



### **Hak cipta dan penggunaan kembali:**

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

### **Copyright and reuse:**

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

## BAB III

### PERANCANGAN APLIKASI

#### 3.1 Spesifikasi Umum Kebutuhan Aplikasi

Aplikasi yang baik, dikembangkan dengan membutuhkan perancangan yang benar dan matang. Perancangan aplikasi ini dijelaskan melalui diagram-diagram yang merepresentasikan alur kerja aplikasi sebagai berikut.

##### 3.1.1 Use Case

Aplikasi penerjemahan kata dapat diakses oleh *user* melalui *gadget* Android-nya. Aktivitas yang dapat dilakukan oleh *user*, yaitu memilih menu pencarian terjemahan kata dari Bahasa Mandarin ke dalam Bahasa Indonesia dengan *format pinyin* dan memilih menu bantuan.

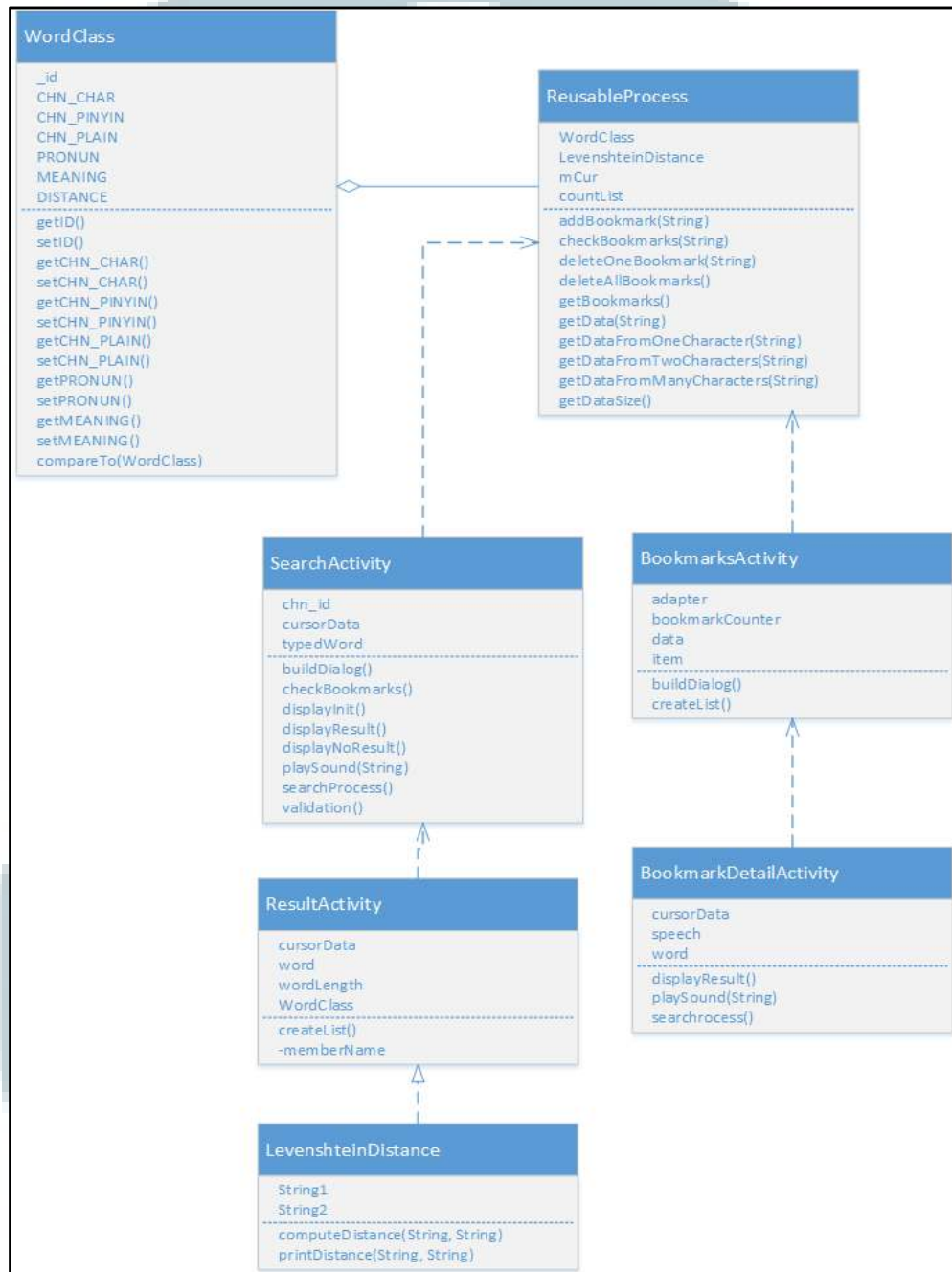


Gambar 3.1 Use case sistem

##### 3.1.2 Class Diagram

Prosedur kerja aplikasi secara umum dapat dijelaskan dengan menggunakan diagram. Proses bermula dari *pinyin* yang *diinput* oleh *user*.

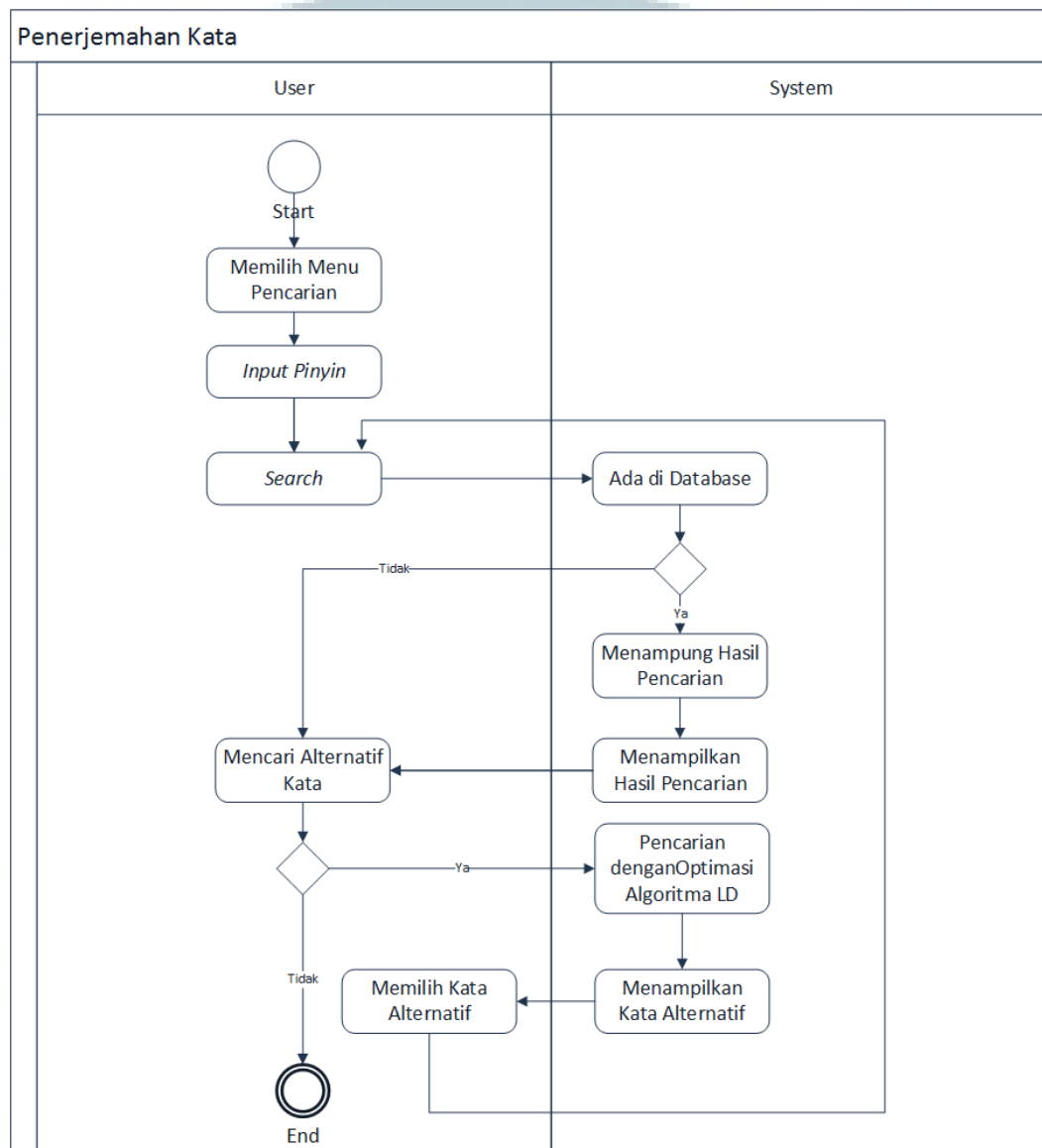
Kemudian sistem melakukan pencarian kata dengan menggunakan optimasi algoritma Levenshtein Distance. Berikutnya akan muncul hasil pencarian yang dilakukan, berikut dengan alternatif kata yang telah diseleksi oleh algoritma Levenshtein Distance. Berikut ini adalah representasi dari *class diagram* aplikasi pencarian kata Bahasa Mandarin ke Bahasa Indonesia.



Gambar 3.2 *Class diagram* sistem

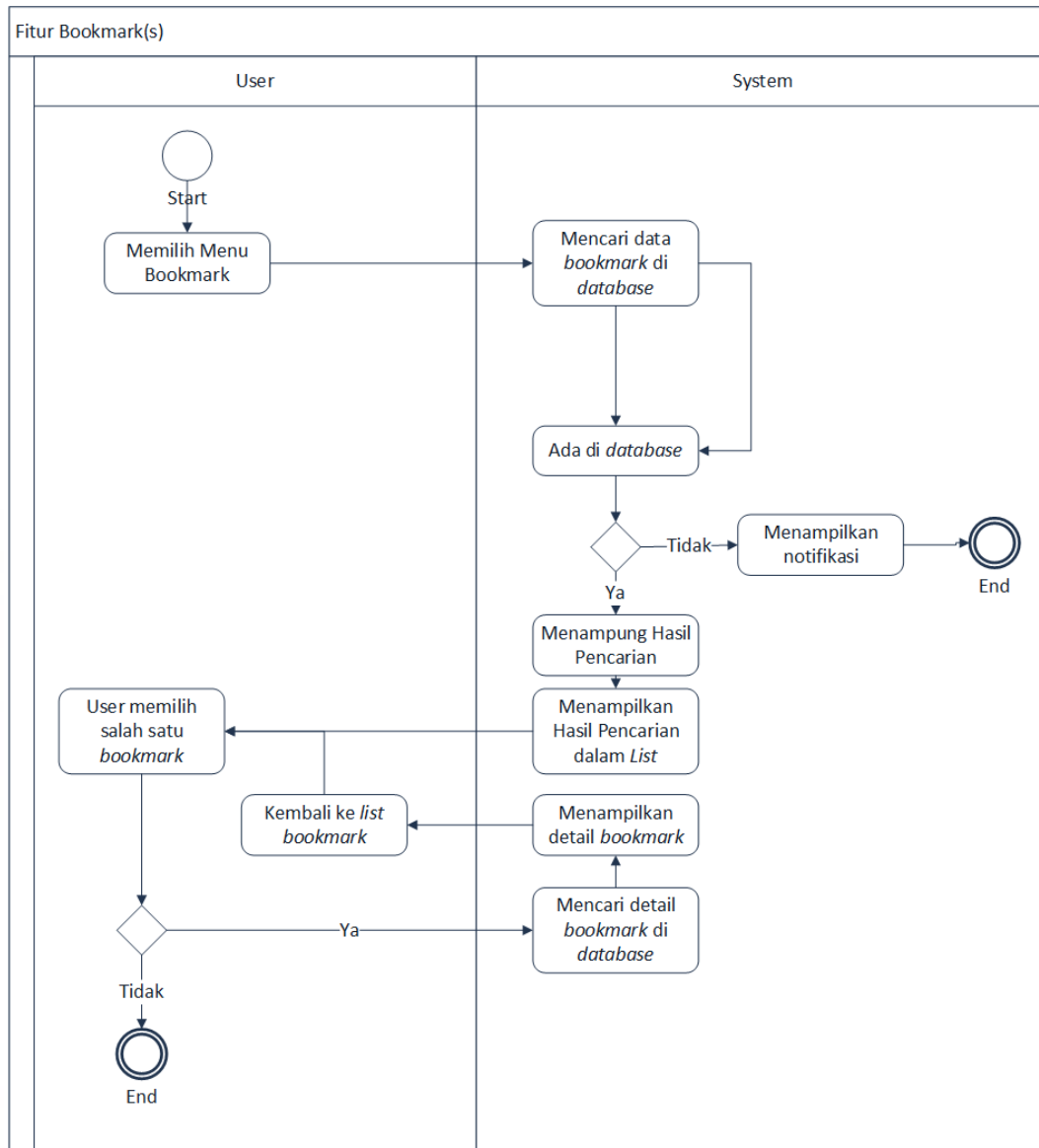
### 3.1.3 Activity Diagram

Aplikasi ini membutuhkan data yang berasal dari *input*-an oleh *user*. Berikut ini adalah *activity diagram* untuk alur data yang terjadi di dalam aplikasi penerjemah kata ini.



Gambar 3.3 *Activity diagram* proses pencarian

Pada aplikasi ini juga terdapat fitur *bookmark* yang memungkinkan *user* untuk menyimpan kata yang telah dicari ke dalam data *bookmarks*. Dengan demikian, *activity diagram* untuk fitur *bookmark* dipaparkan berikut ini.



Gambar 3.4 Activity diagram fitur bookmarks

### 3.1.4 Fungsionalitas Sistem

Fungsionalitas yang dimiliki oleh aplikasi sebagai berikut.

1. Melakukan pencarian berupa *pinyin* ke dalam *database* dengan algoritma pencarian kata.
2. Mengambil kata alternatif yang mirip dengan *pinyin* yang di-input.

3. Melakukan perhitungan dengan algoritma Levenshtein Distance untuk mengurutkan kata mana yang memiliki selisih perbedaan kata terkecil.
4. Mengurutkan kata-kata menurut selisih terkecil. Selisih yang paling kecil adalah kata prioritas.
5. Menampilkan *output* seperti arti kata, *pinyin*, dan karakter Cina dari hasil pencarian kata.

### 3.1.5 Masukan dan Keluaran Sistem

Masukan yang diperlukan oleh sistem adalah *pinyin* yaitu penulisan *chinese character* ke dalam *alphabetical*.

Keluaran dari sistem adalah sebagai berikut.

1. Arti kata dari *pinyin* yang telah diinput oleh *user*.
2. Alternatif *pinyin* yang memiliki selisih perbandingan yang paling kecil.

## 3.2 Desain Sistem

Dalam membangun sebuah aplikasi yang baik dan benar, perancangan sistem harus spesifik sehingga dapat mengurangi resiko kegagalan. Berikut ini adalah analisa dan perancangan sistem.

### 3.2.1 Desain Modul

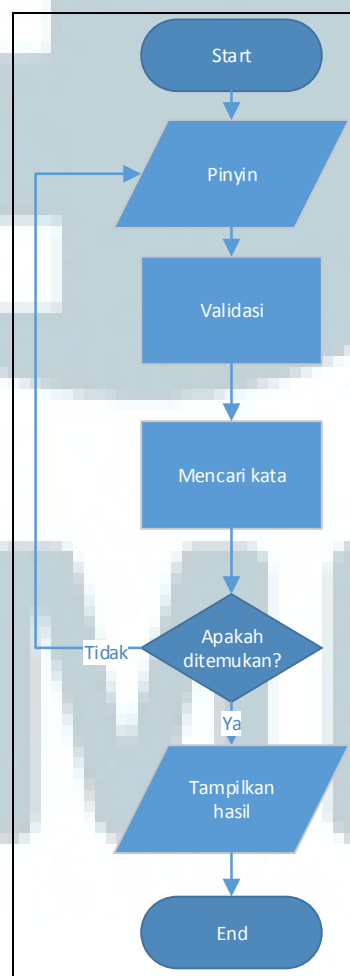
Desain modul terdiri dari desain modul pencarian terjemahan dan desain modul *bookmarks*.

1. Desain modul pencarian terjemahan

Dalam mencari terjemahan kata yang pertama kali dilakukan adalah melakukan *input pinyin* pada kolom *text field* dan menekan tombol *search*. Kemudian, *input-an* akan dilakukan validasi. Pada proses validasi, nilai *true*

menunjukkan bahwa data siap untuk diproses sedangkan nilai *false* menunjukkan bahwa terdapat karakter lain selain alfabet yang ada di dalam *string*, misalnya *special character*. Jika karakter yang tersisip pada validasi yang bernilai *false* adalah spasi, maka spasi akan dihapus.

Pada saat validasi bernilai *true*, maka akan dilanjutkan ke tahap selanjutnya. Pencarian ke dalam *database* dilakukan berdasarkan data yang sudah dilakukan validasi. Untuk data yang ditemukan maka hasilnya akan langsung ditampilkan di dalam *activity*. Namun, untuk data yang tidak ditemukan akan berlanjut ke modul pencarian alternatif kata. Gambar 3.5 adalah desain sistem modul pencarian kata.



Gambar 3.5 *Flowchart* penerjemahan kata

## 2. Desain modul pencarian alternatif kata

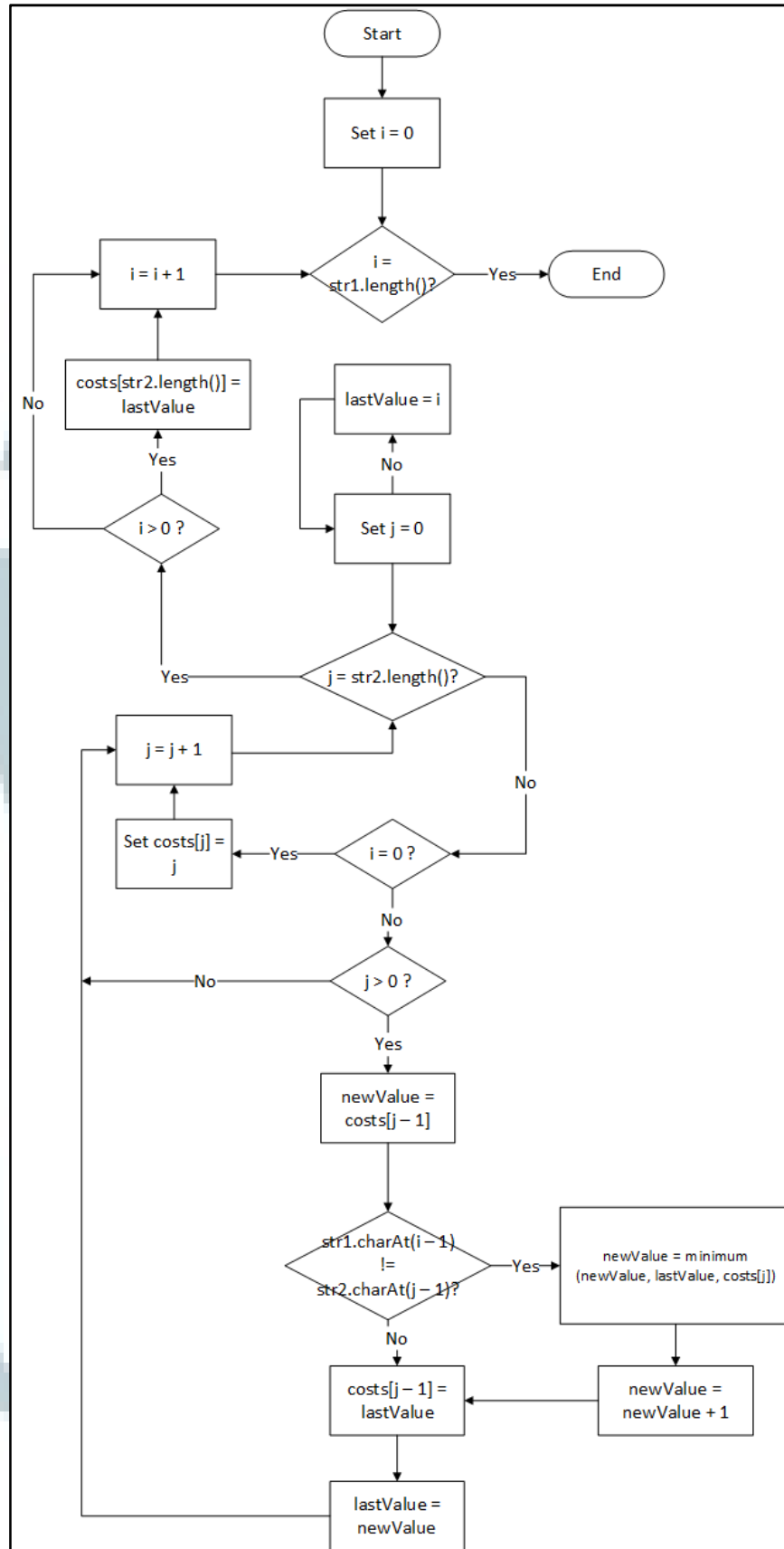
Modul pencarian alternatif kata merupakan pengembangan dari modul pencarian kata. Kata yang tidak ditemukan pada proses pencarian kata dilanjutkan pada proses pencarian alternatif kata.

Prosesnya bermula dengan pencarian kata-kata alternatif yang serupa ke dalam database. Syaratnya adalah jumlah karakter *string* kata alternatif sama dengan jumlah panjang karakter dari *string* yang dicari. Selanjutnya kata-kata alternatif ditampung di dalam sebuah *array of class* yang dapat menampung atribut-atribut dari hasil *query* sebelumnya.

Proses selanjutnya adalah melakukan perhitungan *distance* dengan algoritma Levenshtein Distance antara *string* yang dicari dengan semua data pada *array of class* tersebut. Proses perhitungan ini melibatkan tiga proses dasar dalam perhitungan Levenshtein Distance yaitu penyisipan, penggantian, dan penghapusan.

Proses berikutnya adalah melakukan *sort* dimulai dari kata dengan *distance* yang paling kecil sampai dengan *distance* yang paling besar. Proses perhitungan ini membentuk sebuah matriks dengan besar panjang *string* pertama ditambah dengan 1 dikali dengan panjang *string* kedua ditambah dengan 1. Misalnya, untuk panjang *string* pertama dan *string* kedua secara berturut-turut adalah 2 dan 3, maka matriks yang dihasilkan adalah matriks 3x4. Gambar 3.5 adalah *flowchart* yang memaparkan alur kerja algoritma Levenshtein Distance.



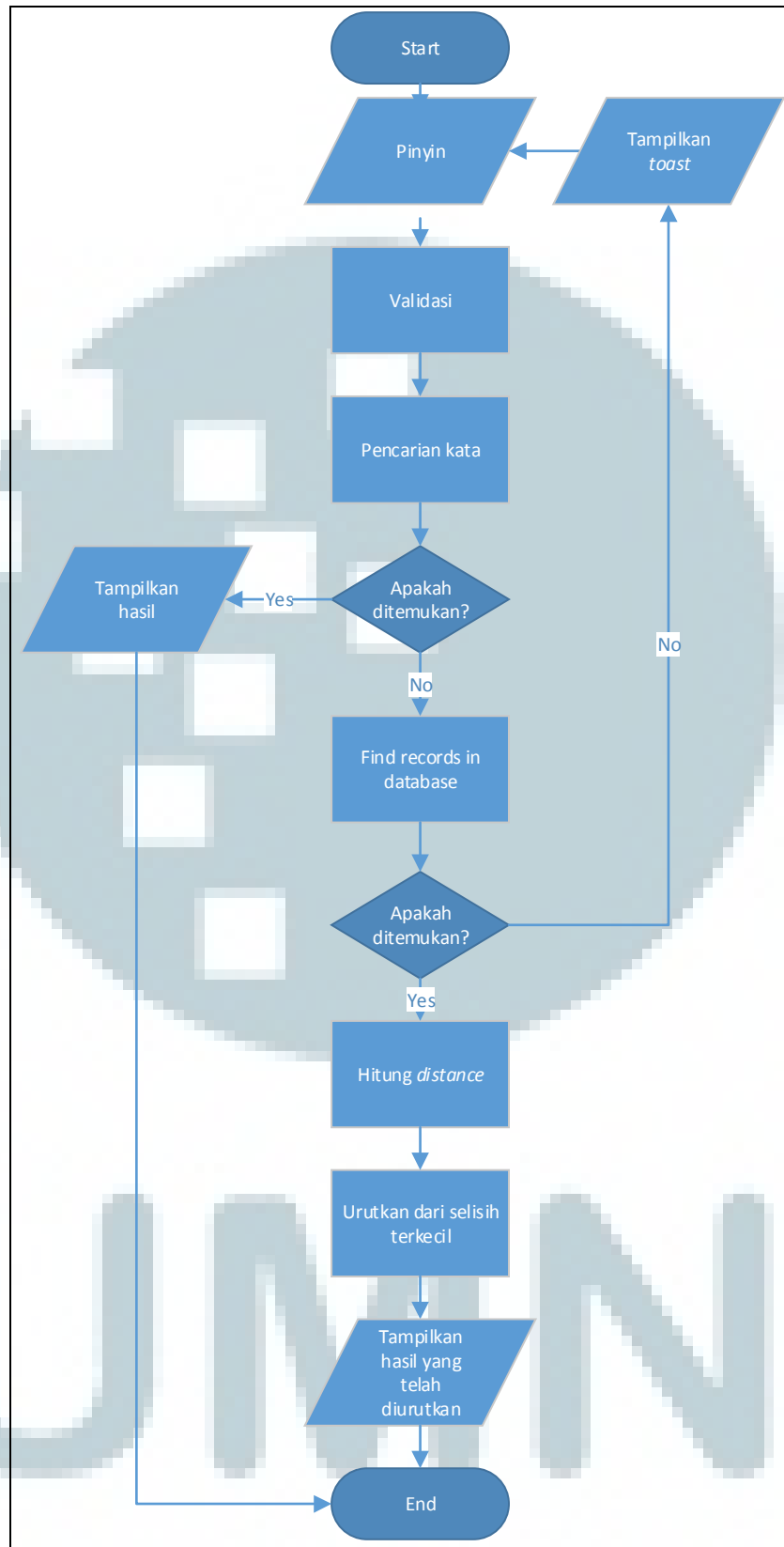


Gambar 3.6 Flowchart algoritma Levenshtein Distance

Pemrograman Android berbasis pada pemrograman Java sehingga proses sorting dilakukan dengan menggunakan *interface Comparable*. Proses *sorting* dilakukan dengan melakukan *override method compareTo* pada *class* yang berisi data tampung. Sebuah *array* yang telah diurutkan ditampilkan dalam sebuah *activity result*. Daftar kata-kata alternatif yang ditampilkan dapat diklik oleh *user*. Kata yang dipilih akan ditampilkan terjemahannya pada *activity* awal.

Pada saat kembali ke *activity* awal dengan menampilkan terjemahan kata alternatif yang dipilih, *user* dapat memilih kembali apakah ingin dilakukan pencarian lanjutan. Jika ya, maka proses pencarian kata alternatif akan berulang dan jika tidak, maka proses pencarian kata alternatif selesai dilakukan. *Flowchart* proses pencarian kata alternatif dapat dilihat pada gambar 3.7.

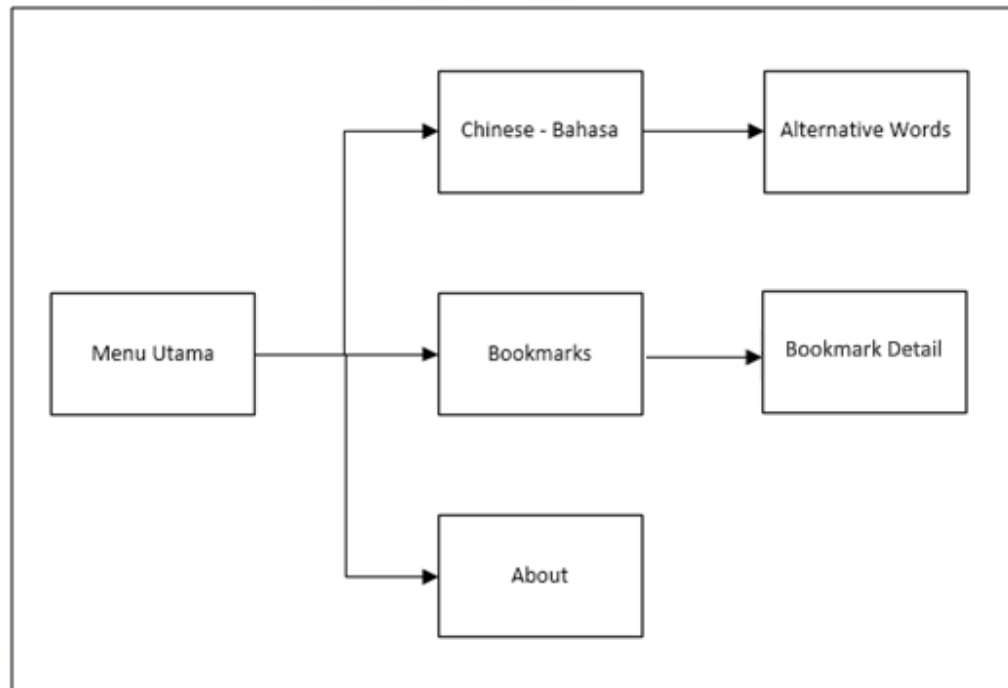
UMMN



Gambar 3.7 *Flowchart* pencarian alternatif kata

### 3.2.2 Hirarki Menu

Hirarki menu dari aplikasi penerjemah kata Bahasa Mandarin ke dalam Bahasa Indonesia dipaparkan pada berikut ini.



Gambar 3.8 Hirarki menu

Pada aplikasi penerjemahan kata ini terdapat tiga menu utama pada halaman awal, yaitu menu “Chinese – Bahasa”, “Bookmarks”, dan “About”. Setiap aplikasi pada Android menggunakan halaman kerja yang disebut dengan *activity*. Pada setiap menu dalam aplikasi ini juga direpresentasikan ke dalam masing-masing *activity* yang mempunyai kinerja dan fungsi yang berbeda-beda.

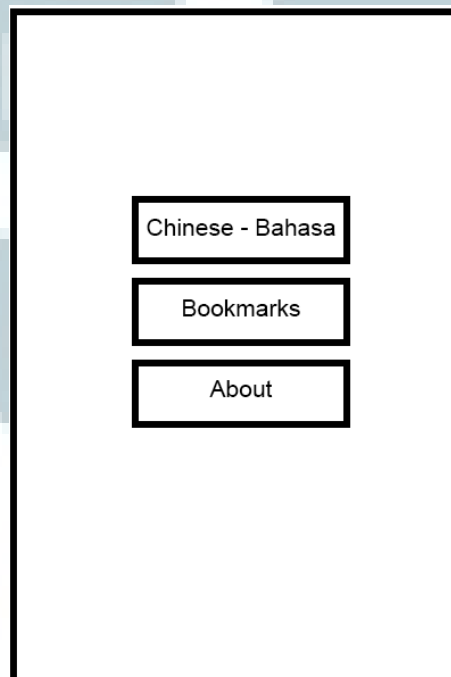
Menu “Chinese – Bahasa” merupakan menu utama yang digunakan untuk proses penerjemahan. Proses penerjemahan melibatkan *input* dari *user* untuk dilakukan proses penerjemahan. Untuk setiap proses pencarian yang tidak menghasilkan terjemahan akan diberikan navigasi ke *activity* lain, yang tidak lain adalah menu “Alternative Words”.

Menu “Bookmarks” adalah fitur tambahan yang menyimpan kata-kata yang disimpan oleh *user*. Fitur ini berfungsi untuk mempermudah *user* dalam mencari kata yang sebelumnya sudah pernah dicari. Selain itu, fitur ini juga mempermudah *user* untuk menyimpan kata yang sering kali digunakan.

### 3.2.3 Tampilan Antarmuka

#### a. Menu Utama

Halaman menu utama memuat tiga tombol navigasi ke halaman lainnya, yaitu halaman “Chinese – Bahasa”, “Bookmarks”, dan “About”. Gambar 3.9 merupakan rancangan tampilan dari menu utama.

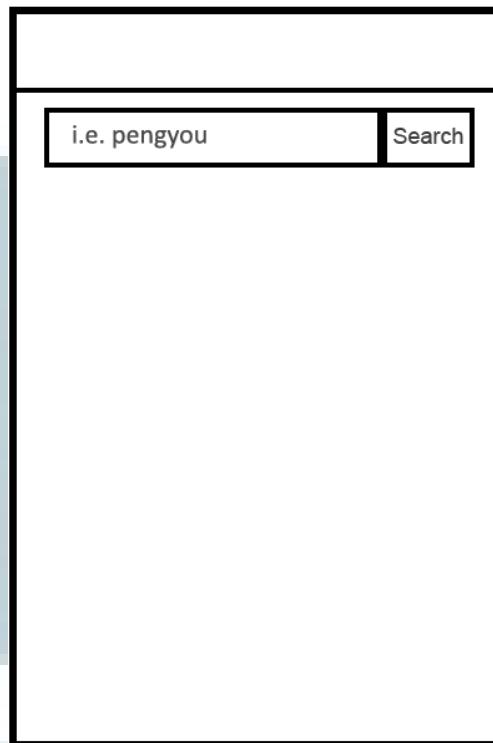


Gambar 3.9 Rancangan antarmuka menu utama

#### b. Halaman “Chinese – Bahasa”

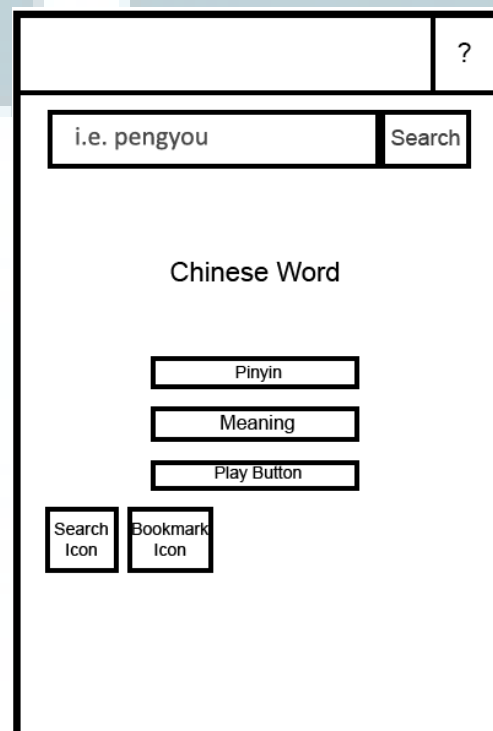
Halaman Chinese – Bahasa merupakan halaman yang digunakan untuk proses penerjemahan kata dalam Bahasa Mandarin ke Bahasa Indonesia. Pada halaman ini terdapat sebuah *text field* yang digunakan untuk menerima *input* dari

*user*. Selain *text field*, pada halaman ini juga terdapat sebuah *button* yang akan memproses *input*-an.



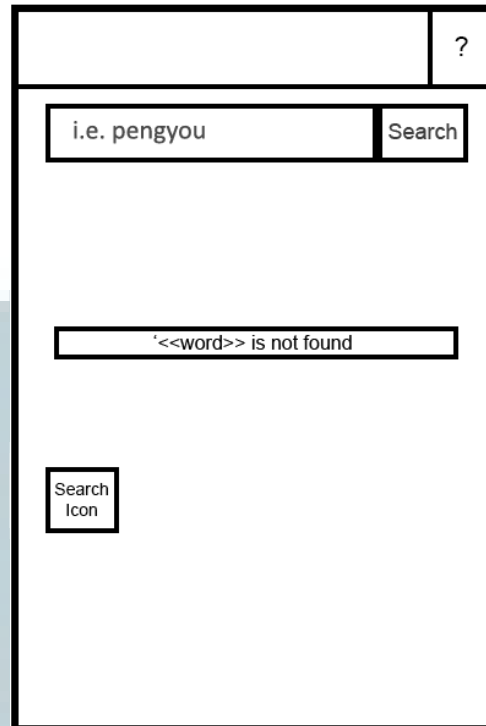
A wireframe diagram of a search menu interface. It consists of a rectangular frame with a header bar at the top. Below the header, there is a search input field containing the text "i.e. pengyou" and a "Search" button to its right.

Gambar 3.10 Rancangan antarmuka menu pencarian



A wireframe diagram of a word detail interface. It features a header bar with a question mark icon on the right. Below the header is a search input field with "i.e. pengyou" and a "Search" button. The main content area is titled "Chinese Word" and contains three stacked buttons: "Pinyin", "Meaning", and "Play Button". At the bottom left, there are two buttons: "Search Icon" and "Bookmark Icon".

Gambar 3.11 Rancangan antarmuka untuk kata yang ditemukan

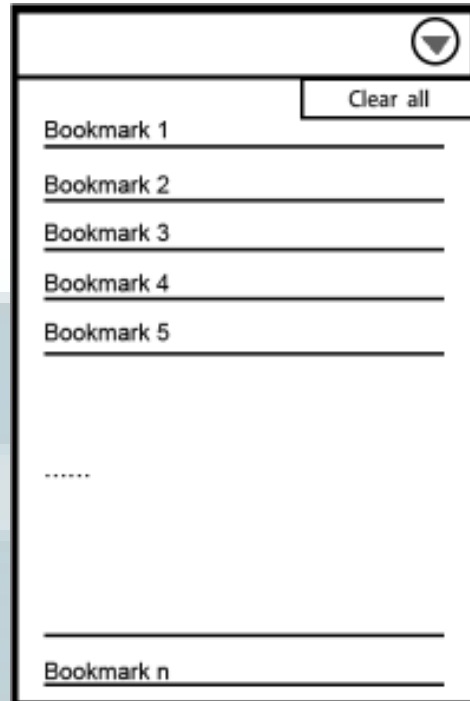


Gambar 3.12 Rancangan antarmuka untuk kata yang tidak ditemukan

c. Halaman “Bookmarks”

Halaman “Bookmarks”, berisi daftar kata yang telah disimpan. Halaman ini berfungsi agar memudahkan *user* untuk melihat kata-kata yang sering digunakan tanpa harus mencari, melalui proses pencarian makna. Gambar 3.13 adalah rancangan tampilan antarmuka untuk halaman “Bookmarks”.

U M N

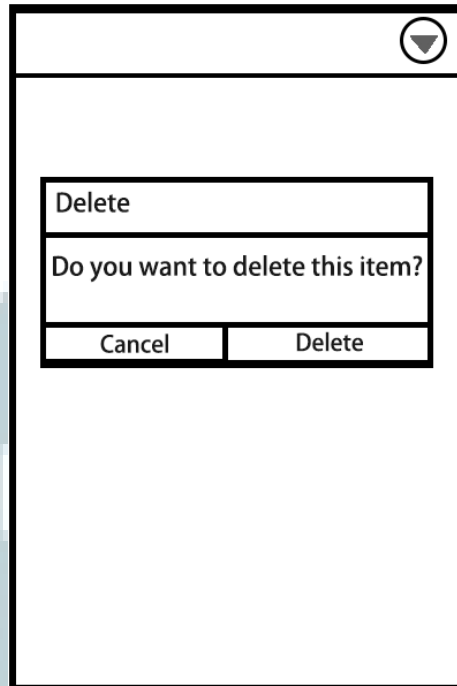


Gambar 3.13 Rancangan antarmuka halaman Bookmarks

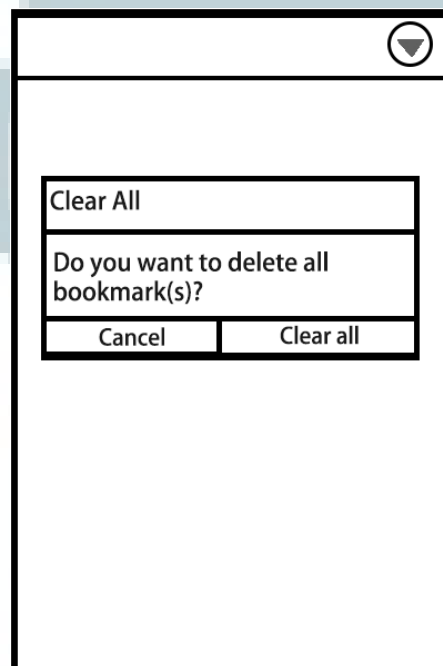
Setiap *bookmark* dalam *activity* ini disusun dalam sebuah daftar *bookmark* dalam bentuk *list view*. Setiap kata yang telah disimpan dalam *bookmark* dapat dihapus atas keinginan *user*. Untuk itu, untuk proses penghapusan akan ditampilkan kotak dialog pada setiap saat *user* melakukan penghapusan, baik untuk menghapus satu data maupun semua data.

UMN





Gambar 3.14 Rancangan antar muka delete item



Gambar 3.15 Rancangan antar muka delete all item

Setiap *item* di dalam *list view* dapat dipilih untuk melihat terjemahan kata.

Pada saat detail dari *item* yang dipilih, maka akan beralih ke *activity* baru yang

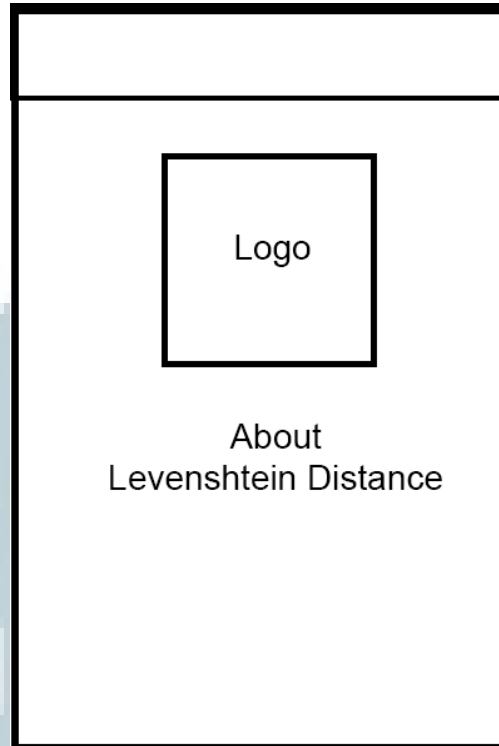
bernama *activity* detail *bookmark*. *Activity* detail *bookmark* berisi detail karakter, *pinyin*, arti kata, dan *button* untuk memutar *audio* pengucapan dalam Bahasa Mandarin. Gambar 3.16 merupakan rancangan tampilan untuk *activity* detail *bookmark*.



Gambar 3.16 Rancangan antar muka halaman detail bookmark

d. Halaman *About*

Halaman *About* berisi penjelasan mengenai seluk beluk aplikasi berikut dengan sedikit ulasan tentang algoritma yang digunakan. Pada halaman ini juga dipaparkan bagaimana algoritma Levenshtein Distance berperan dalam sebuah aplikasi seperti aplikasi penerjemahan kata ini. Gambar 3.17 adalah rancangan antarmuka halaman *About*.



Gambar 3.17 Rancangan antarmuka halaman “About”

Halaman ini terdiri dari sebuah *image* yang menampilkan logo aplikasi yang diberi nama CI Translator. Di bawah *image*, terdapat teks yang berisi informasi seperti yang sudah dijelaskan sebelumnya. Halaman ini tidak melibatkan interaksi dengan *user*.

UMN