



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.



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 Metodologi Penelitian

Dalam perancangan sistem rekomendasi tempat makan dengan metode *Simple Additive Weighting* (SAW), dilakukan beberapa tahap dalam penelitian dan penulisan, yaitu sebagai berikut.

1. Studi Literatur

Pada tahap ini, studi mengenai metode SAW dan perancangan sistem rekomendasi dari berbagai sumber seperti buku, jurnal, maupun yang *online* ataupun yang tercetak dilakukan untuk memperdalam pengetahuan mengenai perancangan sistem rekomendasi dan metode SAW.

2. Perancangan Sistem

Tahap perancangan sistem dimulai dengan merancang *Data Flow Diagram* (DFD), lalu pembuatan *flowchart* untuk menentukan alur kerja dari sistem.

Setelah pembuatan *flowchart*, *mockup* dari tampilan sistem dibuat.

3. Pemrograman Sistem

Tahap selanjutnya setelah sistem dirancang adalah pemrograman sistem. Bahasa pemrograman yang digunakan adalah Java. Fitur-fitur dari sebuah sistem rekomendasi tempat makan dan penerapan dari metode SAW juga dibuat di tahap ini. Untuk penyimpanan nilai dari preferensi *user*, sistem menyimpan data tersebut di sebuah *text file*. Pembuatan tampilan dari sistem (*user interface*) dikerjakan berdasarkan *mockup* yang telah dibuat.

4. Testing dan Evaluasi

Pada tahap ini, testing dilakukan untuk menguji perhitungan dengan metode SAW yang dilakukan oleh sistem dengan membandingkan hasil perhitungan dari sistem dengan perhitungan manual. Evaluasi sistem dilakukan dengan menyebarkan aplikasi dan kuesioner kepada *user*. Setelah *user* menggunakan aplikasi, *user* diberikan kuesioner untuk diisi sebagai bahan evaluasi penelitian.

5. Dokumentasi

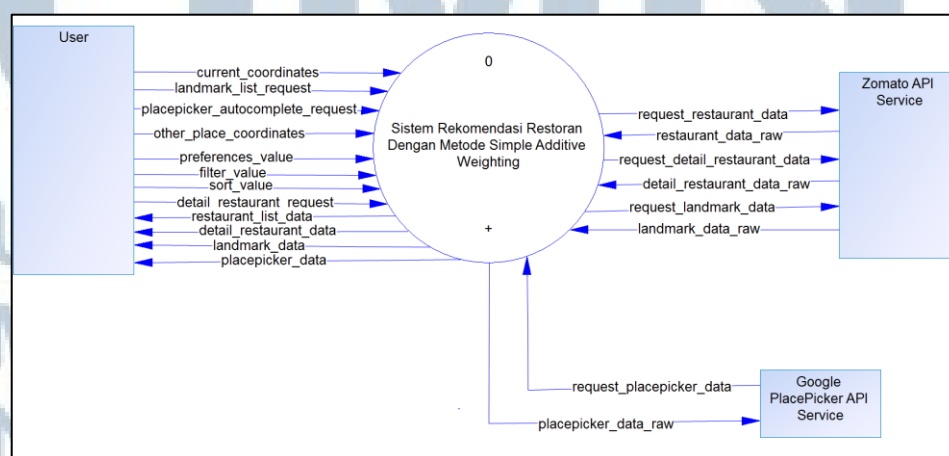
Penulisan laporan dilakukan pada tahapan ini dengan tujuan untuk mencatat segala bentuk proses penelitian serta menyimpulkan hasil akhir yang didapat.

3.2 Perancangan Sistem

Dalam perancangan sistem, terdapat dua diagram yang digunakan, yaitu *data flow diagram* dan *flowchart*. Penjelasan mengenai kedua diagram tersebut akan dijelaskan pada subbab berikut.

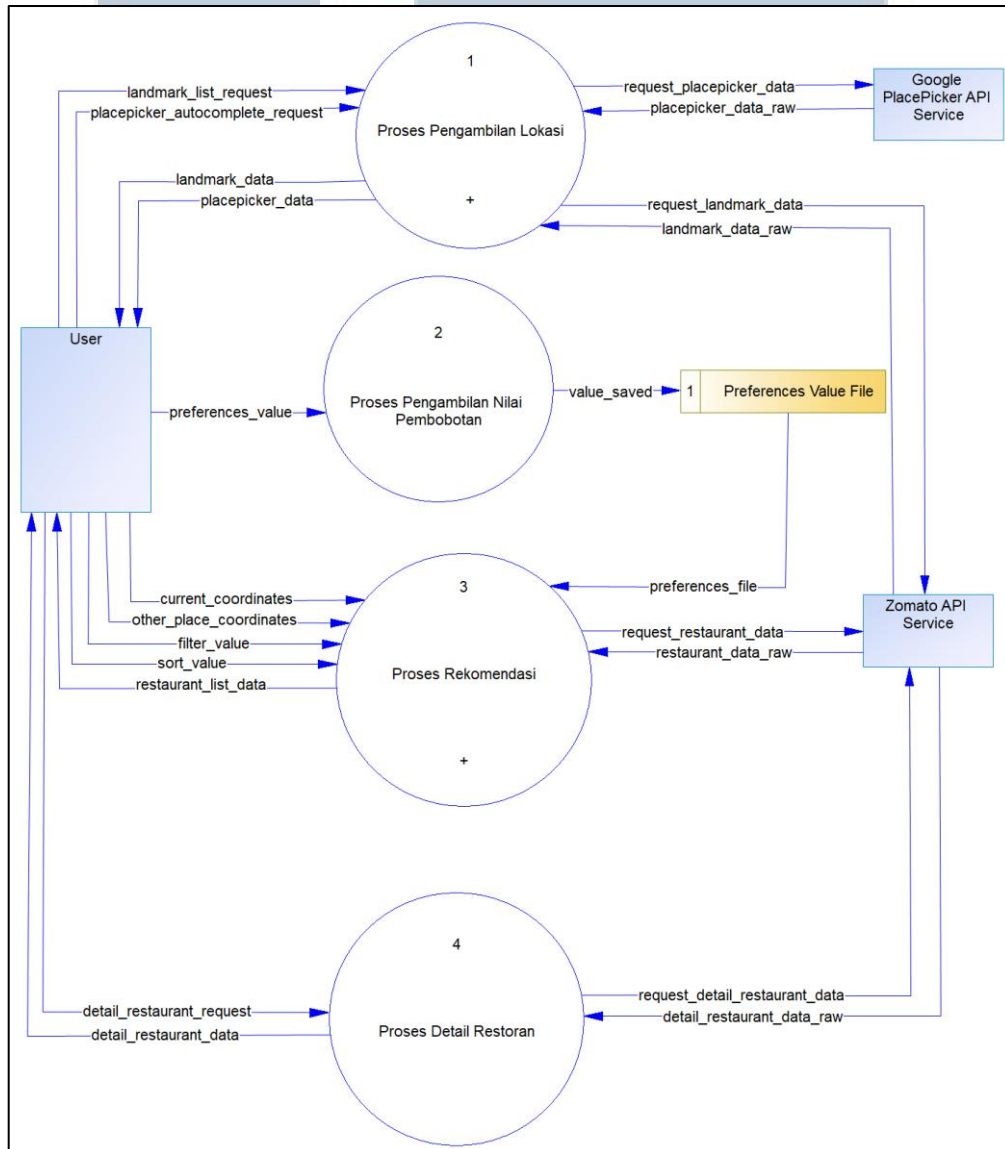
3.2.1 Data Flow Diagram

Data Flow Diagram adalah diagram yang menunjukkan interaksi dan proses pengiriman atau penerimaan data antar entitas dan proses yang ada di dalam sistem.



Gambar 3.1 Data Flow Diagram Level 0

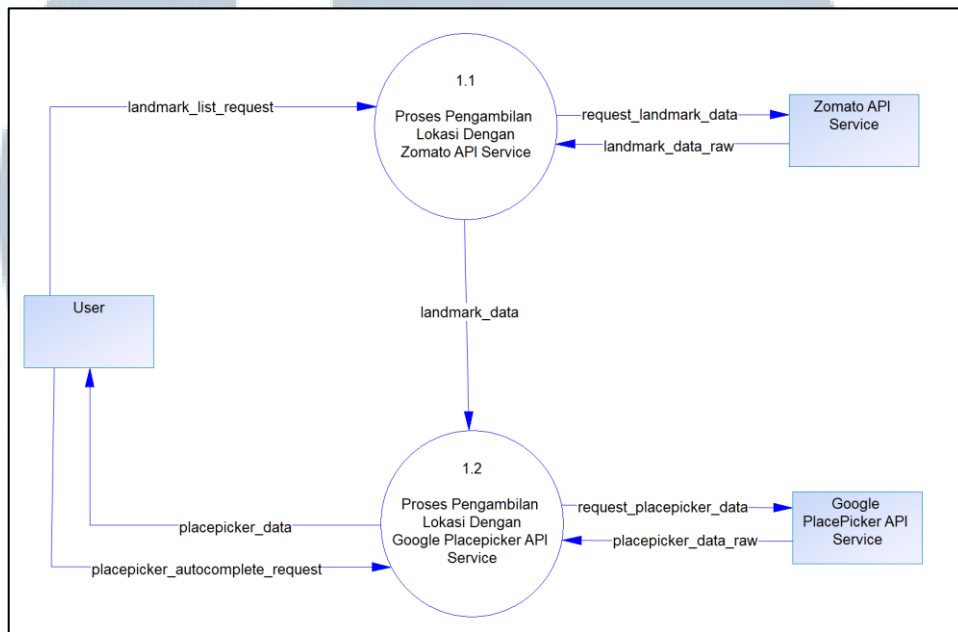
Gambar 3.1 merupakan DFD level 0 atau *Context Diagram* dari sistem rekomendasi restoran dengan metode *Simple Additive Weighting* berbasis Android. Terdapat 3 entitas, yaitu *user*, Google PlacePicker API Service, dan Zomato API Service. Ketiga entitas ini berinteraksi dengan sistem dengan memberikan *input* menuju ke proses dan *feedback* dari sistem menuju ketiga entitas.



Gambar 3.2 DFD Level 1

Gambar 3.2 menunjukkan DFD Level 1 dari sistem rekomendasi restoran dengan metode *Simple Additive Weighting* berbasis Android. Di gambar tersebut,

terdapat 4 proses, yaitu proses pengambilan lokasi, proses pengambilan nilai pembobotan, proses rekomendasi, dan proses detail restoran. Terdapat satu *data store object* di sistem ini, yaitu *Preferences Value File*, sebuah *text file* yang berisi bobot dari preferensi pribadi yang telah dimasukkan oleh *user*.

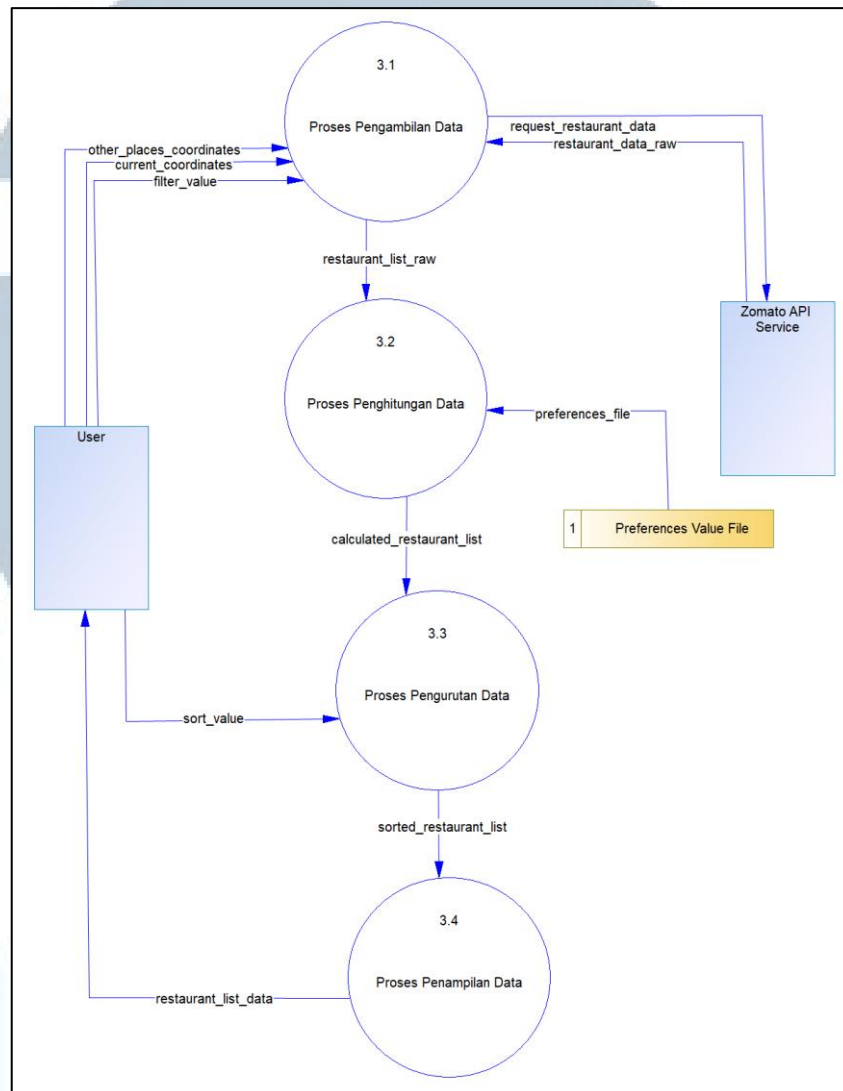


Gambar 3.3 DFD Level 2 Proses Pengambilan Lokasi

Gambar 3.3 adalah diagram subproses untuk pengambilan lokasi. Di sistem ini, *user* dapat mengambil lokasi menggunakan dua cara, yaitu dengan *landmark database* yang disediakan oleh *Zomato API Service*, atau dengan *database* yang disediakan oleh *Google PlacePicker API Service*.

Jika menggunakan *Zomato API Service*, sistem akan mengambil data koordinat dari *database Zomato*, lalu akan mengirim koordinat tersebut ke *Google PlacePicker API Service*, agar *user* dapat melihat lokasi dari koordinat tersebut di sebuah peta. Jika langsung menggunakan *Google PlacePicker API Service*, maka *user* akan memilih koordinat dari lokasi yang diinginkan langsung dari peta yang disediakan oleh *Google PlacePicker API Service*. Koordinat tersebut akan

digunakan untuk mendapatkan data dari restoran-restoran yang ada di sekitar lokasi tersebut.



Gambar 3.4 DFD Level 2 Proses Rekomendasi

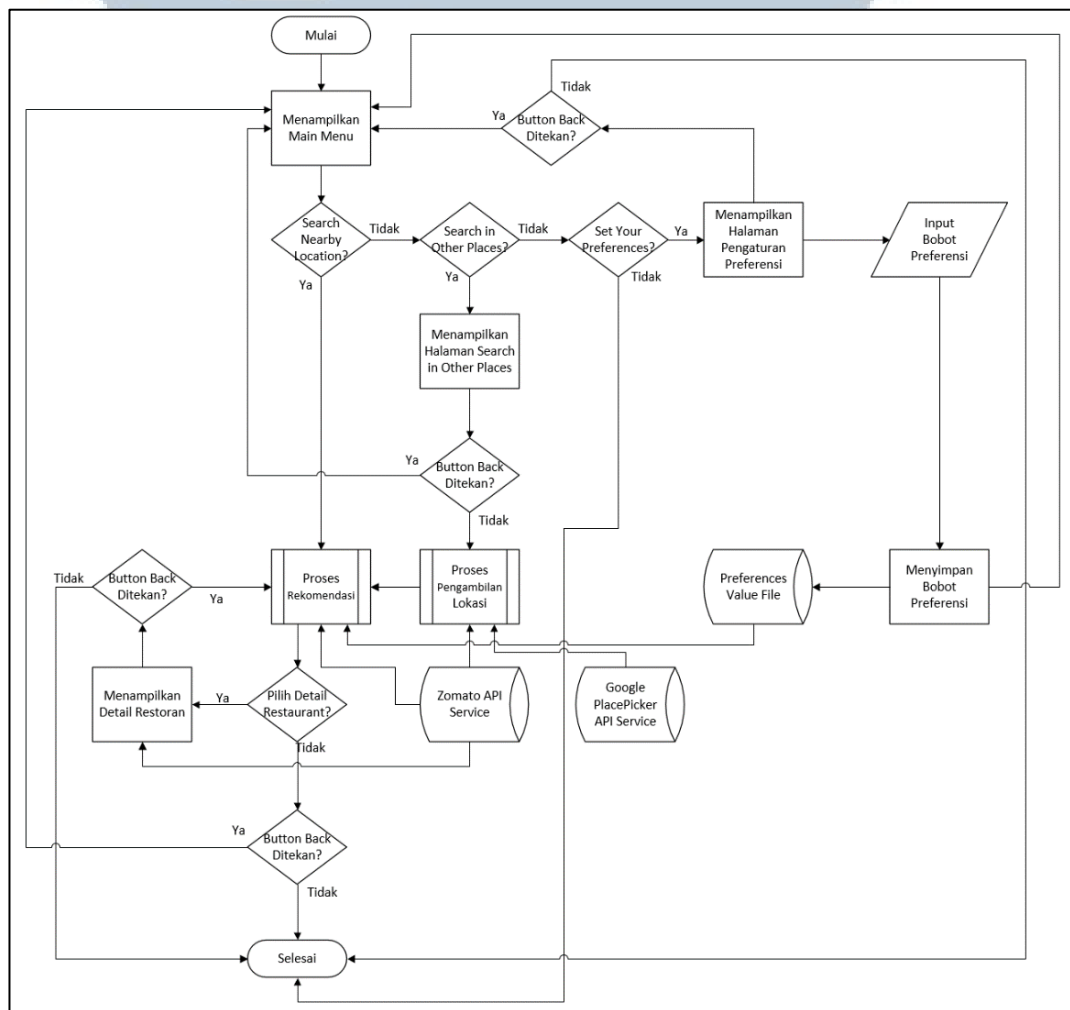
Gambar 3.4 adalah diagram subproses untuk rekomendasi. Setelah mendapatkan koordinat dari lokasi yang diinginkan, sistem akan meminta data restoran yang ada di sekitar lokasi tersebut ke *Zomato API Service*. Data akan disimpan dalam sebuah *list* untuk diproses dalam perhitungan SAW dan pengurutan. Jika *user* menggunakan filter, filter yang dipilih oleh *user* juga akan dikirimkan ke *Zomato API Service*. Data yang didapatkan akan dihitung

menggunakan metode *Simple Additive Weighting* dan nilai dari pembobotan didapat dari *Preferences Value File*.

Setelah data dihitung, data akan diurutkan berdasarkan jarak lalu ditampilkan ke *user*. *User* juga bisa mengurutkan restoran berdasarkan harga, *rating*, jumlah *upvote*, atau berdasarkan nilai preferensi pribadi setelah dihitung.

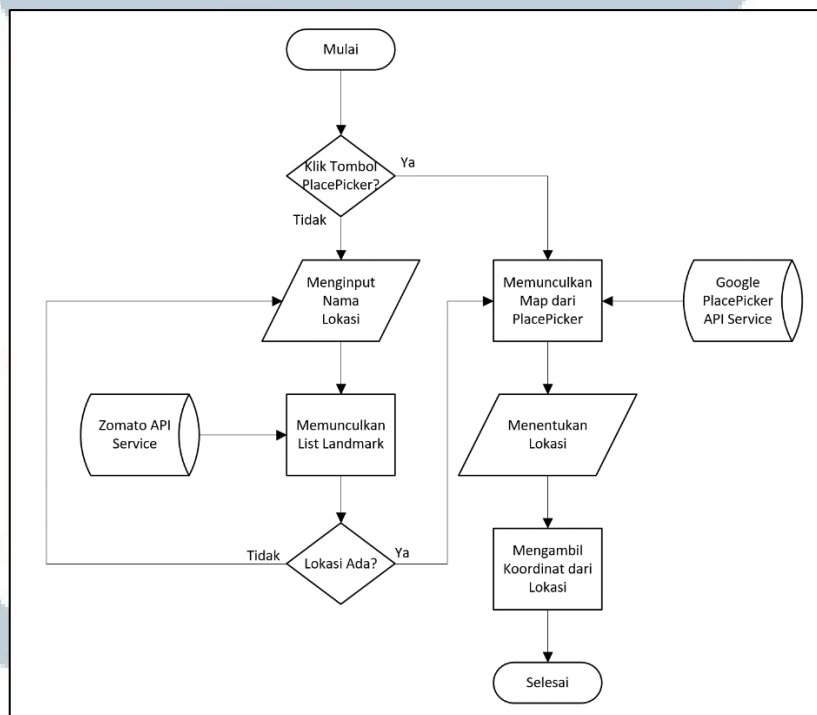
3.2.2 Flowchart

Flowchart adalah diagram dengan langkah-langkah yang berurutan untuk melambangkan alur sebuah proses dalam sebuah sistem menggunakan bentuk atau simbol.



Gambar 3.5 *Flowchart* Utama Aplikasi

Gambar 3.5 adalah *flowchart* utama dari Aplikasi. Aplikasi akan menampilkan *main menu* ke *user*, lalu *user* dapat memilih untuk mencari restoran di sekitar posisi *user*, mencari restoran di tempat lain, atau memberikan nilai pembobotan. Jika *user* memilih untuk mencari restoran di sekitar *user*, sistem akan melakukan proses rekomendasi, yang akan lebih dijelaskan di Gambar 3.7. Jika *user* memilih untuk mencari restoran di tempat lain, sistem akan melakukan proses pengambilan lokasi, yang akan di lebih dijelaskan di Gambar 3.6. Setelah mendapatkan data restoran, *user* juga bisa melihat informasi lebih dari restoran tersebut. Sistem akan meminta informasi lebih mengenai restoran tersebut ke Zomato API Service.

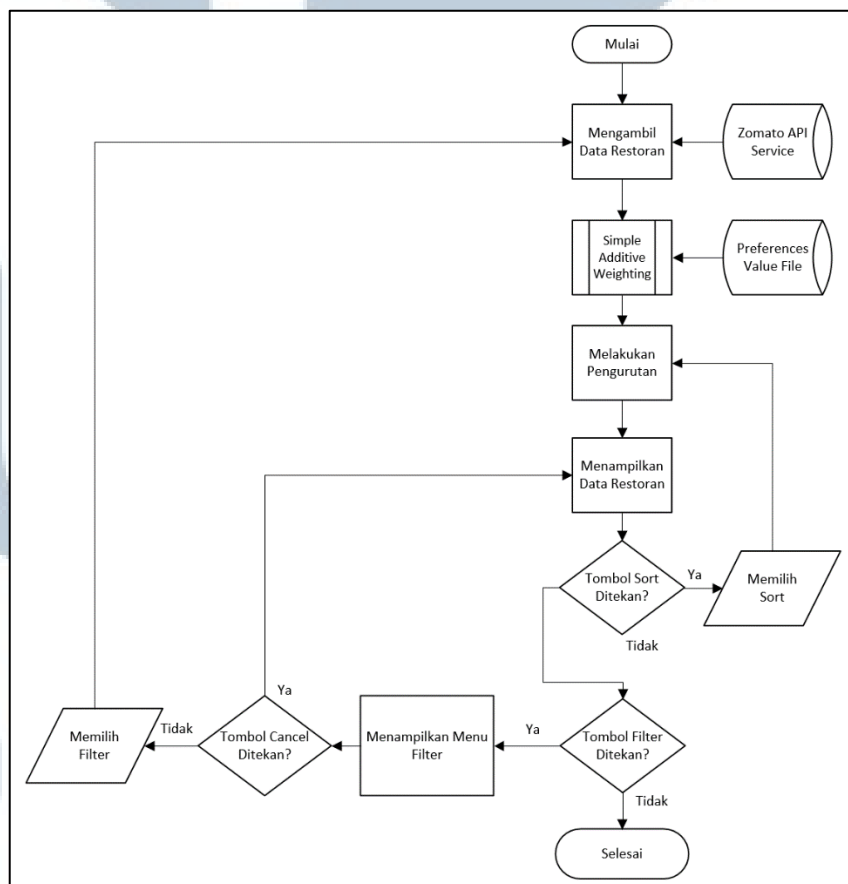


Gambar 3.6 *Flowchart* Proses Pengambilan Lokasi

Gambar 3.6 merupakan *flowchart* untuk proses pengambilan lokasi. Jika *user* memilih untuk mencari restoran di tempat lain, sistem akan menampilkan menu untuk mencari lokasi ke *user*.

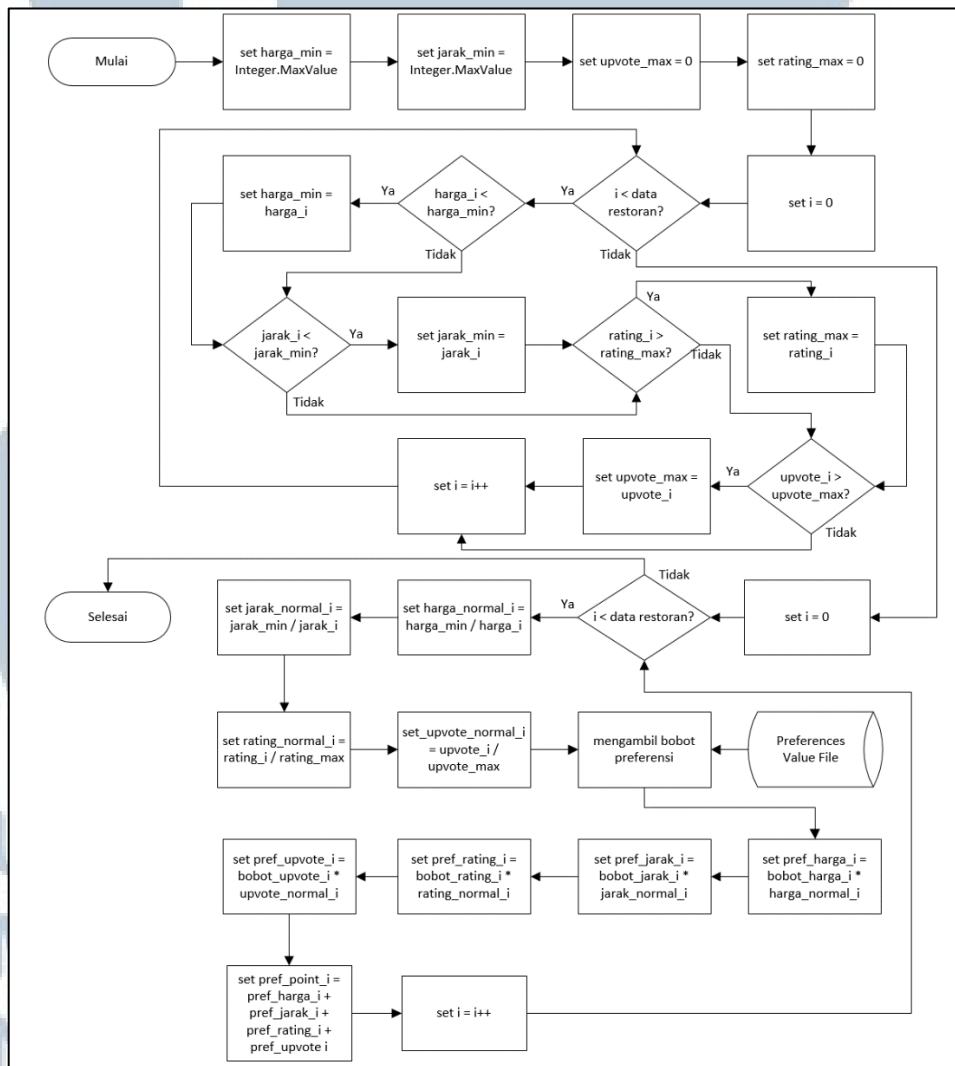
User dapat mencari lokasi yang diinginkan berdasarkan nama lokasi dan sistem akan mengembalikan maksimal 10 lokasi dengan nama yang sesuai atau mirip dengan permintaan user dari *database* Zomato. Setelah user memilih lokasi yang diinginkan, sistem akan mengirim koordinat dari lokasi tersebut ke Google PlacePicker API Service, yang akan memunculkan peta agar user dapat melihat lokasi tersebut.

User juga bisa langsung memilih lokasi Google PlacePicker API Service. Sistem akan menampilkan peta ke user dan user bisa memilih lokasi yang diinginkan. Setelah mendapatkan lokasi yang diinginkan, sistem akan mengambil koordinat dari garis bujur dan lintang lokasi tersebut dan akan mengirimkannya ke Zomato API Service untuk mendapatkan restoran yang ada di sekitar lokasi.



Gambar 3.7 Flowchart Proses Rekomendasi

Gambar 3.7 adalah *flowchart* untuk proses rekomendasi. Sistem akan mengirimkan koordinat tersebut ke *Zomato API Service* untuk mendapatkan data restoran yang ada di sekitar lokasi tersebut, lalu sistem melakukan proses perhitungan dengan metode SAW. Setelah perhitungan dilakukan, sistem akan mengurutkan data berdasarkan jarak yang paling dekat dengan koordinat tersebut. *User* juga bisa mengurutkan berdasarkan harga, *rating*, jumlah *upvote*, atau nilai preferensi pribadi. *User* juga bisa melakukan *filtering* dengan menekan tombol “Filter”. Sistem akan menampilkan menu filter yang berisikan pilihan untuk mencari restoran berdasarkan tipe restoran dan jenis masakan.



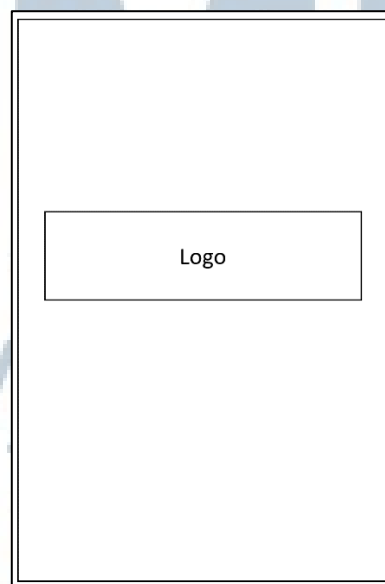
Gambar 3.8 *Flowchart* Metode *Simple Additive Weighting*

Gambar 3.8 adalah *flowchart* untuk perhitungan menggunakan metode SAW. Pertama, sistem akan mencari nilai terendah untuk kriteria jarak dan harga dan nilai tertinggi untuk kriteria rating dan jumlah *upvote*. Sistem akan melakukan normalisasi untuk setiap variabel kriteria dari setiap restoran. Untuk kriteria jarak dan harga, nilai jarak dan harga restoran tersebut akan membagi nilai jarak dan harga terendah, sementara untuk kriteria *rating* dan jumlah *upvote*, nilai *rating* dan jumlah *upvote* akan dibagi oleh nilai *rating* dan jumlah *upvote* tertinggi.

Pembobotan dilakukan dengan mengalikan nilai preferensi pribadi dari *user* dengan hasil nilai normalisasi. Setelah pembobotan dilakukan, nilai pembobotan dari empat kriteria tersebut akan dijumlahkan untuk mendapatkan nilai preferensi akhir.

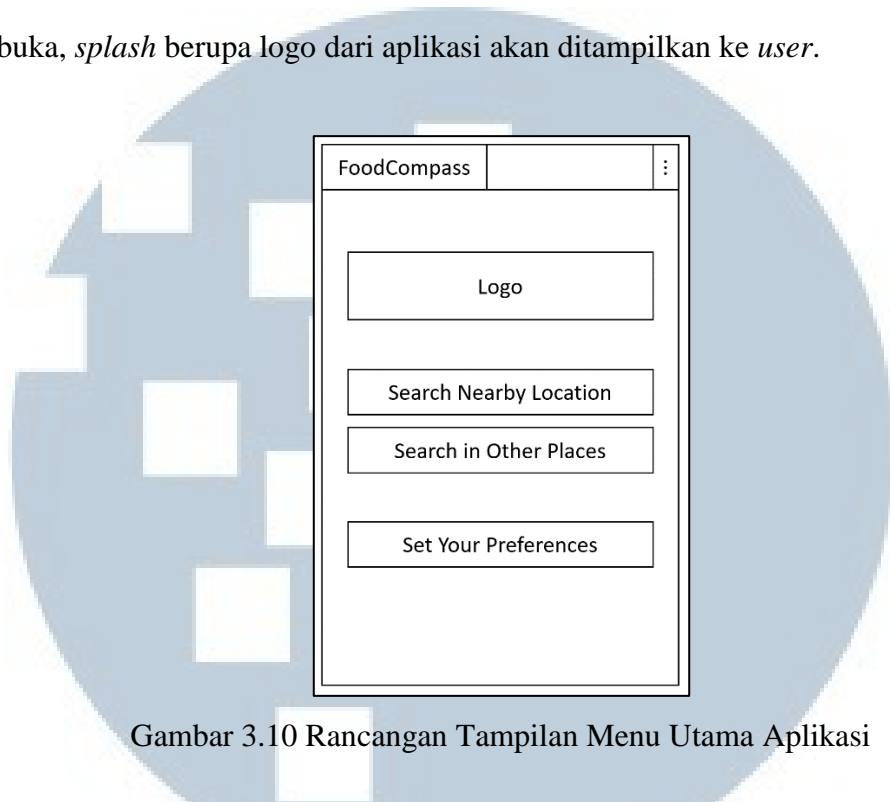
3.2.3 Perancangan Antarmuka Aplikasi

Setelah proses perancangan sistem dilakukan, dibuat pula rancangan antarmuka sistem yang dibentuk sedemikian rupa untuk memudahkan pengguna dalam menggunakan sistem tersebut.



Gambar 3.9 Rancangan Tampilan *Splash* Aplikasi

Gambar 3.9 adalah rancangan tampilan *splash* pada aplikasi. Saat aplikasi dibuka, *splash* berupa logo dari aplikasi akan ditampilkan ke *user*.

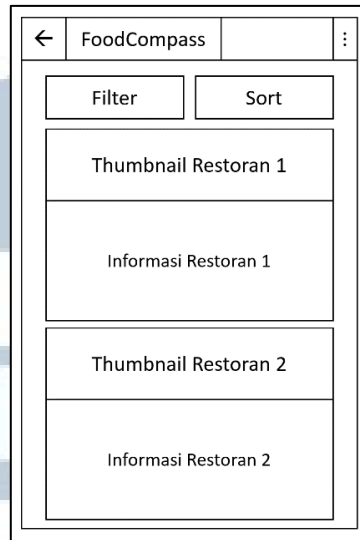


Gambar 3.10 Rancangan Tampilan Menu Utama Aplikasi

Gambar 3.10 adalah rancangan tampilan untuk menu utama aplikasi. Di bagian kiri *navbar* terdapat nama aplikasi, sementara di bagian kanan *navbar* terdapat *action bar*.

Terdapat juga logo aplikasi di bawah *navbar* dan tiga buah *button*. *Button* pertama untuk mencari restoran di sekitar lokasi *user*, *button* kedua untuk mencari restoran berdasarkan lokasi lain yang diinginkan, dan *button* ketiga untuk mengatur bobot dari kriteria yang digunakan untuk merekomendasikan restoran berdasarkan preferensi pribadi.

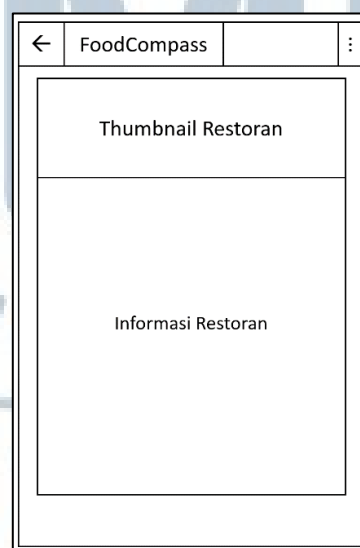
U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A



Gambar 3.11 Rancangan Tampilan Hasil Pencarian Restoran

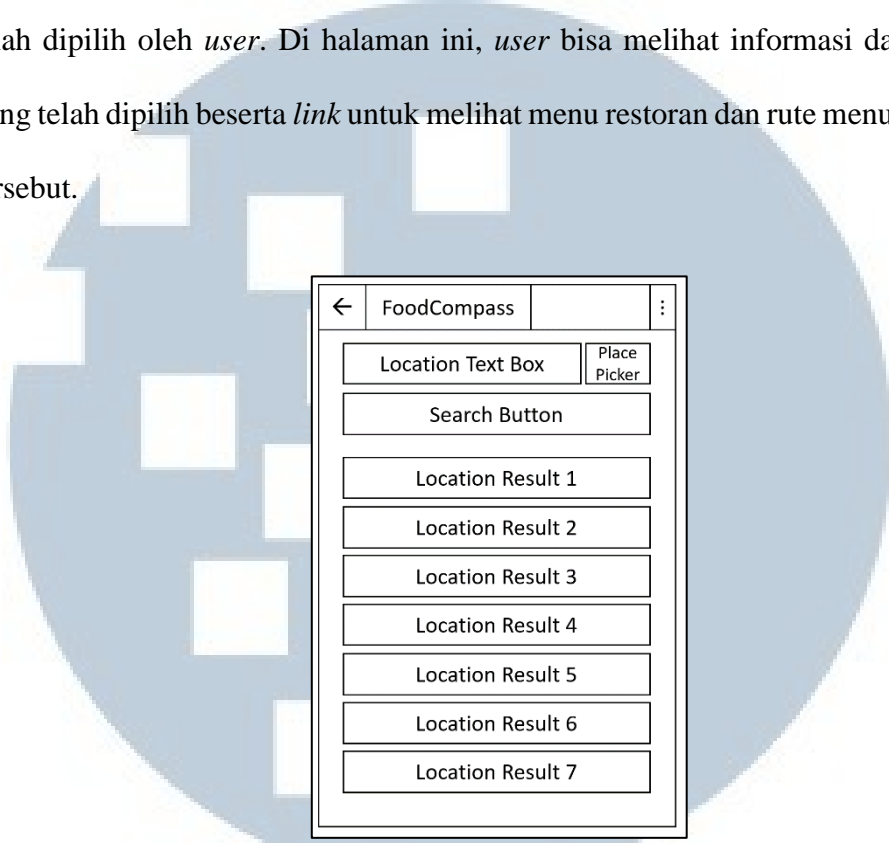
Gambar 3.11 merupakan rancangan tampilan hasil pencarian restoran. Di bagian paling atas terdapat dua *button*, *button* di bagian kiri adalah *button* untuk membuka halaman pengaturan filter, sementara *button* di kanan adalah *button* untuk mengubah urutan hasil rekomendasi restoran.

Di bawah kedua *button* tersebut terdapat *list* restoran. Setiap restoran memiliki *thumbnail* untuk gambar yang disediakan oleh Zomato API Service dan *text box* yang berisi informasi mengenai restoran tersebut.



Gambar 3.12 Rancangan Tampilan Detail Restoran

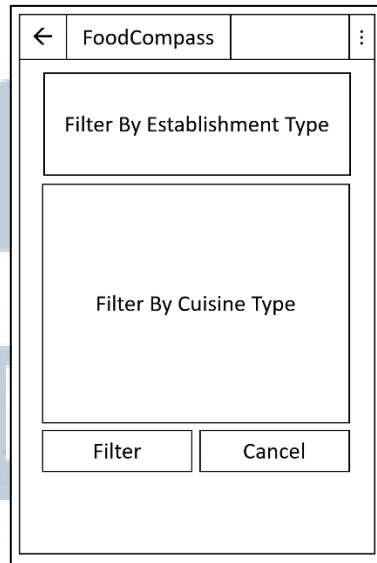
Gambar 3.12 merupakan rancangan tampilan untuk detail restoran yang telah dipilih oleh *user*. Di halaman ini, *user* bisa melihat informasi dari restoran yang telah dipilih beserta *link* untuk melihat menu restoran dan rute menuju restoran tersebut.



Gambar 3.13 Rancangan Tampilan Pencarian Lokasi

Gambar 3.13 merupakan rancangan tampilan untuk pencarian lokasi. Di halaman ini, terdapat *text box* untuk *user* memasukkan nama lokasi yang diinginkan dan juga *button* untuk memproses pencarian.

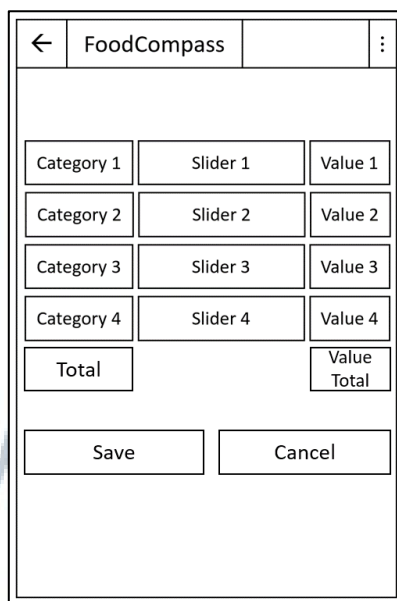
Di samping kanan *text box*, terdapat *button* yang berguna untuk menampilkan peta dari Google PlacePicker API Service, yang bisa digunakan oleh *user* untuk memilih lokasi di peta secara manual. Di bawah *button search* adalah *list* dari *landmark* Zomato API Service yang cocok dengan nama tempat yang dicari oleh *user*.



Gambar 3.14 Rancangan Tampilan Menu Filter

Gambar 3.14 adalah rancangan tampilan untuk menu filter. *User* dapat melakukan *filtering* berdasarkan dua hal, yaitu berdasarkan *establishment type* dan *cuisine type*.

Di bawah *filtering* berdasarkan *cuisine type*, ada dua *button*, yaitu *button filter* untuk memproses *filtering* dan *button cancel* untuk membatalkan.



Gambar 3.15 Rancangan Tampilan Pengaturan Bobot Preferensi

Gambar 3.15 merupakan rancangan tampilan pengaturan bobot preferensi. Di bagian kiri terdapat nama dari masing-masing kriteria, lalu terdapat *slider* yang dapat digeser oleh *user* untuk menentukan bobot dari kriteria tersebut, dan *text box* yang berisi angka dari bobot kriteria tersebut. *User* juga bisa mengubah nilai dari bobot kriteria yang diinginkan lewat *text box* tersebut.

Di bawah kategori keempat terdapat angka yang menunjukkan jumlah bobot dari keempat kriteria. Di bawah dua *button*, yaitu *button save* untuk menyimpan dan *button cancel* untuk membatalkan.

