



Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk mengubah, 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

METODOLOGI DAN PERANCANGAN APLIKASI

3.1 Metodologi Penelitian

Adapun tahap-tahap metodologi penelitian aplikasi repositori tutorial pengembangan piranti lunak berbahasa Indonesia adalah sebagai berikut.

1. Studi literatur

Pada tahap studi literatur akan dilakukan telaah literatur, yaitu membaca dan mempelajari paper, jurnal, buku, dan referensi lainnya mengenai LINE *chatbot*, algoritma *stemming* Nazief & Adriani, algoritma Boyer-Moore, *Technology Acceptance Model* (TAM) dan literatur-literatur lain yang berkaitan dengan aplikasi LINE *chatbot* layanan informasi lembaga bimbingan yang akan dibangun.

2. Analisis Kebutuhan

Pada tahap ini akan dijabarkan usaha terkait pengumpulan data, menjelaskan data yang diperlukan, serta teknik pengumpulan data yang digunakan.

3. Perancangan

Pada tahap perancangan akan dibuat *flowchart* dan struktur tabel.

4. Implementasi

Hasil perancangan yang telah dibuat selanjutnya akan diimplementasikan ke dalam kode dengan bahasa pemrograman yang dipilih, yaitu PHP.

5. Uji Coba

Setelah aplikasi selesai dirancang dan dibangun, akan dilaksanakan pengujian terhadap aplikasi tersebut dengan melakukan pengujian penerimaan aplikasi.

Menurut Roscoe dalam Sugiyono (2012), ukuran sampel yang layak dalam

penelitian adalah 30 sampai dengan 500. Pada tahap ini, dilakukan pengujian sistem kepada sejumlah mahasiswa secara langsung dengan teknik *simple random sampling*, serta memberikan kuesioner dalam bentuk *google form* guna mendapatkan umpan balik terhadap sistem yang telah dibangun. Kuesioner menggunakan metode *Technology Acceptance Model (TAM)* dengan skala Likert.

6. Penulisan Laporan

Pada tahap ini akan dilakukan pencatatan dari hasil uji coba. Setiap hasil yang didapat akan dihitung keberhasilannya untuk mengetahui dampak dari aplikasi yang telah dibangun.

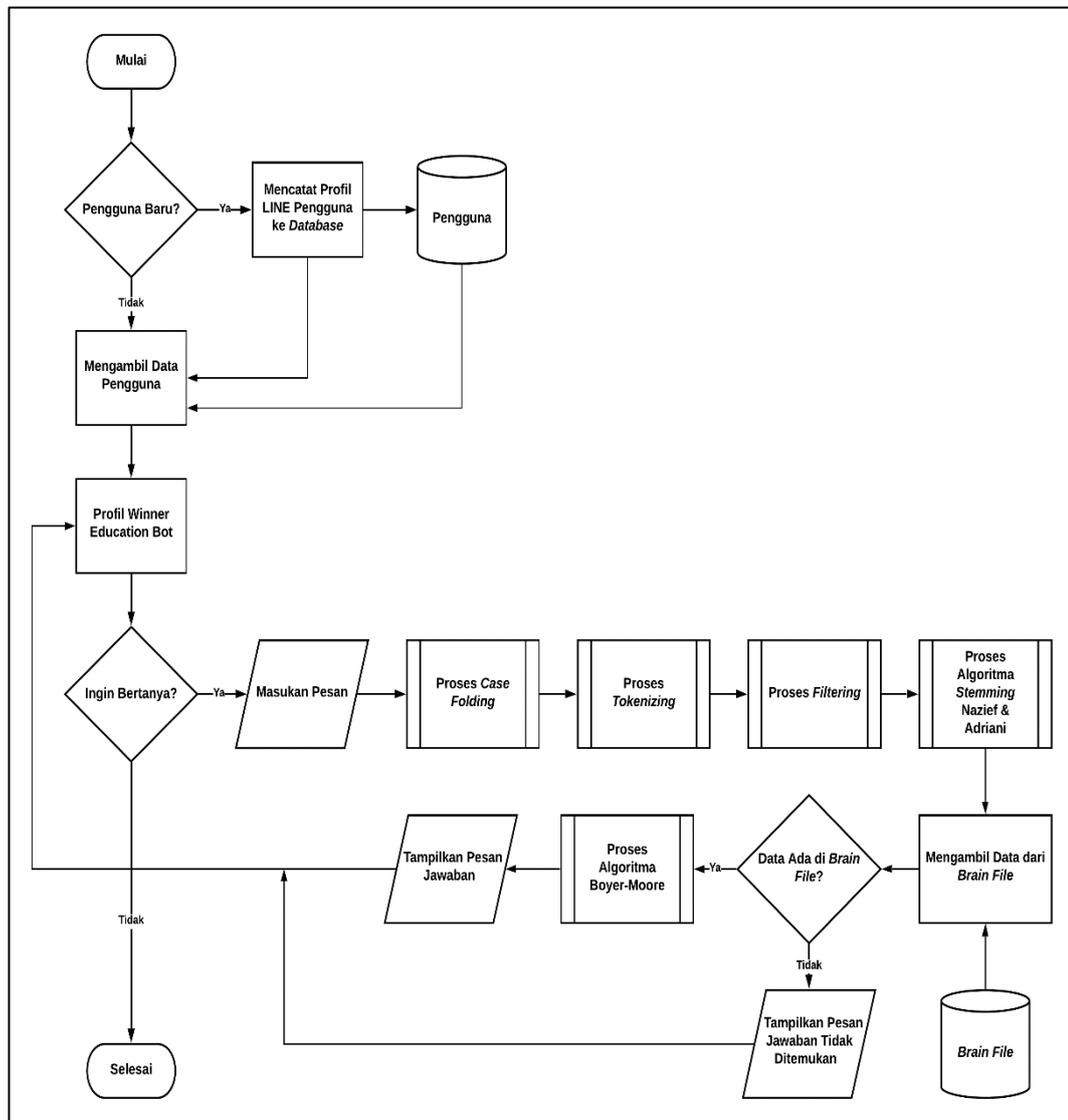
3.2 Perancangan Aplikasi

Proses perancangan aplikasi dimulai dengan membuat *Flowchart*, dan struktur tabel.

3.2.1 Flowchart Chatbot

Alur utama dari Winner Education Bot adalah, jika *terdapat input* dari *user* akan menjalani proses *case folding* yang bertujuan untuk membuat setiap kata menjadi huruf kecil. Proses selanjutnya adalah *tokenizing* yang bertujuan untuk menghilangkan tanda baca, simbol-simbol, dan membagi kalimat berdasarkan *whitespace* atau spasi sehingga diperoleh kata-kata dalam bentuk *array*. Setelah proses *tokenizing* adalah proses *filtering* atau *stopwords removal* yaitu proses di mana *stopwords* akan dihapuskan dari teks sehingga didapatkan teks yang lebih sederhana dan bersih dari kata-kata yang tidak penting. Selanjutnya, hasil dari proses *filtering* atau *stopwords removal* akan menjalani proses *stemming* dengan menggunakan algoritma Nazief & Andriani untuk mendapatkan kata dasar dari

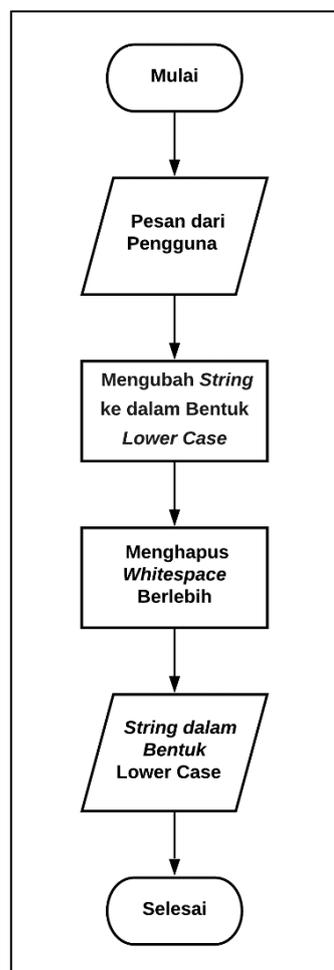
setiap teks. Setelah didapatkan hasil dari *stemming*, maka hasil tersebut digunakan untuk mengambil data dari *database*. Jika setelah proses algoritma Boyer-Moore terdapat data yang sesuai, *chatbot* akan menampilkan jawaban ke pengguna, tapi jika data tidak sesuai maka akan menampilkan pesan jawaban tidak ditemukan dan daftar pertanyaan yang bisa ditanyakan pengguna terkait Winner Education Bot. *Flowchart chatbot* dari aplikasi LINE *chatbot* layanan informasi lembaga bimbingan belajar Winner Education ditunjukkan pada Gambar 3.1.



Gambar 3.1 Flowchart *Chatbot*

3.2.2 Flowchart Case Folding

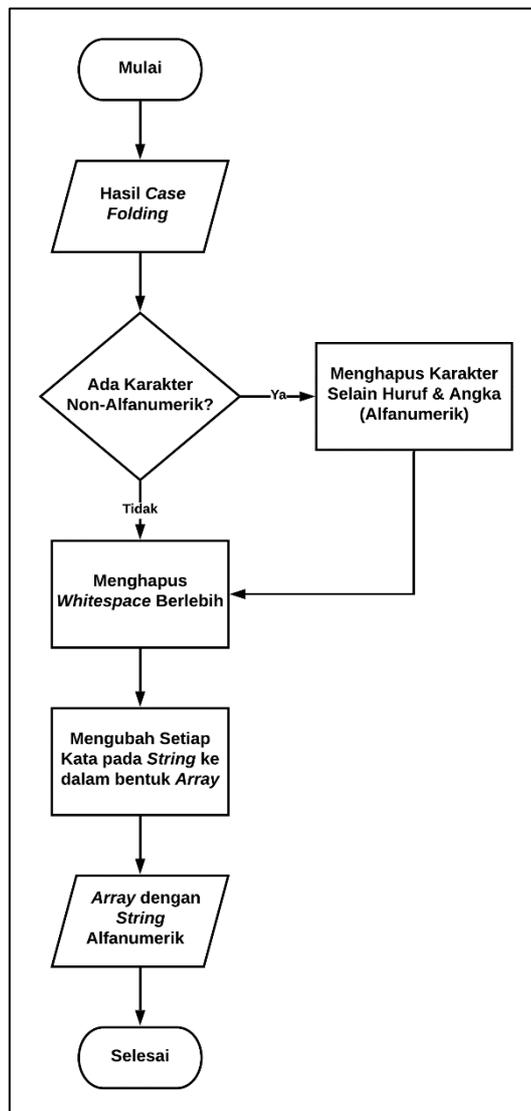
Proses *case folding* merupakan sebuah proses yang bertujuan untuk membuat seluruh kalimat menjadi huruf kecil. Pertama-tama pesan dari pengguna diubah menjadi huruf kecil atau *lower case* dengan menggunakan fungsi *strtolower()* kemudian spasi berlebih pada awal dan akhir *string* akan dihapus dengan menggunakan fungsi *trim()*. Hasil dari *case folding* berupa teks *lower case* dan tidak ada spasi berlebih di awal dan akhir teks. *Flowchart case folding* dari aplikasi LINE *chatbot* layanan informasi lembaga bimbingan belajar Winner Education ditunjukkan pada Gambar 3.2.



Gambar 3.2 Flowchart Proses *Case Folding*

3.2.3 Flowchart Tokenizing

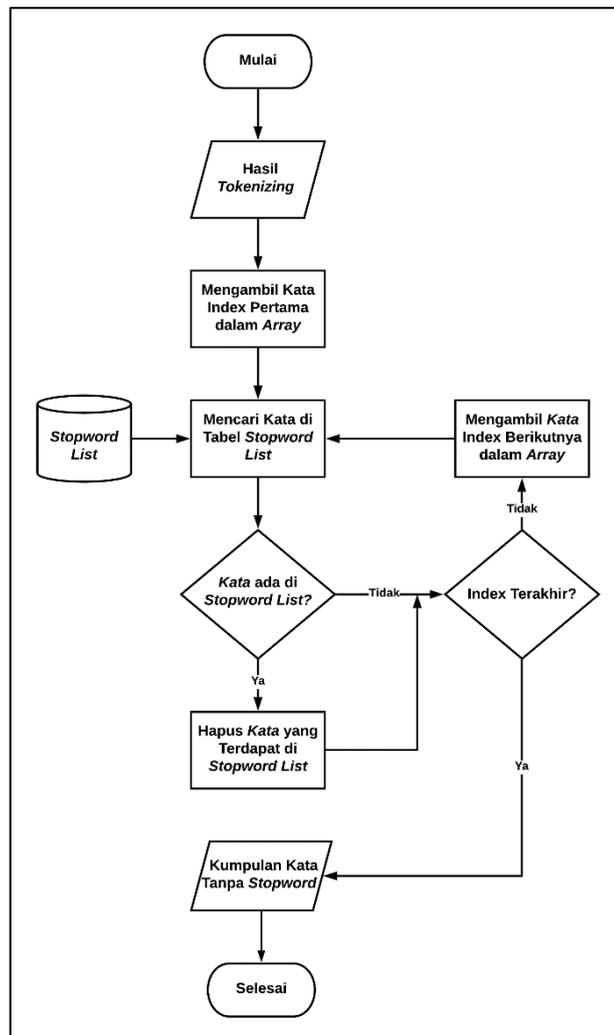
Proses *tokenizing* merupakan sebuah proses yang bertujuan untuk menghilangkan tanda baca, simbol-simbol atau karakter *non-alphanumeric*, dan membagi kalimat berdasarkan *whitespace* atau spasi sehingga diperoleh kata-kata dalam bentuk *array*. *Flowchart tokenizing* dari aplikasi LINE *chatbot* layanan informasi lembaga bimbingan belajar Winner Education ditunjukkan pada Gambar 3.3.



Gambar 3.3 Flowchart Proses *Tokenizing*

3.2.4 Flowchart Filtering

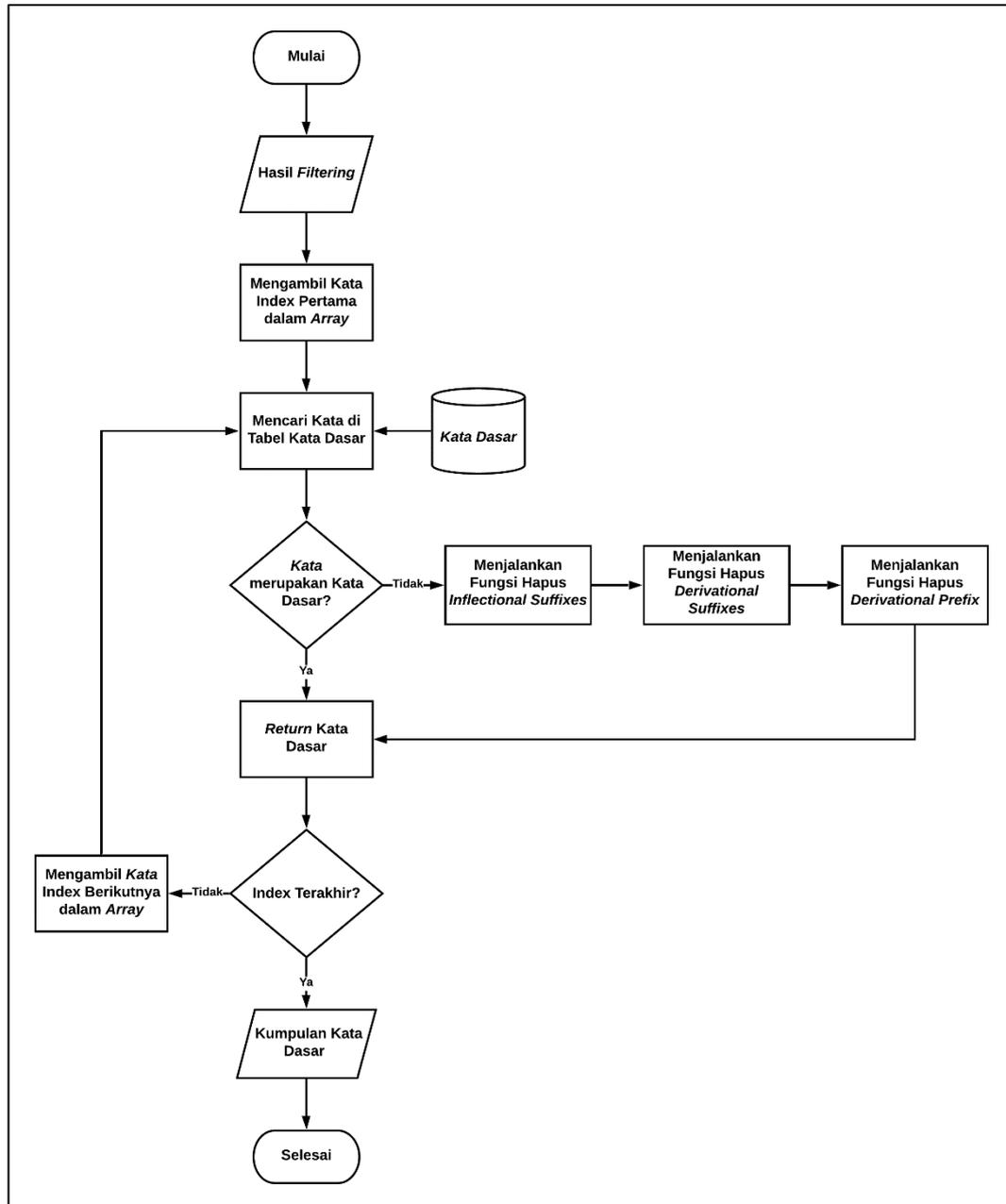
Proses *filtering* atau *stopwords removal* berfungsi untuk menghapus semua *stopword* yang ada. Hasil dari proses *tokenizing* akan dicocokkan dengan daftar *stopword* yang ada dalam tabel *stopword_list*. Kemudian teks yang terdapat di *stopword_list* akan dihapus, sedangkan teks yang tidak terdapat dalam *stopword_list* akan di kembalikan. *Flowchart filtering* dari aplikasi LINE *chatbot* layanan informasi lembaga bimbingan belajar Winner Education ditunjukkan pada Gambar 3.4.



Gambar 3.4 Flowchart Proses *Filtering*

3.2.5 Flowchart Stemming

Proses *stemming* berfungsi untuk mencari bentuk kata dasar dari setiap kata. Hasil dari proses filtering akan dilakukan penghapusan awalan (prefiks) dan akhiran (suffiks). Penghapusan akhiran yang pertama adalah *inflectional suffixes*, yaitu teks dengan akhiran “-lah”, “-kah”, “-ku”, “-mu”, atau “-nya” akan dihapus akhirannya. Penghapusan akhiran yang kedua adalah *derivational suffixes*, yaitu teks dengan akhiran “-i”, “-an” atau “-kan” akan dihapus akhirannya. Pada penghapusan *derivational suffixes* juga dilakukan pengecekan kombinasi awalan dan akhiran yang tidak diizinkan seperti be- dengan -i, di- dengan -an, ke- dengan -i dan -kan, me- dengan -an, se- dengan -i dan -kan. Penghapusan yang terakhir adalah *derivational prefix*, yaitu teks dengan awalan “di-”, “ke-”, “se-”, “te-”, “be-”, “me-”, atau “pe-” akan dihapus awalannya. Pada penghapusan *derivational prefix* juga dilakukan penggantian awalan “meng-” dan “peng-” dengan huruf “k”, “meny-” dan “peny-” dengan huruf “s”, “men-” dan “pen-” dengan huruf “t”, “mem-” dan “pem-” dengan huruf “p”, dan terdapat kasus khusus seperti kata seseorang dan terpelajar akan dihapus awalannya, yaitu “sese” dan “terpel”. Jika semua langkah telah dilakukan tetapi kata dasar tidak ditemukan pada kamus, maka kata yang asli sebelum dilakukan *stemming* akan diasumsikan sebagai kata dasar. *Flowchart stemming* dari aplikasi LINE *chatbot* layanan informasi lembaga bimbingan belajar Winner Education ditunjukkan pada Gambar 3.5.

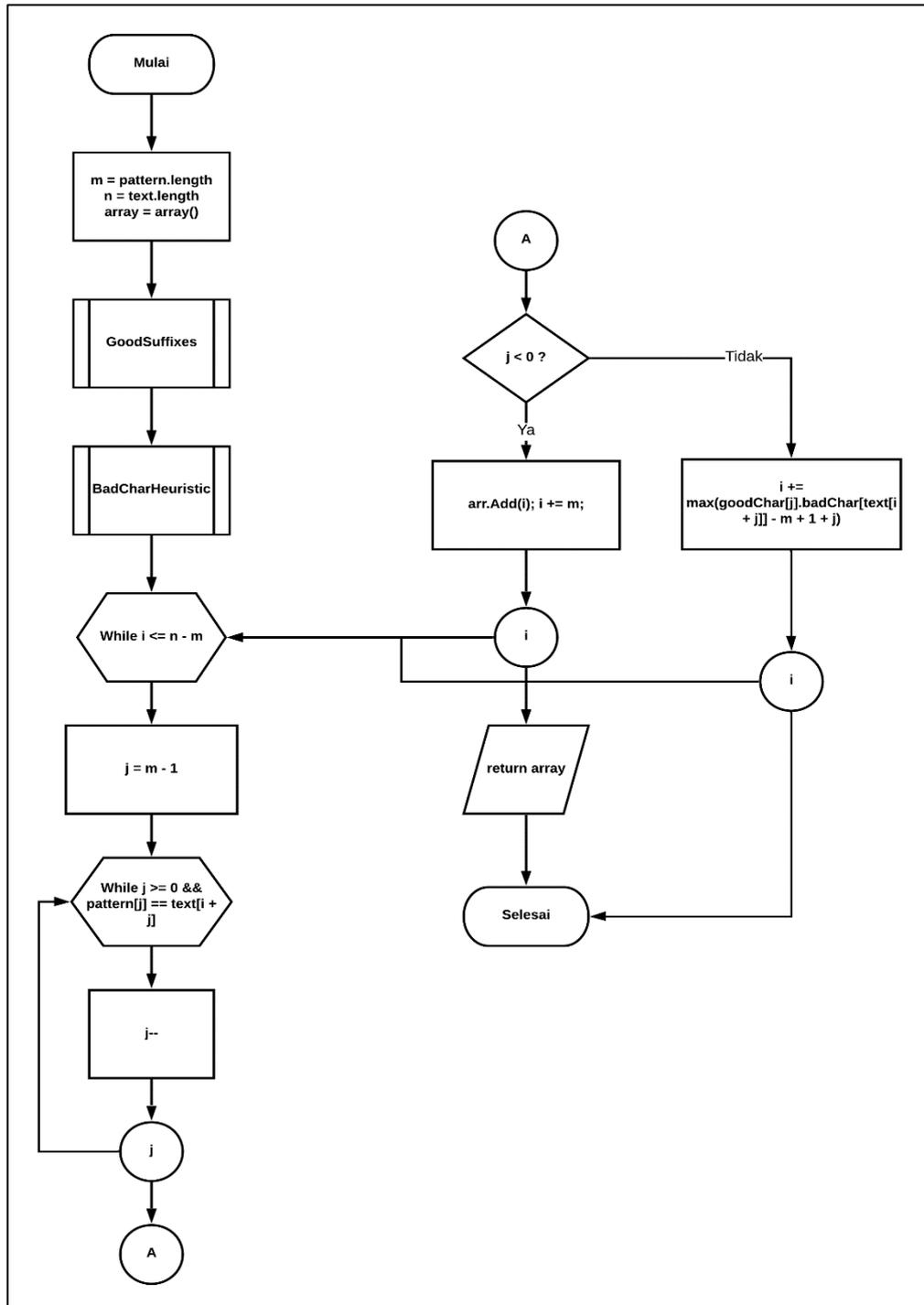


Gambar 3.5 Flowchart Proses *Stemming*

3.2.6 Flowchart Search String Boyer-Moore

Gambar 3.6 menggambarkan *flowchart* fungsi *Search String* pada Algoritma Boyer-Moore. Pada fungsi *Search String* yang pertama dilakukan adalah peletakan *window* dari *pattern* pada teks. Proses pencarian dimulai dari karakter paling kanan *pattern*. Setiap karakter akan dibandingkan satu per satu. Jika terjadi ketidakcocokan, maka akan dicek nilai pergeseran yang mungkin dilakukan ke tabel

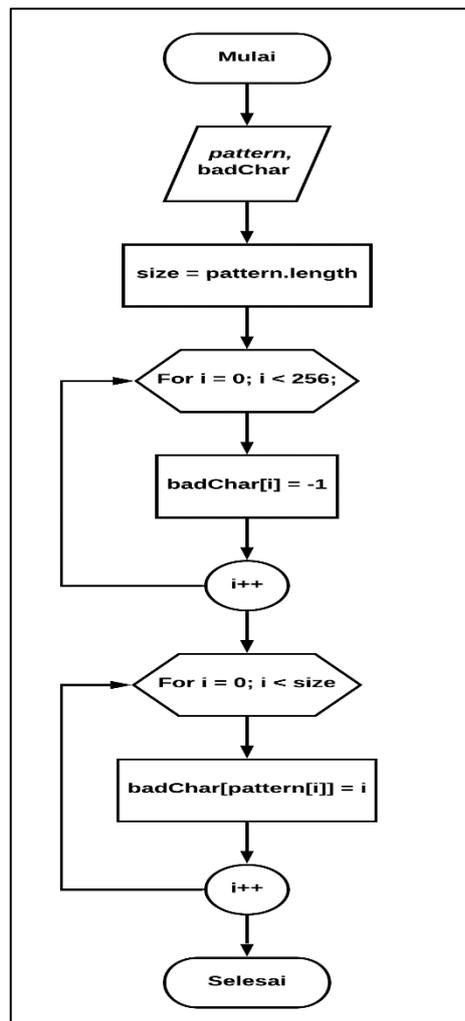
good suffix shift dan *bad character shift*. Nilai terbesar yang didapat akan diambil dan pergeseran *window* akan dilakukan sesuai dengan nilai tersebut.



Gambar 3.6 Flowchart *Search String Boyer-Moore*

3.2.7 Flowchart Fungsi Bad Character

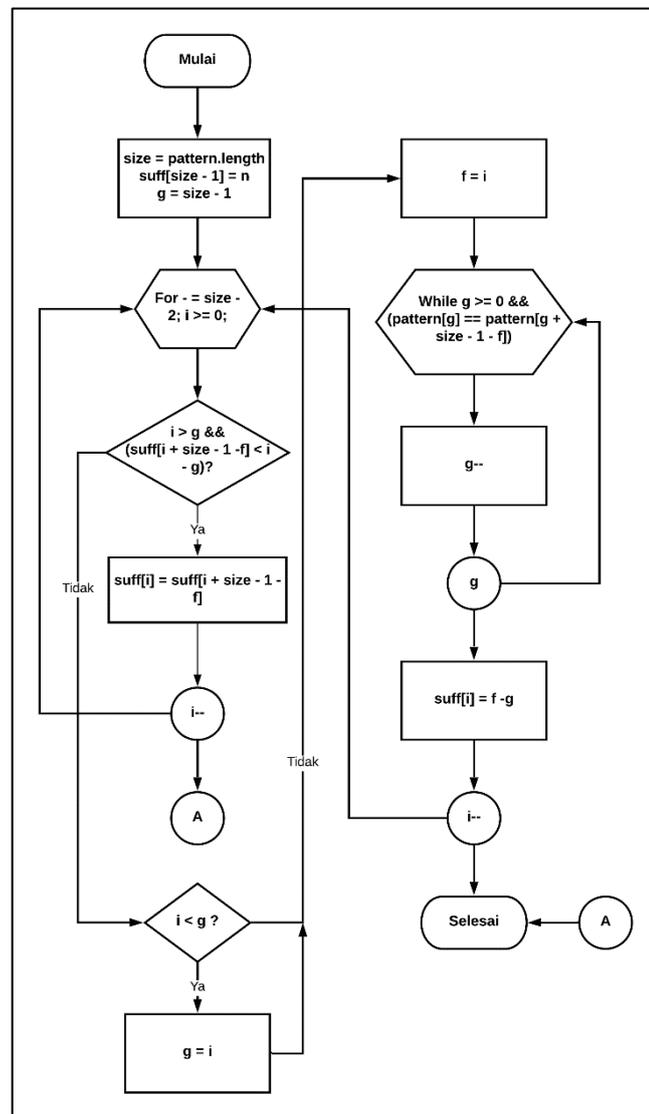
Gambar 3.7 menggambarkan *flowchart* fungsi *Bad-Character Shift*. Fungsi *Bad-Character Shift* berisi nilai-nilai pergeseran yang dapat dilakukan untuk setiap karakter yang terdapat dalam *pattern*. Setiap karakter yang ada di *pattern* diberi nilai sesuai dengan ukuran jauhnya karakter tersebut dari karakter paling akhir dan untuk karakter yang tidak terdapat di dalam *pattern* akan diberi nilai yang sama yaitu sejumlah panjang dari *pattern* yang dimasukan, misalnya *pattern* “mawan” yang berarti panjang *pattern* adalah lima.



Gambar 3.7 Flowchart Fungsi *Bad Character*

3.2.8 Flowchart Fungsi Suffixes

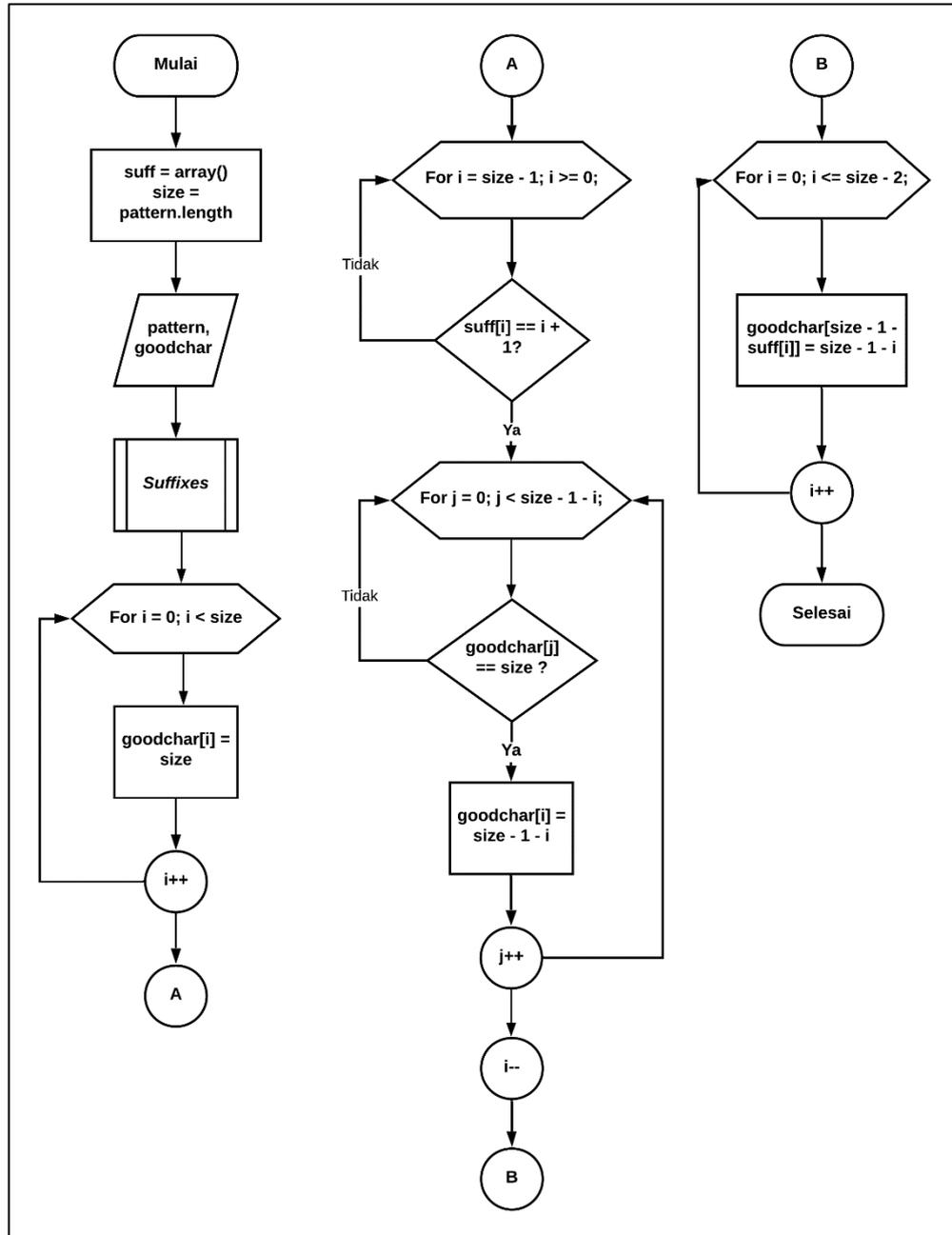
Gambar 3.8 menggambarkan *flowchart* fungsi *suffixes*. Fungsi *suffixes* digunakan untuk menghitung tabel *good-suffixes shift*. Tabel ini berisi nilai-nilai dari tiap karakter yang ada di *pattern* yang menunjukkan adanya perulangan akhiran (*suffix*) atau tidak dan dimana letak perulangan tersebut sehingga ketika proses perhitungan tabel *good-suffix shift*, dapat diketahui seberapa banyak karakter yang dapat digeser untuk pencocokan karakter yang selanjutnya.



Gambar 3.8 Flowchart Fungsi *Suffixes*

3.2.9 Flowchart Fungsi Good Suffixes

Gambar 3.9 menggambarkan fungsi *Good Suffixes*.



Gambar 3.9 Flowchart Fungsi *Good Suffixes*

Pada fungsi *Good Suffixes* akan dibuat tabel yang berisi nilai yang dapat digunakan untuk pergeseran ketika ketidakcocokan ditemukan berdasarkan karakter pada posisi beberapa yang menyebabkan ketidakcocokan. Untuk

menentukan nilai-nilainya, perlu dihitung terlebih dahulu tabel *suffix* yang fungsinya memberi tanda jika terdapat perulangan akhiran. Dari tabel *suffix* inilah tabel *goodsuffix shift* akan dihitung. Nilai dari tiap karakter yang ada dalam *pattern* bergantung pada apakah adanya perulangan akhiran (suffix) atau tidak. Semakin banyak perulangan, semakin kecil nilai pergeseran.

3.2.10 Struktur Tabel

Database yang digunakan pada penelitian ini yaitu PostgreSQL. *Database* yang dibuat memiliki lima tabel, diantaranya yaitu tabel *brain_file*, *kata_dasar*, *stopword_list*, *kata_sapa*, dan *pengguna*. Adapun nama tabel, fungsi serta deskripsi tabel adalah sebagai berikut.

1. Nama Tabel : *brain_file*
 Fungsi : berisi data pengetahuan *chatbot* yang merupakan *Frequently Asked Question (FAQ)* dari lembaga bimbingan belajar Winner Education. Deskripsi tabel ditunjukkan pada Tabel 3.1.

Tabel 3.1 Tabel *brain_file*

Nama Field	Tipe Data	Keterangan
<i>id_pengetahuan</i>	int(11)	Primary key
<i>kata_kunci</i>	varchar(1000)	Kata kunci yang digunakan untuk menentukan jawaban <i>chatbot</i>
<i>jawaban</i>	varchar(1000)	Jawaban dari FAQ untuk pertanyaan dari pengguna

2. Nama Tabel : kata_dasar
- Fungsi : menyimpan kumpulan kata dasar untuk proses *stemming*.
- Deskripsi tabel ditunjukkan pada Tabel 3.2.

Tabel 3.2 Tabel kata_dasar

Nama Field	Tipe Data	Keterangan
id_katadasar	int(11)	Primary key
katadasar	varchar(1000)	Kumpulan kata dasar bahasa Indonesia
stopword	enum('Ya', 'Bukan')	Keterangan kata dasar, apakah merupakan <i>stopword</i> atau bukan

3. Nama Tabel : stopwords_list
- Fungsi : menyimpan *stopword* atau kata umum yang dianggap tidak memiliki makna untuk proses *filtering* atau *stopword removal*. Deskripsi tabel ditunjukkan pada Tabel 3.3.

Tabel 3.3 Tabel stopwords_list

Nama Field	Tipe Data	Keterangan
id_stopword	int(11)	Primary key
stopword	varchar(1000)	Kumpulan <i>stopword</i> (kata umum yang dianggap tidak memiliki makna)

4. Nama Tabel : kata_sapa
- Fungsi : menyimpan data sapaan. Deskripsi tabel ditunjukkan pada Tabel 3.4.

Tabel 3.4 Tabel kata_sapa

Nama Field	Tipe Data	Keterangan
id_sapaan	int(11)	<i>Primary key</i>
katasapa	varchar(100)	Kata kunci yang digunakan untuk menjawab sapaan
responsapa	varchar(100)	Jawaban dari sapaan

5. Nama Tabel : pengguna

Fungsi : menyimpan data pengguna yang menambahkan Winner Education Bot sebagai teman di aplikasi LINE *Messenger*.

Deskripsi tabel ditunjukkan pada Tabel 3.5.

Tabel 3.5 Tabel pengguna

Nama Field	Tipe Data	Keterangan
id	int(11)	<i>Primary key</i>
display_name	varchar(100)	Nama LINE pengguna (sesuai dengan nama profil pengguna di LINE <i>Messenger</i>)
user_id	varchar(100)	Id LINE pengguna