



### **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 DAN PERANCANGAN SISTEM

#### 3.1 Analisa Kebutuhan

Untuk melakukan penelitian, dibutuhkan perangkat keras, piranti lunak, dan bahasa pemrograman yang digunakan. Rincian yang dibutuhkan sebagai berikut.

##### 3.1.1 Perangkat Keras yang Digunakan

Perangkat keras yang digunakan sebagai berikut :

1. Laptop Dell Vostro 3450 , dengan spesifikasi
  - a. Intel® Core™ i5-2450M CPU @ 2.50GHz (4 CPUs), ~2.50 GHz
  - b. 4096 Mbytes DDR3 Memory
  - c. Video Intel®HD Graphics 3000 2048 Mbytes & AMD Radeon 6700 HD  
1024 Mbytes
  - d. Display 14.1" WXGA LED, *max resolution* 1366 x 768
  - e. Hard Drive Seagate 500 GB Serial ATA 5400 RPM

##### 3.1.2 Piranti Lunak yang Digunakan

1. Netbeans IDE 7.2
2. MySQL Database
3. Google Chrome versi 31.0.1650.63

### 3.1.3 Bahasa Pemrograman yang Digunakan

1. PHP
2. CSS
3. JavaScript
4. JQuery

## 3.2 Metodologi Penelitian

### 3.2.1 Studi Literatur

Tahap pertama penelitian ini adalah dengan melakukan studi literatur menggunakan referensi buku, artikel, jurnal, mengenai algoritma *Squeezer*, metode *Term Frequency ranking*, dan berbagai sumber untuk mendukung pembangunan dan pengembangan *website*. Literatur tersebut dijadikan pedoman dalam melakukan penelitian.

### 3.2.2 Pencarian dan Pendataan Kebutuhan

Pengumpulan data-data berupa tempat makan di kota-kota yang sudah ditentukan sebelumnya. Data-data tempat makan tersebut diberikan beberapa atribut yang merupakan kriteria makanan tersebut.

### 3.2.3 Perancangan Desain Website dan Database

Setelah mendapatkan semua data yang dibutuhkan, dilakukan perancangan dan pembuatan *database* untuk menampung semua data yang telah didapatkan. Selain itu, dilakukan juga perancangan *website* yang nantinya akan digunakan untuk mempresentasikan data.

### 3.2.4 Pembangunan Website dan Menerapkan Algoritma

Melakukan pembangunan *website* dan implementasi algoritma *Squeezer* yang digunakan dalam sistem rekomendasi ini.

### 3.2.5 Uji Coba

*Website* yang telah jadi dan diimplementasikan dengan Algoritma *Squeezer* untuk sistem rekomendasinya, akan dicoba seberapa akuratnya rekomendasi yang dihasilkan oleh Algoritma *Squeezer* ini. Pengujian ini menggunakan beberapa skenario yang telah disusun sesuai dengan kemungkinan – kemungkinan yang akan terjadi untuk melihat hasil yang dikeluarkan oleh algoritma *Squeezer* ini dalam pemberian rekomendasi kepada *user*. Pengujian terhadap metode *Term Frequency Ranking* dilakukan dengan memasukkan *string* kata pencarian dengan panjang yang bervariasi. Keakuratan dari metode *Term Frequency Ranking* ini dapat diukur dari nilai *Precision* dan *Recall* yang akan keluar sesuai dengan *string* kata pencari yang dimasukkan.

### 3.3 Perancangan Sistem

Perancangan sistem sebagai gambaran proses-proses utama yang ada pada aplikasi.

#### 3.3.1 Diagram Alir

Diagram alir merupakan bagan-bagan yang memiliki arus yang menggambarkan langkah-langkah dan proses dari suatu sistem. Diagram alir sistem dipecah menjadi tiga bagian besar, yaitu diagram alir untuk sistem umum, diagram alir sistem rekomendasi dengan menggunakan algoritma *Squeezer*, dan diagram alir metode pemberian *ranking*. Diagram alir sistem rekomendasi dengan menggunakan algoritma *Squeezer* dibagi lagi menjadi dua, yaitu diagram alir untuk membuat populasi, dan diagram alir untuk memberikan rekomendasi.

##### 1. Diagram Alir Sistem

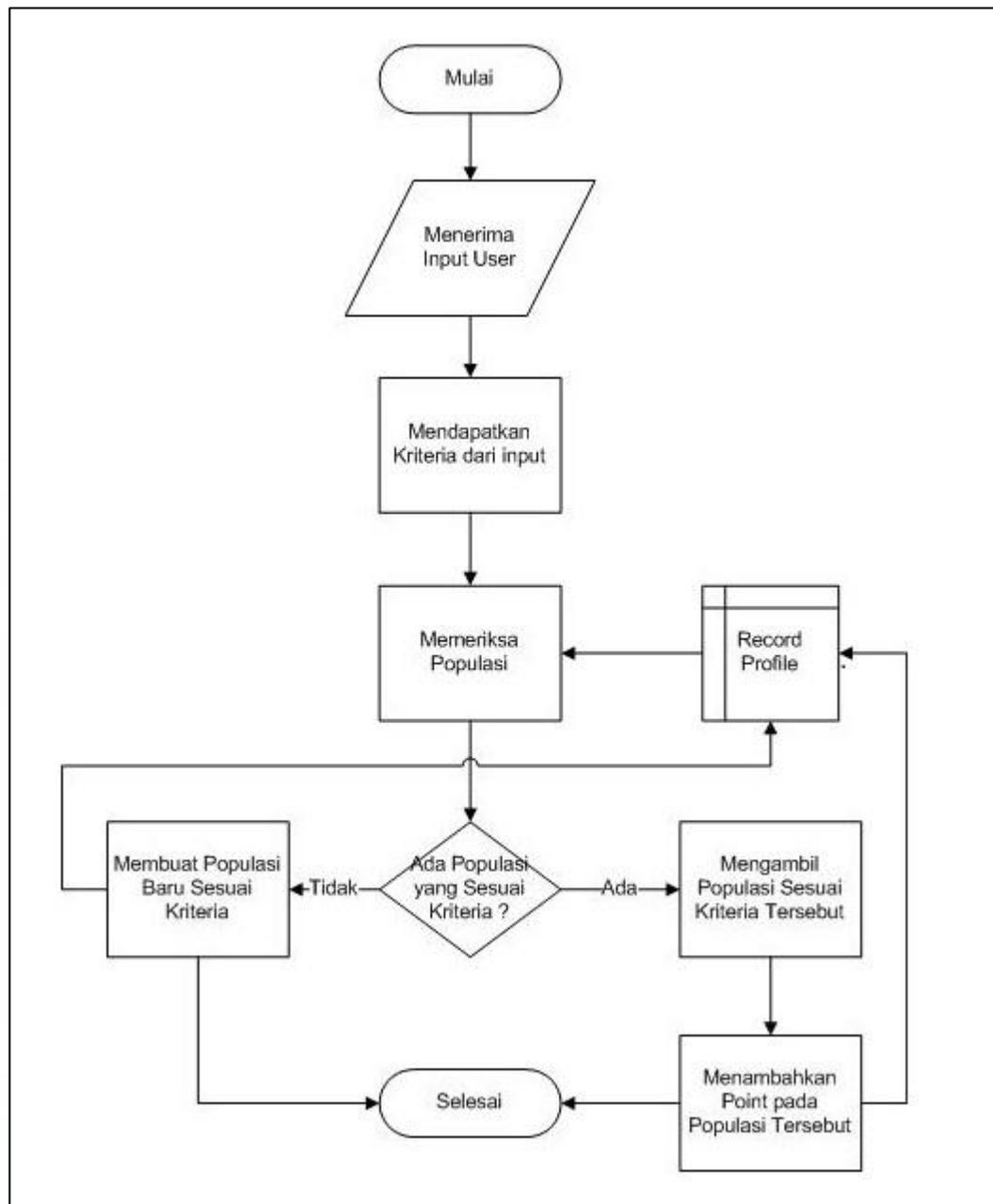
UMMN



ini dikarenakan rekomendasi yang diberikan kepada *user* berbeda satu sama lain tergantung dari profil dan kriteria si *user* tersebut.

2. Diagram Alir Sub-proses *Squeezer*
  - a. Diagram alir pembentukan populasi

The image contains a large, semi-transparent watermark of the UMMN logo. The logo consists of a circular emblem with a stylized building or tower structure inside, and the letters 'UMMN' written in a bold, sans-serif font below it.

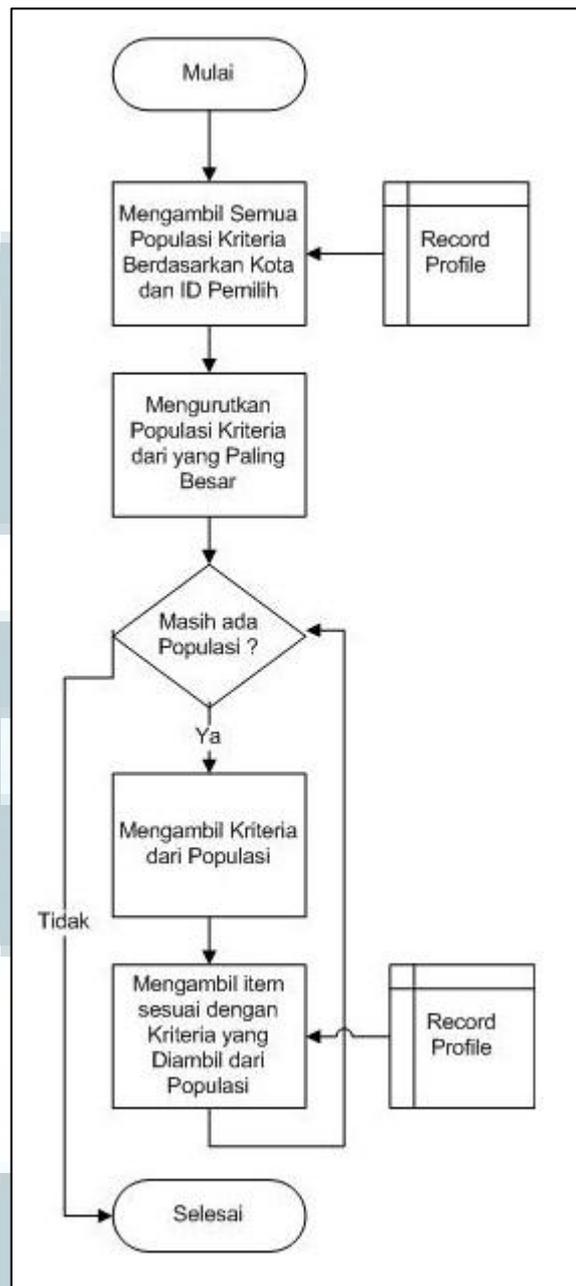


Gambar 3.2 Diagram alir pembentukan populasi

Pada sub-proses ini, dimulai dari penerimaan *input* dari *user*, yaitu pilihan tempat makan. Pilihan tempat makan tersebut akan diambil kriteria yang terdapat didalamnya. Setelah itu, akan dilakukan pemeriksaan apakah terdapat populasi yang sesuai dengan kriteria tersebut. Apabila tidak terdapat populasi yang sesuai dengan kriteria tersebut, maka akan dibuatkan populasi baru yang sesuai dengan kriteria tersebut. Apabila sudah ada populasi yang sesuai dengan kriteria tersebut, maka akan ditambahkan *point* pada populasi tersebut.

b. Diagram alir pemberian rekomendasi

UMMN

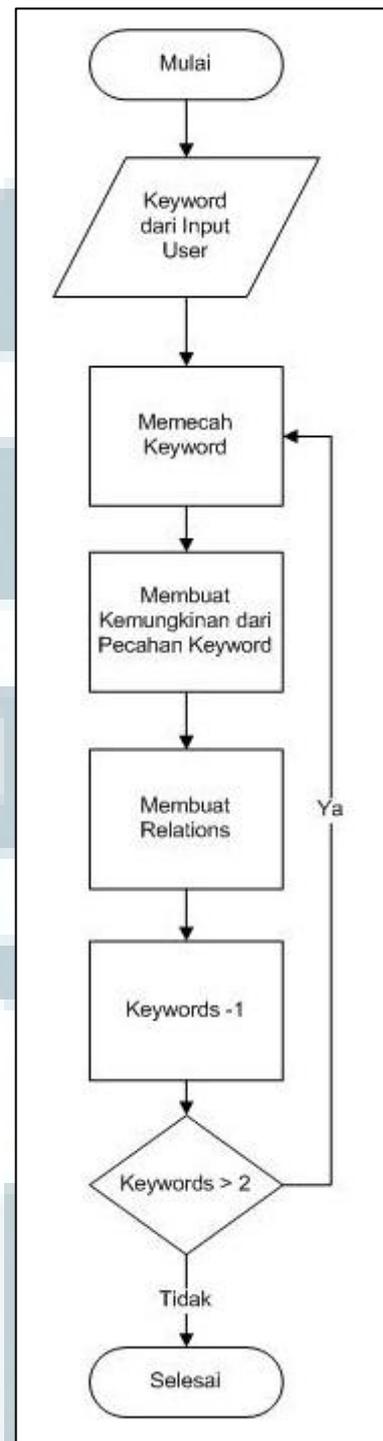


Gambar 3.3 Diagram alir pemberian rekomendasi

Pada bagian pemberian rekomendasi ini, dimulai dari pengambilan semua populasi kriteria yang sesuai dengan *user* yang sedang aktif dan kota tempat makan yang dipilih. Populasi-populasi tersebut diurutkan dari yang memiliki *point* paling tinggi sampai ke *point* yang paling rendah. Setelah itu, kriteria-kriteria pada setiap populasi akan diambil dan dijadikan acuan untuk memberikan rekomendasi kepada *user*, yaitu dengan cara merekomendasikan tempat makan yang memiliki kriteria yang sama dengan kriteria pada populasi tersebut.

### 3. Diagram alir sub-proses *Ranking*

UMMN

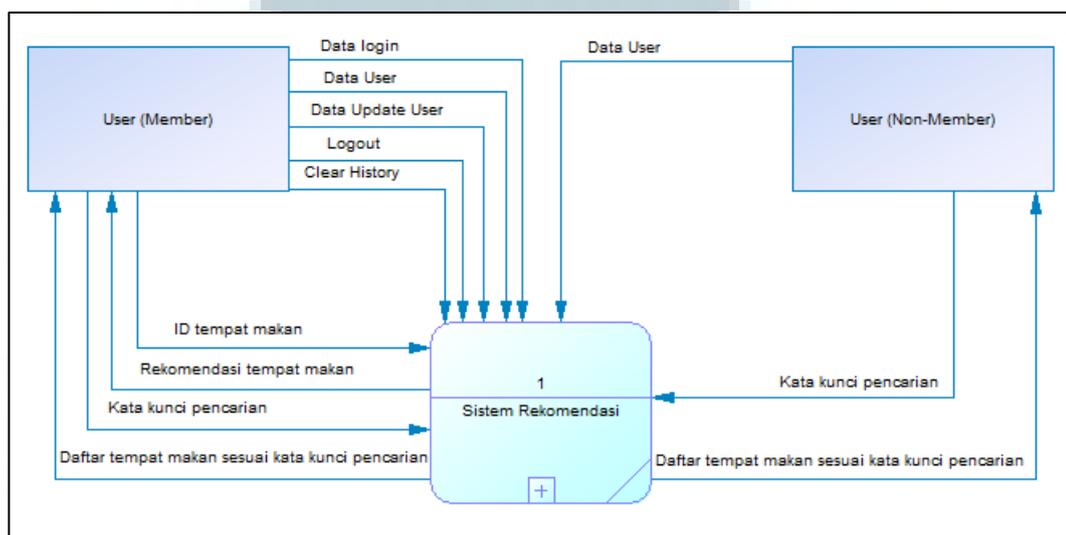


Gambar 3.4 Diagram alir sub-proses *Ranking*

Pemberian *ranking* untuk menentukan relevansi dokumen dimulai dengan penerimaan kalimat pencari dari *user*. Kalimat pencari tersebut kemudian dipecah menjadi kata-kata dan kemudian dilakukan dan dicari kemungkinan-kemungkinan kombinasi kata yang terbentuk. Setelah semua kemungkinan kombinasi kata terbentuk, maka jumlah panjang kata untuk kemungkinan kombinasi akan dikurangi dan dilakukan pencarian kembali kombinasinya. Data yang memiliki kata pencari yang lebih banyak didalamnya, memiliki nilai relevansi lebih tinggi. Setelah itu, dibuatlah pengurutan dari data yang memiliki relevansi paling tinggi ke yang paling rendah.

### 3.3.2 Data Flow Diagram

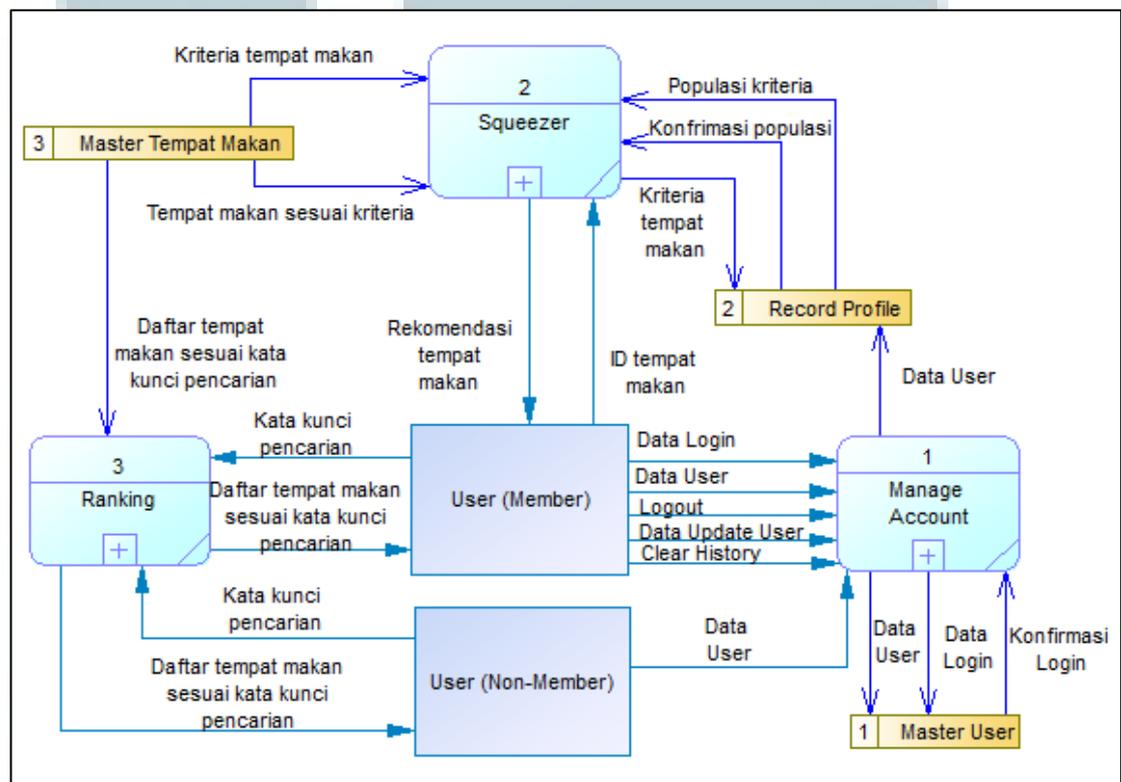
#### 1. Context Diagram



Gambar 3.5 Context Diagram

Dikarenakan pembuatan sistem hanya pada sistem *user*, maka pada diagram tersebut, sistem digambarkan sepenuhnya berinteraksi dengan *user*. Mengeluarkan *output* kepada *user* berupa rekomendasi dengan algoritma *Squeezer* dan hasil pencarian kata dengan metode *Term Frequency ranking*. Interaksi sistem lebih dalam dan lengkapnya dapat dilihat pada gambar selanjutnya yang menggambarkan *input* dan *output user* dan interaksi sistem dengan *table* dalam *database*.

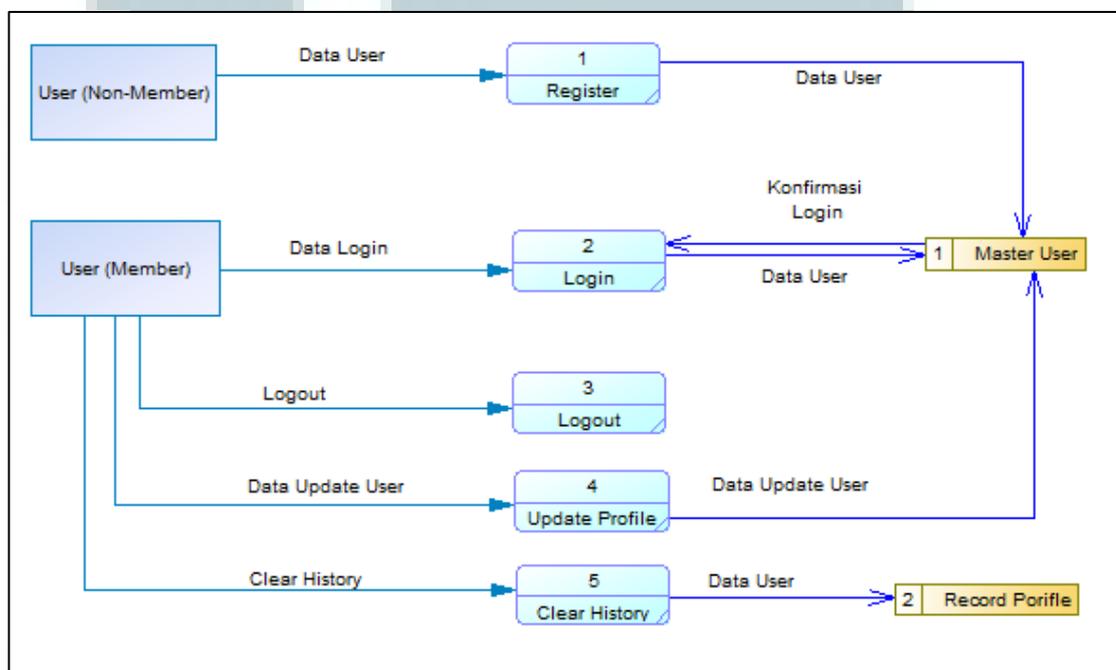
## 2. DFD Level 1



Gambar 3.6 DFD Level 1

Pada DFD *level 1* ini, seperti pada Gambar 3.6, terdapat tiga proses besar dalam sistem, yaitu *Manage Account*, *Squeezer*, dan *Ranking*. *Squeezer* dan *Ranking* ini merupakan implementasi dan cara kerja dari algoritma yang dipakai dalam sistem.

### 3. DFD Level 1.1 (*Manage Account*)

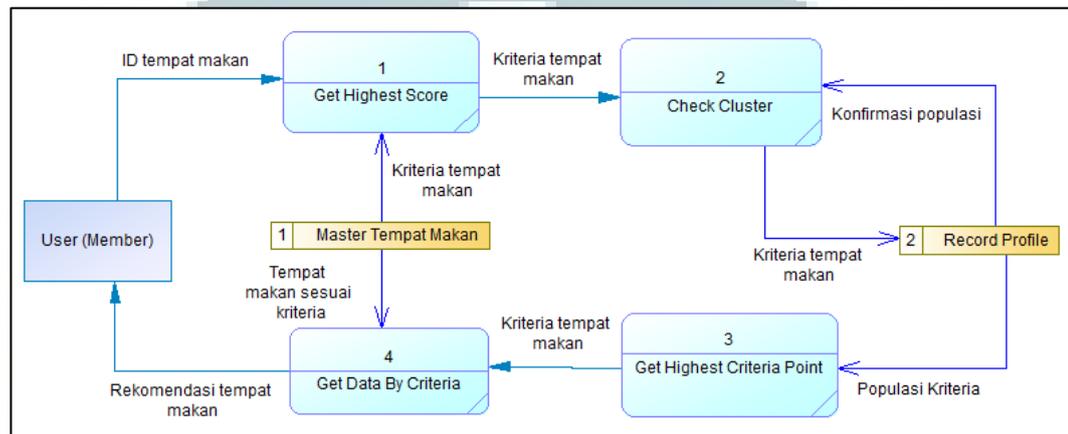


Gambar 3.7 DFD Level 1.1 (*Manage Account*)

Pada DFD *Manage Account* pada Gambar 3.7, terdapat lima proses utama yang dapat dilakukan oleh *user* seperti *Login*, *Register*, *Logout*, *Update Profile*, dan *Clear History*. Proses *Clear History* adalah proses untuk menghapus profil *user* tentang populasi kriteria tempat makan sebelumnya, sehingga dapat memberikan rekomendasi yang berbeda apabila *user* membuat populasi yang baru. Proses *Login*, *Register*, dan *Update Profile* terhubung dengan tabel *Master*

User, sedangkan proses *Clear History* hanya terhubung dengan tabel *Record Profile*.

#### 4. DFD Level 2.1 (*Squeezer*)

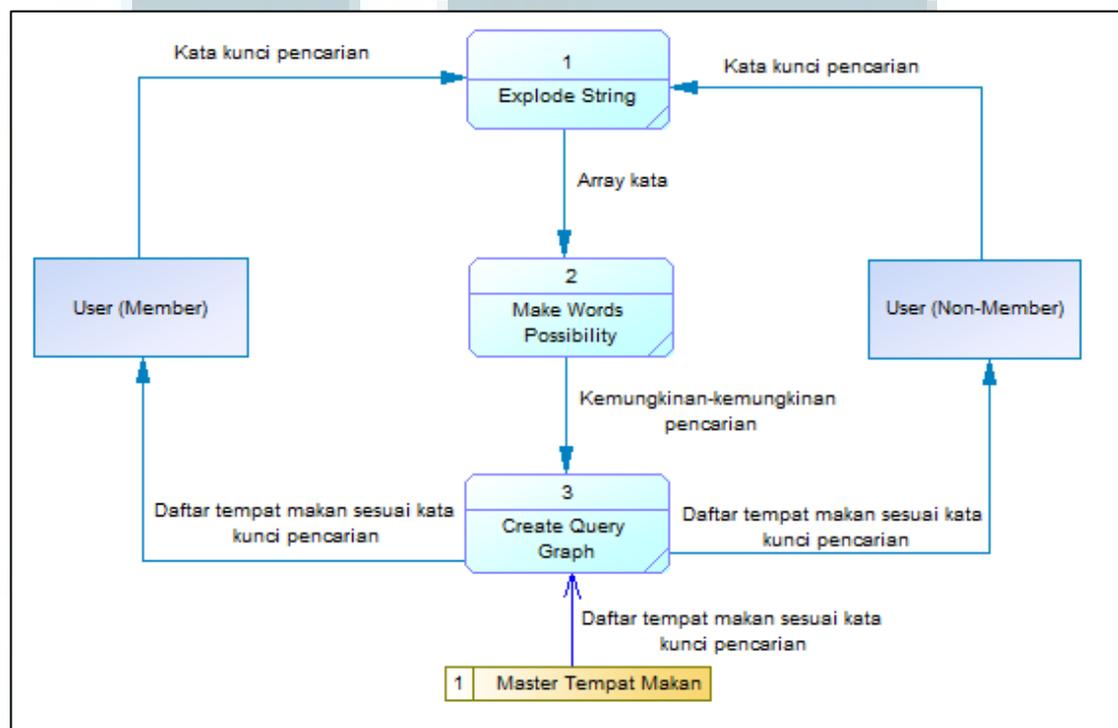


Gambar 3.8 DFD Level 2.1 (*Squeezer*)

Pada DFD *Squeezer* seperti pada Gambar 3.8, terdapat empat proses utama pemberian rekomendasi, yaitu *Get Highest Score*, *Check Cluster*, *Get Highest Criteria Point*, dan *Get Data by Criteria*. Proses dimulai dari user yang memilih sebuah tempat makan, secara otomatis *ID* tempat makan tersebut dikirimkan. Kemudian, pilihan tempat makan oleh user tersebut diolah untuk diambil nilai kriteria tertingginya pada proses *Get Highest Score*. Setelah nilai kriteria tertingginya diperoleh, dilakukan pemeriksaan pada populasi apakah sudah ada populasi dengan kriteria yang didapat tersebut atau belum. Apabila belum ada, maka dibuat populasi baru, dan apabila sudah ada, maka ditambahkan 1 *point* pada populasi tersebut. Proses selanjutnya adalah *Get Highest Criteria Point*

dimana pemeriksaan terhadap populasi yang ada. Populasi-populasi tersebut diambil dan diurutkan berdasarkan *point* yang paling tinggi ke yang paling rendah. Setelah itu, dilakukan pengambilan data-data yang sesuai dengan kriteria dari populasi-populasi yang diambil untuk diberikan kepada *user* pada proses *Get Data by Criteria*.

### 5. DFD Level 3.1 (*Ranking*)



Gambar 3.9 DFD Level 3.1 (*Ranking*)

Proses *ranking* ini digunakan pada saat *user* memberikan kata kunci pencarian untuk mencari tempat makan yang diinginkan. Dimulai pada proses *Explode String*, dimana kata kunci oleh *user* dipecah menjadi pecahan-pecahan kata. Kemudian, pecahan-pecahan kata tersebut dikirimkan ke proses *Make Words Possibility* untuk dibuatkan kemungkinan-kemungkinan kombinasi kata. Setelah semua kemungkinan kombinasi kata sudah terbentuk, maka kata-kata tersebut dijadikan *query* untuk melakukan penarikan dari *database* pada proses *Create Query Graph*. Kemudian hasil *query* tersebut diberikan kepada *user* pada halaman *Search Result*.

### 3.3.3 Perancangan Database

*Database* yang digunakan adalah MySQL. Berikut struktur tabel-tabel yang terdapat di dalamnya.

1. Nama tabel : *Master Jenis Makanan*

Fungsi : Tabel ini digunakan untuk menyimpan daftar jenis makanan yang akan digunakan sebagai acuan untuk tabel *Master Tempat Makan*.

Tabel 3.1 *Master Jenis Makanan*

No	Nama Field	Tipe Data	Panjang	Keterangan
1	ID	int		UNIQUE_IDENTITY
2	JENIS_MAKANAN	varchar	20	Jenis Makanan

2. Nama tabel : *Master Tempat Makan*

Fungsi : Tabel ini digunakan untuk menyimpan daftar tempat makan beserta dengan keterangan-keterangannya yang digunakan.

Tabel 3.2 *Master Tempat Makan*

No	Nama Field	Tipe Data	Panjang	Keterangan
1	ID	int		UNIQUE_IDENTITY
2	NAMA	varchar	100	Nama Tempat Makan
3	ALAMAT	varchar	200	Alamat Tempat makan
4	ID_KOTA	int		Mengarah ke tabel MASTER_KOTA
5	KETERANGAN	text		Keterangan Tempat Makan
6	HARGA_MIN	int		Harga Terendah di Tempat Makan
7	HARGA_MAX	int		Harga Tertinggi di Tempat Makan
8	POSX	float		Posisi X untuk Google Maps
9	POSY	float		Posisi Y untuk Google Maps
10	ID_JENIS_MAKANAN	int		Mengarah ke table MASTER_JENIS_MAKANAN
11	IMAGE_NAME	varchar	50	Menyimpan Nama Gambar
12	VIEW	int		Jumlah yang Melihat
13	KRITERIA_PEDAS	float		Nilai Kriteria Pedas
14	KRITERIA_MANIS	float		Nilai Kriteria Manis
15	KRITERIA_ASIN	float		Nilai Kriteria Asin
16	KRITERIA_ASAM	float		Nilai Kriteria Asam
17	Tag	text		Tag untuk Mengoptimalkan Pencarian

3. Nama tabel : *Master Kota*

Fungsi : Tabel ini digunakan untuk menyimpan daftar kota yang akan digunakan sebagai acuan kota pada Tabel *Master Tempat Makan*.

Tabel 3.3 *Master Kota*

No	Nama Field	Tipe Data	Panjang	Keterangan
1	ID	int		UNIQUE_IDENTITY
2	NAMA_KOTA	varchar	20	Nama Kota

4. Nama tabel : *Master User*

Fungsi : Tabel ini digunakan untuk menyimpan daftar pengguna beserta dengan keterangan-keterangannya.

Tabel 3.4 *Master User*

No	Nama Field	Tipe Data	Panjang	Keterangan
1	ID	int		UNIQUE_IDENTITY
2	USERNAME	varchar	50	Username
3	EMAIL	varchar	50	Email User
4	PASSWORD	varchar	50	Password User
5	NAMA	varchar	50	Nama Lengkap User
6	NOMOR_TELEPON	varchar	25	Nomor Telepon user
7	ALAMAT	text		Alamat User

5. Nama tabel : *Record Profile*

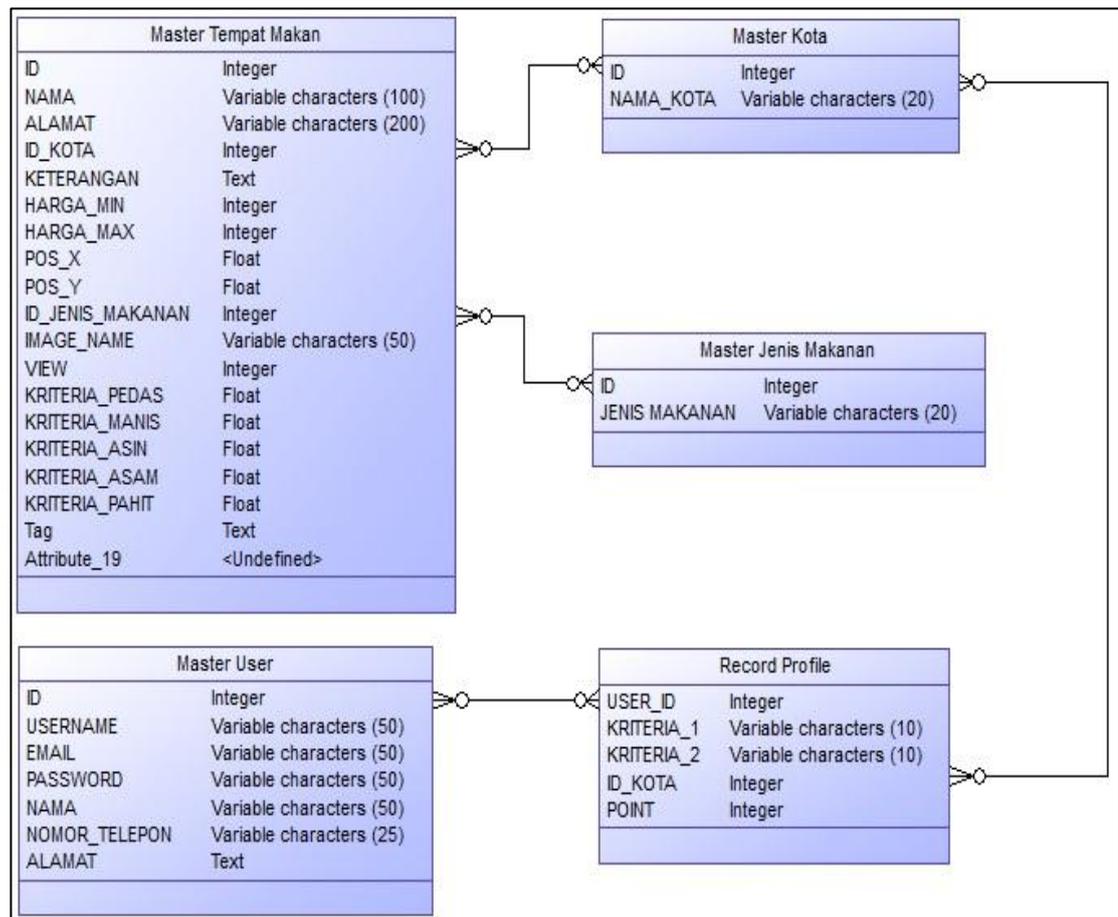
Fungsi : Tabel ini digunakan untuk menyimpan daftar populasi kriteria yang dimiliki oleh setiap *user*.

Tabel 3.5 *Record Profile*

No	Nama Field	Tipe Data	Panjang	Keterangan
1	USER_ID	int		ID User
2	KRITERIA_1	varchar	10	Kriteria Pertama
3	KRITERIA_2	varchar	10	Kriteria Kedua
4	ID_KOTA	int		ID Kota
	POINT	int		Bobot Nilai Populasi

Dan berikut adalah relasi antar tabel dalam *database*.

UMMN



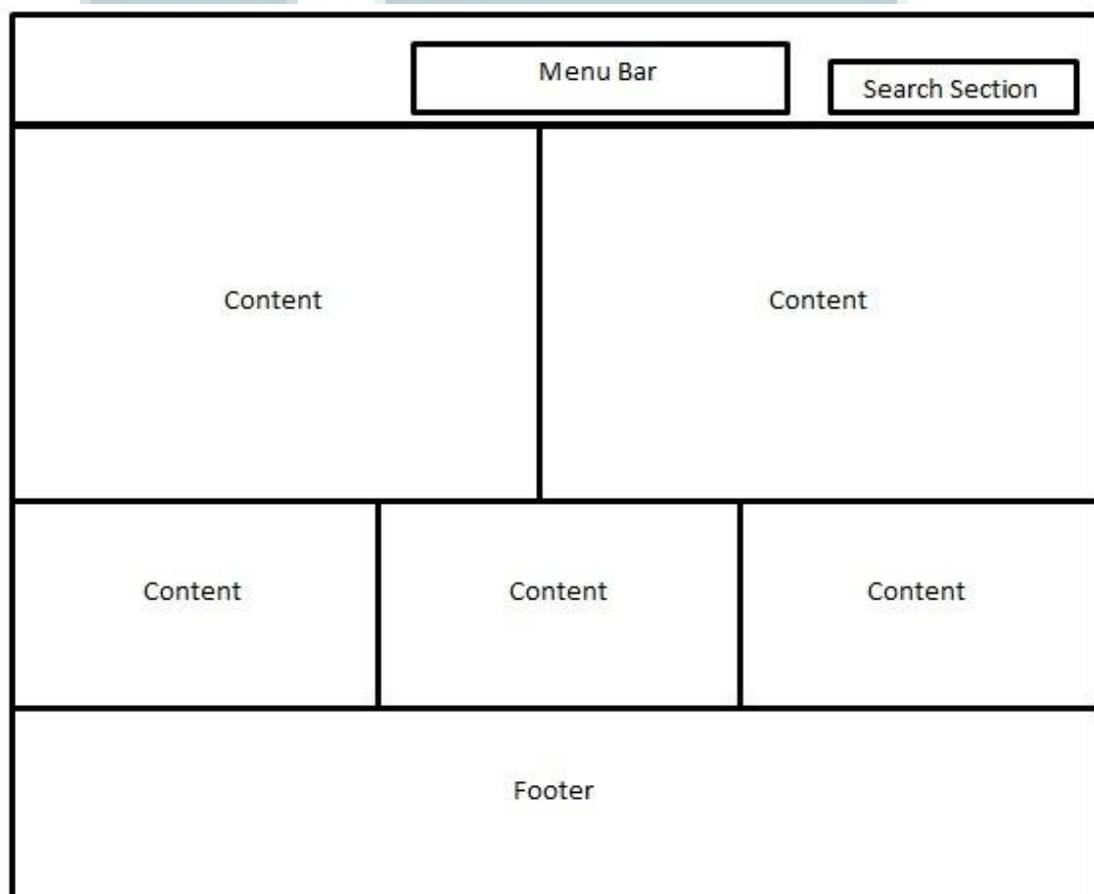
Gambar 3.10 ERD

# UMMN

### 3.3.4 Perancangan Antarmuka

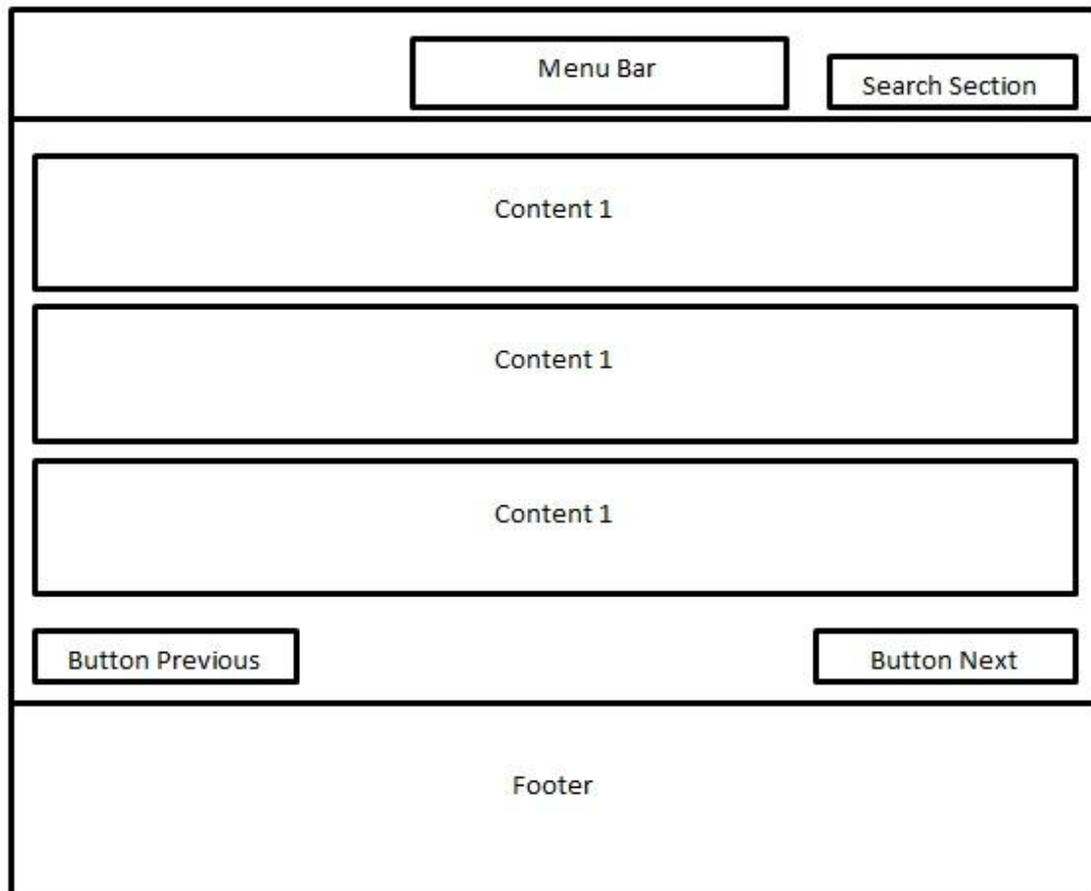
Pada bagian ini, ditampilkan beberapa sketsa antarmuka sistem dari sisi pengguna akhir. Berikut adalah beberapa perancangan antarmuka *website* Sistem Rekomendasi ini.

#### 1. Halaman Utama



Gambar 3.11 Sketsa Halaman Utama

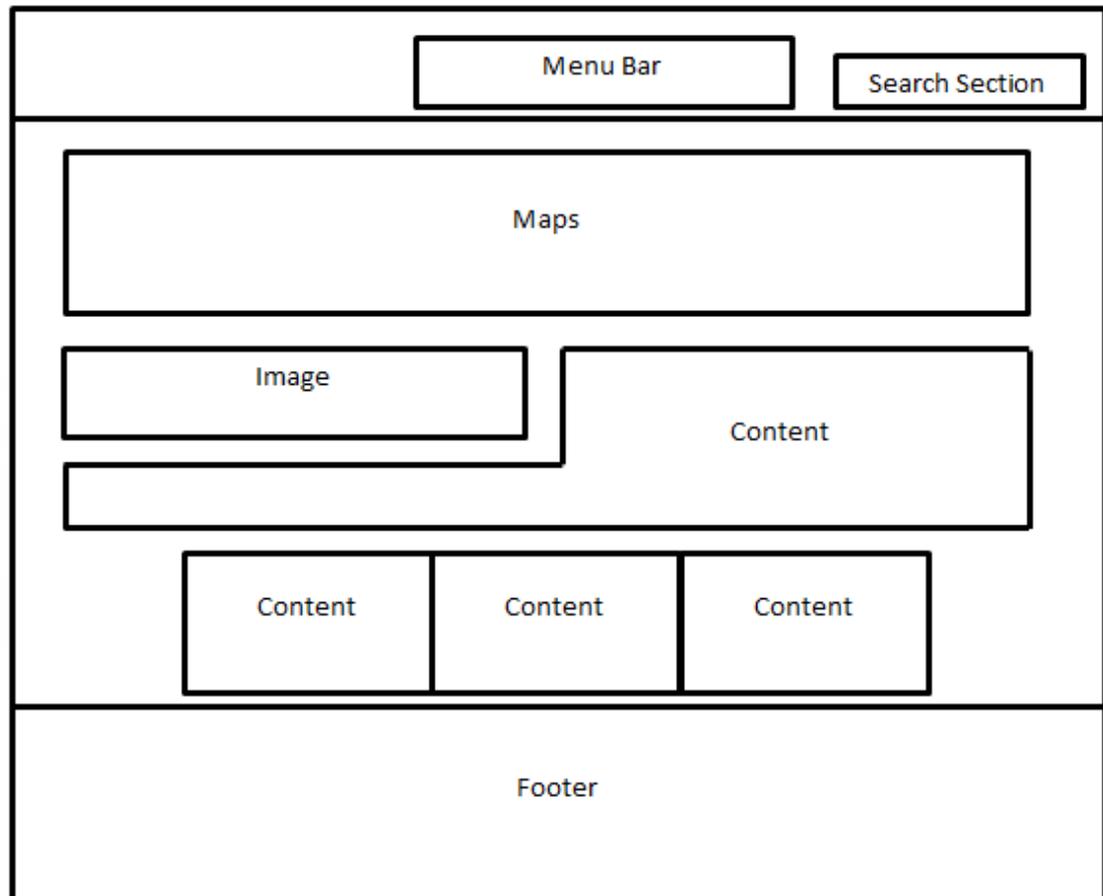
## 2. Halaman Daftar Restoran



Gambar 3.12 Sketsa Halaman Daftar Restoran

Halaman ini dapat diakses dengan memilih menu “*Restaurants*” pada *menu bar*. Setelah memilih kota mana yang akan ditampilkan, maka data restoran pada kota tersebut akan ditampilkan secara berurutan. Terdapat tombol “*Previous*” dan “*Next*” sebagai navigasi melihat isi selanjutnya atau sebelumnya.

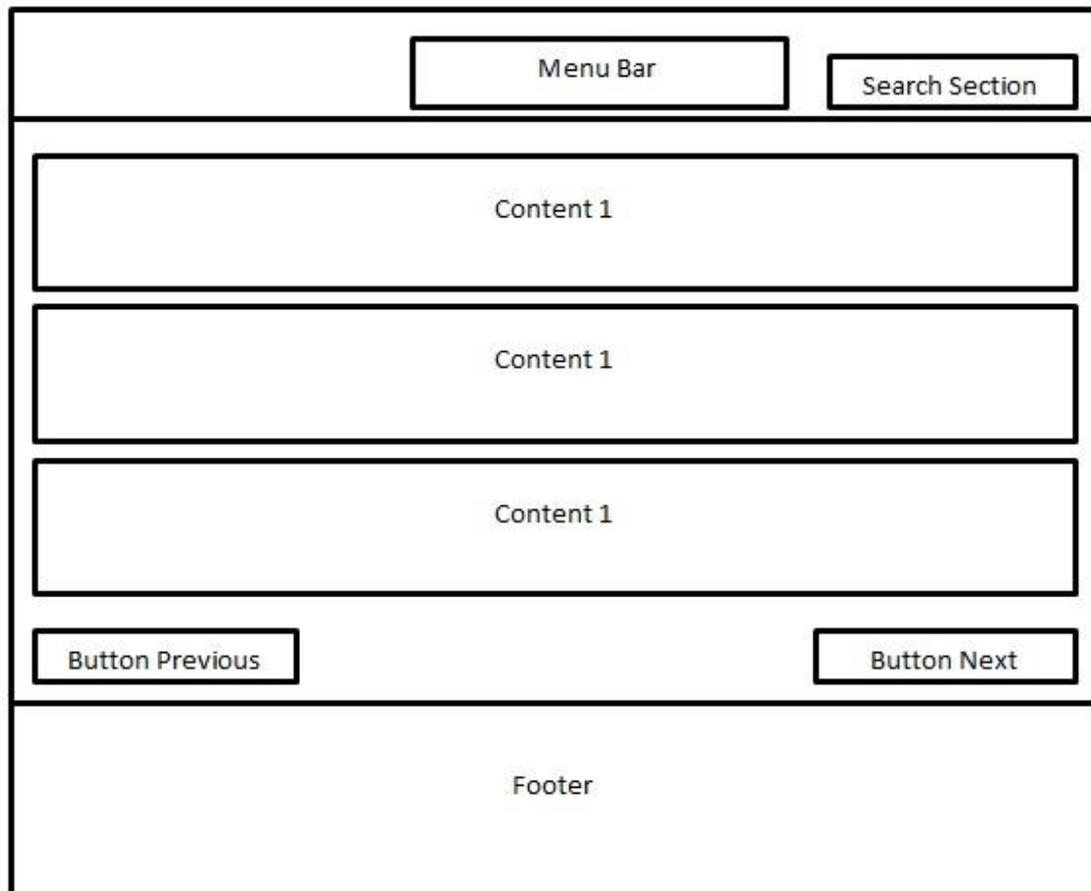
### 3. Halaman *Detail* Restoran



Gambar 3.13 Sketsa Halaman *Detail* Restoran

Halaman ini dapat diakses setelah *user* memilih tempat makan yang *user* inginkan. Halaman detail restoran ini menampilkan lokasi pada bagian atas dalam fitur *google maps*. Gambar pada bagian kiri, dan keterangannya pada bagian kanan. Pada bagian paling bawah, terdapat rekomendasi konten untuk *user*. Untuk mendapatkan rekomendasi tersebut, *user* harus melakukan *login* terlebih dahulu.

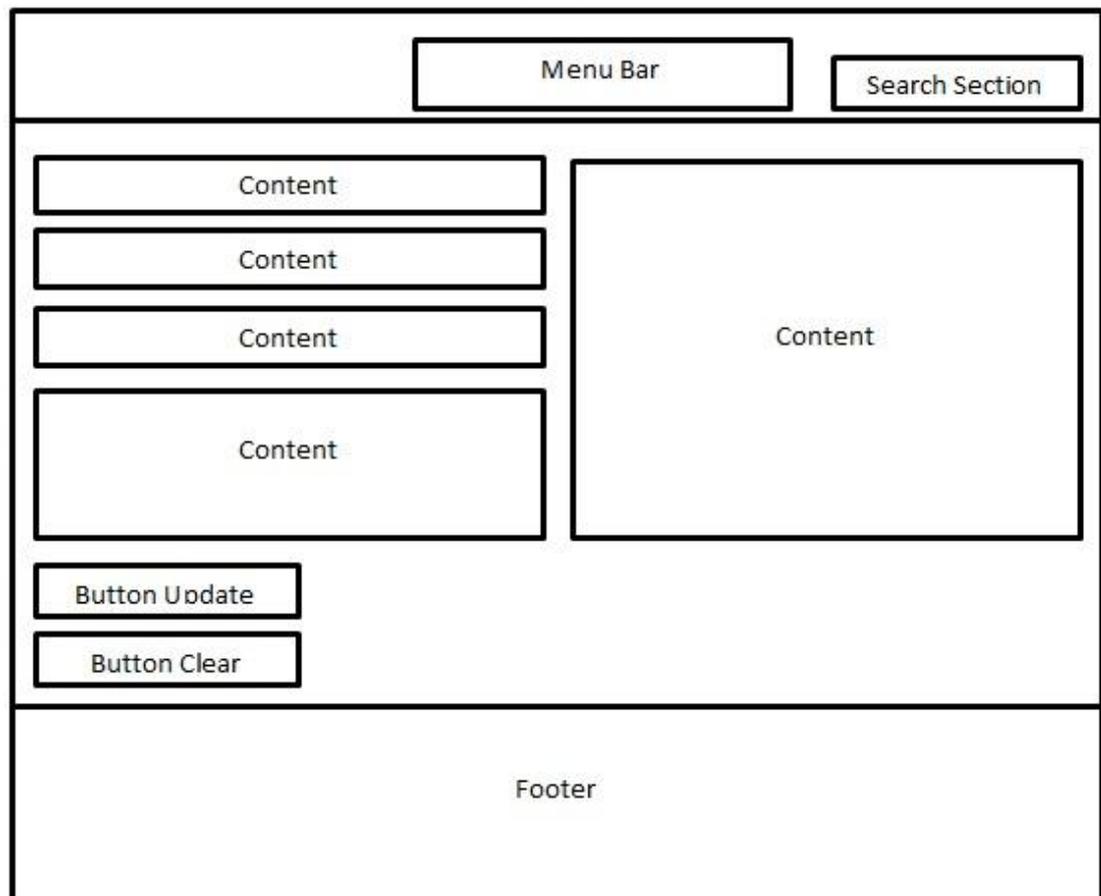
#### 4. Halaman Hasil Pencarian



Gambar 3.14 Sketsa Halaman Hasil Pencarian

Halaman ini dapat diakses apabila kita sudah memasukkan kata kunci pencarian. Halaman hasil pencarian ini menampilkan dua jenis hasil pencarian, yaitu pencarian yang sesuai dengan kata kunci, dan pencarian yang berhubungan (*similar search*) yang didapat dari proses *ranking*.

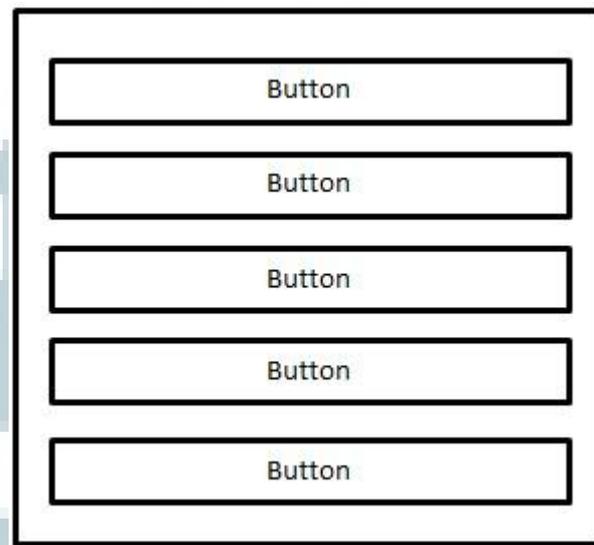
## 5. Halaman *Profile*



Gambar 3.15 Sketsa Halaman *Profile*

Halaman *profile* ini bertujuan untuk mengatur data pengguna. Kolom *Content* pada halaman *profile* ini untuk menampilkan data dari *user*. Terdapat tombol “*Update*” untuk mengubah data pengguna, dan “*Clear History*” untuk menghapus profil *user* tentang kriteria tempat makan yang sebelumnya.

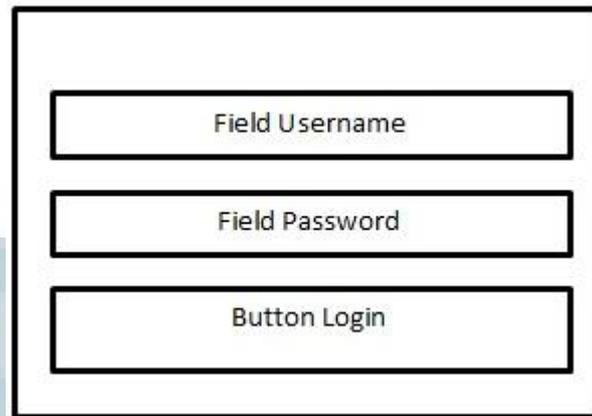
## 6. *Pop-up* Pilihan Kota



Gambar 3.16 Sketsa *Pop-up* Pilihan Kota

*Pop-up* ini muncul ketika *user* menekan tombol “*Restaurants*” pada *menu bar*. Akan muncul pilihan kota yang terbatas untuk ditampilkan pada halaman Daftar Restoran.

## 7. *Pop-up* Login



Field Username

Field Password

Button Login

Gambar 3.17 Sketsa *Pop-up Login*

*Pop-up login* muncul apabila *user* menekan tombol “*login*” pada *menu bar* untuk mengatur akunnya, atau mendapatkan rekomendasi pada halaman *detail* restoran. Terdapat dua *field* untuk memasukkan *username* dan *password*, serta tombol *Login*.

UMMN