



### **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 3

### METODOLOGI PENELITIAN

#### 3.1 Metode Penelitian

Dalam penelitian ini, terdapat beberapa tahap yang digunakan dalam perancangan dan pembangunan sistem rekomendasi restoran *all you can eat* menggunakan metode ELECTRE berbasis *mobile*, yaitu studi literatur, observasi awal, perancangan sistem, implementasi, serta pengujian sistem.

##### 1) Studi Literatur

Pada tahap ini, dilakukan pencarian mengenai referensi teori-teori pendukung penelitian, seperti *Haversine Formula*, Kategorisasi Ordinal, ELECTRE, Sistem Rekomendasi, *End User Computing Satisfaction (EUCS)*, dan skala Likert. Referensi dapat dicari melalui sumber jurnal, buku, artikel, dan internet. Tujuan dari studi literatur adalah sebagai dasar teori dalam menjalankan penelitian ini.

##### 2) Observasi Awal

Pada tahap ini, dilakukan perencanaan dan pengumpulan data awal yang nantinya akan digunakan dalam sistem rekomendasi. Aspek-aspek penelitian atau kriteria pemilihan restoran *all you can eat* didapatkan dari jurnal-jurnal penelitian yang telah meneliti faktor-faktor yang dapat mempengaruhi konsumen dalam memilih restoran *all you can eat*. Pengumpulan data restoran *all you can eat* di Tangerang sebanyak 35 restoran dimasukkan ke dalam basis data *firebase* yang didapatkan melalui situs ulasan restoran [www.pergikuliner.com](http://www.pergikuliner.com).

### 3) Perancangan sistem

Pada tahap ini, perancangan aplikasi yang terdiri dari pembuatan *data flow diagram*, *flowchart*, dan rancangan antarmuka aplikasi atau *mockup* dilakukan untuk memudahkan proses pembangunan aplikasi agar sesuai rencana.

### 4) Implementasi

Pada tahap implementasi, seluruh rancangan yang sebelumnya telah dibuat diimplementasikan ke dalam bentuk kode, sehingga akan menciptakan sebuah sistem yang dapat menentukan rekomendasi restoran *all you can eat* yang sesuai dengan bobot kriteria yang diinginkan pengguna.

### 5) Pengujian Sistem

Pengujian dilakukan untuk mendapatkan tingkat kepuasan pengguna terhadap sistem dengan cara menyebarkan kuisioner ke pengguna dengan model kuisioner menggunakan *End User Computing Satisfaction (EUCS)*, serta skala Likert.

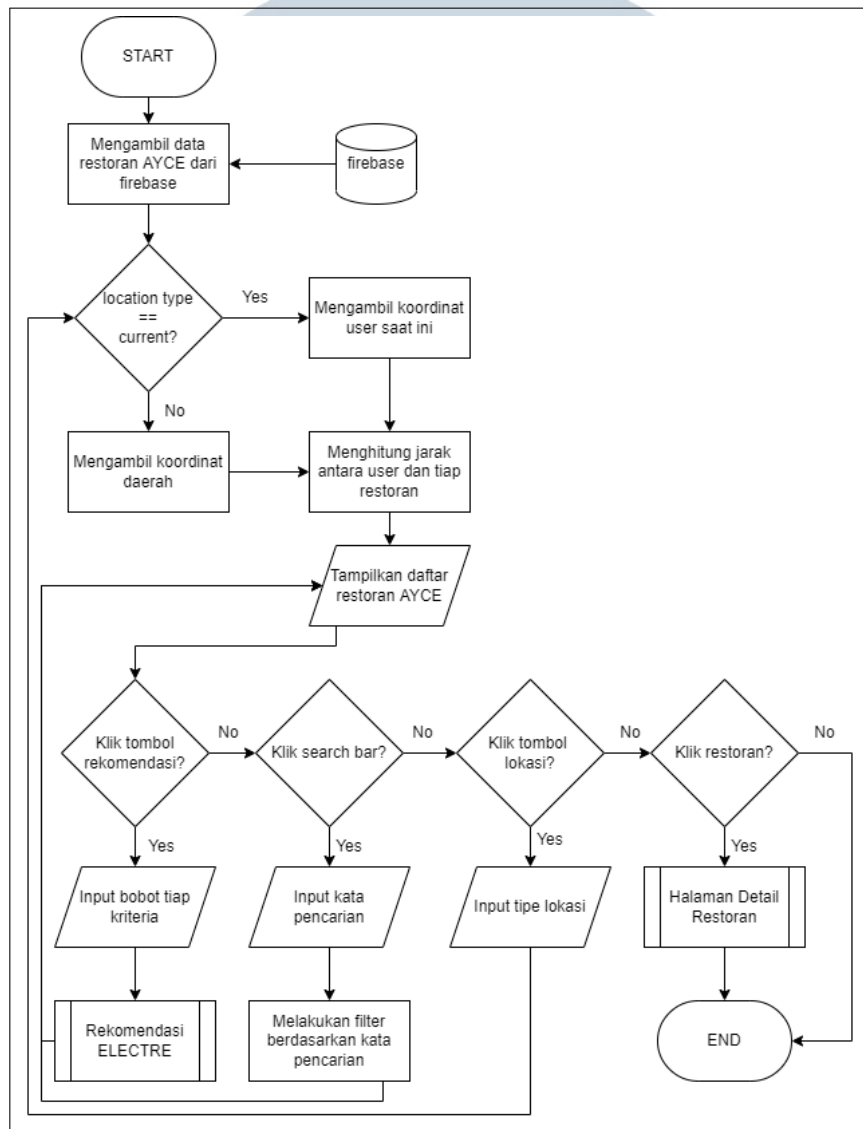
## 3.2 Perancangan Sistem

Perancangan sistem dalam penelitian ini dilakukan agar dapat memahami alur kerja dari sistem secara keseluruhan. Perancangan pada sistem yang akan dibangun dijelaskan dalam *flowchart*, *data flow diagram (DFD)*, objek diagram, serta rancangan antarmuka aplikasi (*mockup*).

### 3.2.1 Flowchart

Bagian ini menguraikan alur kerja dari rancang bangun sistem rekomendasi restoran *all you can eat* di Tangerang dengan metode ELECTRE yang telah digambarkan ke dalam sebuah *flowchart*. Adapun *flowchart* yang dibuat yaitu Halaman *Home*, Rekomendasi ELECTRE,

Halaman Detail Restoran, Halaman *Favorite*, Halaman *Login*, Halaman *Register*, Halaman *Profile*, Halaman *Help*.



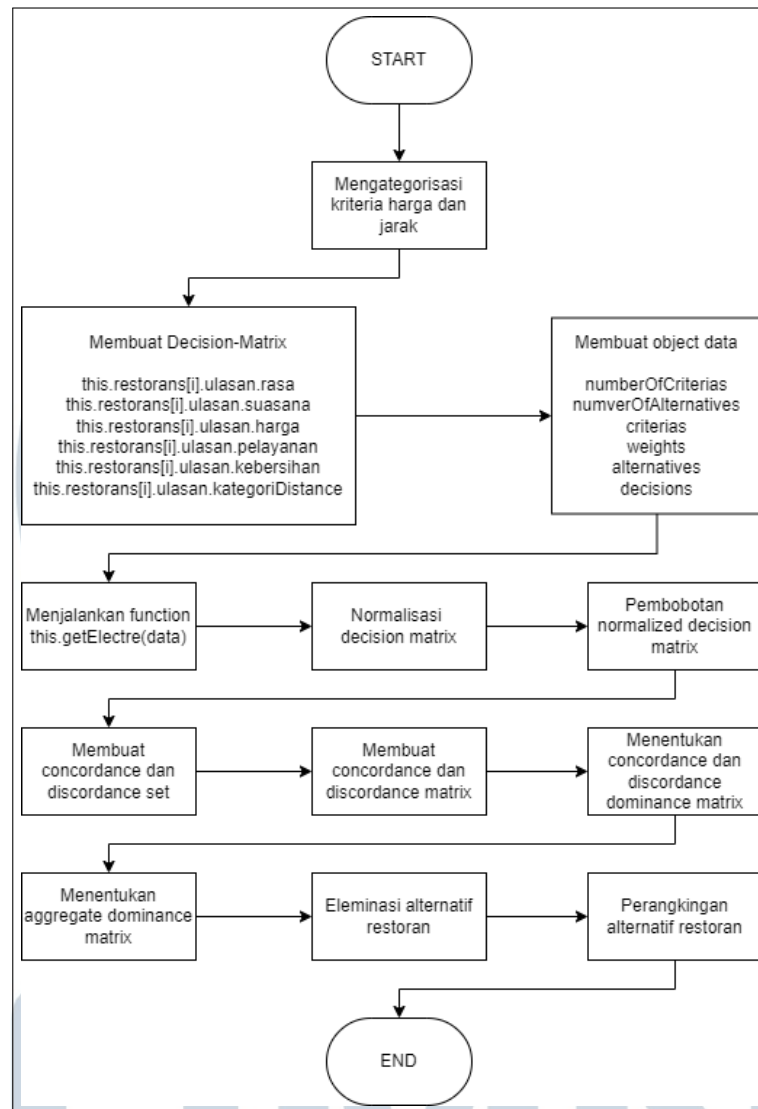
Gambar 3.1. Flowchart Halaman Home

Pada Gambar 3.1 merupakan alur kerja halaman *home*. Halaman ini merupakan halaman yang pertama kali dibuka saat *user* menggunakan aplikasi. Hal yang pertama kali dilakukan pada halaman ini yaitu mengambil data restoran AYCE dari *firebase*. Selanjutnya, jika tipe lokasi *user* menggunakan lokasi saat ini atau *current* maka sistem akan

menggunakan koordinat *user* saat ini menggunakan GPS, namun jika bukan sistem akan menggunakan salah satu koordinat dari 3 daerah yaitu Kota Tangerang, Kota Tangerang selatan, dan Kabupaten Tangerang. Setelah koordinat didapatkan, selanjutnya koordinat ini akan digunakan untuk menghitung jarak dari masing-masing restoran. Kemudian, sistem akan menampilkan daftar restoran AYCE dari data *firebase* sebelumnya.

Setelah itu apabila *user* meng-klik tombol rekomendasi maka *user* akan diminta untuk memasukkan bobot dari masing-masing kriteria, dimana bobot ini akan digunakan dalam perhitungan rekomendasi ELECTRE. Selanjutnya, apabila *user* meng-klik *search bar*, maka *user* akan diminta memasukkan kata pencarian kemudian sistem akan melakukan *filter* berdasarkan kata pencarian tersebut. Selanjutnya jika *user* meng-klik tombol lokasi, maka *user* akan diminta untuk memasukkan tipe lokasi. Kemudian, apabila *user* meng-klik salah satu restoran dari daftar restoran yang sudah ditampilkan sebelumnya maka *user* akan diarahkan ke halaman detail restoran tersebut.

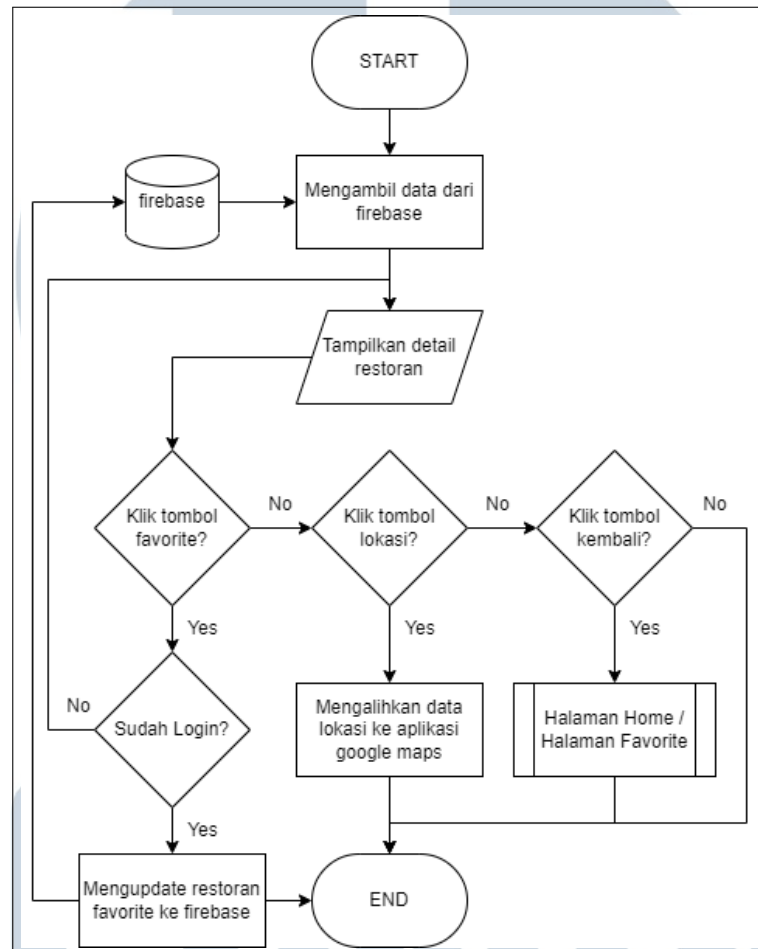
Pada Gambar 3.2 menjelaskan alur kerja dari rekomendasi ELECTRE. Setelah *user* memasukkan bobot tiap kriteria, sistem akan memulai perhitungan metode ELECTRE. Pertama sistem akan mengategorikan kriteria harga dan jarak pada tiap alternatif menggunakan metode kategorisasi ordinal, kemudian setelah mendapatkan kategori harga dan jarak tiap alternatif sistem akan membuat *decision matrix* dari kriteria rasa, suasana, kategori harga, pelayanan, kebersihan, dan kategori jarak.



Gambar 3.2. Flowchart Rekomendasi ELECTRE

Setelah *decision matrix* terbentuk sistem akan membuat data berupa objek yang berisi jumlah kriteria, jumlah alternatif, nama kriteria, bobot tiap kriteria, *key* dari tiap alternatif, dan *decision matrix* dimana data objek ini akan dikirimkan ke *function* ELECTRE. Selanjutnya, sistem akan menjalankan *function* *getElectre* dimana *function* ini akan menormalisasi *decision matrix*, melakukan pembobotan terhadap *normalized decision matrix*, membuat *concordance* dan *discordance set*, membuat *concordance* dan *discordance matrix*, menentukan *concordance* dan *discordance*

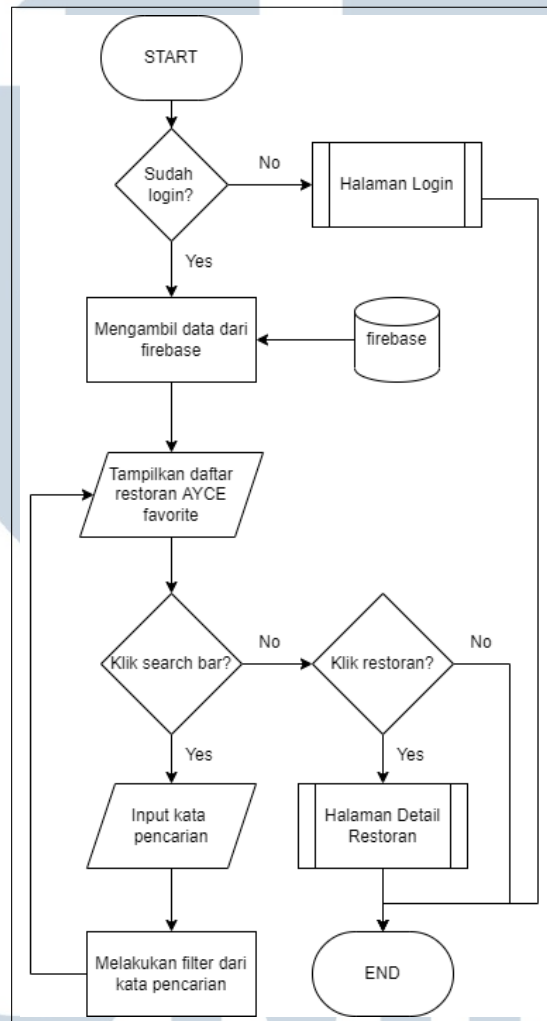
*dominance matrix*, menentukan *aggregate dominance matrix*, mengeleminasi alternatif restoran, dan terakhir melakukan perangkingan alternatif restoran dimana data ini akan ditampilkan pada halaman *home*.



Gambar 3.3. Flowchart Halaman Detail Restoran

Gambar 3.3 menguraikan alur kerja dari halaman detail restoran. Halaman ini dapat diakses apabila *user* meng-klik salah satu restoran di halaman *home* maupun *favorite*. Pertama sistem akan mengambil data dari *firebase*. Selanjutnya sistem akan menampilkan detail restoran seperti galeri foto, nama, ulasan, daerah, dll. Jika *user* belum *login* maka *user* tidak dapat meng-klik tombol *favorite* namun jika sudah maka sistem akan memperbaharui data restoran *favorite user* ke *firebase*. Kemudian, apabila

*user* meng-klik tombol lokasi maka sistem akan mengalihkan data lokasi ke aplikasi Google Maps. Setelah itu, apabila *user* meng-klik tombol kembali maka sistem akan mengalihkan *user* ke halaman yang memanggilnya.

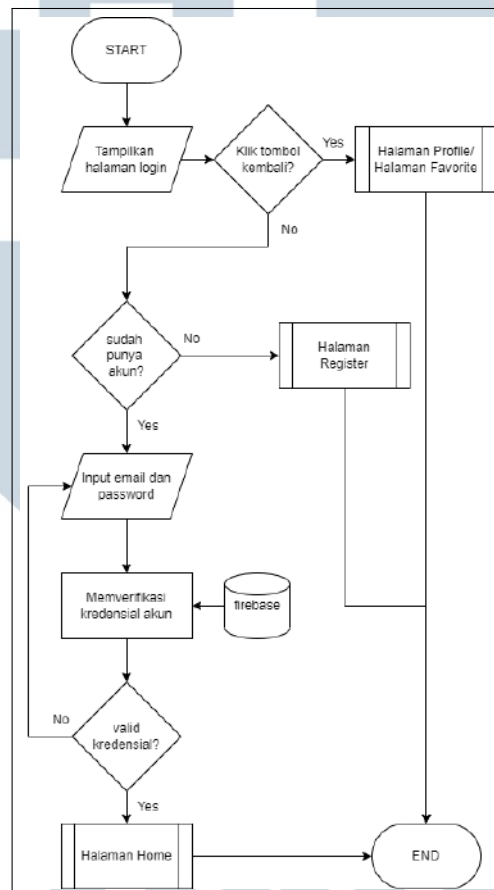


Gambar 3.4. Flowchart Halaman Favorite

Pada Gambar 3.4 menjelaskan tentang alur kerja dari halaman *favorite*. Pada halaman ini dilakukan validasi pertama jika *user* belum *login* maka *user* akan disarankan untuk *login* terlebih dahulu, namun jika sudah sistem akan mengambil data dari *firebase* berupa daftar restoran *favorite user*. Selanjutnya, sistem akan menampilkan daftar restoran *favorite user*. Kemudian, apabila *user* meng-klik *search bar* maka *user* akan diminta



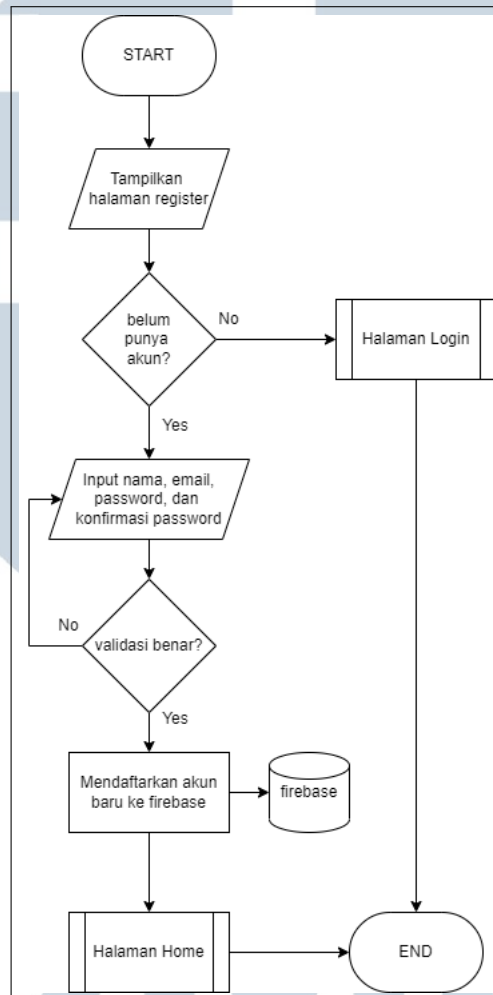
memasukkan kata pencarian yang nantinya sistem akan menggunakannya untuk melakukan *filter* berdasarkan kata pencarian tersebut. Setelah itu, apabila *user* meng-klik salah satu restoran maka sistem akan mengalihkan ke halaman detail restoran.



Gambar 3.5. Flowchart Halaman Login

Pada Gambar 3.5, merupakan alur kerja dari halaman *login*. Pada halaman *login*, hal yang pertama dilakukan yaitu menampilkan formulir *email* dan *password*. Jika *user* belum mempunyai akun dapat masuk ke halaman *register* untuk menambahkan akun, namun apabila sudah memiliki akun *user* dapat memasukkan *email* dan *password*-nya ke formulir *login*. Selanjutnya, sistem akan mengambil data dari *firebase* dan melakukan verifikasi kredensial akunnya valid atau tidak valid. Jika *email* dan

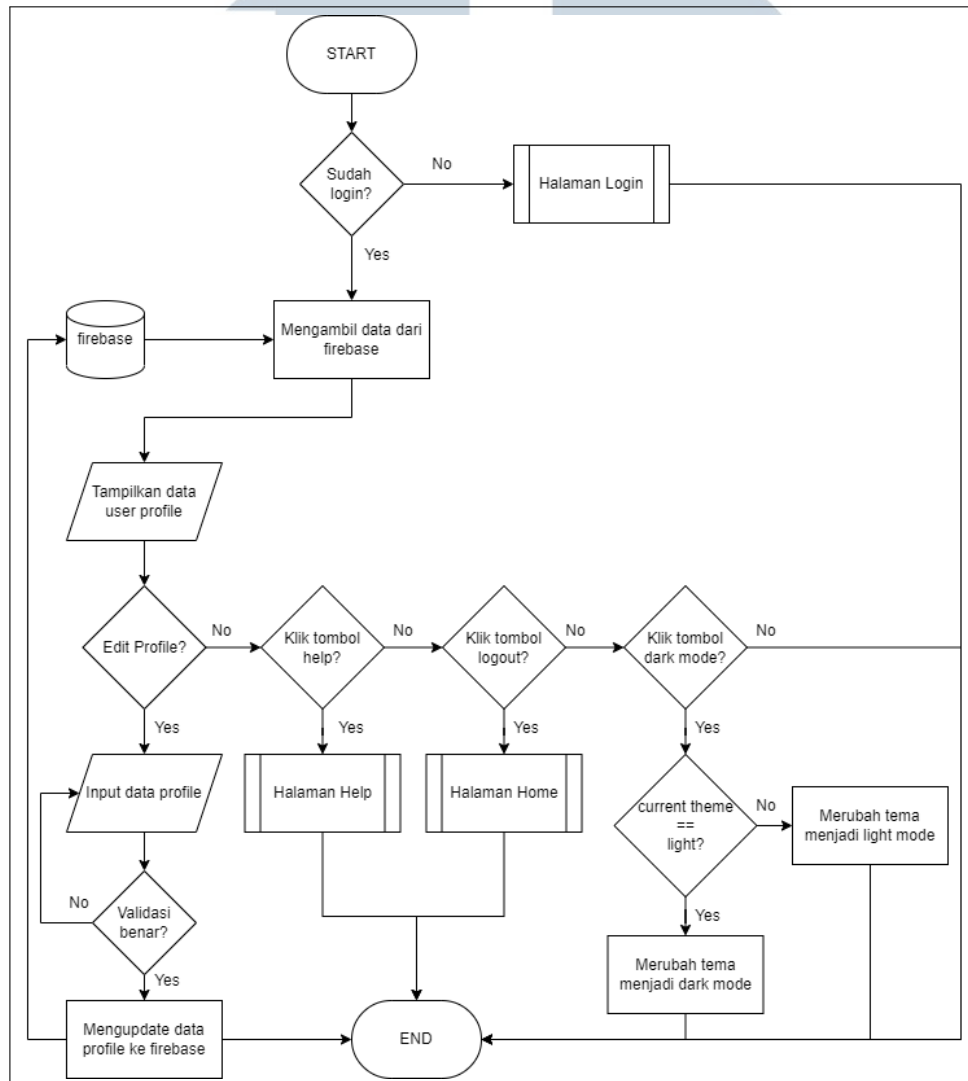
*password* tidak valid maka *user* akan diminta untuk memasukkan ulang *email* dan *password* yang valid, namun apabila valid akan diarahkan ke halaman *home*.



Gambar 3.6. Flowchart Halaman Register

Gambar 3.6 menjelaskan alur kerja halaman *register*. Halaman ini diakses ketika *user* belum memiliki akun atau ingin mendaftarkan akun baru. Hal pertama yang dilakukan yaitu menampilkan formulir *register* berupa nama, *email*, *password*, dan konfirmasi *password*. Selanjutnya, setelah *user* memasukkan data ke formulir, sistem akan memvalidasi data formulir terlebih dahulu. Jika validasi salah maka *user* akan diminta untuk

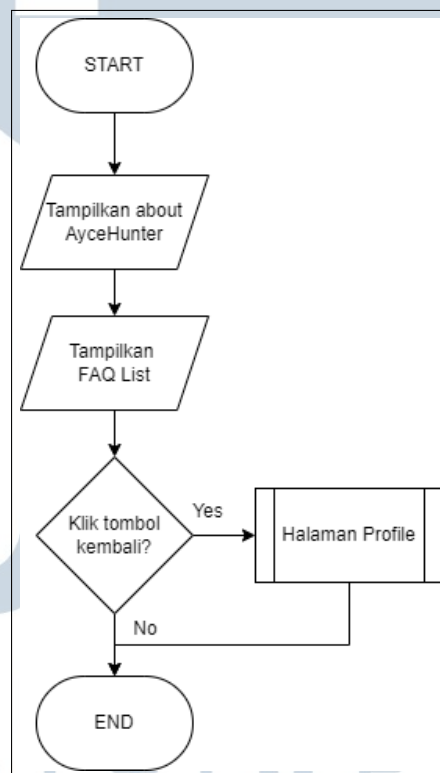
memasukkan formulir yang valid, namun apabila validasi benar maka *user* akan mendaftarkan akun baru ke *firebase*. Setelah sistem berhasil mendaftarkan akun, *user* akan diarahkan ke halaman *home*.



Gambar 3.7. Flowchart Halaman Profile

Gambar 3.7 menguraikan alur kerja di halaman *profile*. Pada halaman ini dilakukan validasi terlebih dahulu apakah *user* sudah *login* atau belum, jika *user* belum *login* maka *user* disarankan untuk *login* terlebih dahulu. Namun jika sudah *login*, sistem akan mengambil data dari *firebase* dimana data ini akan dipakai untuk menampilkan data *profile*.

Selanjutnya, apabila *user* melakukan *edit profile* maka *user* akan diminta memasukkan data *profile* yang ingin diganti. Kemudian, apabila validasi benar sistem akan melakukan *update* terhadap data *profile* ke *firebase*. Kemudian jika *user* meng-klik tombol *help* maka *user* akan dialihkan ke halaman *help*. Selanjutnya, jika *user* meng-klik tombol *dark mode* maka sistem akan melakukan pengecekan pada tema yang diaplikasikan, jika tema *light* maka sistem akan merubah tema ke *dark mode* sebaliknya apabila tema *dark* maka sistem akan merubah tema ke *light mode*. Terakhir, apabila *user* meng-klik tombol *logout* maka sistem akan mengalihkan *user* ke halaman *login*.



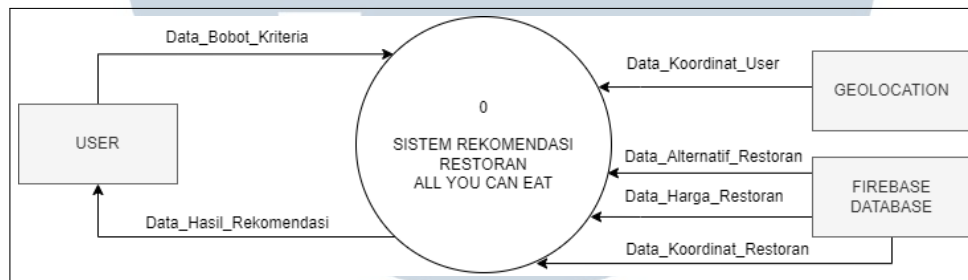
Gambar 3.8. Flowchart Halaman Help

Pada Gambar 3.8 merupakan alur kerja di halaman *help*. Pada halaman ini ditampilkan informasi tentang aplikasi ini yaitu bernama *ayce hunter*, lalu ditampilkan juga *FAQ list* atau list pertanyaan yang sering

ditanyakan. Kemudian jika *user* meng-klik tombol kembali, maka *user* akan diarahkan ke halaman *profile*.

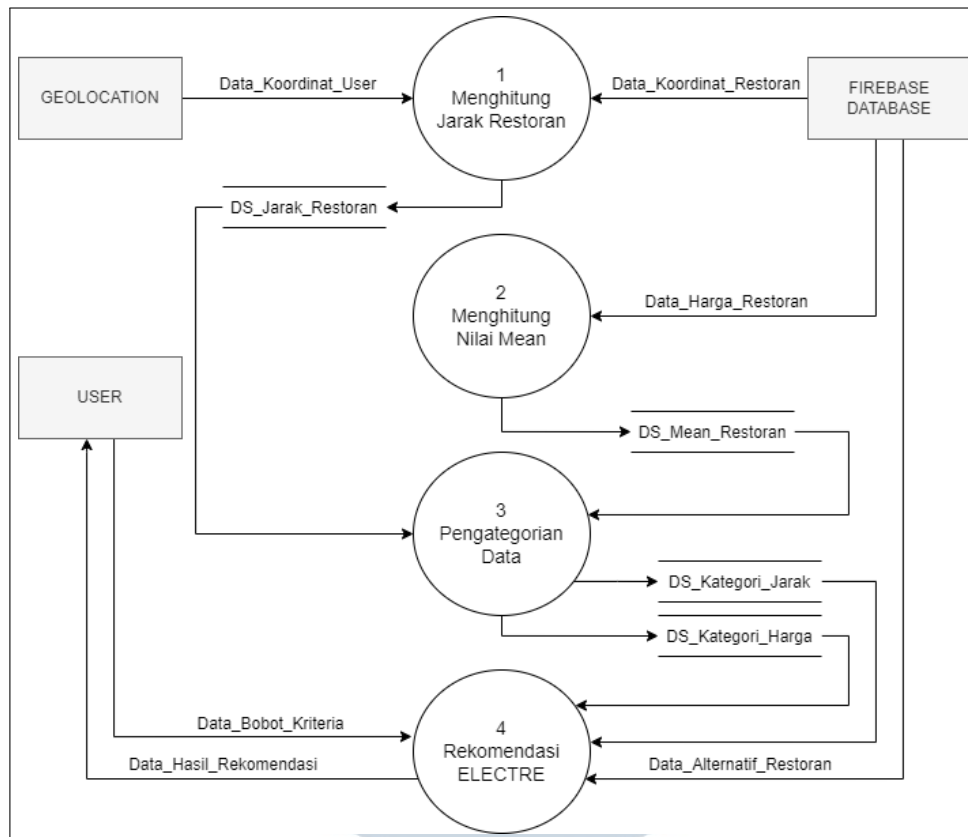
### 3.2.2 Data Flow Diagram

Pada bagian ini, akan menguraikan alur data sistem pada saat proses rekomendasi sedang terjadi. Alur akan direpresentasikan menggunakan *Data Flow Diagram*. Pada DFD level 0, terdapat tiga entitas, yaitu *Firestore Database*, *Geolocation* (GPS), dan *User*. Proses utama pada DFD level 0 yaitu sistem rekomendasi restoran *all you can eat*.



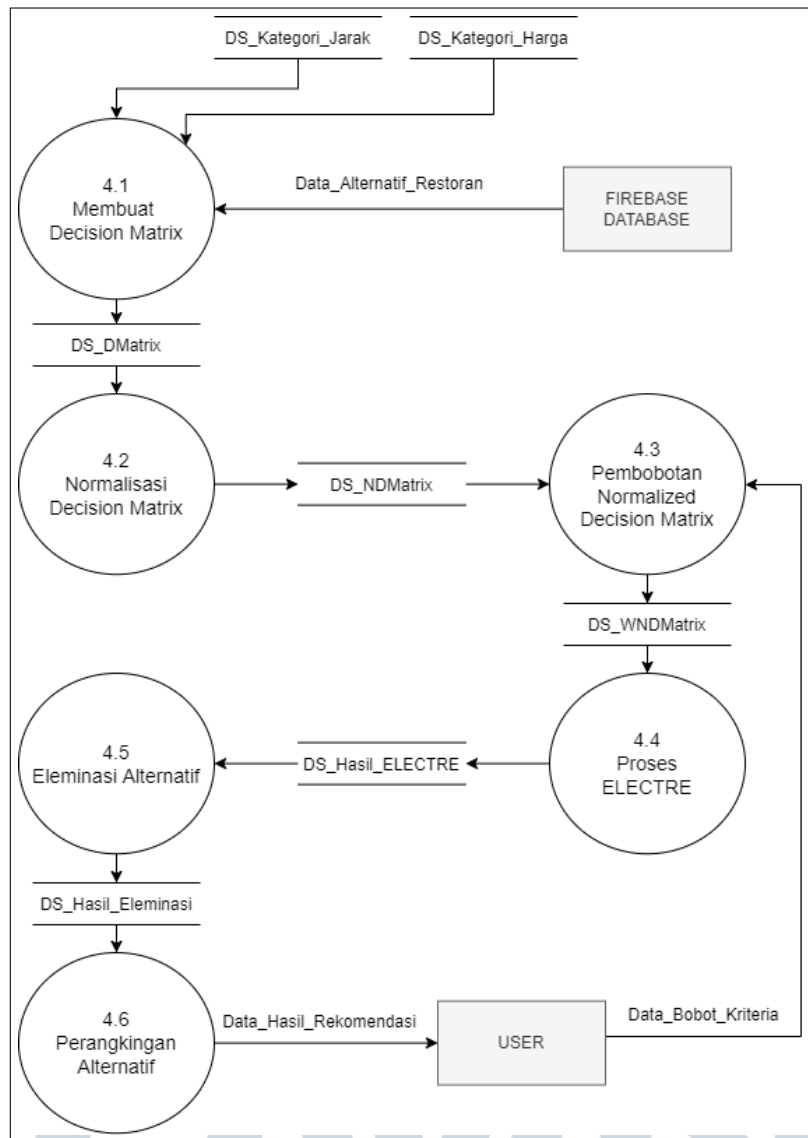
Gambar 3.9. Data Flow Diagram level 0

Gambar 3.9 menafsirkan bahwa entitas *user* memberikan data ke proses sistem rekomendasi berupa data bobot tiap kriteria. Selain itu entitas *geolocation* mengirimkan data berupa titik koordinat *user* saat ini dan entitas *firebase database* juga mengirimkan data alternatif restoran, data harga restoran, dan data koordinat restoran. Data-data ini nantinya akan digunakan pada proses sistem rekomendasi untuk mendapatkan data hasil rekomendasi restoran yang akan dikirimkan ke entitas *user*.



Gambar 3.10. Data Flow Diagram level 1

Gambar 3.10 menjelaskan alur data pada proses sistem rekomendasi restoran *all you can eat* secara detail. Pada level ini, proses sistem rekomendasi restoran *all you can eat* terbagi menjadi empat proses yaitu menghitung jarak restoran, menghitung nilai mean, pengategorian data, dan rekomendasi ELECTRE. Pada proses menghitung jarak restoran dibutuhkan data koordinat *user* dan restoran, sehingga akan menghasilkan data jarak restoran yang diberikan ke proses pengategorian data. Pada proses menghitung nilai mean, dibutuhkan data harga restoran sehingga menghasilkan data mean restoran yang diberikan ke proses pengategorian data. Pada proses rekomendasi ELECTRE dibutuhkan data kategori jarak dan harga, data alternatif restoran, serta data bobot kriteria sehingga menghasilkan data hasil rekomendasi yang diberikan ke *user*.



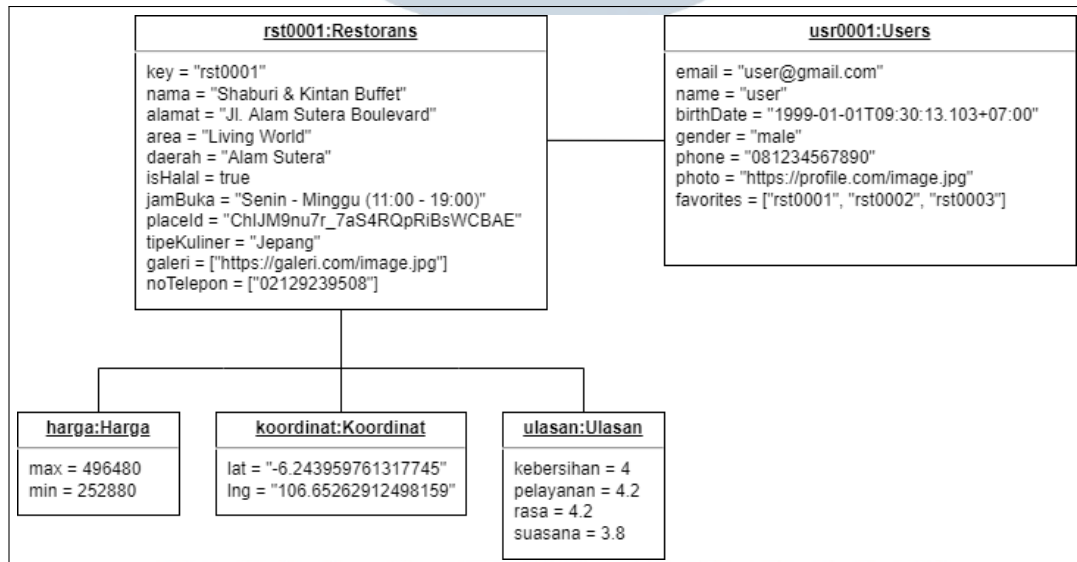
Gambar 3.11. Data Flow Diagram level 2 - Rekomendasi ELECTRE

Pada Gambar 3.11 menjelaskan detail yang terjadi pada proses rekomendasi menggunakan metode ELECTRE. Pada proses ini, data kategori jarak dan harga yang telah dibuat sebelumnya serta data alternatif restoran dibutuhkan untuk membuat sebuah *decision matrix* yang kemudian data yang dihasilkan akan diproses oleh normalisasi *decision matrix*. Setelah didapatkan data *normalized decision matrix*, langkah selanjutnya yaitu melakukan pembobotan pada *normalized decision matrix* dengan data

bobot kriteria yang diberikan *user*. Sehingga didapatkan data *weighted normalized decision matrix* yang digunakan dalam proses ELECTRE. Setelah proses ELECTRE selesai maka akan mengirimkan data hasil ELECTRE yang diberikan ke proses eliminasi alternatif sehingga menghasilkan data hasil eliminasi yang diberikan ke proses perankingan alternatif. Setelah proses perankingan alternatif selesai maka data hasil rekomendasi dapat dikirimkan ke entitas *user*.

### 3.2.3 Diagram Objek

Bagian ini akan menjelaskan struktur *database* sistem rekomendasi restoran *all you can eat* menggunakan diagram objek. Diagram objek merupakan diagram yang memperlihatkan objek-objek serta relasi antar objek yang ada pada suatu sistem [28].



Gambar 3.12. Objek Diagram

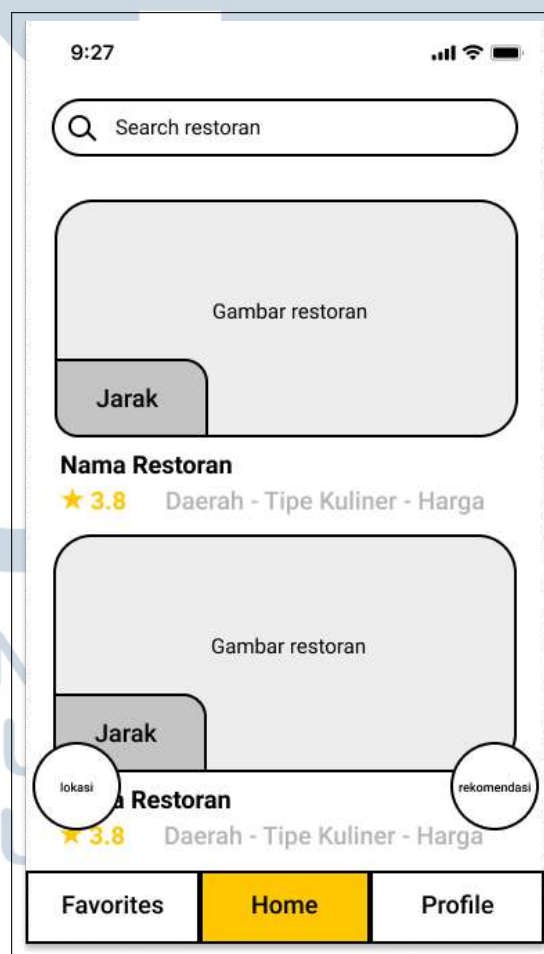
Gambar 3.12 merupakan struktur database yang ditafsirkan menggunakan objek diagram yang memiliki 5 tipe objek yaitu Restorans, Users, Harga, Koordinat, dan Ulasan. Objek rst0001 yang bertipe



Restorans memiliki 14 atribut, dengan 8 atribut bertipe *string*, satu atribut bertipe *boolean*, dua atribut bertipe *array of string*, dan tiga atribut bertipe objek. Sedangkan objek *usr0001* yang bertipe *Users* memiliki 7 atribut, dengan 6 atribut bertipe *string* dan satu atribut bertipe *array of string*.

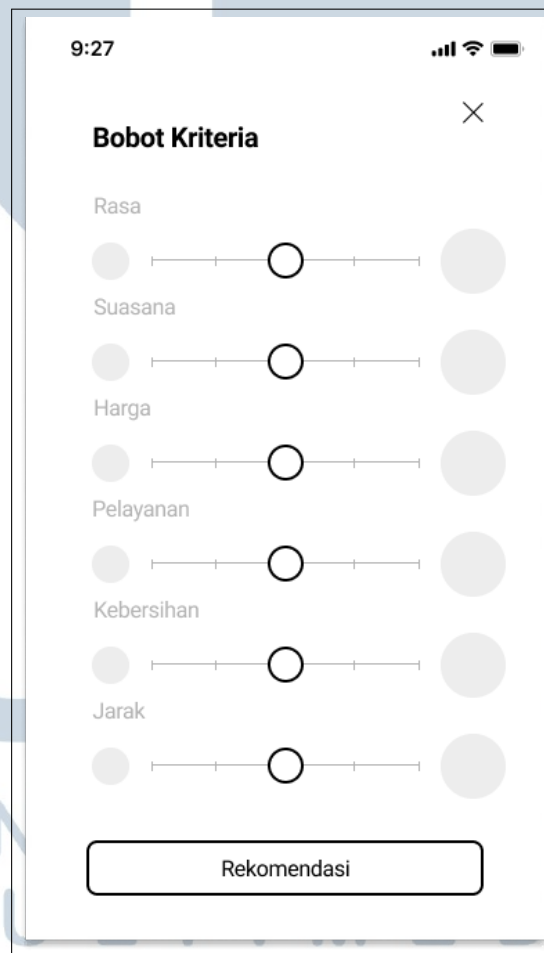
### 3.2.4 Rancangan Antarmuka Aplikasi

Bagian ini akan menerangkan rancangan antarmuka/tampilan sistem rekomendasi restoran *all you can eat* di Tangerang dengan metode ELECTRE. Rancangan tampilan pada aplikasi ini ada 7 yaitu rancangan tampilan halaman *home*, lokasi, rekomendasi, restoran detail, *favorite*, *profile*, *login*, *register*, *help*.



Gambar 3.13. Rancangan Tampilan Halaman Home

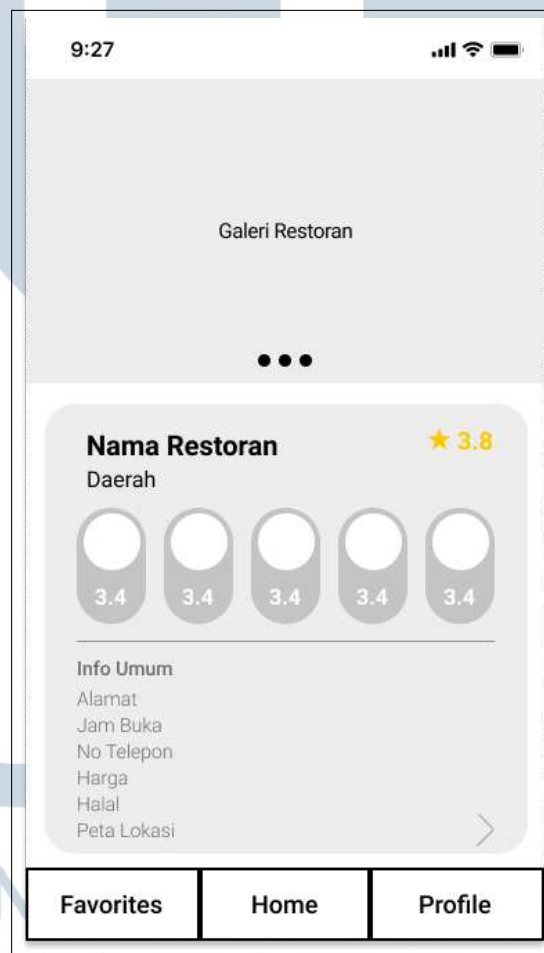
Gambar 3.13 merupakan rancangan tampilan halaman *home*. Pada halaman ini menampilkan daftar restoran berupa *card*, yang tiap *card* nya menampilkan data restoran yaitu gambar, jarak, nama, total ulasan, daerah, tipe kuliner, dan harga. Terdapat juga *tabs* untuk memudahkan *user* dalam melakukan navigasi halaman. Kemudian di bagian atas halaman terdapat *search bar* yang berguna sebagai *input user* untuk mencari restoran yang diinginkan. Selain itu, *floating action button* di sisi kanan bawah halaman berguna untuk menampilkan modal rekomendasi.



Gambar 3.14. Rancangan Tampilan Modal Rekomendasi

Modal rekomendasi ini berisi *inputan* bobot kriteria berupa *slider input* di tiap kriteria. Terlihat pada Gambar 3.14 terdapat *slider input* yang

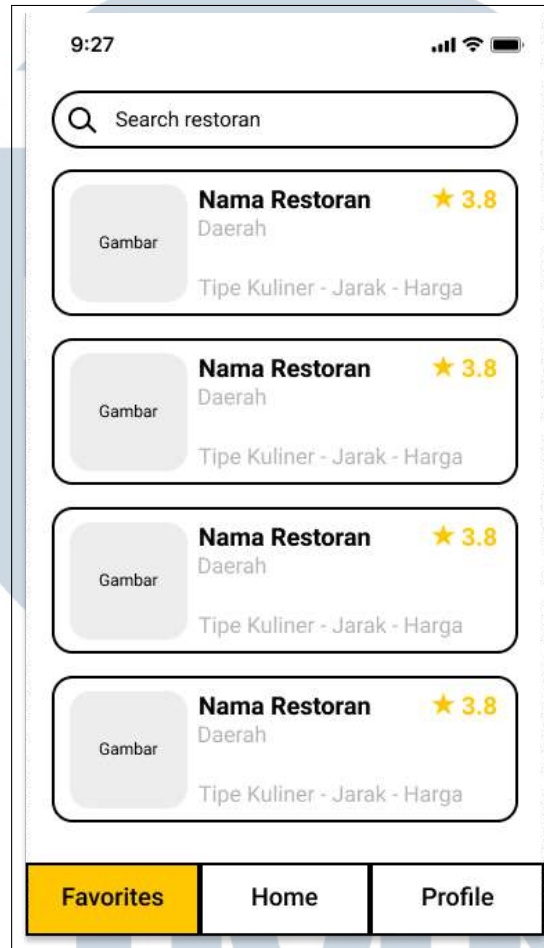
memiliki nilai minimum satu dan nilai maksimum lima. Adapula icon kecil dan icon besar di tiap kriteria melambangkan jika *slider* semakin ke kiri maka nilai bobot kriteria tersebut semakin kecil, sebaliknya jika *slider* semakin ke kanan maka nilai bobot kriteria tersebut semakin besar. Kemudian terdapat tombol silang di bagian atas kanan halaman untuk menutup modal rekomendasi dan juga terdapat tombol rekomendasi untuk memproses rekomendasi restoran *all you can eat*.



Gambar 3.15. Rancangan Tampilan Halaman Detail Restoran

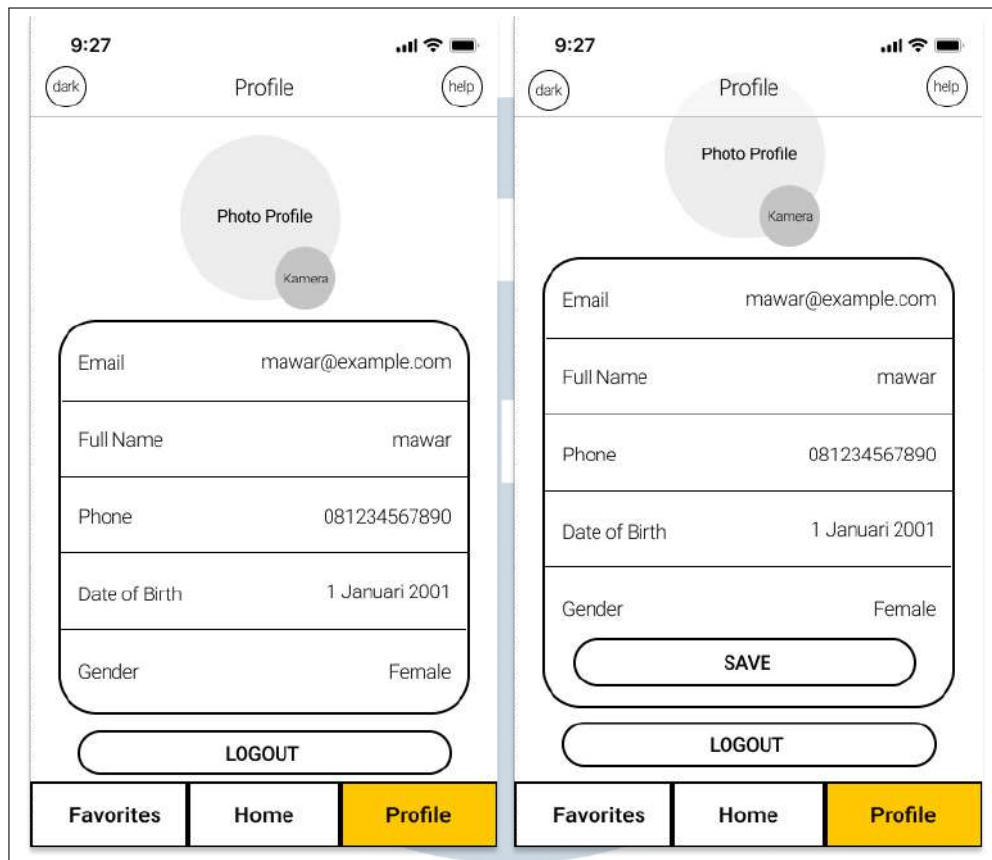
Pada Gambar 3.15 menunjukkan rancangan tampilan detail restoran. Pada halaman ini menampilkan detail restoran seperti galeri restoran berupa *image slides*, nama restoran, daerah, total ulasan, nilai ulasan tiap kriteria,

alamat, jam buka, nomor telepon, harga, halal, dan peta lokasi berupa tombol yang jika diklik *user* akan diarahkan ke aplikasi Google Maps.



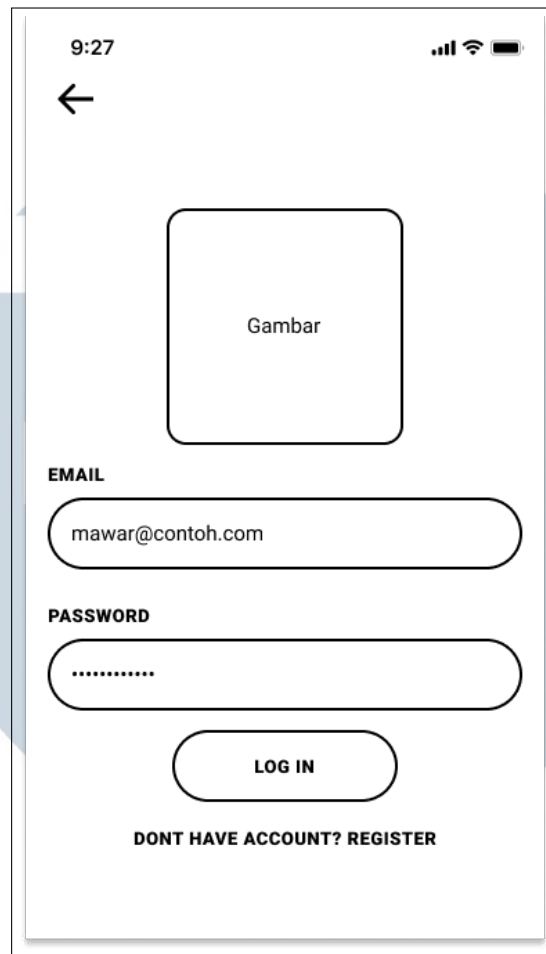
Gambar 3.16. Rancangan Tampilan Halaman Favorite

Gambar 3.16 merupakan rancangan tampilan halaman *favorite*. Pada halaman ini menampilkan daftar restoran *favorite user* yang dibentuk berupa *card*. Tiap *card* memiliki data restoran berupa gambar, nama, daerah, total ulasan, tipe kuliner, jarak, dan harga. Selain itu terdapat *search bar* di bagian atas halaman yang berupa *text input*.



Gambar 3.17. Rancangan Tampilan Halaman Profile

Pada Gambar 3.17 menunjukkan rancangan tampilan *profile*. Pada halaman ini menampilkan data *user* berupa *photo profile*, *email*, *full name*, *phone*, *date of birth*, dan *gender*. Pada Gambar 3.17 juga terdapat formulir yang memiliki fungsi selain untuk memasukkan data juga dapat menampilkan data, dimana jika *user* merubah data dalam formulir tersebut akan muncul tombol *save* untuk merubah data *profile*. Selain itu terdapat tombol *kamera* pada *photo profile* untuk merubah *photo profile*, kemudian terdapat tombol *dark mode* di bagian kanan atas halaman untuk merubah tema aplikasi, selanjutnya terdapat juga tombol *logout* yang jika diklik *user* akan diarahkan ke halaman *login*.



Gambar 3.18. Rancangan Tampilan Halaman Login

Gambar 3.18 merupakan rancangan tampilan halaman *login*. Pada halaman ini terdapat formulir yang memiliki dua buah *input*. Input pertama yaitu *email* dengan *type email* dan yang kedua *input password* dengan *type password*. Selain itu ada tombol *login* untuk melakukan proses otentikasi, apabila *user* belum memiliki akun, telah tersedia *link* yang jika diklik akan mengalihkan *user* ke halaman *register*. Kemudian terdapat gambar di bagian atas halaman untuk menambah nilai estetika aplikasi.

9:27

**REGISTER**

**NAME**

Mawar

**EMAIL**

mawar@example.com

**PASSWORD**

.....

**CONFIRM PASSWORD**

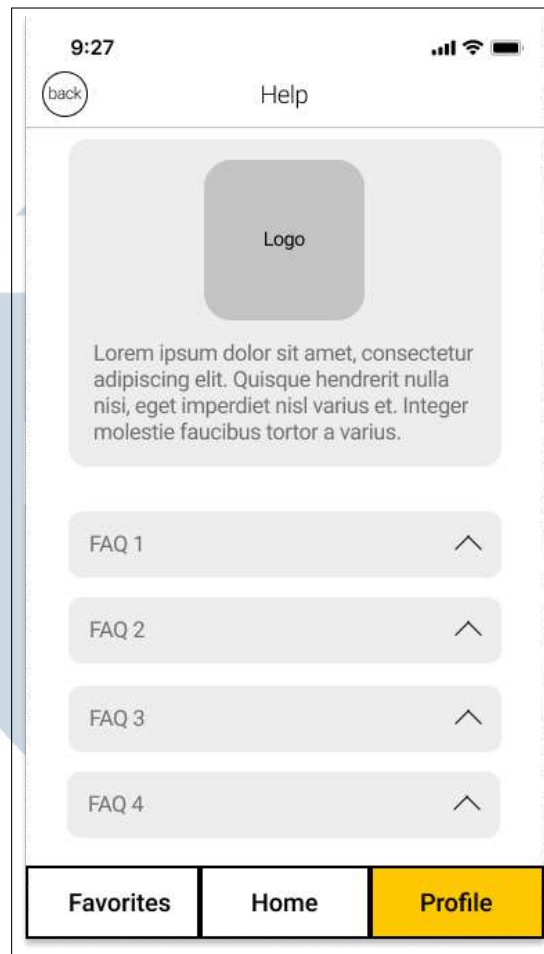
.....

**SIGN UP**

**ALREADY HAVE AN ACCOUNT? LOGIN**

Gambar 3.19. Rancangan Tampilan Halaman Register

Pada Gambar 3.19 menunjukkan rancangan tampilan *register*. Rancangan tampilan pada halaman ini memiliki formulir *register* yang berisi 4 *input* yaitu *name* dengan *type text*, *email* dengan *type email*, *password* dengan *type password*, dan *confirm password* dengan *type password*. Apabila *user* sudah memiliki akun tersedia *link* yang jika diklik akan diarahkan ke halaman *login*. Setelah *user* mengisi semua *inputan* pada formulir *register*, *user* dapat menekan tombol *sign up* untuk mendaftarkan akun baru.



Gambar 3.20. Rancangan Tampilan Halaman Help

Gambar 3.20 merupakan rancangan tampilan halaman *help*. Rancangan tampilan ini memiliki *card container* yang berisi informasi tentang aplikasi dan logo aplikasi, serta juga terdapat *accordion list* yang berisi informasi list pertanyaan yang sering ditanyakan.

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA