan, konversi suara ke teks atau sebaliknya, serta penyimpanan percakapan
6	Favourite	*Control*	Merupakan *class* untuk menampilkan daftar *Favourites*
7	FavouriteItem	*Entity*	Merupakan *class* data yang digunakan untuk memproses segala pengaksesan ke *database Favourites*
8	FavouriteAdapter	*Control*	Merupakan *class* yang digunakan untuk mengkontrol tampilan (*widget*) agar dapat tampil kedalam *view* android
9	Phrasebook	*Control*	Merupakan *class* untuk menampilkan menu kategori *Phrasebook*
10	PhrasebookItem	*Entity*	Merupakan *class* data yang digunakan untuk memproses segala pengaksesan ke *File Phasebook*
11	PhrasebookDetail	*Control*	Merupakan *class* untuk menampilkan detail *Phrasebook*
12	PhrasebookAdapter	*Control*	Merupakan *class* yang digunakan untuk mengkontrol tampilan (*widget*) agar dapat tampil kedalam *view* android
13	SearchSentence	*Control*	Merupakan *class* yang berisi *method-method* untuk menangani proses pencarian data *Favourites* dan pencarian data *Phrasebook*

3.2 Perancangan Sistem

Perancangan sistem bertujuan untuk menspesifikasikan aspek-aspek teknik yang menjadi solusi dalam perencanaan. Pada tahap ini perancangan akan didefinisikan secara detail untuk mengatasi masalah-masalah yang lebih teknis, berkaitan dengan kegiatan implementasi seperti perancangan menu, perancangan antarmuka, dan jaringan semantik.

3.2.1 Perancangan Struktur Menu

Perancangan struktur menu merupakan gambaran alur pemakaian aplikasi. Perancangan struktur menu dari aplikasi ini dapat dilihat pada Gambar 3.46.

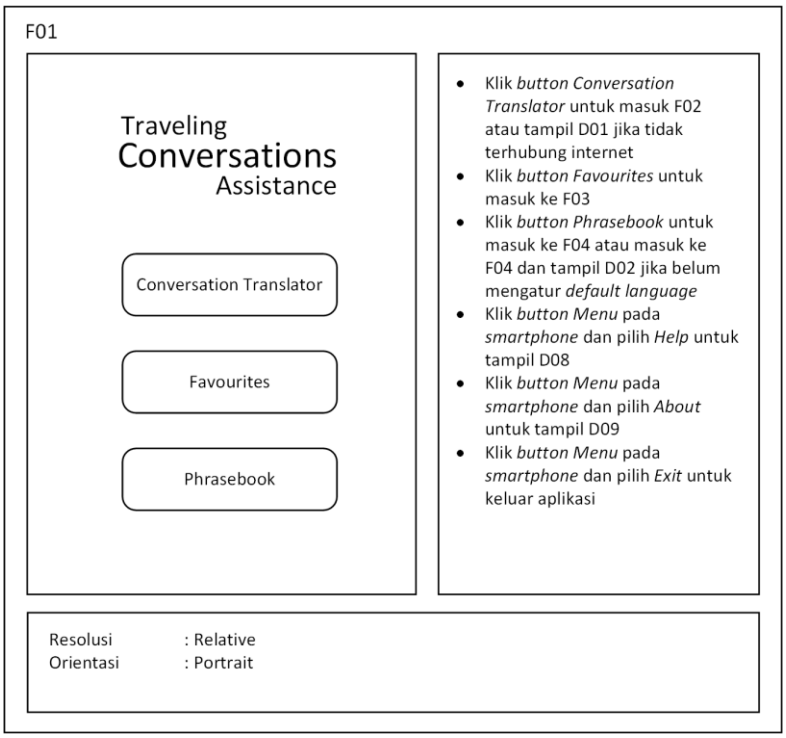


Gambar 3.46 Struktur Menu

3.2.2 Perancangan Antarmuka

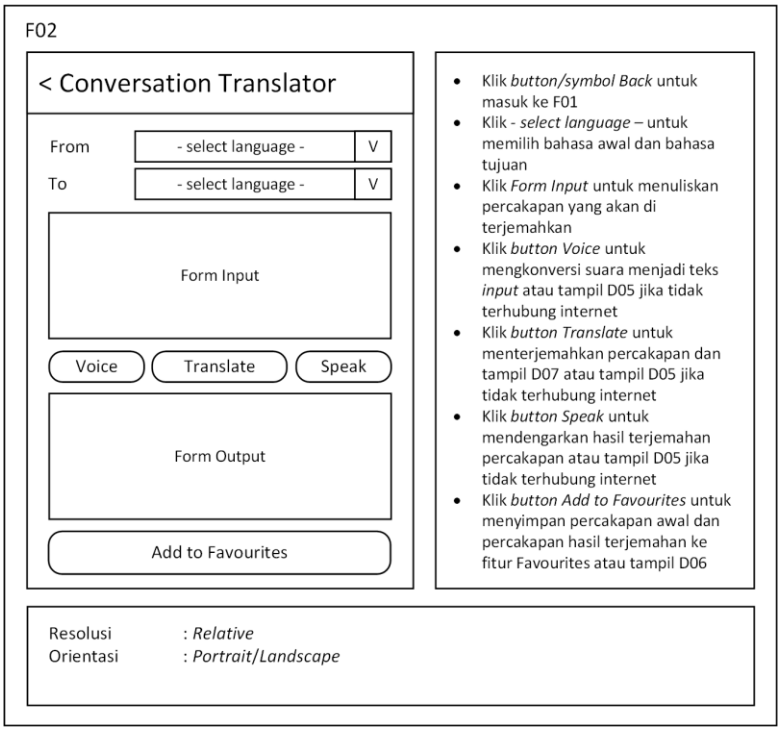
Perancangan antarmuka bertujuan untuk memberikan gambaran tentang aplikasi yang akan dibangun, sehingga mempermudah dalam mengimplementasikan aplikasi serta memudahkan dalam pembuatan aplikasi.

1. Perancangan Antarmuka *Main Menu*



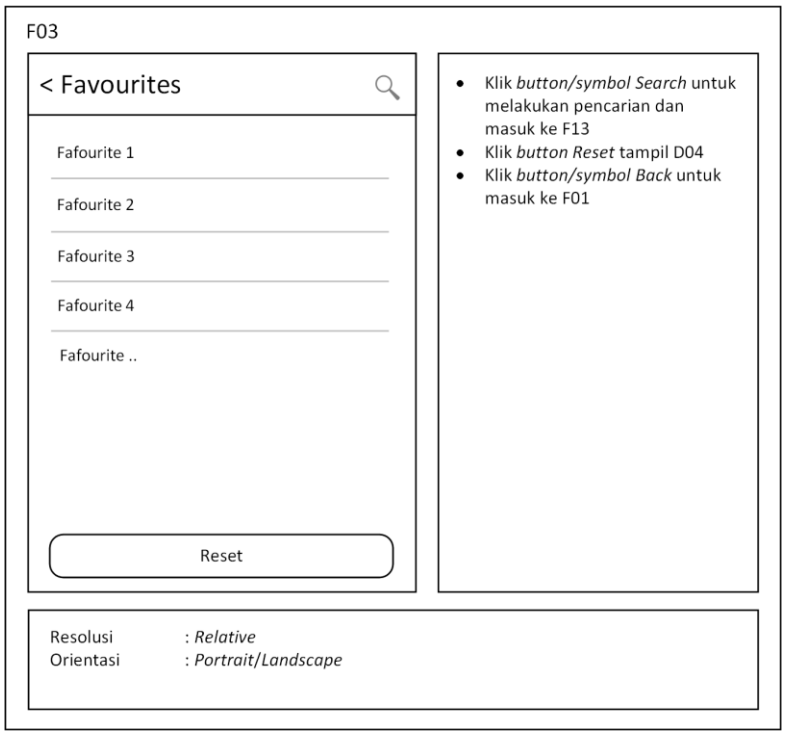
Gambar 3.47 Perancangan Antarmuka *Main Menu*

2. Perancangan Antarmuka *Conversation Translator*



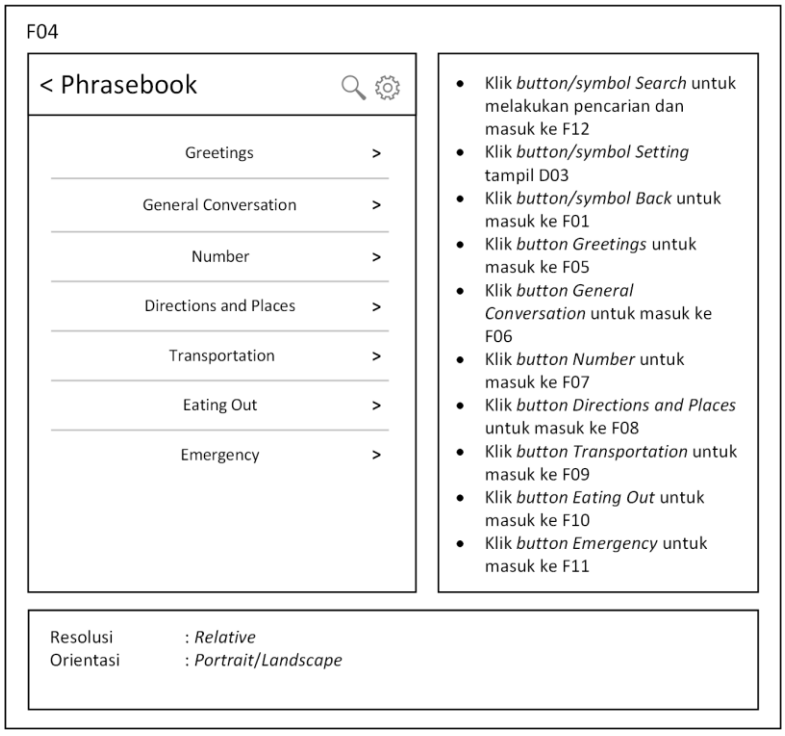
Gambar 3.48 Perancangan Antarmuka *Conversation Translator*

3. Perancangan Antarmuka *Favourites*



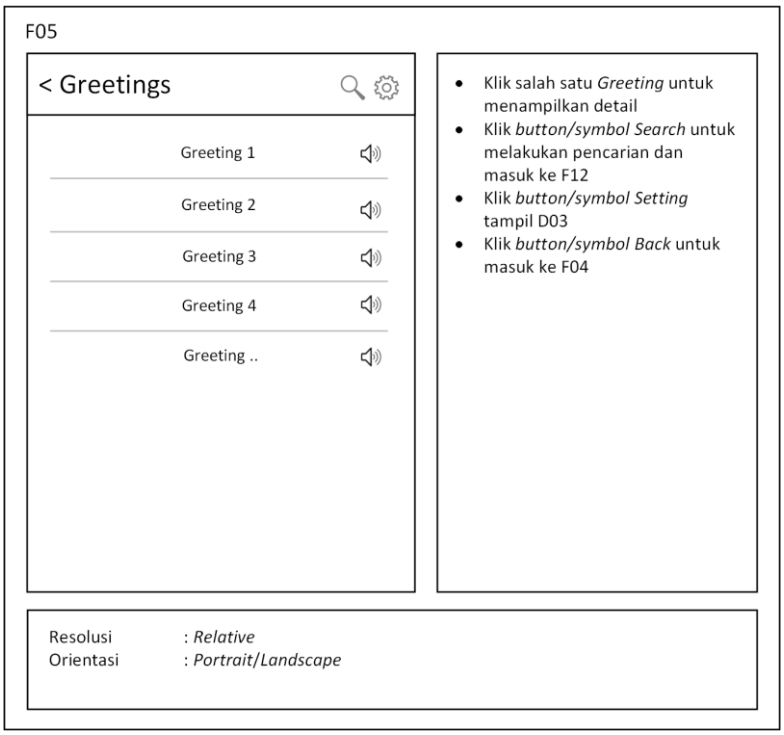
Gambar 3.49 Perancangan Antarmuka *Favourites*

4. Perancangan Antarmuka *Phrasebook*



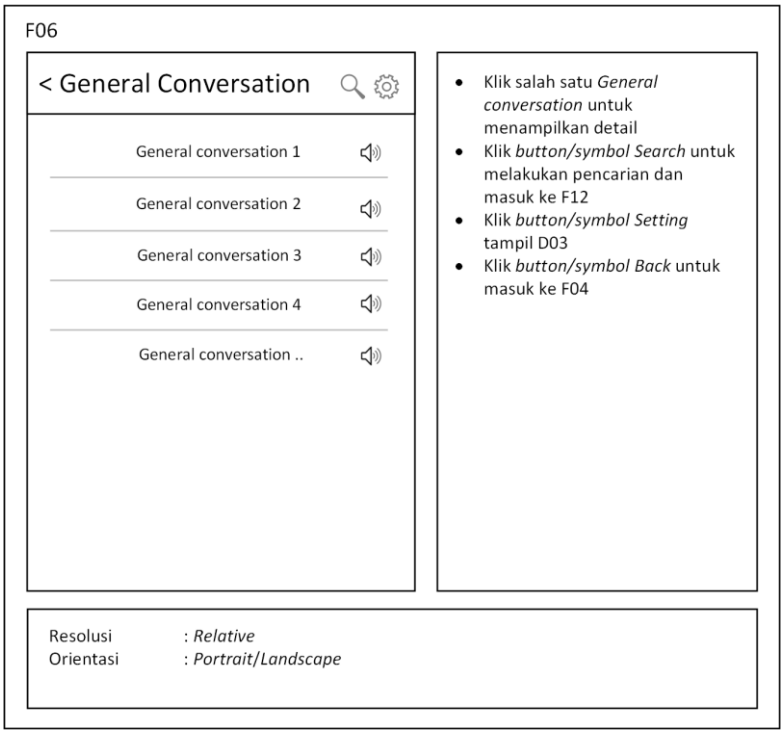
Gambar 3.50 Perancangan Antarmuka *Phrasebook*

5. Perancangan Antarmuka *Greetings*



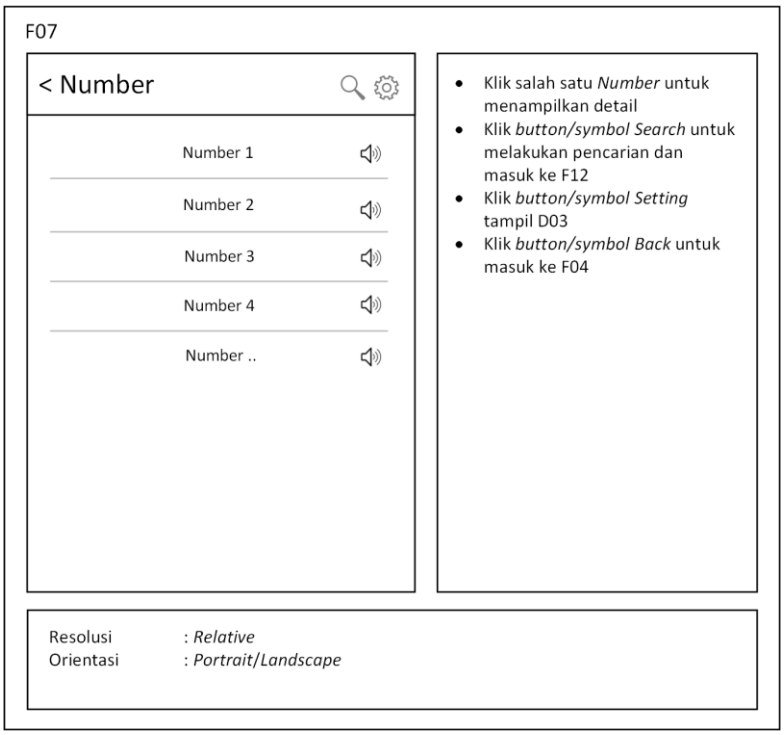
Gambar 3.51 Perancangan Antarmuka *Greetings*

6. Perancangan Antarmuka *General Conversation*



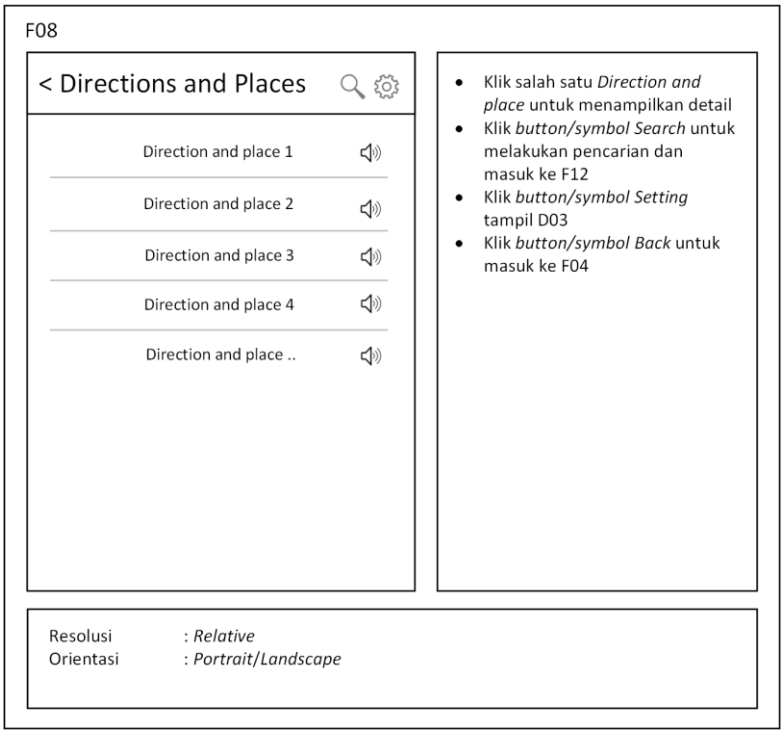
Gambar 3.52 Perancangan Antarmuka *General Conversation*

7. Perancangan Antarmuka *Number*



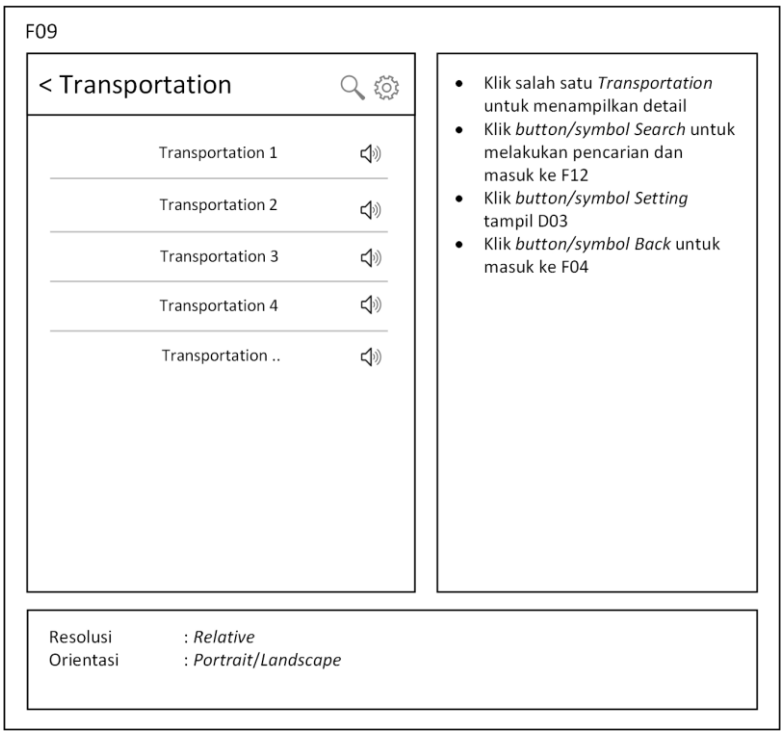
Gambar 3.53 Perancangan Antarmuka *Number*

8. Perancangan Antarmuka *Directions and Places*



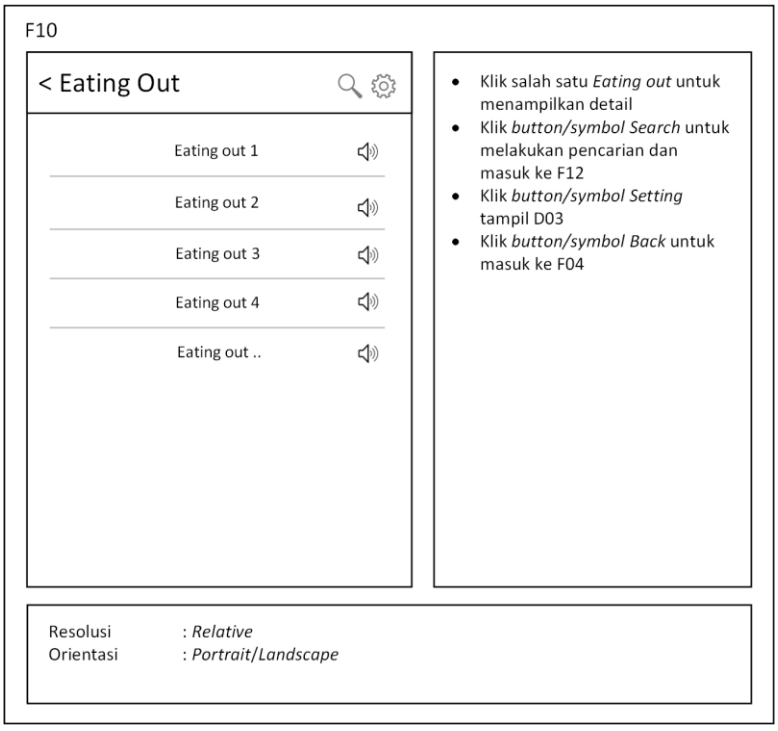
Gambar 3.54 Perancangan Antarmuka *Directions and Places*

9. Perancangan Antarmuka *Transportation*



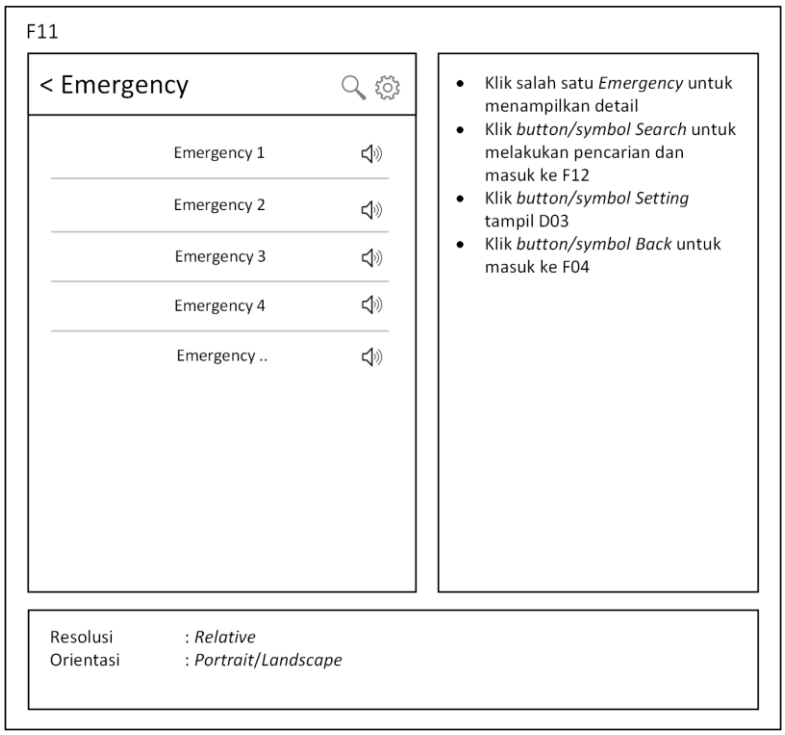
Gambar 3.55 Perancangan Antarmuka *Transportation*

10. Perancangan Antarmuka *Eating Out*



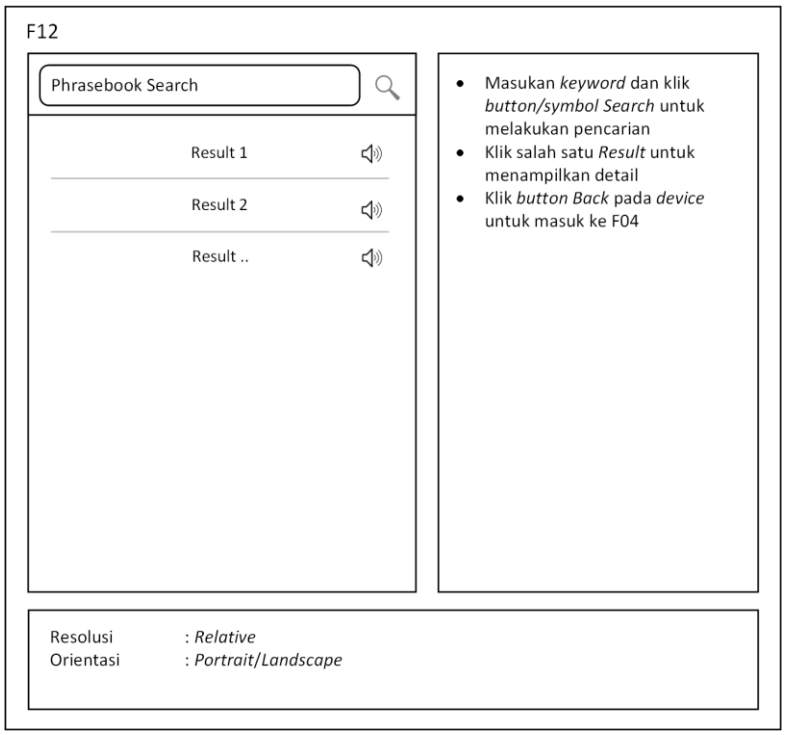
Gambar 3.56 Perancangan Antarmuka *Eating Out*

11. Perancangan Antarmuka *Emergency*



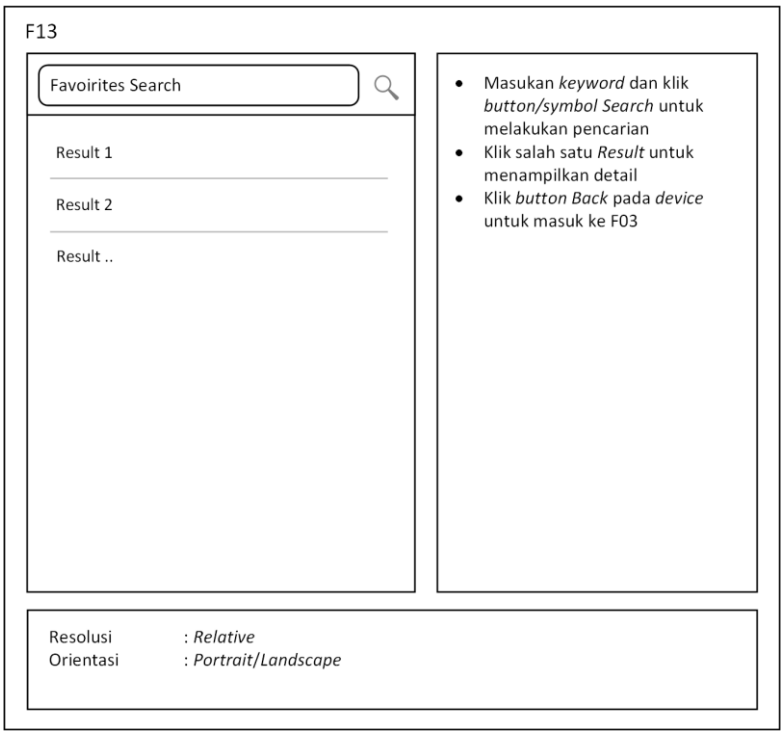
Gambar 3.57 Perancangan Antarmuka *Emergency*

12. Perancangan Antarmuka *Phrasebook Search*



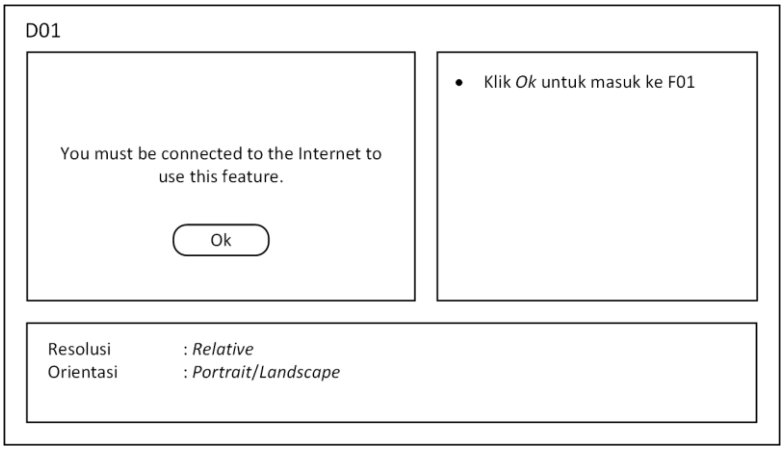
Gambar 3.58 Perancangan Antarmuka *Phrasebook Search*

13. Perancangan Antarmuka *Favourites Search*



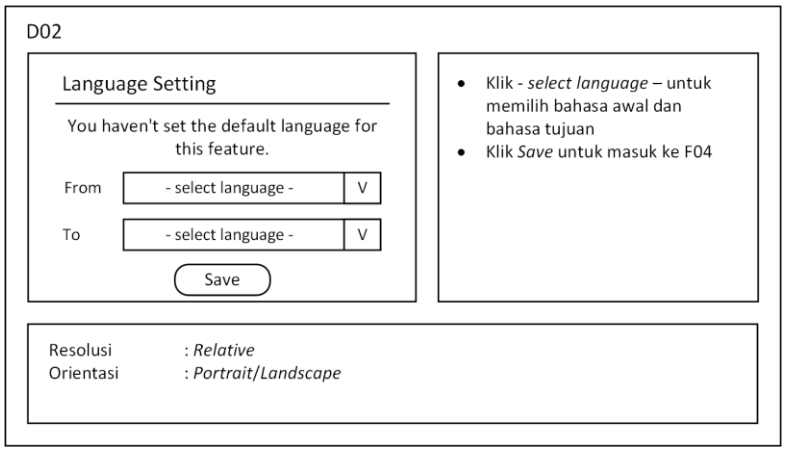
Gambar 3.59 Perancangan Antarmuka *Favourites Search*

14. Perancangan Antarmuka Dialog Koneksi Internet (*Feature*)



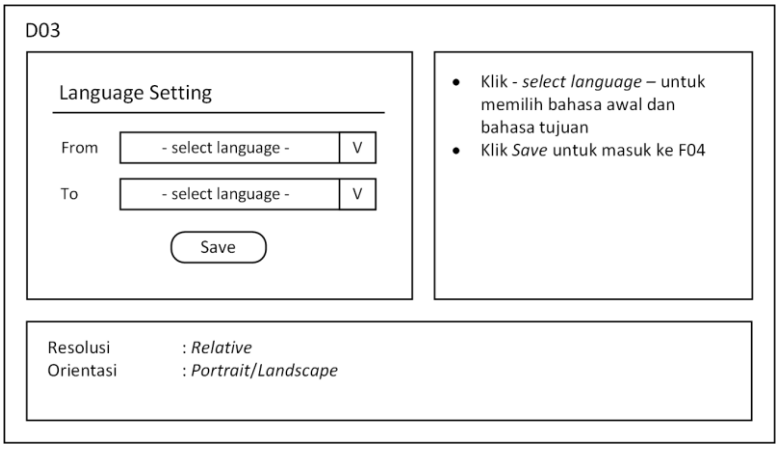
Gambar 3.60 Perancangan Antarmuka Dialog Koneksi Internet (*Feature*)

15. Perancangan Antarmuka Dialog *Default Language*



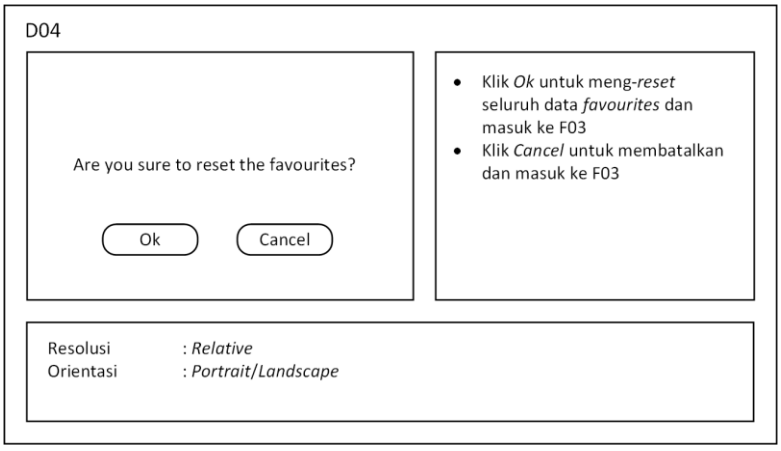
Gambar 3.61 Perancangan Antarmuka Dialog *Default Language*

16. Perancangan Antarmuka Dialog *Language Setting*



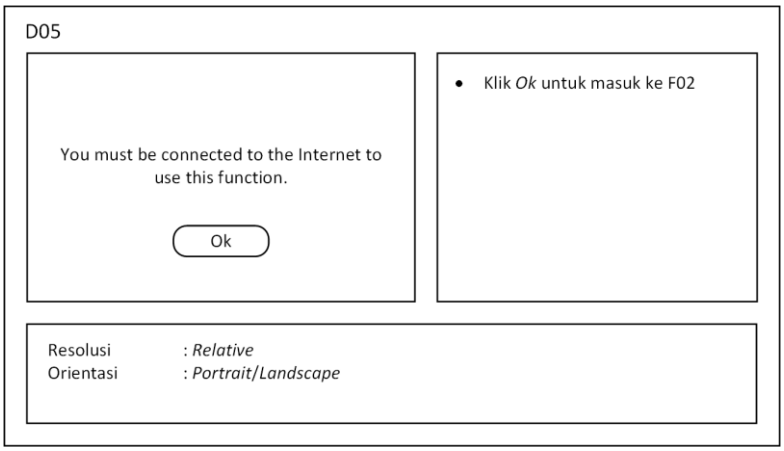
Gambar 3.62 Perancangan Antarmuka Dialog *Language Setting*

17. Perancangan Antarmuka Dialog Penghapusan



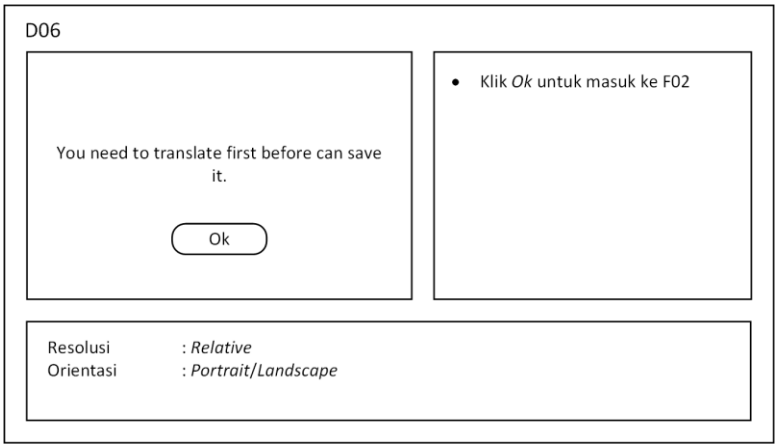
Gambar 3.63 Perancangan Antarmuka Dialog Penghapusan

18. Perancangan Antarmuka Dialog Koneksi Internet (*Function*)



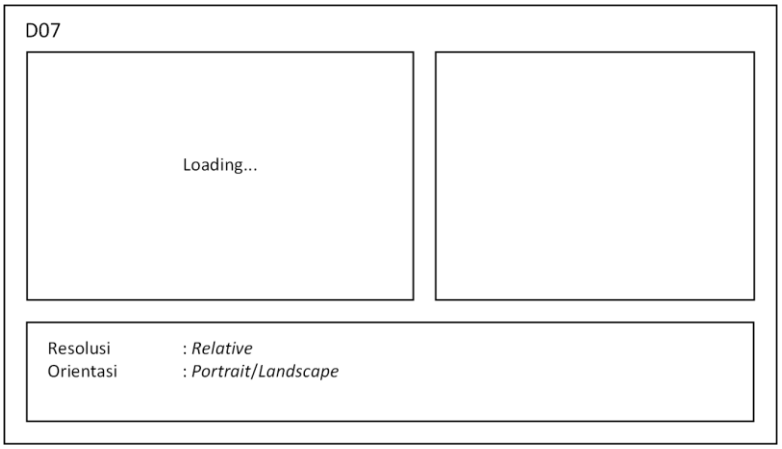
Gambar 3.64 Perancangan Antarmuka Dialog Penghapusan (*Function*)

19. Perancangan Antarmuka Dialog *Add to Favourites*



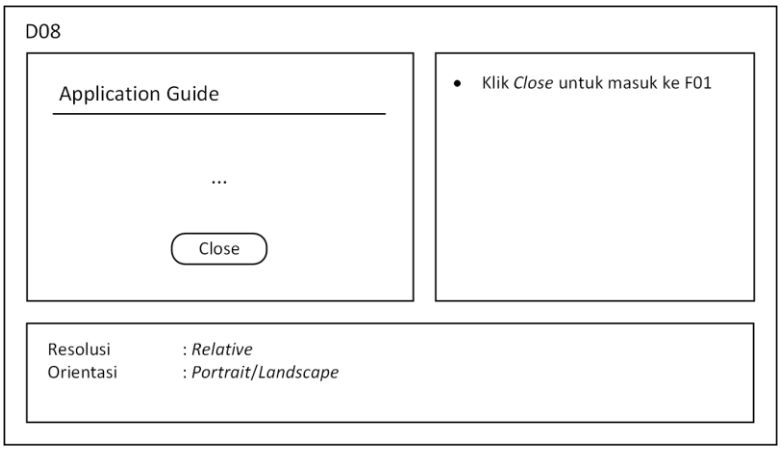
Gambar 3.65 Perancangan Antarmuka Dialog *Add to Favourites*

20. Perancangan Antarmuka Dialog *Loading*



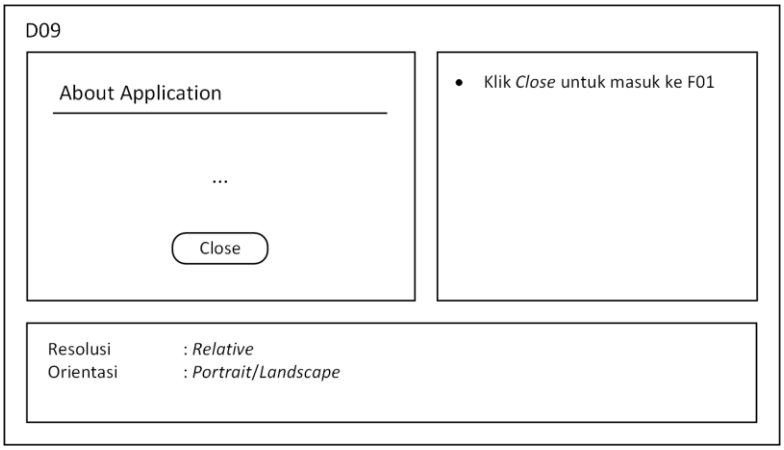
Gambar 3.66 Perancangan Antarmuka Dialog *Loading*

21. Perancangan Antarmuka Dialog *Help*



Gambar 3.67 Perancangan Antarmuka Dialog *Help*

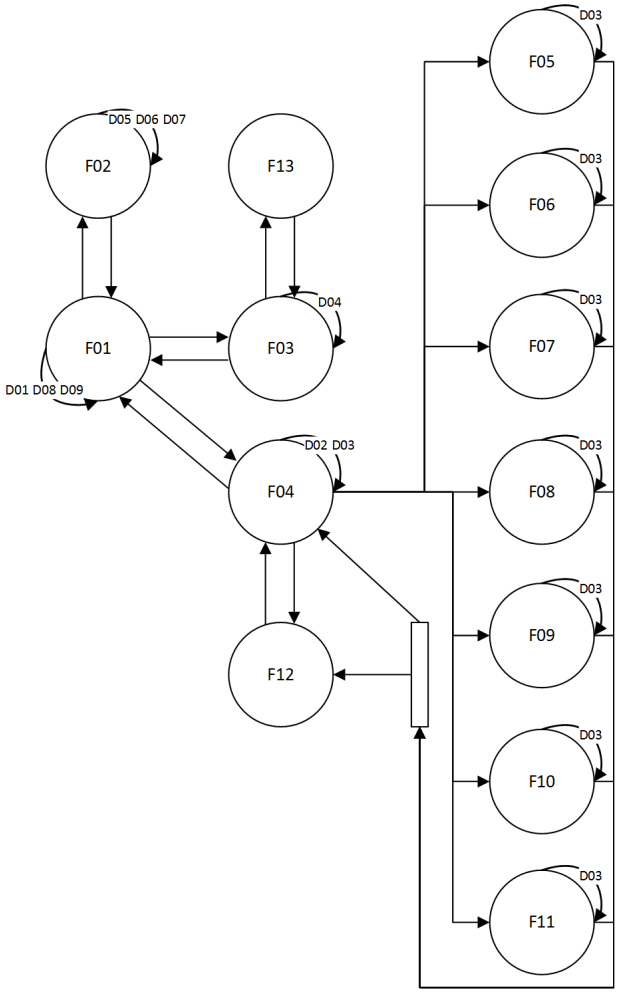
22. Perancangan Antarmuka Dialog *About*



Gambar 3.68 Perancangan Antarmuka Dialog *About*

3.2.3 Jaringan Sematik

Jaringan semantik merupakan gambaran tentang relasi dari masing-masing halaman yang bisa diakses oleh pengguna. Jaringan semantik ini, digunakan sebagai panduan penggunaan menu-menu dalam sebuah aplikasi. Perancangan semantik dapat dilihat pada Gambar 3.69.



Gambar 3.69 Jaringan Semantik