



### **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

### METODOLOGI PENELITIAN DAN PERANCANGAN SISTEM

#### 3.1 Metode Penelitian

Metode penelitian yang digunakan untuk merancang dan membangun LINE *chatbot* menggunakan algoritma Bigram akan digambarkan pada Gambar 3.1, serta penjelasannya yaitu sebagai berikut.



Gambar 3.1. Metode Penelitian yang Digunakan

##### 1. Telaah Literatur

Mempelajari dan mengumpulkan literatur yang relevan dengan tujuan penelitian untuk mencapai hasil yang sesuai dengan teori yang ada.

##### 2. Analisis Kebutuhan

Menganalisis kebutuhan berdasarkan permasalahan yang terjadi, serta mengumpulkan data yang diperlukan dalam pembuatan sistem dengan melakukan wawancara terhadap perwakilan *Student Service* yang ada di Universitas Multimedia Nusantara. Hasil wawancara dapat dilihat pada Lampiran 3 dan 4.

##### 3. Desain Sistem

Membuat desain database yang akan digunakan, membuat *flowchart* bagaimana sistem berjalan, mencari API maupun *library* yang berguna dalam pembangunan sistem.

#### 4. Pemrograman Sistem

Membuat database yang dapat diakses oleh sistem, melakukan pemrograman sesuai dengan *flowchart* yang ada, mengimplementasikan API dan library yang mendukung pembangunan sistem.

#### 5. Pengujian Sistem

Menurut Roscoe dalam Sugiyono (2013), 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. Jenis kuesioner yang digunakan adalah kuesioner tertutup, yaitu kuesioner yang sudah disediakan jawabannya dengan tujuan untuk mempermudah responden dalam memberikan jawaban. Hasil kuesioner dapat dilihat pada Lampiran 5.

#### 6. Evaluasi

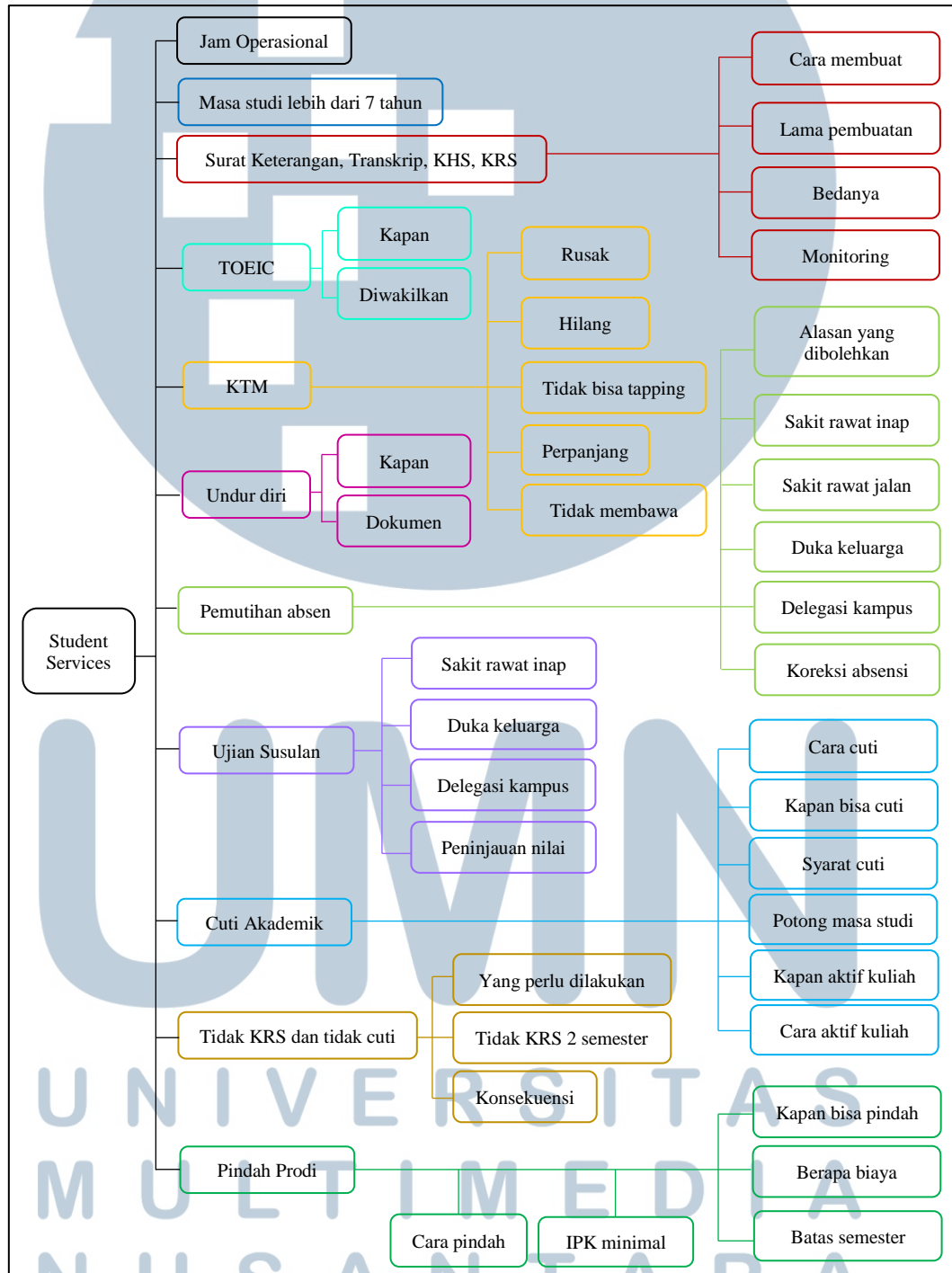
Evaluasi dilakukan dengan menganalisa umpan balik yang diberikan oleh pengguna untuk mengetahui tingkat kepuasan terhadap sistem yang telah dirancang dan dibangun menggunakan perhitungan berdasarkan skala Likert.

### 3.2 Perancangan Aplikasi

Proses perancangan aplikasi dimulai dengan menentukan area topik, serta membuat *flowchart* dan struktur tabel.

### 3.2.1 Area Topik

Dalam penelitian ini terdapat beberapa pertanyaan seputar Student Services yang dapat dijawab oleh chatbot, lebih jelasnya dapat dilihat seperti pada Gambar 3.2.



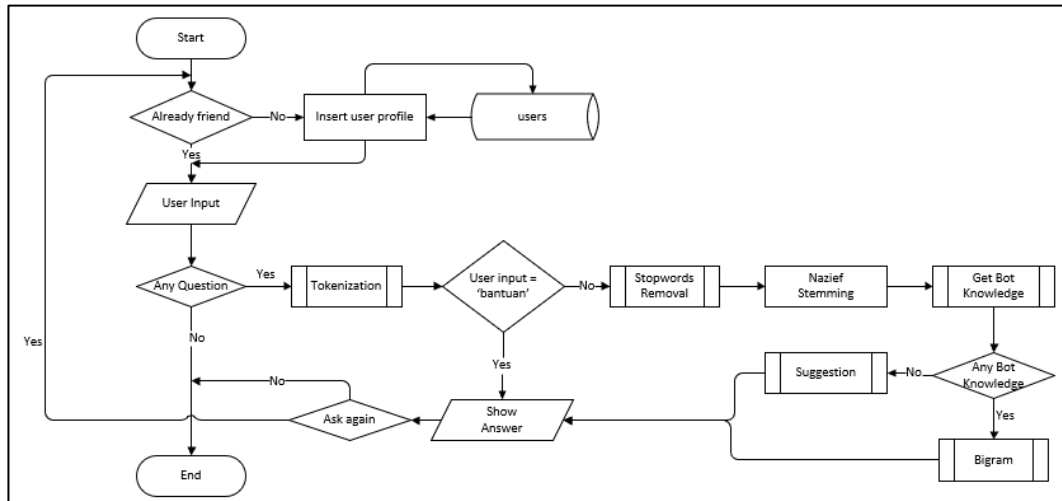
Gambar 3.2 Area Topik

### 3.2.2 Flowchart Umum

Untuk memulai percakapan dengan *chatbot*, *user* harus menambahkan *chatbot* menjadi teman terlebih dahulu. Jika belum berteman maka ketika menekan tombol berteman, data *profile user* akan disimpan ke tabel *users* yang ada di *database*. Lalu *user* dapat memberikan *input*, jika terdapat *input* dari *user* akan menjalani proses *tokenization* yang bertujuan untuk menyamaratakan *case* menjadi huruf kecil, menghilangkan tanda baca, simbol-simbol, serta membagi kalimat berdasarkan spasi sehingga diperoleh kata-kata dalam bentuk array.

Pada kasus *user* mengetikkan kata “bantuan”, maka *chatbot* akan segera menampilkan jawaban, namun pada kasus biasa hasil dari proses *tokenization* akan dimasukkan ke dalam proses *stopwords removal*, yaitu proses di mana *stopwords* akan dihapuskan atau dibuang dari teks sehingga didapatkan teks yang lebih sederhana. Selanjutnya, hasil dari proses *stopwords removal* akan menjalani proses *stemming* dengan menggunakan algoritma Nazief & Andriani dari *library* yang sudah ada sebelumnya.

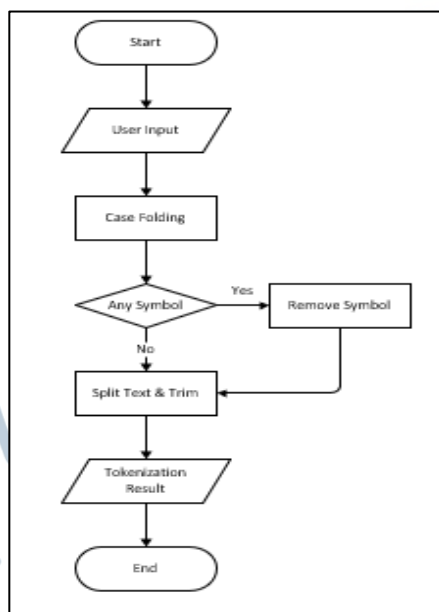
Setelah didapatkan hasil dari *stemming*, maka hasil tersebut digunakan untuk mengambil data dari *database* pada proses *get bot knowledge*. Jika tidak terdapat data yang sesuai, *chatbot* akan memasuki proses *suggestion* yang bertujuan untuk memberikan saran kata-kata yang sesuai kepada *user*, namun jika terdapat data yang sesuai, *chatbot* akan melanjutkan kepada proses bigram, pada proses ini hasil *stemming* dan *bot knowledge* akan dipotong-potong per dua karakter lalu dihitung nilai *similarity*-nya. Jawaban yang ditampilkan sesuai dengan hasil *similarity* yang tertinggi. Alur proses selengkapnya dapat dilihat pada Gambar 3.3.



Gambar 3.3 Flowchart Umum

### 3.2.3 Flowchart Tokenization

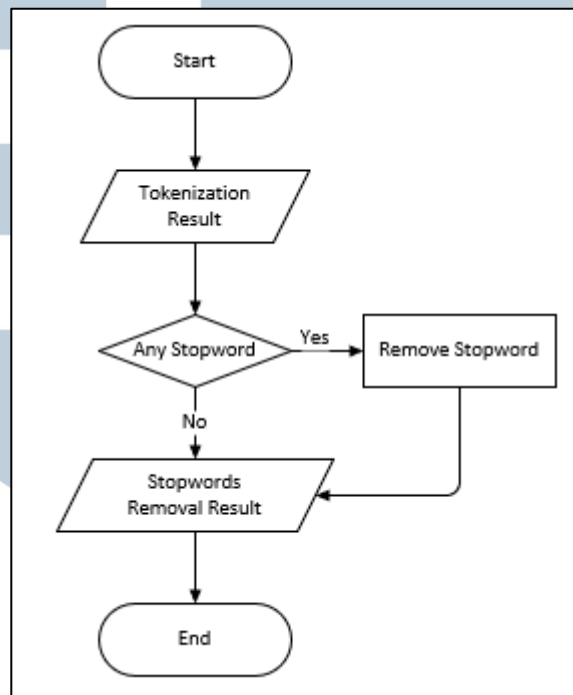
Proses *tokenization* merupakan sebuah proses yang bertujuan untuk menyamaratakan *case* menjadi huruf kecil, menghilangkan tanda baca, simbol-simbol, serta membagi kalimat berdasarkan spasi sehingga diperoleh kata-kata dalam bentuk array. Alur proses seperti pada Gambar 3.4.



Gambar 3.4 Flowchart Tokenization

### 3.2.4 Flowchart Stopwords Removal

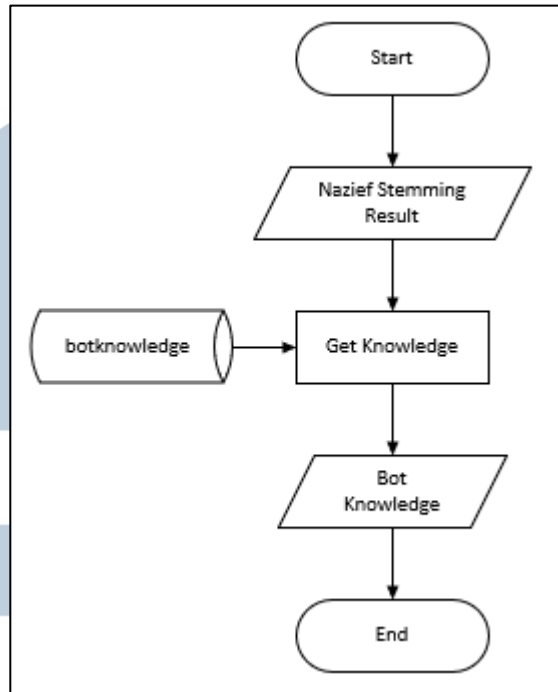
Proses *stopwords removal* berfungsi untuk menghapus atau membuang semua stopword yang ada. Tujuan dihilangkannya stopword ini agar didapatkan kalimat yang lebih sederhana sehingga menampilkan hasil jawaban yang sesuai dengan harapan. Alur proses seperti pada Gambar 3.5.



Gambar 3.5 Flowchart Stopwords Removal

### 3.2.5 Flowchart Get Bot Knowledge

Pada proses *get bot knowledge*, hasil *stemming* menggunakan algoritma Nazief & Andriani akan digunakan untuk mengambil data dari tabel *botknowledge* yang ada di *database*. Ada kemungkinan tidak ditemukan data yang sesuai, untuk itu pada tahap selanjutnya akan dipisahkan apakah terdapat data atau tidak terdapat data. Alur proses seperti pada Gambar 3.6.

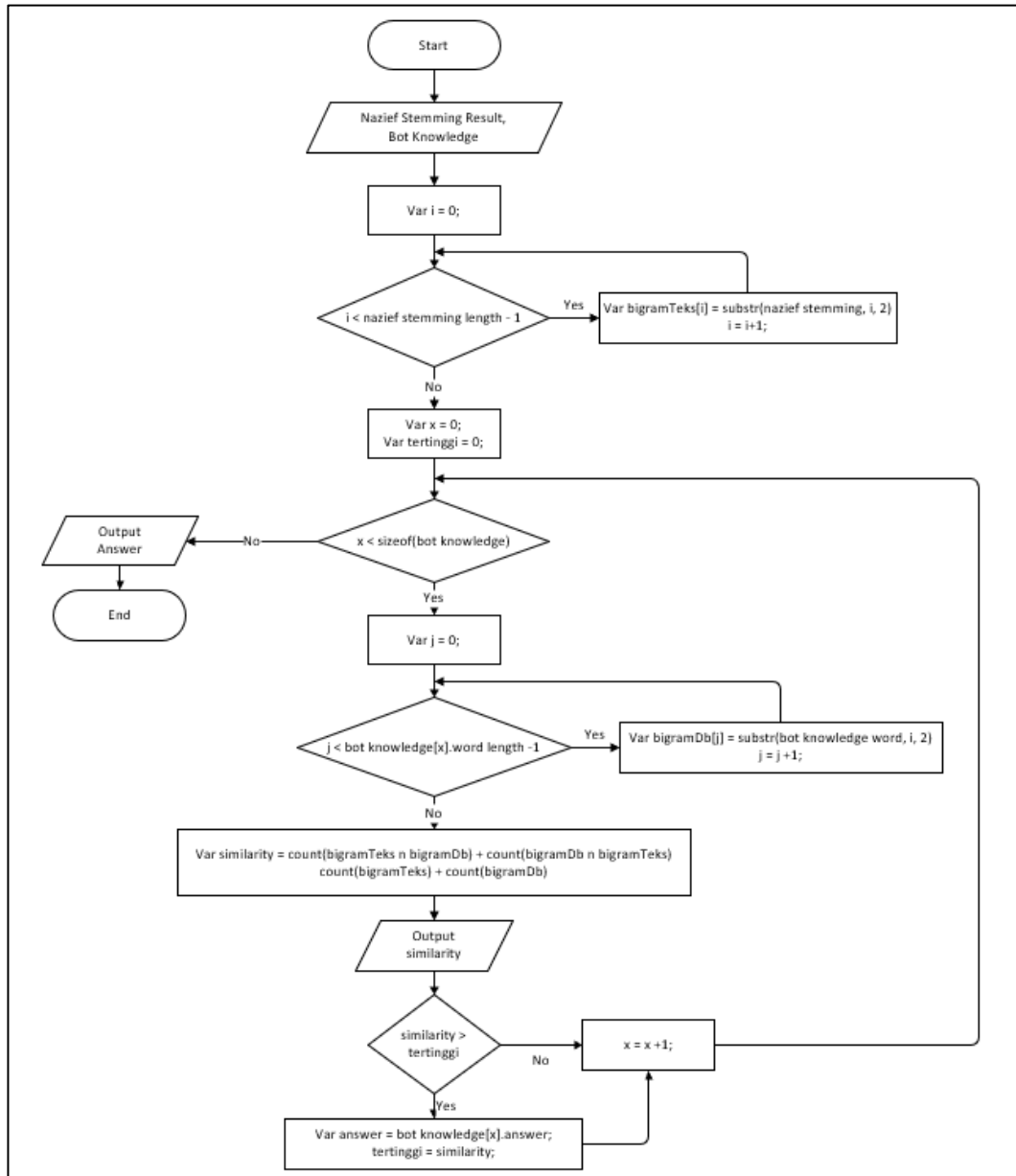


Gambar 3.6 Flowchart Get Bot Knowledge

### 3.2.6 Flowchart Bigram

Bigram merupakan sebuah algoritma untuk memotong 2 karakter dalam suatu string tertentu atau potongan kata dalam suatu kalimat tertentu. Pada proses ini hasil dari *stemming* akan dibagi-bagi sebanyak panjang kalimat *stemming* – 1 lalu diambil 2 karakter dan disimpan ke dalam *variable* bigramTeks dalam bentuk *array*. Setelah itu hasil yang diperoleh dari *bot knowledge* akan diambil satu per satu lalu dibagi-bagi lagi menjadi 2 karakter dan disimpan ke dalam *variable* bigramDb dalam bentuk *array*. Selama masa pengulangan sebanyak *size* dari *bot knowledge*, dilakukan perhitungan *similarity* untuk mendapatkan hasil kesamaan semantik yang tertinggi. Jawaban yang diambil merupakan jawaban dari bot knowledge yang memiliki hasil *similarity* tertinggi antara bigramDb dengan bigramTeks. Alur proses dapat dilihat pada Gambar 3.7.



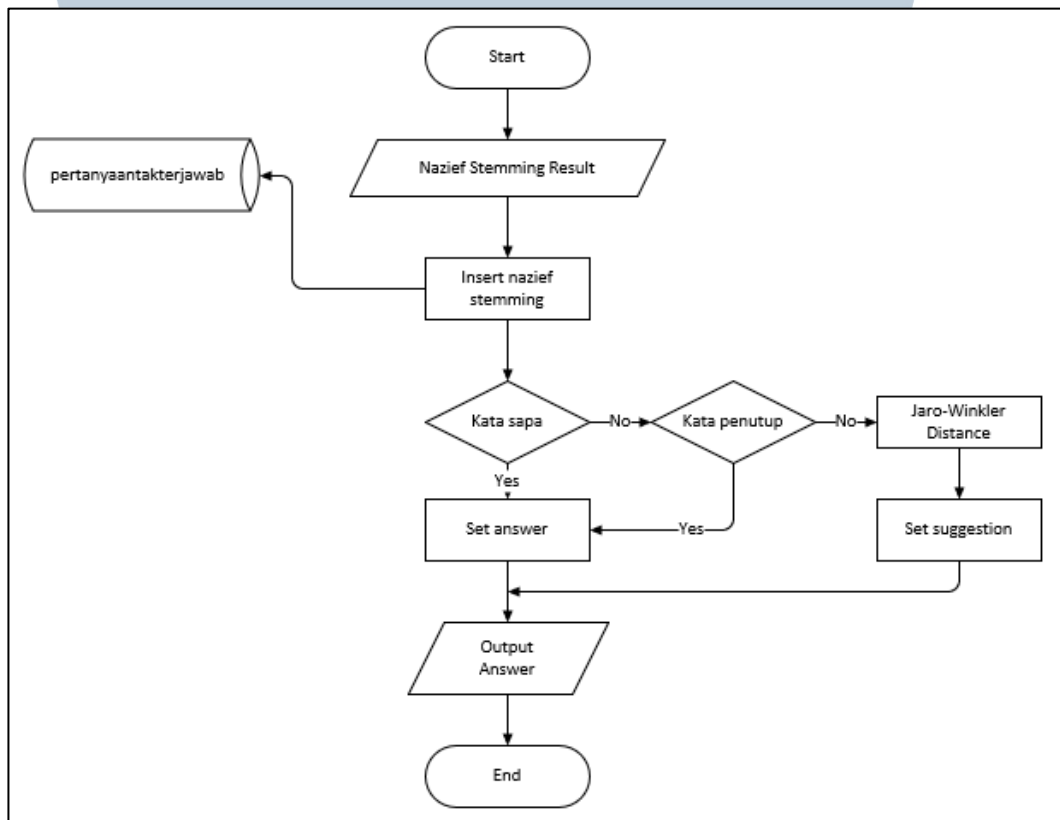


Gambar 3.7 Flowchart Bigram

### 3.2.7 Flowchart Suggestion

Ketika memasuki proses suggestion artinya tidak ada data yang ditemukan di database sehingga hasil dari stemming akan dimasukkan ke dalam tabel pertanyaanterjawab. Setelah itu dilakukan pengecekan apakah hasil stemming merupakan kata sapa, beberapa contoh dari kata sapa seperti “Hai”, “Hallo”, “Pagi”,

“Siang”, “Sore”, “Malam”. Jika benar terdapat kata sapa maka akan di-*set* jawabannya, namun jika tidak dilanjutkan dengan pengecekan lagi terhadap kata penutup, beberapa contoh dari kata penutup yaitu “Terima Kasih”, “Makasih”, “Trims”, “Thx”, “Tq”. Jika benar terdapat kata penutup maka akan di-*set* jawabannya dan jika tidak maka dilakukan pengecekan menggunakan algoritma Jaro-Winkler *Distance* dari library yang sudah ada untuk mendapatkan saran kata. Selanjutnya output dari masing” pengecekan akan dilanjutkan pada tahap selanjutnya. Alur proses dapat dilihat pada Gambar 3.8.



Gambar 3.8 Flowchart Suggestion

### 3.2.8 Struktur Tabel

Database yang digunakan pada penelitian ini adalah PostgreSQL. Database yang dibuat memiliki empat tabel, yaitu botknowledge, dictionary,

pertanyaanterjawab, dan users. Berikut ini akan dijelaskan struktur, serta fungsi dari tabel-tabel yang ada.

1. Nama Tabel : botknowledge  
 Fungsi : menyimpan Frequently Ask Question dari *Student Service* beserta jawabannya sebagai dasar pengetahuan dari *chatbot*. Struktur tabel ditunjukkan pada Tabel 3.1.

Tabel 3.1 Tabel botknowledge

Nama Field	Tipe Data	Keterangan
id	int(11)	Primary key
word	varchar(1000)	Kata-kata kunci pertanyaan yang sudah dilakukan proses <i>text pre-processing</i>
answer	varchar(1000)	Jawaban dari pertanyaan-pertanyaan yang ada

2. Nama Tabel : dictionary  
 Fungsi : menyimpan data *rootword* atau kata dasar yang berguna untuk proses *stemming* menggunakan algoritma Nazief & Andriani. Data pada tabel dictionary ini diambil dari *library* yang sudah ada. Struktur tabel ditunjukkan pada Tabel 3.2.

U N I V E R S I T A S  
 M U L T I M E D I A  
 N U S A N T A R A

Tabel 3.2 Tabel dictionary

Nama Field	Tipe Data	Keterangan
id_word	int(11)	Primary key
word	varchar(1000)	Kata dasar atau <i>rootword</i>
stopword	enum('Ya', 'Bukan', '')	Daftar <i>stopword</i>
category	enum('kapabilitas', 'integritas', '')	Kategori <i>stopword</i> , tetapi tidak digunakan untuk proses <i>stemming</i>
sentiment	enum('positif', 'negatif', 'netral', '')	Menunjukkan sifat kata, tetapi tidak digunakan untuk proses <i>stemming</i>

3. Nama Tabel : pertanyaantakterjawab

Fungsi : menyimpan daftar pertanyaan yang datanya tidak

ditemukan dalam *database*. Struktur tabel ditunjukkan pada

Tabel 3.3.

Tabel 3.3 Tabel pertanyaantakterjawab

Nama Field	Tipe Data	Keterangan
id	int(11)	Primary key
signature	varchar(64)	Tanda atau kunci dari suatu event
events	text	Paket event yang dikirimkan user kepada chatbot
timestamp	timestamp	Tanggal dan waktu ketika event dikirimkan

4. Nama Tabel : users

Fungsi : menyimpan data *profile user* yang sudah berteman dengan *chatbot*. Struktur tabel ditunjukkan pada Tabel 3.4.

Tabel 3.4 Tabel users

Nama Field	Tipe Data	Keterangan
id	int(11)	Primary key
user_id	varchar(100)	Id user yang dikirim dari LINE
display_name	varchar(100)	Nama dari user sesuai dengan nama <i>profile</i> di LINE

UMMN

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA