BAB III

METODOLOGI PENELITIAN DAN PERANCANGAN SISTEM

3.1 Metodologi Penelitian

Dalam penelitian ini, terdapat beberapa tahap yang digunakan dalam perancangan dan pembangunan sistem rekomendasi restoran menggunakan metode AHP dan VIKOR pada *platform* LINE, yaitu studi literatur, perancangan dan pembangunan aplikasi, pengujian aplikasi, dan evaluasi.

1. Studi Literatur

   Dalam studi literatur, pembelajaran terhadap teori-teori dan metode yang berhubungan dengan perancangan dan pembangunan sistem rekomendasi restoran menggunakan metode AHP dan metode VIKOR pada *platform* LINE. Teori dan metode yang dimaksud adalah studi literatur, LINE *Message API*, metode AHP, metode VIKOR, dan SUS.

2. Perancangan dan Pembangunan Aplikasi

   Tahap dalam perancangan dan pembangunan aplikasi dilakukan dengan merancang dan membangun sistem rekomendasi restoran menggunakan metode AHP dan metode VIKOR dalam bahasa pemrograman PHP. Data restoran yang dimaksud akan diambil dari *website* Zomato. Setelah itu akan diimplementasikan kedalam *platform* LINE beserta fitur-fitur yang akan dibangun.

3. Pengujian Aplikasi

   Tahap pengujian aplikasi ini dilakukan dengan menambahkan akun LINE sebagai teman, dimana akun LINE tersebut sudah diimplementasikan dengan

sistem rekomendasi restoran yang telah dibuat. Selanjutnya, *user* diberikan kuesioner untuk menguji kepuasan pengguna.

4. Evaluasi

   Tahap evaluasi akan dilakukan dengan menganalisis hasil kuesioner yang didapat dari pengujian aplikasi.

3.2 Teknik Pengumpulan Data

Data yang digunakan dalam penelitian ini adalah data restoran yang terdapat di *website* Zomato. Data tersebut akan diambil menggunakan API yang disediakan oleh Zomato kepada pengembang dengan format JSON (*JavaScript Object Notation*). Berikut potongan format JSON data restoran pada Gambar 3.1.


```
"restaurants": [
 {
   "restaurant": {
     "R": {
       "res_id": 7424239
     },
     "apikey": "acbcab2eb2dae8e1f46810ec3c14ed55",
     "id": "7424239",
     "name": "Ayam Mercon Kongko2",
     "url": "https://www.zomato.com/jakarta/ayam-mercon-kongko2-serpong-utara?utm_source=api_basic_user&utm_medium=api&u
     "location": {
       "address": "Summarecon Digital Center Mall, Garden Walk, Jl. Scientia Boulevard Gading, Serpong Utara, Tangerang"
       "locality": "Summarecon Digital Center Mall, Serpong",
       "city": "Tangerang",
       "city_id": 74,
       "latitude": "-6.2567010000",
       "longitude": "106.6163520000",
       "zipcode": "",
       "country_id": 94,
       "locality_verbose": "Summarecon Digital Center Mall, Serpong, Tangerang"
```


Gambar 3.1 Format JSON Data Restoran

Berdasarkan data yang diambil pada Gambar 3.1, berikut penjelasan mengenai data apa saja yang digunakan dalam merancang dan membangun sistem rekomendasi restoran.

Tabel 3.1 Data *requirement*

#	Key	Tipe Data	Deskripsi
1	name	String	Nama restoran
2	url	String	Link URL halaman restoran di Zomato
3	locality	String	Nama daerah restoran
4	latitude	Number	Titik koordinat lintang restoran

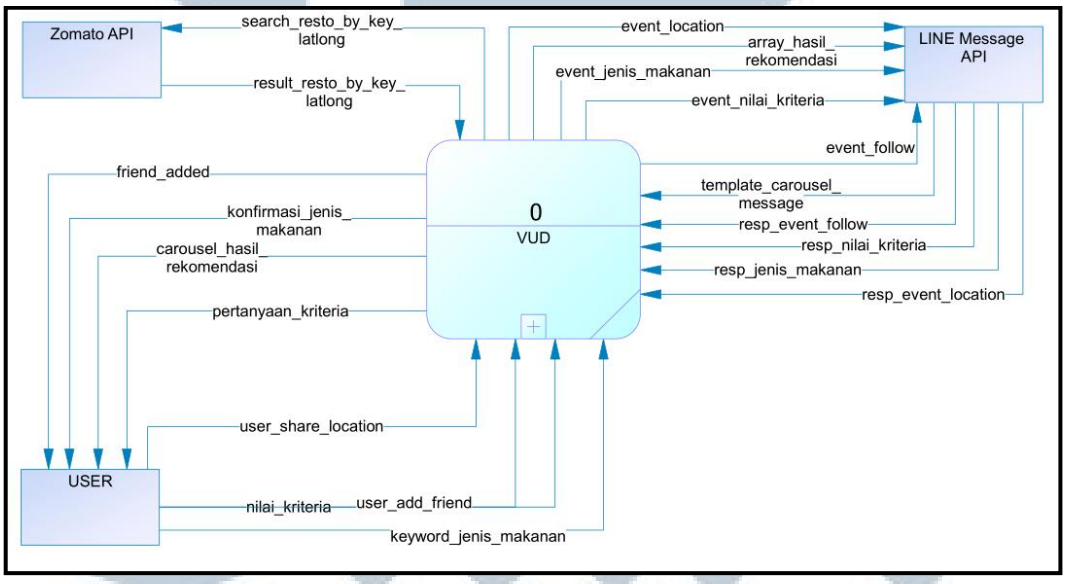
Tabel 3.1 Data *requirement* (Lanjutan)

#	Key	Tipe Data	Deskripsi
5	longitude	Number	Titik koordinat bujur restoran
6	average_cost_for_two	Integer	Harga rata-rata makanan untuk dua orang
7	aggregate_rating	String	Nilai rata-rata rating restoran
8	featured_image	String	Link URL gambar utama restoran

3.3 Perancangan Sistem

Perancangan sistem yang dibuat dalam penelitian ini meliputi *Data Flow Fiagram* (DFD), *Flowchart*, *Database Schema*, Struktur Tabel, dan rancangan tampilan antarmuka yang menggambarkan tampilan dari aplikasi yang akan dibangun.

3.3.1 Data Flow Diagram



Gambar 3.1 Context Diagram Sistem Rekomendasi Restoran

Pada Gambar 3.1 menjelaskan tentang diagram konteks dari sistem rekomendasi restoran. Terdapat tiga entitas, yaitu entitas *User* yang merupakan pengguna dari sistem, entitas Zomato API yang merupakan sumber data restoran, dan LINE *Message* API yang berfungsi untuk menerima *event* dari *user* serta

menampilkan hasil rekomendasi ke dalam *platform* LINE. Pada proses sistem rekomendasi restoran terdapat sembilan proses turunan, yaitu *Receive Events*, *Add Friend*, Tampil Pertanyaan, *Update Data*, *Save* Jenis Makanan, *Share Location*, *Fetch dFta Zomato*, Metode AHP dan VIKOR, dan C*arousel Message*.

Berdasarkan Gambar 3.2 dapat dijelaskan mengenai aliran data yang terjadi antar tiap proses dan entitas. Saat entitas *user* melakukan proses *Add Friend*, *user* mengirimkan data berupa *user_add_friend* ke proses tersebut, selanjutnya proses *Add Friend* mengirimkan data *event_follow* ke entiras LINE *Message* API yang dimana data tersebut dikirimkan ke proses *Receive Event* dan kemudian menyimpannya ke dalam *data store eventlogs*. Saat waktu yang bersamaan proses *Add Friend* menyimpan data-data *user* ke dalam *data store users* dan kemudian proses tersebut mengirimkan data ke *user* berupa *friend_added*.

Pada proses Tampil Pertanyaan, proses tersebut menerima *data_pertanyaan* yang dikirimkan dari *data store questions* yang kemudian proses tersebut mengiriman data ke *user* berupa *pertanyaan_kriteria*. Ketika user melakukan proses *Update Data*, proses tersebut menerima data *nilai_k*riteria yang dikirimkan *user* dan kemudian meneruskannya ke entitas LINE *Message* API berupa *event_nilai_kriteria* untuk disimpan sebagai *logs* pada proses *receive events*. Proses *Update Data* juga mengirimkan data berupa *data_nilai_kriteria* ke *data store users* untuk disimpan.

Ketika user melakukan proses *Save* Jenis Makanan, *user* mengirimkan data berupa *keyword_jenis_makanan* ke proses *Save* Jenis Makanan dan kemudian mengirimkannya ke LINE *Message* API untuk diteruskan ke proses *Receive Events* dengan data *event_jenis_makanan* yang nantinya akan disimpan ke dalam *data*

store eventlogs berupa *RE_jenis_makanan*. Proses *save* jenis makanan melakukan penyimpan data berupa *data_jenis_makanan* kedalam *data store users* dan proses tersebut mengirimkan data ke entitas *user* berupa *konfirmasi_jenis_makanan*. Proses selanjutnya adalah *share location*, dimana proses ini *user* mengirimkan data berupa *user_share_location* dan proses tersebut mengirimkan data *event_location* ke LINE *Message* API untuk di teruskan ke proses *receive events* untuk disimpan ke *data store eventlogs* berupa data *RE_event_location*. Selanjutnya pada proses *Share Location*, proses tersebut mengirimkan data *send_lat_long* ke proses *fetch data zomato*. Data tersebut berupa nilai dari titik koordinat *latitude* dan *longitude*.

Proses selanjutnya adalah *Fetch Data Zomato*, dimana proses tersebut menerima data dari proses *Share Location* yaitu titik koordinat *latitude* dan *longitude* berupa *send_lat_long* dan data dari *data store users* yaitu *data_jenis_makanan*. Setelah menerima data-data tersebut, proses *Fetch Data Zomato* mengirimkan data ke entitas Zomato API berupa *search_resto_by_key_latlong* yaitu berfungi untuk mencari daftar restoran berdasarkan jenis makanan dan titik koordinat *latitude longitude*.

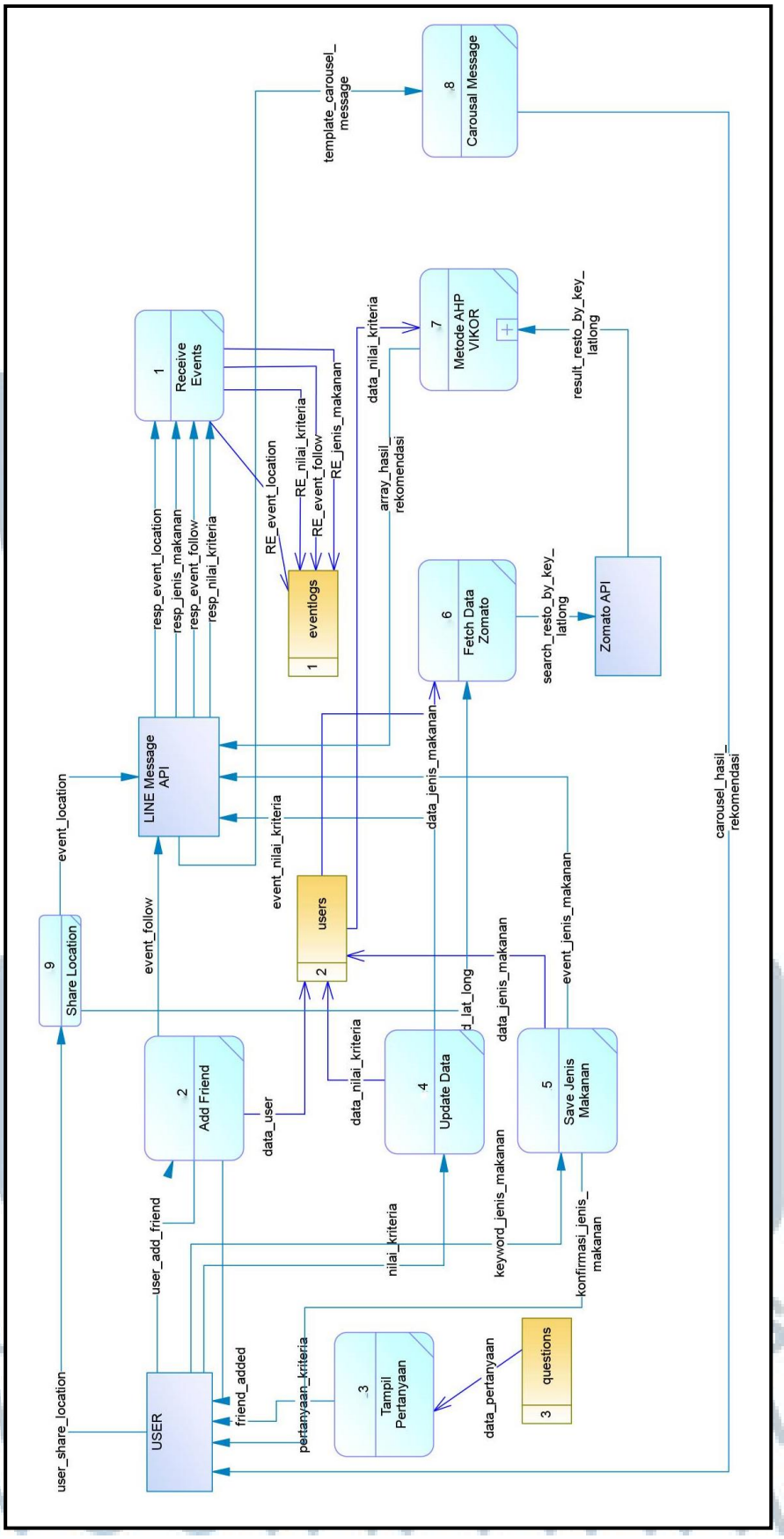
Pada entitas Zomato API akan mengirimkan data hasil pencariam retoran berdasarkan jenis makanan dan titik koordinat *latitude longitude* ke proses Metode AHP dan VIKOR berupa *result_resto_by_key_latlang*. Proses Metode AHP dan VIKOR juga menerima data untuk melakukan proses rekomendasi berupa *data_nilai_kriteria*. Setelahnya proses tersebut mengirimkan data *array_hasil_rekomendasi* ke entitas LINE *Message* API. Entitas LINE *Message* API mengirimkan data *template_carousel_message* ke proses *carousel message* dan meneruskannya ke *user* berupa data *carousel_hasil_rekomendasi*. Data tersebut

berisikan hasil rekomendasi restoran berupa *carousel message* ke *platform* LINE *user*.

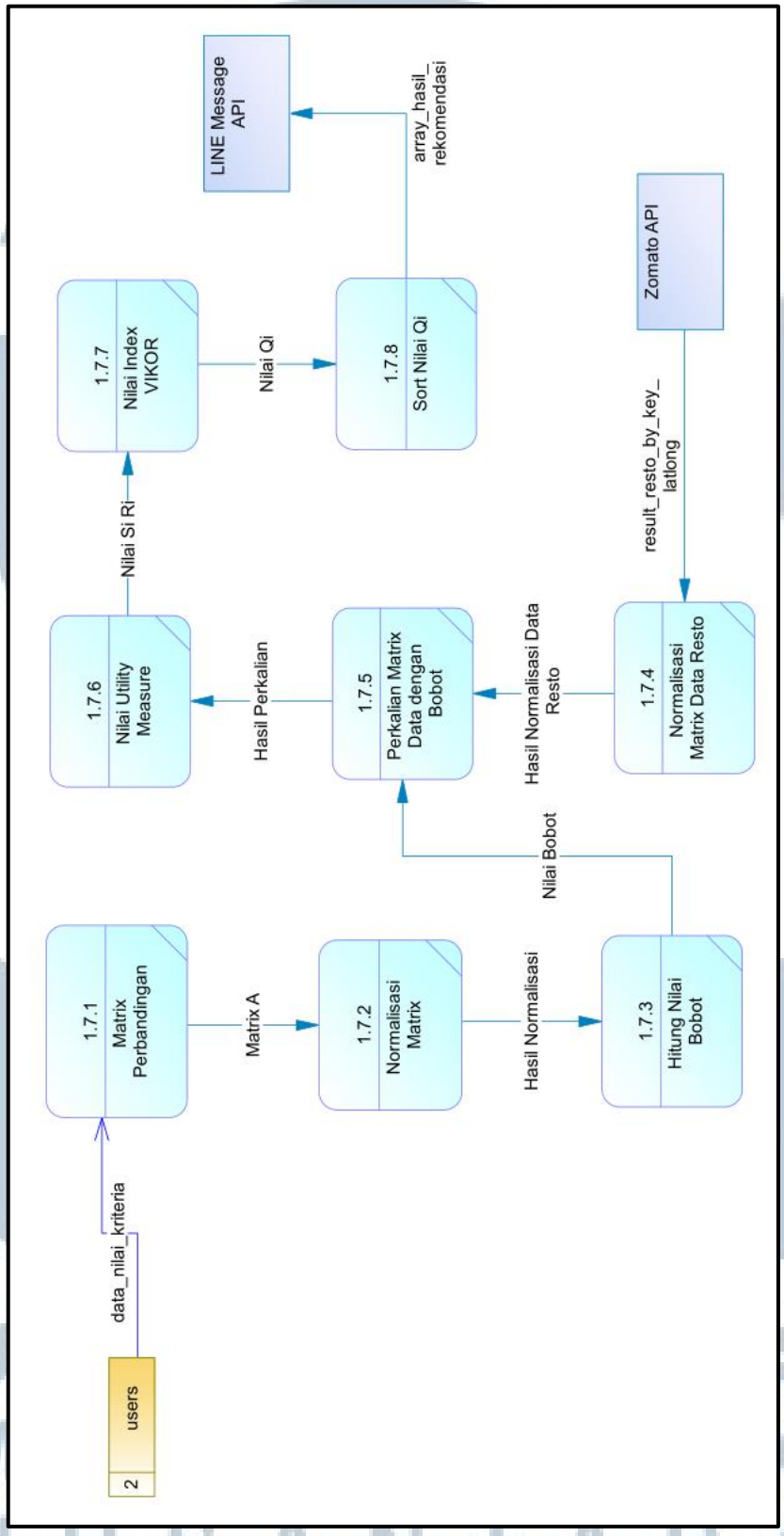
Gambar 3.3 merupakan proses turunan dari proses Metode AHP dan VIKOR dimana dalam proses turunan ini terdapat delapan proses yag saling berubungan satu dengan lainnya. *Data store users* mengirimkan data, yaitu *data_nilai_kriteria* ke proses Matriks Perbandingan. Selanjutnya proses tersebut mengirimkan Matrix A ke proses Normalisasi Matrix. Hasil dari normalisasi tersebut akan di hitung nilai bobotnya di proses Hitung Nilai Bobot.

Pada proses Normalisasi Matrix Data Resto, proses tersebut menerima data dari Zomato API berupa *result_resto_by_key_latlong*, kemudian hasil normalisasi terhadap data yang diterima tersebut dikirim ke proses Perkalian Matriks dengan nlai bobot. Nilai bobot tersebut didapatkan dari proses Hitung Nilai Bobot. Selanjutnya hasil perkalian tersebut dikirim ke proses Nilai *Utility Measure* yang berfungsi untuk menghitung nilai Si dan Ri.

Nilai Si dan Ri dari proses Nilai *Utility Measure* dikirimkan ke proses Nilai *Index* VIKOR, dimana proses tersebut bertujuan untuk menghitung nilai Qi. Selanjutnya nilai Qi tersebut di *sort* dari nilai terkecil sampai terbesar dalam proses S*ort* Nilai Qi. Setelah itu hasil *sort* terserbut berupa *array_hasil_rekomendasi* dikirimkan ke entitas Line *Message* API.



Gambar 3.2 DFD Level 1 Sistem Rekomendasi Restoran



Gambar 3.3 DFD Level 2 Metode AHP VIKOR

3.3.2 Flowchart

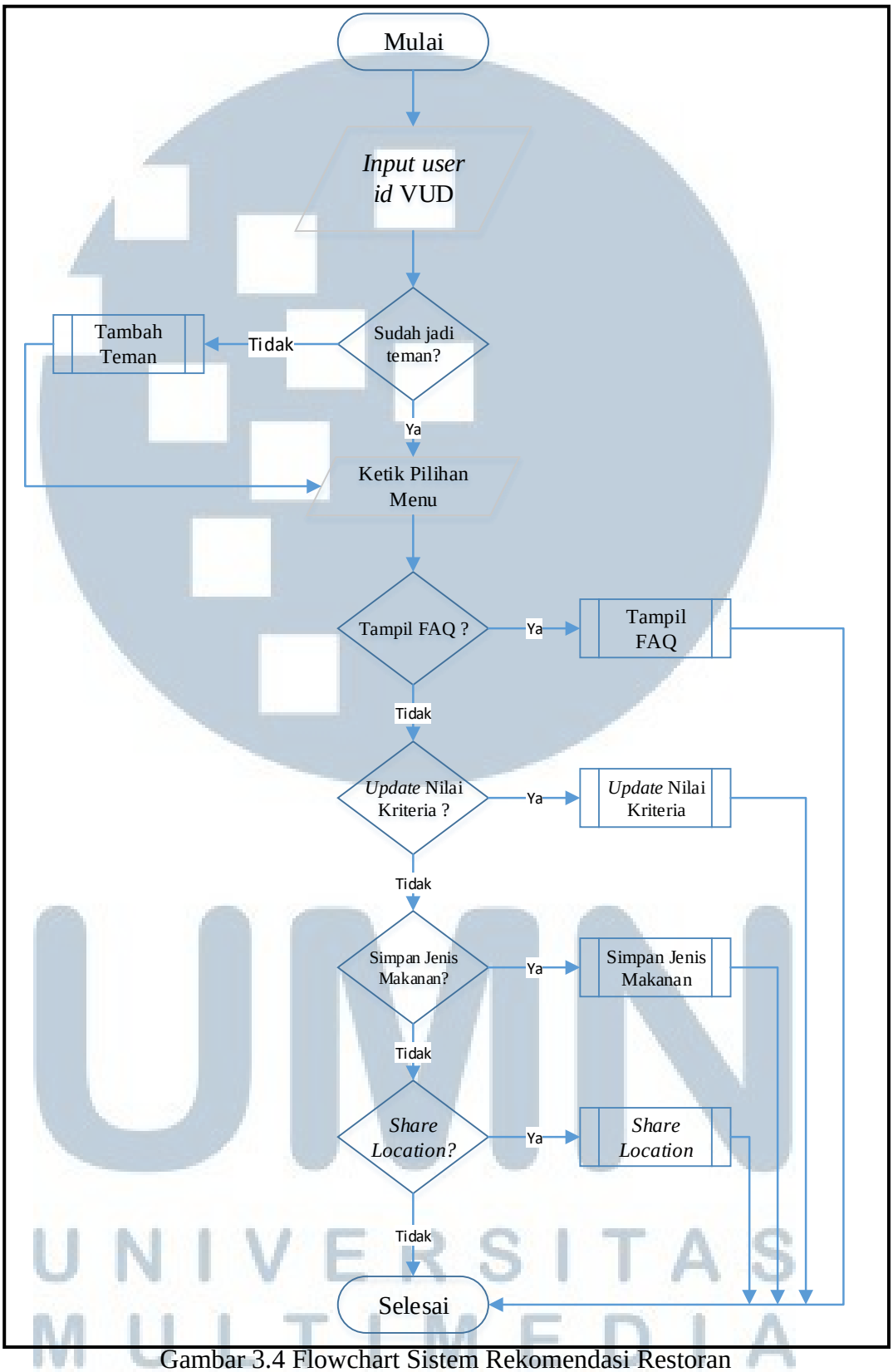
Flowchart pada sistem rekomendasi restoran terbagi menjadi 8 bagian, yaitu.

A. Flowchart Sistem Rekomendasi Restoran

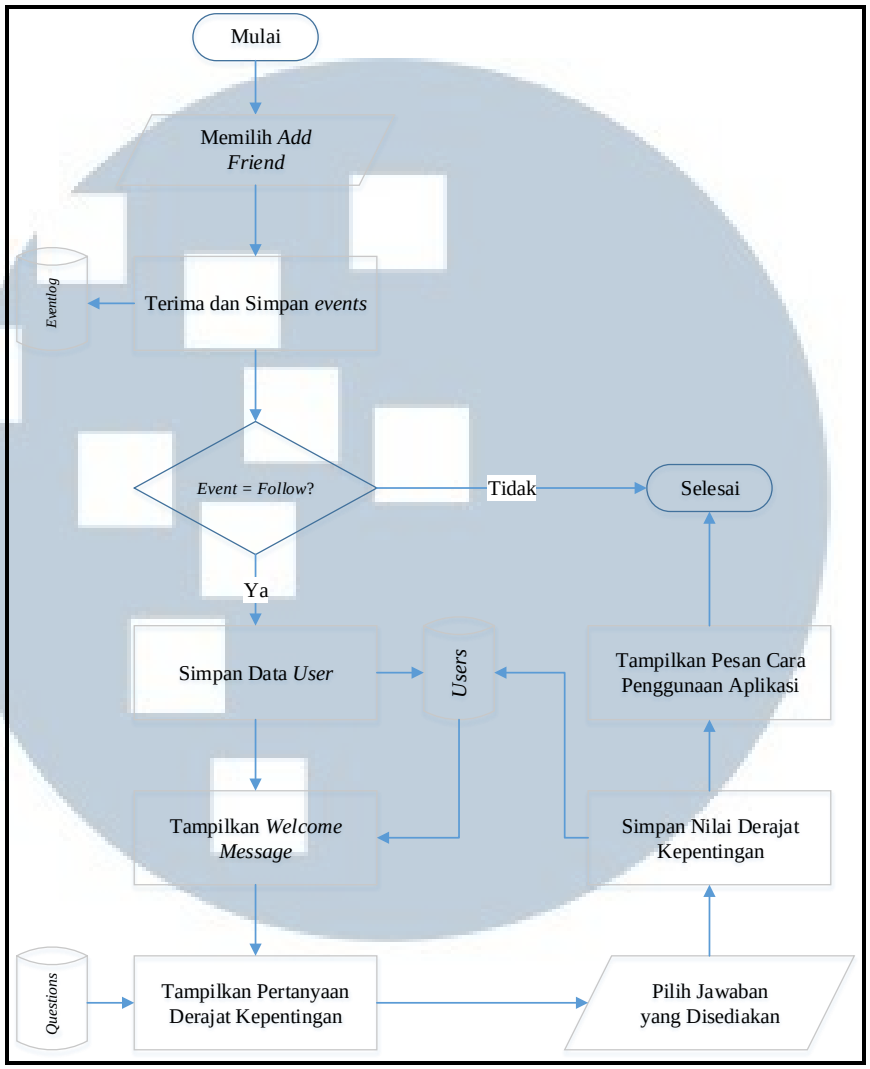
Berdasarkan Gambar 3.4 dapat dijelaskan bahwa untuk menjalankan sistem rekomendasi restoran, hal pertama yang dilakukan adalah meng-*input user id* dari akun VUD. Jika akun VUD sudah menjadi teman, maka selanjutnya dapat menggunakan fitur-fitur yang disediakan. Jika belum berteman dengan akun VUD, maka akan dilanjutkan dengan sub-proses Tambah Teman.

B. Flowchart Tambah Teman

Berdasarkan Gambar 3.5, untuk menggunakan fitur yang disediakan hal pertama yang dilakukan adalah menambahkan teman dengan memilih *add friend*. Selanjutnya pada proses Terima dan Simpan *events* berfungsi untuk menangani *event* apa saja yang dikirim *user* ke *platform* LINE dan kemudian menyimpannya ke dalam *database* sebagai *logs*. Jika *events* yang di terima adalah *follow*, data *user* akan di simpan ke dalam *database users.* Setelah itu akan ditampilkan *welcome message* beserta pertanyaan untuk menentukan nilai derajat suatu kriteria, dimana pertanyaan tersebut di ambil dari *database questions*. Proses selanjutnya adalah memilih jawaban dari pertanyaan tersebut dan menyimpannya ke dalam *database users,* kemudian menampilkan panduan penggunaan aplikasi.



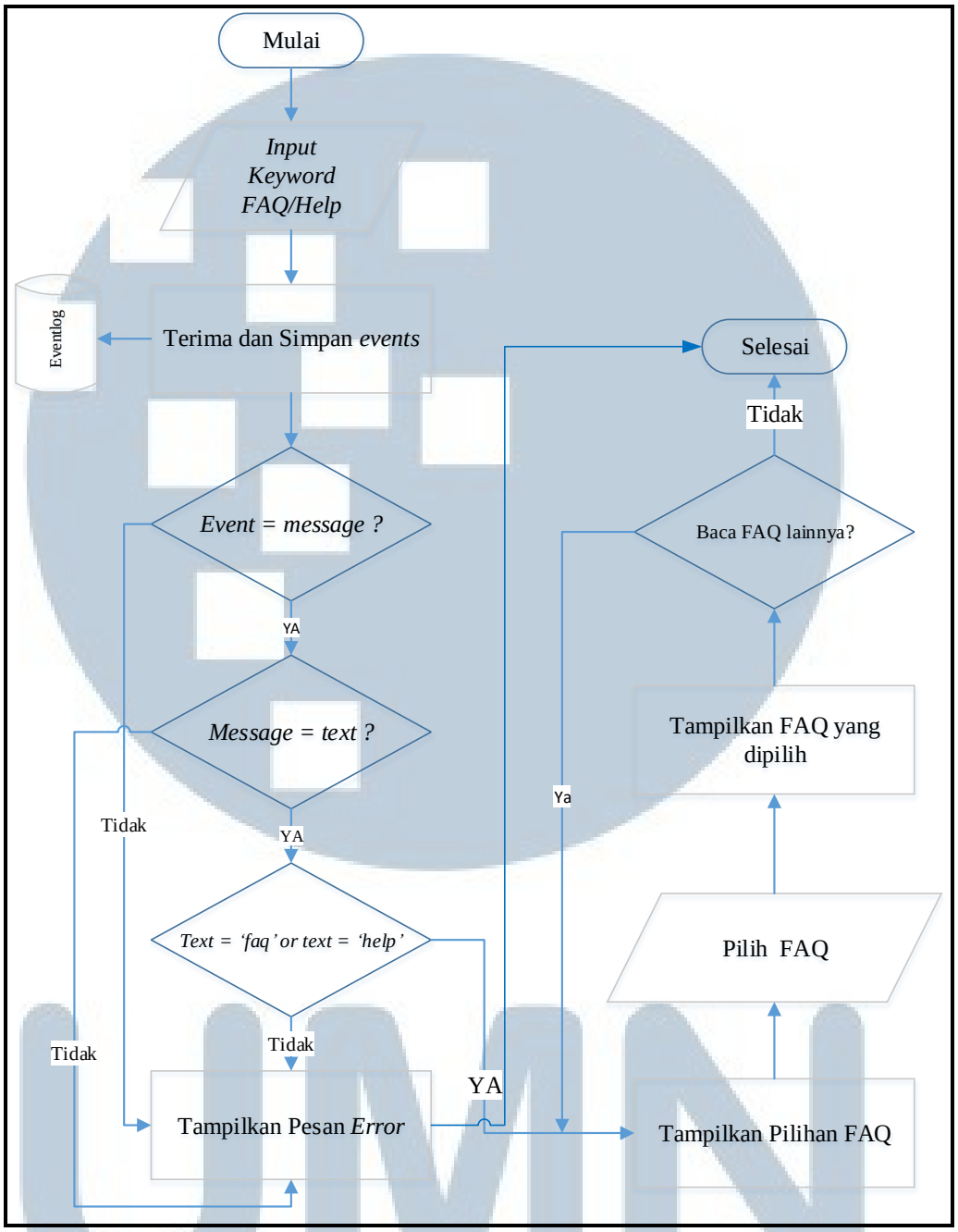
Gambar 3.4 Flowchart Sistem Rekomendasi Restoran



Gambar 3.5 Flowchart Tambah Teman

C. Flowchart Tampil FAQ

Pada Gambar 3.6, untuk menampilkan FAQ langkah pertama adalah memasukkan *keyword help* atau FAQ dan kemudian *events* yang di terima tersebut dilihat jika *events*-nya adalah *message* dengan tipe *text* dan *keyword* yang dimasukkan adalah *help* atau FAQ maka akan menampilkan pilihan FAQ. Proses selanjutnya adalah memilih FAQ yang akan di baca dan menampilkan isi dari FAQ tersebut. Setelah itu, jika ingin membaca FAQ lainnya maka akan ditampilkan pilihan FAQ. Jika *keyword* yang dimasukkan tidak sesuai maka akan menampilkan pesan *error*.

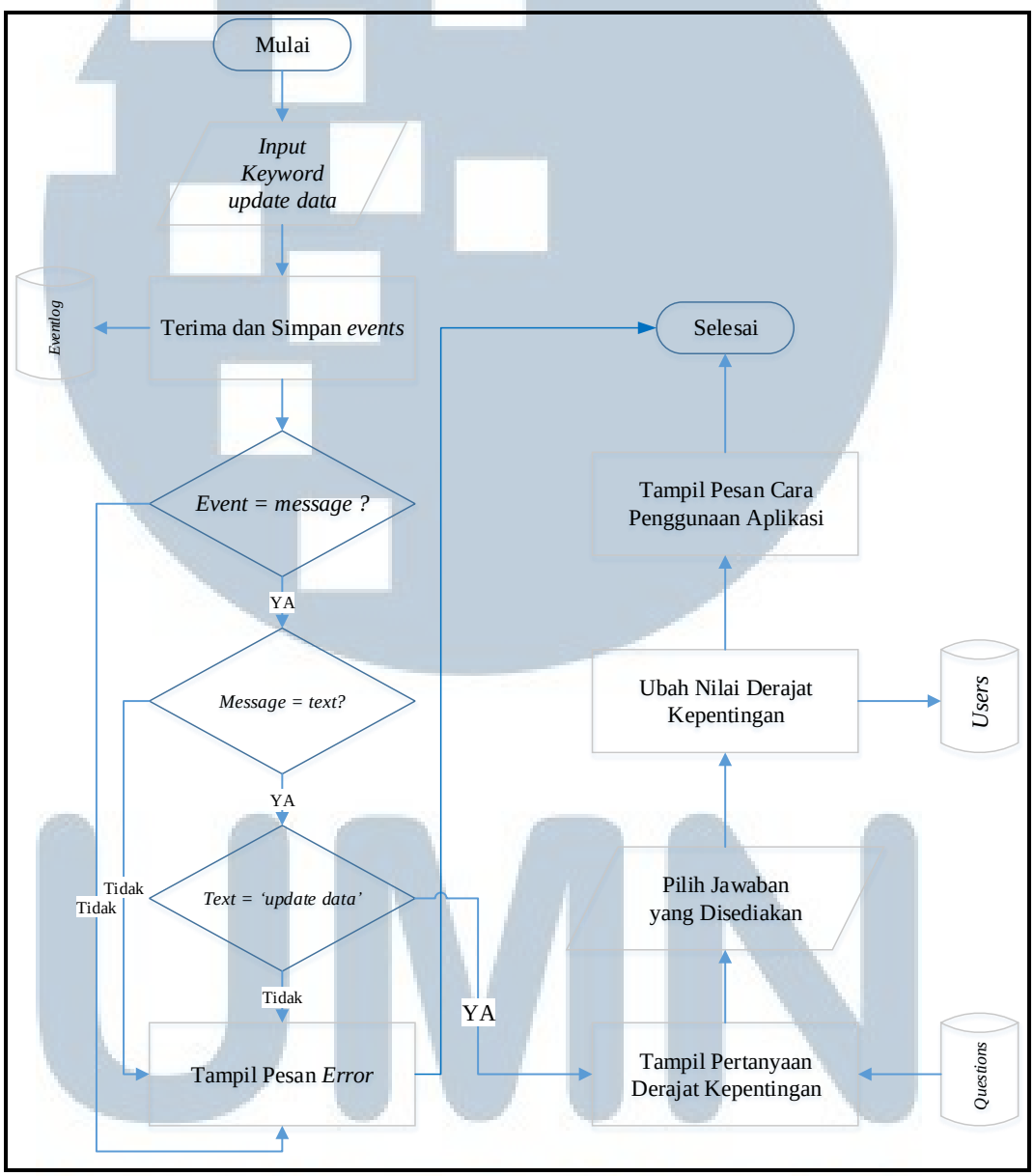


Gambar 3.6 Flowchart Tampil FAQ

D. Flowchart Update Nilai Kriteria

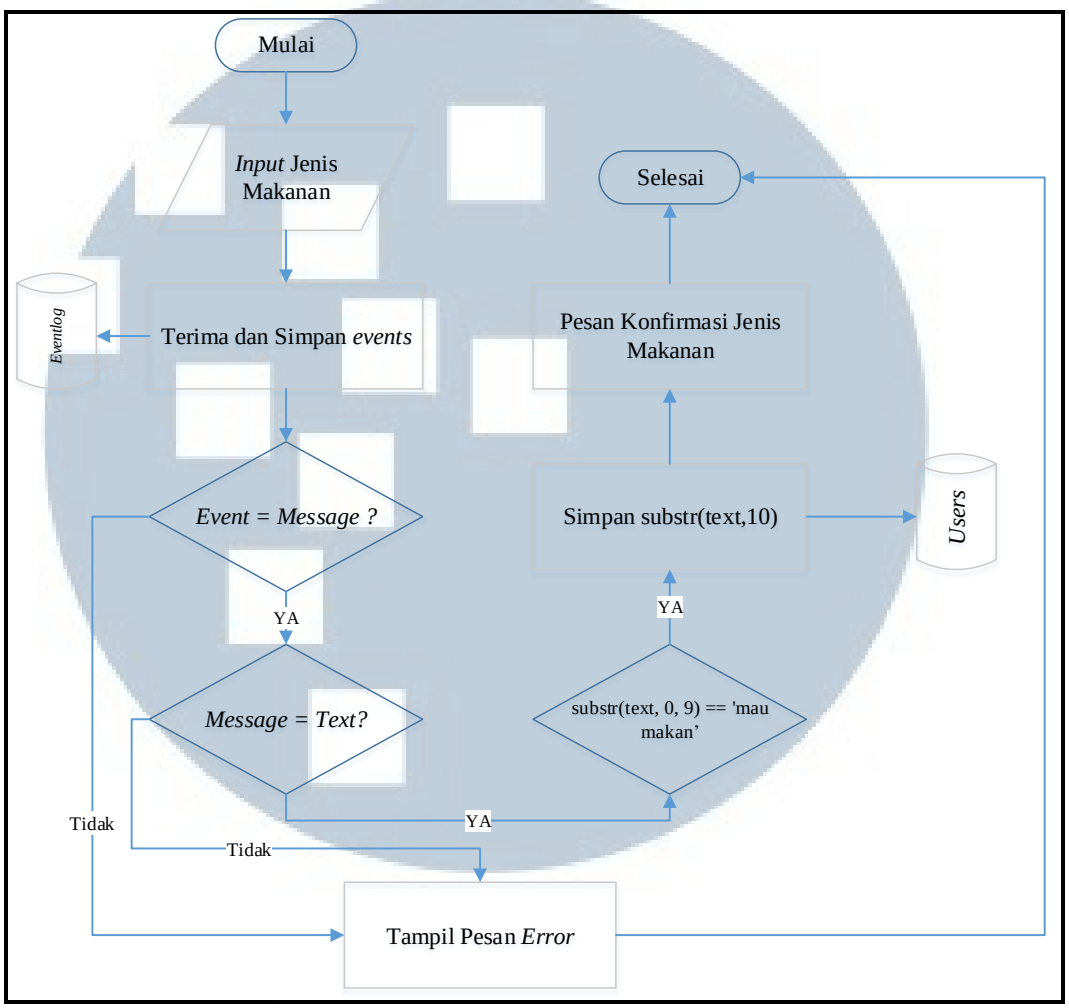
Berdasarkan Gambar 3.7, untuk merubah nilai kriteria dapat dilakukan dengan memasukkan *keyword* update data dan kemudian pada proses *Receive Events* akan dilihat jika *events* yang dikirim berupa *message* dengan tipe *text* dan *keyword* yang dimasukkan benar maka akan ditampilkan pertanyaan yang diambil

dari *database questions*. Selanjutnya adalah proses memilih jawaban yang disediakan dan kemudian merubah nilai kriteria pada *database* sesuai dengan pilihan jawaban tersebut.



Gambar 3.7 Flowchart Update Nilai Kriteria

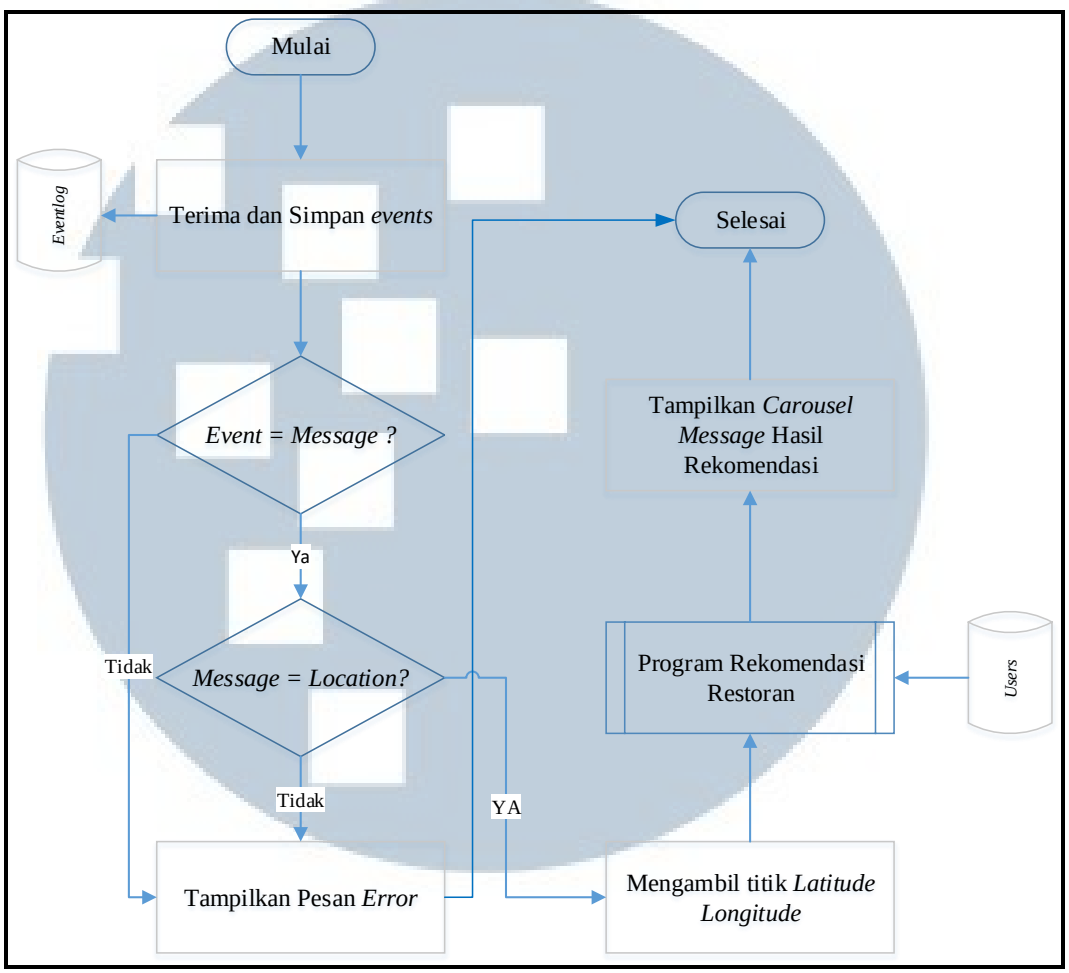
E. Flowchart Simpan Jenis Makanan



Gambar 3.8 Flowchart Simpan Jenis Makanan

Berdasarkan Gambar 3.8, dimasukkan *keyword* mau makan diikuti jenis makanan yang akan dicari. Selanjutnya, jika *events* yang masuk berupa *message* dengan tipe *text* dan *keyword* pada kata ke-1 sampai 9 adalah mau makan maka kata ke-10 dan seterusnya akan disimpan pada *database users*.

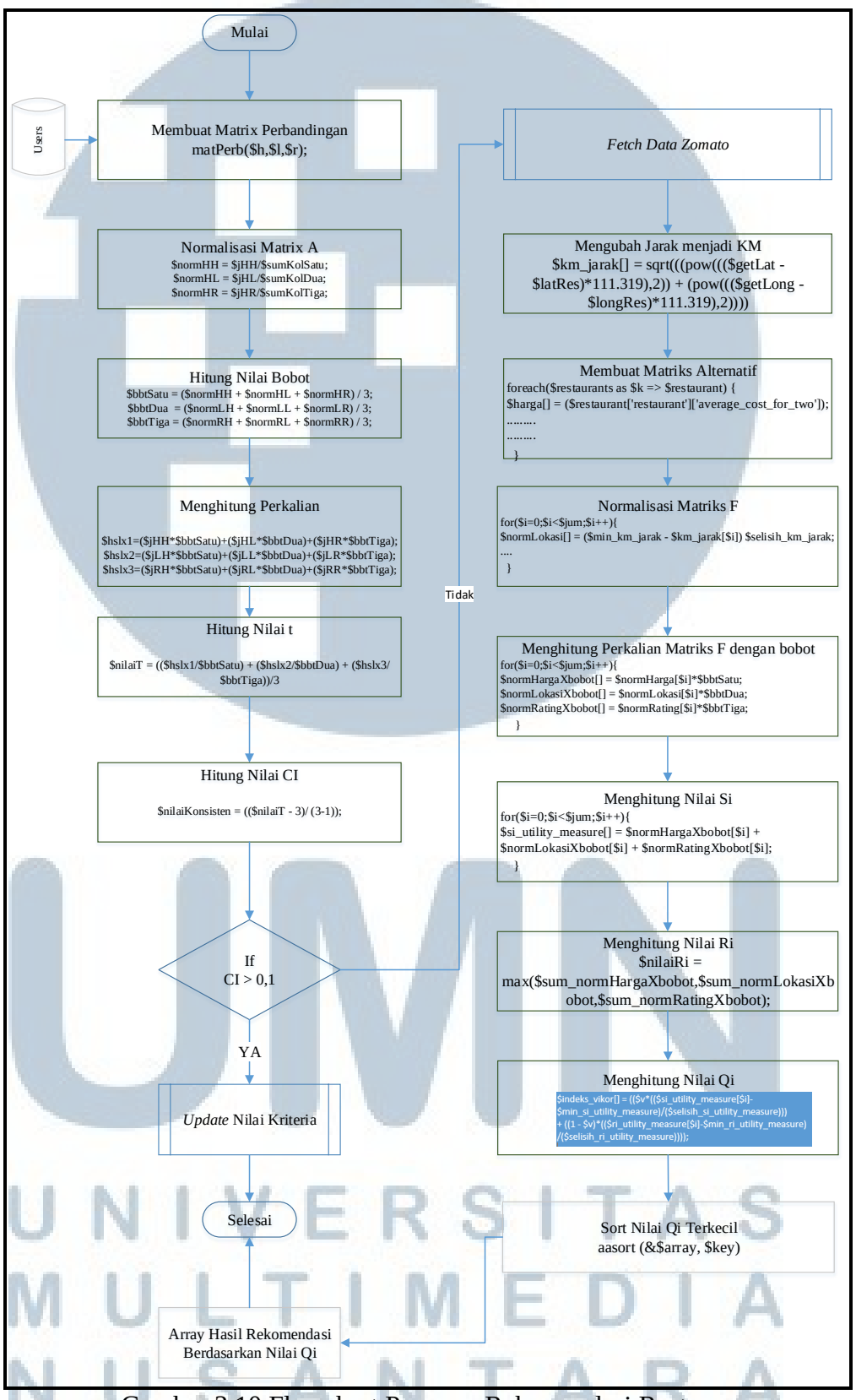
F. Flowchart Share Location



Gambar 3.9 Flowchart Share Location

Berdasarkan Gambar 3.9, saat *events* yang diterima adalah *message* dengan tipe *location* maka proses yang akan dilakukan selanjutnya adalah mengirim nilai koordinat *latitude* dan *longitude* ke proses pembuatan rekomendasi restoran. Setelah itu aka ditampilkan hasil rekomendasi restoran berupa *carousel message* di *platform* LINE.

G. Flowchart Program Rekomendasi Restoran



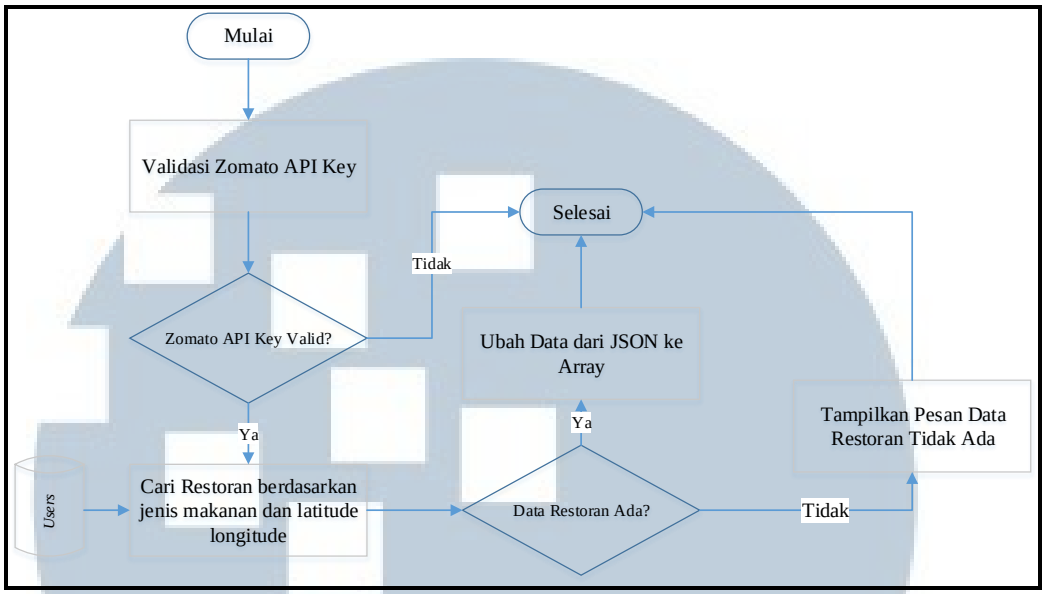
Gambar 3.10 Flowchart Program Rekomendasi Restoran

Gambar 3.10 merupakan *flowchart* dari program dalam merekomendasikan restoran. Dalam membuat matrix perbandingan berdasarkan nilai yang diambil dari *database users* yaitu nilai derajat kepentingan. Setelahnya, matrix tersebut dinormalisasi dan kemudian menghitung nilai bobotnya. Proses selanjutnya adalah mengkalikan matrix perbandingan tersebut dengan nilai bobot yang didapat dan kemudian menghitung nilai t dari hasil perkalian matrix tersebut. Jika nilai t > 0,1 maka akan dijalankan proses *update* nilai kriteria. Jika nilai t < 0,1, maka proses yang dilakukan adalah data restoran menggunakan Zomato API berdasarkan jenis makanan dan titik koordinat *latitude* dan *longitude*.

Berdasarkan data yang sudah diambil, titik koordinat *latitude* dan *longitude* diubah satuannya menjadi KM. Selanjutnya, membuat matrix alternatif dari data yang diambil kemudian menormalisasikan matrix tersebut. Selanjutnya hasil matrix normalisasi dikalikan dengan nilai bobot yang sudah didapat sebelumnya. Proses selanjutnya adalah menghitung nilai Si, Ri, dan Qi. Setelah itu mengurutkan nilai Qi dari yang terkecil dan merubahnya dalam bentuk format *array*.

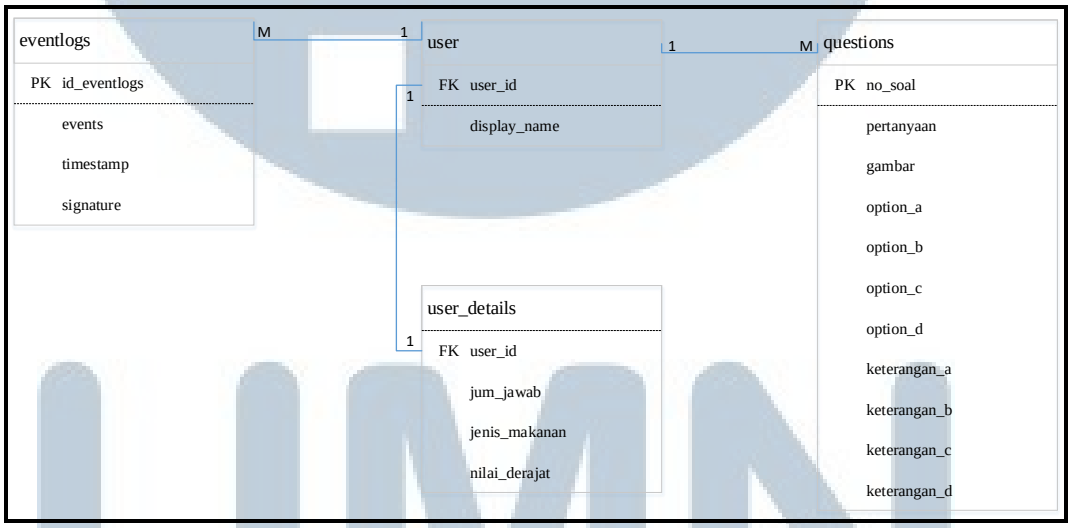
H. Flowchart Fetch Data Zomato

Pada Gambar 3.11, untuk mengambil data restoran menggunakan Zomato API, proses yang harus dilakukan adalah memvalidasi *Zomato API Key*. Jika *Zomato API Key* tersebut benar maka akan dilakukan proses pencarian restoran berdasarkan jenis makanan yang diambil pada *database users* dan titik koordinat *latitude* dan *longitude* dari *events location*. Jika data yang dicari tersedia, maka diubah formatnya dari JSON ke *array*. Jika data restoran yang dicari tidak tersedia, maka akan ditampilkan pesan bahwa data restoran yang dicari tidak tersedia.



Gambar 3.11 Flowchart Fetch Data Zomato

3.3.3 Database Schema



Gambar 3.2 Database Schema Sistem Rekomendasi Restoran

Berdasarkan Gambar 3.12, tabel *eventlogs* berfungsi untuk menyimpan semua *events* yang dikirim *user* ke *platform* LINE. Data yang disimpan yaitu *events* yang berisikan *event* tiap *message* yang dikirim *user*, *timestamp* merupakan waktu *event* tersebut di terima, dan *signature* yang digunakan untuk menvalidasi bahwa pesan yang dikirim *user* berasal dari *platform* LINE. Pada tabel *users* berfungsi untuk menyimpan data *user* yang sudah menjadi teman di *platform* LINE. Data yang

dimaksud yaitu nama di *platform* LINE disimpan di kolom *display_name*. Pada tabel *user_details* berisikan informasi tambahan dari *user,* seperti *jum_jawab* yaitu menyimpan informasi mengenai jumlah pertanyaan yang sudah dijawab. Kolom *jenis_makanan* berisikan jenis makanan yang akan di cari *user*, dan kolom *nilai_kriteria* merupakan nilai dari hasil jawaban *user* saat menjawab pertanyaan yang diberikan.

Tabel *questions* berisikan pertanyaan yang akan diberikan kepada *user* untuk mendapatkan nilai kriteria dari masing-masing *user*. Untuk setiap pertanyaan disimpan pada kolom pertanyaan, untuk nilai dari setiap pertanyaan disimpan pada kolom *opsi_a* sampai *opsi_d*. Kolom *keterangan_a* sampai *keterangan_d* merupakan kolom yang berisikan informasi mengenai keterangan dari nilai pada *opsi_a* sampai *opsi_d*. kolom *no_soal* merupakan nomor urut dari pertanyaan dan kolom gambar yaitu berisikan *link url* gambar.

3.3.4 Struktur Tabel

Struktur tabel database sistem rekomendasi restoran adalah sebagai berikut.

1. Tabel eventlogs

   Fungsi Tabel : Menyimpan *event* yang dikirim *user*.

   Primary Key : id

Tabel 3.2 Struktur Tabel Eventlogs

#	Nama Kolom	Tipe	Panjang	Keterangan
1	Id_eventlogs	serial	-	Identifier events
2	Events	text	-	Isi event yang dikirim user
3	Timestamp	timestamp	-	Waktu event yang diterima
4	Signature	varchar	100	Hasil enkripsi dari validasi platform LINE

2. Tabel *user_detail*

   Fungsi Tabel : Menyimpan informasi tambahan dari *user*

Foreign Key : userID

Tabel 3.3 Struktur Tabel User_details

#	Nama Kolom	Tipe	Panjang	Keterangan
1	UserID	Varchar	100	UserID LINE
2	jum_jawab	int	-	Jumlah pertanyaan yang sudah dijawab
3	Jenis_makanan	varchar	100	Jenis makanan yang dicari user
4	Nilai_derajat	int	-	Jawaban user dari pertanyaan

3. Tabel questions

Fungsi Tabel : Berisikan daftar pertanyaan.

Primary Key : No_soal

Tabel 3.4 Struktur Tabel Questions

#	Nama Kolom	Tipe	Panjang	Keterangan
1	No_soal	int	-	Identifier No Soal
2	Pertanyaan	text	-	Pertanyaan
3	Gambar	varchar	200	Link url gambar
4	Option_a	varchar	100	Pilihan jawaban A
5	Option_b	varchar	100	Pilihan jawaban B
6	Option_c	varchar	100	Pilihan jawaban C
7	Option_d	varchar	100	Pilihan jawaban D
8	Keterangan_a	varchar	100	Keterangan dari pilihan jawaban A
9	Keterangan_b	varchar	100	Keterangan dari pilihan jawaban B
10	Keterangan_c	varchar	100	Keterangan dari pilihan jawaban C
11	Keterangan_d	varchar	100	Keterangan dari pilihan jawaban D

4. Tabel *Users*

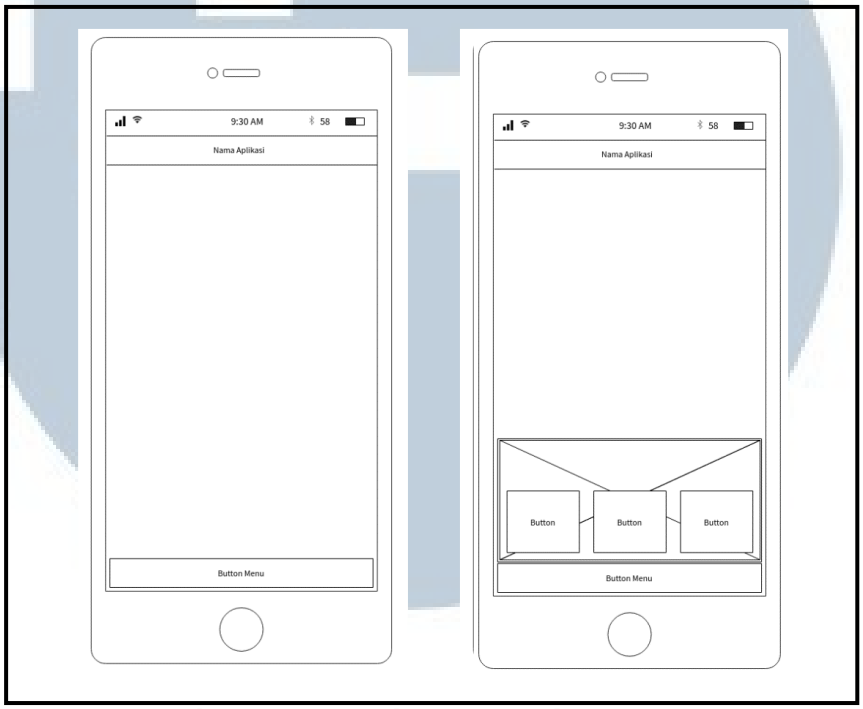
Fungsi Tabel : Berisikan informasi *user*.

Primary Key : UserID

Tabel 3.5 Struktur Tabel Users

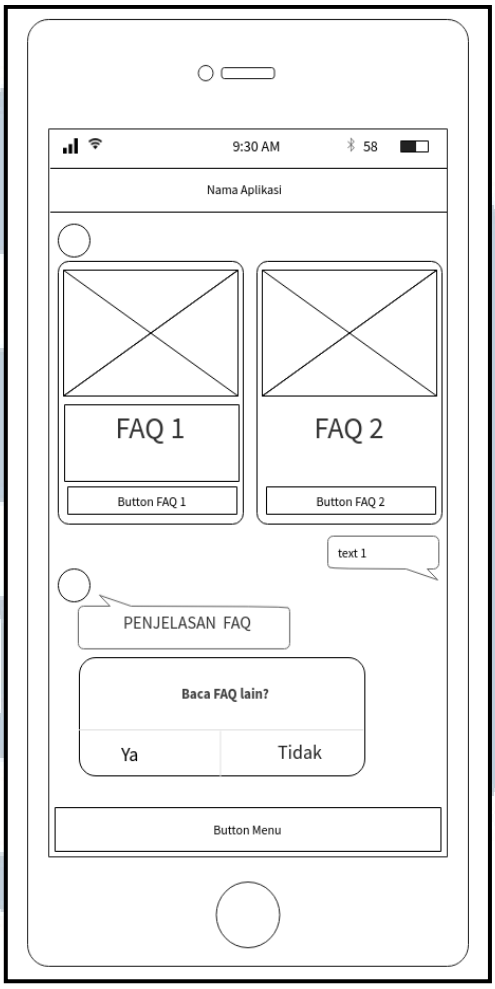
#	Nama Kolom	Tipe	Panjang	Keterangan
1	UserID	Varchar	100	Identifier *userID*
2	Display_name	Varchar	100-	Nama *User* dalam platform LINE

4.3.5. Rancangan Tampilan Antarmuka



Gambar 3.14 Tampilan Antarmuka Menu Utama

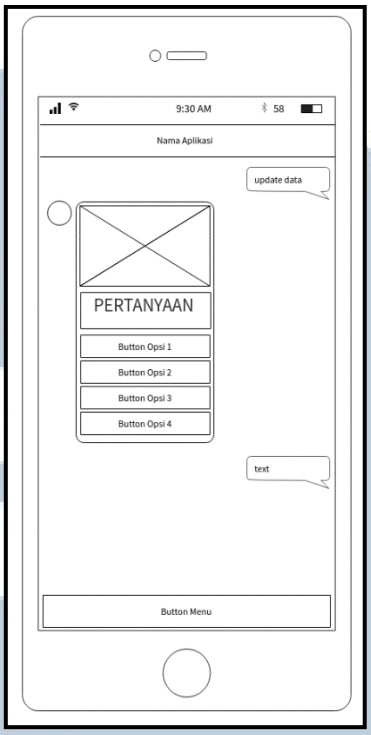
Berdasarkan Gambar 3.14 merupakan tampilan awal saat memulai *chating* dengan akun sistem rekomendasi yang dibuat. Terdapat empat *button* yang memiliki fungsi yang berbeda, dimana *button* menu yaitu berfungsi untuk menampilkan atau menghilangkan menu-menu pilihan dari aplikasi. *Button help* merupakan *button* yang berfungsi untuk menampilkan cara pemakaian aplikasi sistem rekomendasi, *button collection* berfungsi untuk menampilkan *restaurant of the week* langsung dari *browser*, sedangkan *button feedback* berfungsi untuk menampilkan *form* kritik dan saran langsung dari *browser*.



Gambar 3.15 Tampilan Antarmuka FAQ

Pada Gambar 3.15 merupakan tampilan saat *user* memilih menu *help* yang dimana terdapat beberapa FAQ berbentuk *carousel message* yang dapat dipilih *user*. Saat *user* memilih FAQ yang diinginkan, maka akan ditampilkan penjelasan FAQ tersebut dan kemudian menampilkan *confirm message* .

Pada Gambar 3.16 menampilkan *button message* berupa pertanyaan untuk mendapatkan nilai kriteria dari *user*. Tampilan ini akan muncul saat *user* melakukan *update data* atau saat *user* pertama kali berteman dengan akun sistem rekomendasi restoran.



Gambar 3.16 Tampilan Antar Muka Pertanyaan

Pada Gambar 3.17 merupakan hasil rekomendasi restoran berupa *carousel message*. Dalam *carousel message* tersebut terdapat gambar restoran, deskripsi restoran serta harga. Selain itu, terdapat juga *button* untuk melihat informasi detail restoran melalui *browser*

Gambar 3.17 Tampilan Antar Muka Hasil Rekomendasi Restoran

.