

## **BAB III**

### **METODOLOGI PENELITIAN**

Pada BAB III dijelaskan mengenai metodologi penelitian yang digunakan pada tugas akhir ini. Metodologi yang ada termasuk analisis kebutuhan sistem, desain aplikasi, dan nilai *entrepreneurship* dari tugas akhir ini.

#### **3. 1. Analisis Kebutuhan Sistem**

Analisis dilakukan untuk mengetahui kebutuhan aplikasi yang dirancang pada tugas akhir ini. Metode analisis yang digunakan adalah metode analisis data wawancara terhadap kebutuhan rekomendasi dan analisis perbandingan aplikasi serupa untuk kebutuhan aplikasi.

##### **3. 1. 1. Analisis Kebutuhan Rekomendasi**

Analisis ini dilaksanakan untuk mendapatkan kebutuhan rekomendasi pada sistem, khususnya mengenai kebutuhan kalori manusia dan makanan alternatif yang sesuai. Metode pelaksanaan ini adalah wawancara terhadap tiga orang *nutritionist* yang memiliki kualifikasi pekerjaan dalam bidang penelitian gizi dengan minimal gelar sarjana yang fokus pada kesehatan dan gizi. Informan *nutritionist* pertama menjabat sebagai *consultant nutritionist* selama empat tahun terakhir pada perusahaan X, *nutritionist* kedua menjabat sebagai ahli gizi dengan gelar A.Md.Gz. pada rumah sakit Y selama empat tahun terakhir, dan *nutritionist* ketiga menjabat sebagai dosen fakultas teknologi pangan pada universitas Z dengan fokus nutrisi selama dua tahun terakhir.

Wawancara dijalankan dengan aspek pertanyaan sebagai berikut:

- 1) Mengetahui kebutuhan kalori tubuh manusia.
- 2) Mengetahui kemungkinan adanya makanan alternatif lebih sehat yang didapatkan dari bahan makanan serupa.

Dua aspek pertanyaan tersebut merupakan fokus dari wawancara yang telah diadakan dengan *nutritionist* dengan aspek pertanyaan pertama untuk memastikan perhitungan kebutuhan kalori yang sesuai untuk tubuh manusia dan aspek pertanyaan kedua untuk memastikan rekomendasi hasil makanan alternatif cukup didapat dari kalori dan bahan makanan saja. Transkrip wawancara terlampir pada Lampiran B.

Hasil wawancara yang didapatkan dari setiap *nutritionist* sebagai berikut:

- 1) *Nutritionist* pertama menjawab kebutuhan kalori tubuh manusia dilihat dari kebutuhan energi yang diperlukan sehingga kebutuhan setiap orang berbeda, terutama orang sehat dengan orang berkebutuhan khusus. Persamaan TEE yang dapat menjawab kebutuhan kalori tersebut dan merupakan persamaan terbaik untuk perhitungannya. Untuk rekomendasi makanan alternatif, cukup fokus pada kalori makanan karena kalori adalah pengukuran pertama dalam makanan untuk banyak orang (dalam keadaan sehat). Untuk makanan sehat juga bisa didapatkan dari resep makanan sehat atau substitusi bahan makanan yang lebih sehat secara langsung tetapi tergantung preferensi orang.

- 2) *Nutritionist* kedua menjawab kebutuhan kalori tubuh manusia dapat dijawab melalui persamaan TEE karena sudah mempertimbangkan aspek kebutuhan energi dan tingkat aktivitas. Untuk makanan alternatif, lebih baik dipertimbangkan nutrisi lainnya seperti lemak, karbohidrat, dan protein untuk lebih baik karena komposisi setiap makanan berbeda. *Nutritionist* kedua ini menjawab dengan asumsi orang yang mendapatkan rekomendasi merupakan orang yang tidak memiliki kebutuhan khusus.
- 3) *Nutritionist* ketiga menjawab kebutuhan kalori tubuh manusia dapat dilihat melalui persamaan TEE secara umum. Untuk makanan alternatif, *nutritionist* ketiga setuju bahwa dilihat melalui kalori dan bahan makanan karena pertimbangan kalori memang cukup. Untuk menjaga kesehatan tubuh lebih lanjut, dapat orang dapat dianjurkan mengikuti konsep piring sehat, yaitu adanya variasi makanan pokok, lauk-pauk, buah-buahan, dan sayuran yang seimbang. *Nutritionist* ketiga ini menjawab dengan asumsi orang yang mendapatkan rekomendasi merupakan orang yang tidak memiliki kebutuhan khusus.

Kesimpulan hasil wawancara yang didapat untuk pertanyaan tersebut sebagai berikut:

- 1) Untuk pertanyaan pertama, ketiga *nutritionist* setuju bahwa persamaan yang digunakan sudah cukup sesuai untuk menghitung estimasi kebutuhan kalori yang dibutuhkan tubuh manusia.

Penelitian ini fokus terhadap orang dengan tingkat aktivitas minim dan tidak memiliki kebutuhan khusus atau mengidap penyakit apapun sehingga persamaan sudah mencantumkan faktor yang sesuai untuk perhitungan kalori.

- 2) Untuk pertanyaan kedua, dua dari ketiga *nutritionist* setuju bahwa hasil rekomendasi makanan alternatif cukup didapat melalui kalori dan bahan makanan serupa saja. Satu *nutritionist* yang tidak setuju dikarenakan jika makanan dinilai lebih sehat maka harus dipertimbangkan juga kandungan nutrisi makro dan mikro.

### **3. 1. 2. Analisis Kebutuhan Aplikasi**

Analisis ini dilaksanakan melalui komparasi tiga aplikasi sejenis untuk mendapatkan kebutuhan aplikasi yang dirancang. Aplikasi yang dipilih untuk komparasi merupakan aplikasi dengan fungsi perhitungan kalori, memiliki *rating* di Google Play Store minimal 4.5, dan merupakan aplikasi yang banyak dipakai orang berdasarkan jumlah unduhan sepuluh juta lebih. Aplikasi yang dipilih adalah aplikasi Calorie Counter by FatSecret, Calorie Counter – MyFitnessPal, dan Calorie Counter, Home Workout & Immunity Plan (HealthifyMe).

Aspek komparasi yang dilakukan fokus pada dua fitur yang juga diterapkan pada aplikasi yang dirancang, yaitu: 1) pengambilan informasi pengguna, yang ada sebagai fitur aplikasi untuk menghitung kebutuhan kalori pengguna, dan 2) tampilan pencarian makanan, yang ada sebagai fitur utama aplikasi untuk memberi masukan pilihan makanan yang akan diberi rekomendasi.

Hasil aspek komparasi aplikasi sebagai berikut:

- 1) Pengambilan informasi pengguna aplikasi
  - a. FatSecret: mengambil informasi tujuan pengguna, berat badan yang ingin dihilangkan, jenis kelamin, tingkat aktivitas pengguna, berat badan pengguna sekarang, tinggi badan, tanggal lahir, dan wilayah tempat tinggal pengguna.
  - b. MyFitnessPal: tujuan pengguna, tingkat aktivitas, jenis kelamin, tanggal lahir, wilayah tempat tinggal, berat badan, tinggi badan, *goal weight*, *weekly goal*, dan informasi akun user.
  - c. HealthifyMe: nama, jenis kelamin, nomor telepon, usia, berat badan, tinggi badan, tingkat aktivitas, kondisi medis, dan kota pengguna.
- 2) Tampilan pencarian makanan
  - a. FatSecret: nama makanan, porsi makanan, kalori makanan, dan kandungan RDI (*Recommended Dietary Intake*).
  - b. MyFitnessPal: nama makanan, brand makanan, porsi makanan, dan kalori makanan.
  - c. HealthifyMe: nama makanan, porsi makanan, dan kalori makanan.

Dari hasil komparasi yang didapat, pengambilan informasi pengguna difokuskan untuk perhitungan TEE sehingga mengambil informasi jenis kelamin, berat badan, tinggi badan, dan tanggal lahir pengguna. Informasi tersebut dipilih



karena fungsi aplikasi yang sudah dikhususkan untuk pemberian rekomendasi kepada pengguna dengan tingkat aktivitas minim dan fungsi aplikasi tidak mengacu untuk membantu penurunan berat badan pengguna.

Untuk pencarian makanan, sesuai hasil komparasi, informasi yang ditampilkan pada pencarian adalah nama makanan, deskripsi makanan sebagai pengganti porsi karena dalam penelitian makanan yang diambil merupakan makanan dengan data bahan makanan, dan kalori makanan.

### **3. 2. Desain Aplikasi**

Perancangan aplikasi telah dibuat dengan desain yang fokus untuk menyusun aplikasi sistem rekomendasi makanan alternatif berkalori lebih rendah berbasis konten menggunakan *hierarchical clustering* dalam tugas akhir ini.

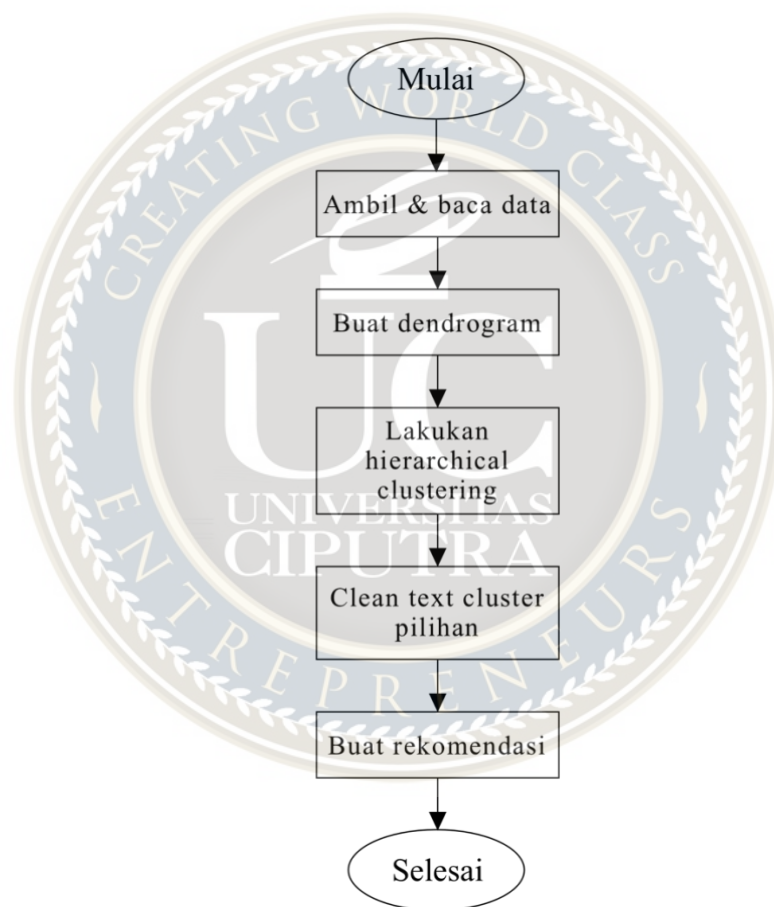
#### **3. 2. 1. Desain Sistem Rekomendasi Berbasis Konten dengan Hierarchical Clustering**

Dalam sistem rekomendasi ini, ada tiga sisi berhubungan, yaitu Python yang melaksanakan perhitungan dan pembuatan rekomendasi, Android yang mencari dan mengambil data makanan dari FatSecret API serta mengirim dan mengambil data makanan, dan PHP yang menangani pelembaran dan pengambilan data ke basis data di *localhost*.

##### **3. 2. 1. 1. Proses Hierarchical Clustering dan Content-Based Filtering di Python**

Sistem rekomendasi yang dirancang merupakan sistem rekomendasi dengan pendekatan *content-based filtering* yang menggabungkan algoritma *agglomerative hierarchical clustering* dan *content-based filtering* dengan parameter kalori dan

bahan makanan untuk membuat rekomendasi makanan. Jalannya sistem *clustering* dan *filtering* ini diawali dengan proses pengambilan dan pembacaan data untuk pembuatan dendrogram dan melaksanakan *agglomerative hierarchical clustering* agar dapat membuat rekomendasi dengan melakukan *clean text* dari *cluster* yang mengandung pilihan makanan pengguna. Jalan sistem dijelaskan pada subbab ini dan digambarkan melalui Gambar 3. 1.



**Gambar 3. 1** Jalan Proses Clustering dan Filtering di Python

### **3. 2. 1. 1. 1. Proses Pengambilan dan Pembacaan Data**

Proses pada *clustering* dan *filtering* dimulai dengan pengambilan data dari basis data makanan di *localhost*. Data yang diambil berupa keseluruhan data makanan yang mengandung ID dan nama makanan, ID dan nama bahan makanan,

kalori makanan, berat bahan makanan, dan deskripsi makanan. Data tersebut memiliki tipe objek *cursor* sehingga dimasukkan ke dalam sebuah DataFrame untuk memudahkan pembacaan dan proses data di Python.

### 3. 2. 1. 1. 2. Proses Pembentukan Dendrogram

Setelah data makanan didapatkan, nilai dari data makanan yang diambil untuk dibuat *cluster* adalah data bahan makanan, kalori makanan, dan berat makanan, yang merupakan nilai *numeric*. Dendrogram disusun melalui *library* *scipy* dengan parameter nilai data yang diambil dan metode *average linkage*, untuk membantu dalam penentuan jumlah *cluster* untuk parameter *hierarchical clustering*. Hasil dendrogram akan menunjukkan penggambaran awal *hierarchical clustering* dan dari data tersebut ditentukan terbentuk empat *cluster*.

### 3. 2. 1. 1. 3. Proses Agglomerative Hierarchical Clustering

*Hierarchical clustering* dilaksanakan oleh *function* *AgglomerativeClustering* dari *library* *sklearn*, untuk dapat membentuk kelompok data sesuai dari data makanan yang ada. *AgglomerativeClustering* diinisiasikan dengan pembentukan sebuah *instance* dari *hierarchical clustering* yang memiliki parameter jumlah kelompok yang didapat berdasarkan hasil dari dendrogram, yaitu empat, *Euclidean distance* sebagai jarak antar *point*, and *linkage average* atau *average distance* untuk memperhitungan jarak antar *cluster*. Akan diterapkan fungsi *\_labels* pada model yang terbentuk untuk akan mengembalikan *array* dari *integers* di mana nilai sesuai dengan kategori dalam pengelompokan, sehingga mengetahui data yang ada pada *cluster*. Nilai pengelompokan (*cluster labels*) yang didapat dari hasil *clustering* akan dimasukkan dalam kolom *cluster* yang ditampung



pada DataFrame baru untuk kemudahan pencarian kelompok yang sesuai dengan masukkan data makanan pengguna.

#### **3. 2. 1. 1. 4. Proses Content-Based Filtering Cluster Pilihan**

Pada proses ini, *cluster* yang mengandung pilihan makanan pengguna diambil untuk persiapan *content-based filtering*. Sebelum dilakukang *filtering*, proses *clean text* dilakukan untuk menghilangkan *stopwords* yang ada agar memudahkan pengukuran tingkat kesamaan antar kata. Pertama, dibentuk kolom deskripsi makanan, yang tersusun atas nama makanan dan bahan makan yang mengandung data bertipe *String*. Data tersebut lalu diproses dengan fungsi *clean\_text* yang menghilangkan karakter yang tidak diinginkan (*stopwords*). Data yang telah diproses lalu dimasukkan ke dalam kolom baru dengan nama *desc\_clean*. Untuk dapat dicari tingkat kesamaan antar makanan, data *desc\_clean* diterapkan perhitungan dengan TF-IDF (*Term Frequency-Inverse Document Frequency*) yang merupakan metode *library* sklearn untuk menghitung bobot setiap kata dalam dokumen dan *cosine similarity* untuk mendapatkan relevansi terdekat.

#### **3. 2. 1. 1. 5. Proses Pembuatan Rekomendasi**

Rekomendasi dibuat dengan mengambil hasil dari *processing data similarity* oleh TF-IDF dan *cosine similarity* lalu memasukkan variabel ID dari makanan pilihan pengguna sebagai *input* untuk menilai kesamaan. Setelah data rekomendasi didapatkan, data tersebut dimasukkan ke dalam DataFrame baru yang hanya mengandung satu kolom berisi ID dari makanan hasil rekomendasi, dan disimpan ke basis data di *localhost*.

### **3. 2. 1. 2. Proses FatSecret API serta Pelemparan Data di Android**

Dalam Android, dilakukan penerapan FatSecret API untuk pencarian dan pengambilan data makanan dari server FatSecret serta adanya pelemparan data makanan yang didapatkan ke server *localhost*.

#### **3. 2. 1. 2. 1. Proses Pencarian Data Makanan FatSecret**

Android melakukan pencarian data makanan FatSecret dengan metode pencarian dari FatSecret API yang diterapkan oleh metode *searchFood*. Pada metode *searchFood*, parameter yang digunakan adalah String *item*, yang merupakan *query* makanan dari pengguna, dan *int page\_num*, yang akan menentukan hasil makanan pada halaman dalam basis data FatSecret API yang dispesifikasikan. Metode ini akan berjalan ketika pengguna ingin mencari makanan pada aplikasi. Informasi yang akan didapatkan adalah data ID, nama, kalori, dan deskripsi makanan. Pencarian ini dilakukan dalam kelas *AysncTask* untuk melaksanakan pencarian data pada *background* agar aplikasi lebih responsif.

#### **3. 2. 1. 2. 2. Proses Pengambilan Data Makanan FatSecret**

Serupa dengan proses pencarian data, Android melakukan pengambilan data makanan FatSecret dengan metode pengambilan dari FatSecret API yang diterapkan oleh metode *getFood*. Pada pengambilan makanan, parameter yang digunakan adalah *long id*, yang merupakan ID makanan yang telah didapat pada hasil pencarian makanan, lebih tepatnya ketika pengguna memilih *item* dari daftar pencarian makanan di *ListView* halaman *Search*. Metode ini berfungsi untuk menampilkan informasi makanan lebih jelas seperti data bahan makanan dan berat

makanan. Pengambilan data ini juga dilakukan dalam kelas `AysncTask` untuk meringankan beban pada aplikasi.

### **3. 2. 1. 2. 3. Proses Pengiriman Data Makanan ke Server**

Pengiriman makanan ke *localhost* dilakukan dengan metode `InsertData` yang memiliki parameter semua data yang diperoleh dari hasil pengambilan makanan `FatSecret`, yaitu ID makanan dan bahan makanan, nama makanan dan bahan makanan, kalori, berat bahan makanan, deskripsi berat bahan makanan. Pengiriman ini dilaksanakan pada kelas `AsyncTask` untuk melaksanakan pengambilan data pada background agar aplikasi lebih responsive dan berfungsi untuk membangun basis data makanan dan mengirim makanan pilihan pengguna.

### **3. 2. 1. 2. 4. Proses Pengambilan Data Makanan dari Server**

Pengambilan makanan dari *localhost* dilakukan dengan metode `GetDataRec` yang memiliki parameter ID makanan. Metode `GetDataRec` dilaksanakan pada kelas `FoodFragment` untuk mengambil data hasil rekomendasi makanan setelah dilakukan *clustering* dan *filtering* oleh Python yang disimpan di server *localhost*. Metode ini juga berada dilaksanakan dalam kelas `AsyncTask` untuk meringankan eksekusinya.

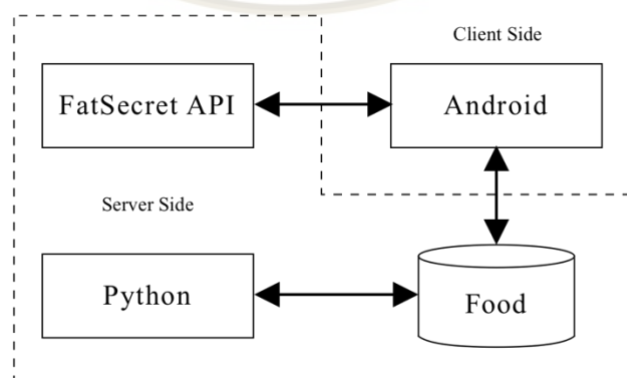
### **3. 2. 1. 3. Proses Sisi PHP**

PHP akan membantu Android dalam membangun koneksi server *localhost* untuk melakukan pengiriman dan pengambilan data pada Android. Dalam pengiriman data, PHP membantu dalam mengirimkan *query* yang mengandung data makanan untuk masuk pada tabel `food` yang menjadi basis data makanan keseluruhan dan *query* yang mengandung data makanan pilihan pengguna pada

tabel user yang memiliki parameter serupa tabel food dengan tambahan variable dt untuk menyimpan waktu pemasukan data. Dalam penyimpanan data, PHP membantu dalam mengirimkan *query* untuk mengambil data hasil rekomendasi berupa ID makanan dari tabel rec pada server *localhost* dari Python.

### 3. 2. 2. Desain Arsitektur

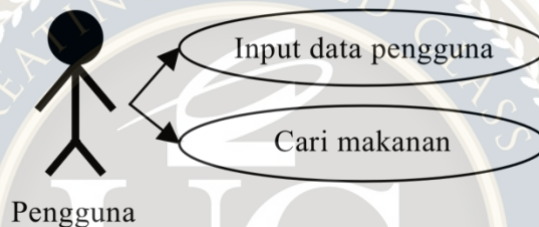
Pada aplikasi ini, Android adalah sisi klien yang akan menjadi perantara pengguna dengan aplikasi itu sendiri dalam koneksi ke sisi server, yaitu FatSecret API untuk pemanggilan basis data makanan, server *localhost* (Food) yang menyimpan basis data makanan dan makanan pengguna, serta Python yang menjalankan algoritma rekomendasi. Android memberikan *query* kepada FatSecret API untuk memanggil basis data makanan yang ada dan ditampilkan dalam aplikasi. Data makanan tersebut dan data makanan yang dipilih oleh pengguna akan dikirim ke *localhost* dengan bantuan PHP kemudian diteruskan ke Python untuk dilakukan *clustering* dan *filtering*. Server *localhost* ditujukan untuk menjadi tempat penampungan dalam pelepasan data yang diproses server ke klien. Desain yang telah dirancang dapat dilihat pada Gambar 3. 2 di bawah ini.



Gambar 3. 2 Desain Arsitektur Aplikasi

### 3. 2. 3. Use Case Diagram

Aplikasi ini dirancang untuk satu pengguna dan data informasi pengguna hanya disimpan pada aplikasi sehingga tidak ada admin lain. Dalam aplikasi ini, pengguna dapat memasukkan data berupa informasi jenis kelamin, berat badan, tinggi badan, dan tanggal lahir, serta pengguna juga dapat mencari makanan yang diinginkan untuk mendapatkan rekomendasi makanan serupa. Hal tersebut disampaikan oleh *use case diagram* pada Gambar 3. 3 di bawah ini.

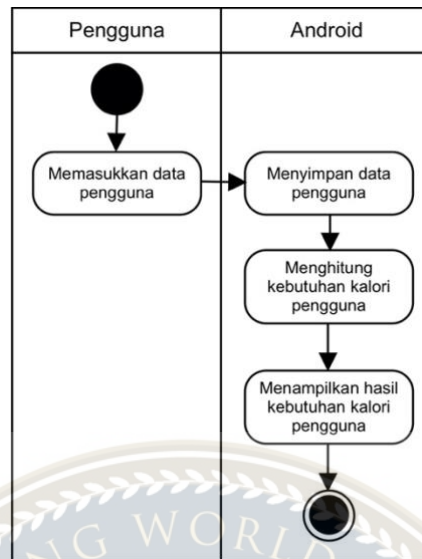


Gambar 3. 3 Use Case Diagram Aplikasi

### 3. 2. 4. Activity Diagram

Untuk input data pengguna, dimulai dengan pengguna memasukkan data pengguna dan akan disimpan oleh Android. Data pengguna yang ada akan diperhitungkan dalam Android untuk menampilkan hasil kebutuhan kalori pengguna. Alur yang ada digambarkan pada Gambar 3. 4 di bawah ini.

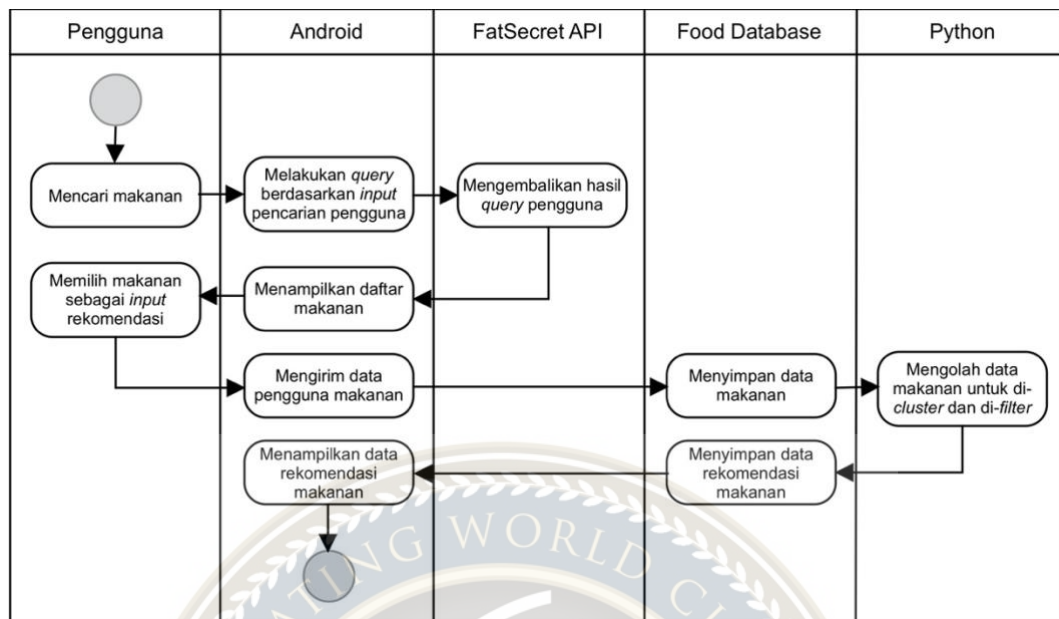




**Gambar 3. 4 Activity Diagram untuk Input Data Pengguna**

Untuk pencarian makanan, pengguna mencari makanan yang sesuai dengan apa yang diinginkan. Dalam Android, aplikasi akan mengirim *query* ke FatSecret API untuk menampilkan data makanan yang dapat dipilih pengguna. Pengguna melakukan pemilihan makanan sebagai parameter rekomendasi makanan. Android akan mengirimkan data pengguna ke *Food Database* berupa server *localhost* untuk mengirimkan data tersebut ke Python agar dapat dilakukan *clustering* lalu *filtering*. Hasil yang diperoleh akan dikirim ke *localhost* untuk diteruskan ke Android agar pengguna dapat melihat data rekomendasi makanan yang tersusun. Alur yang ada digambarkan pada Gambar 3. 5 di bawah ini.





Gambar 3. 5 Activity Diagram untuk Cari Makanan

### 3. 2. 5. Class Diagram

Aplikasi ini memiliki beberapa kelas, yaitu MainActivity, WelcomeFragment, GenderFragment, WeightFragment, HeightFragment, AgeFragment, HomeFragment, SearchFragment, FSSearch, FSGet, FoodRecipe, SearchAdapter, dan FoodFragment. Diagram kelas yang ada digambarkan pada Gambar 3. 6.

Penjelasan mengenai kelas-kelas yang ada pada aplikasi ini sebagai berikut:

1. MainActivity: kelas ini merupakan kelas utama dari aplikasi dan berfungsi sebagai *container* untuk setiap *fragment* yang ada. Pada kelas ini terdapat variabel context dengan tipe Context, sharedPreferences dengan tipe SharedPreferences, dan genderUser dengan tipe String. Variabel context dan sharedPreferences berfungsi untuk mengambil data yang disimpan dalam SharedPreferences Android, dan data tersebut ditampilkan dalam variabel genderUser. Dalam kelas ini diperiksa

apakah ada data yang tertampung sehingga jika pengguna telah menjalankan aplikasi ini sebelumnya, dan data pengguna telah tersimpan, maka MainActivity akan melakukan transaksi ke *fragment* HomeFragment. Jika pengguna baru pertama kali menjalankan aplikasi, maka MainActivity akan melakukan transaksi ke WelcomeFragment.

2. WelcomeFragment: memiliki variabel rootView dengan tipe View dan btnNext dengan tipe Button. Kelas ini memiliki metode onCreateView untuk menampilkan tampilan kelas dan metode onClick yang akan memulai transaksi ke *fragment* GenderFragment jika btnNext ditekan.
3. GenderFragment: memiliki variabel rootView dengan tipe View, nextBtn dengan tipe Button, rbGender dengan tipe RadioGroup, context dengan tipe Context, sharedPref dengan tipe SharedPreferences, dan editor dengan tipe SharedPreferences.Editor. Variabel context, sharedPref, dan editor berfungsi untuk menyimpan data dari variabel rbGender ke dalam SharedPreferences Android. Kelas ini memiliki metode onCreateView untuk menampilkan tampilan kelas, metode onCheckedChange untuk melihat apakah *radio button* dari rbGender telah dipilih, dan metode onClick yang akan memulai transaksi ke *fragment* WeightFragment jika nextBtn ditekan.

4. WeightFragment: memiliki variabel rootView dengan tipe View, nextBtn dengan tipe Button, edWeight dengan tipe EditText, context dengan tipe Context, sharedPred dengan tipe SharedPreferences, dan editor dengan tipe SharedPreferences.Editor. Variabel context, sharedPref, dan editor berfungsi untuk menyimpan data dari variabel edWeight ke dalam SharedPreferences Android. Kelas ini memiliki metode onCreateView untuk menampilkan tampilan kelas, metode onEditorAction untuk melihat apakah edWeight telah mengandung *user input*, dan metode onClick yang akan memulai transaksi ke *fragment* HeightFragment jika nextBtn ditekan.
5. HeightFragment: memiliki variabel rootView dengan tipe View, nextBtn dengan tipe Button, edHeight dengan tipe EditText, context dengan tipe Context, sharedPred dengan tipe SharedPreferences, dan editor dengan tipe SharedPreferences.Editor. Variabel context, sharedPref, dan editor berfungsi untuk menyimpan data dari variabel edHeight ke dalam SharedPreferences Android. Kelas ini memiliki metode onCreateView untuk menampilkan tampilan kelas, metode onEditorAction untuk melihat apakah edHeight telah mengandung *user input*, dan metode onClick yang akan memulai transaksi ke *fragment* AgeFragment jika nextBtn ditekan.
6. AgeFragment: memiliki variabel rootView dengan tipe View, nextBtn dengan tipe Button, tvAge dengan tipe TextView, context dengan tipe Context, sharedPred dengan tipe SharedPreferences,

dan editor dengan tipe `SharedPreferences.Editor`. Variabel `context`, `sharedPref`, dan editor berfungsi untuk menyimpan data dari variabel `tvAge` ke dalam `SharedPreferences` Android. Kelas ini memiliki metode `onCreateView` untuk menampilkan tampilan kelas, metode `onClickText` untuk menampilkan kalender, metode `onDateSet` untuk membaca tanggal yang didapat dari kalender, dan metode `onClick` yang akan memulai transaksi ke *fragment* `HomeFragment` jika `nextBtn` ditekan.

7. `HomeFragment`: memiliki variabel `rootView` dengan tipe `View`, `searchBtn` dan `changeUDBtn` dengan tipe `Button`, `imgUser` dengan tipe `ImageView`, `tvTEE` dan `tvAddInfo` dengan tipe `TextView`, `context` dengan tipe `Context`, `sharedPred` dengan tipe `SharedPreferences`, `tee` dengan tipe `Double`, `genderUser` dengan tipe `String`, serta `result`, `weightUser`, `heightUser`, dan `ageUser` dengan tipe `int`. Variabel `context` dan `sharedPref` berfungsi untuk mengambil data yang disimpan dalam `SharedPreferences` Android, dan data tersebut ditampung dalam variabel `genderUser`, `weightUser`, `heightUser`, dan `ageUser`. Kelas ini memiliki metode `onCreateView` untuk menampilkan tampilan kelas, metode `onClickChangeUD` yang akan memulai transaksi ke *fragment* `GenderFragment` jika `changeUDBtn` ditekan, dan metode `onClickSearch` yang akan memulai transaksi ke *fragment* `SearchFragment` jika `searchBtn` ditekan. Variabel dengan tipe `TextView` dan `ImageView` akan

digunakan untuk menampilkan data dari *input user*. Variabel *tee* dan *result* digunakan untuk melakukan perhitungan kebutuhan kalori dan menampung hasil.

8. SearchFragment: memiliki variabel *rootView* dengan tipe *View*, *listView* dengan tipe *ListView*, *backBtn* dengan tipe *Button*, *searchView* dengan tipe *SearchView*, *mFSSearch* dengan tipe *FSSearch*, *mFSGet* dengan tipe *FSGet*, *linkInsert* dengan tipe *String*, *adapter* dengan tipe *SearchAdapter*, *currentPage* dengan tipe *int*, dan *bundle* dengan tipe *Bundle*. Kelas ini memiliki metode *onCreateView* untuk menampilkan tampilan kelas, metode *onClick* untuk menjalankan transaksi ke *fragment* *HomeFragment* jika *backBtn* ditekan, metode *updateList* untuk mengatur *visibility* dari *ListView*, metode *searchImplementation* untuk mendapatkan metode *searchFood*, untuk mencari makanan, berdasarkan *QueryTextListener* *Android* dari *SearchView*, metode *getImplementation* untuk menjalankan metode *getFood* berdasarkan *Listener item* di *ListView*, metode *getFood* untuk mencari makanan dan menjalankan transaksi ke *fragment* *FoodFragment*, metode *InsertData* untuk mengirim data makanan ke basis data makanan di *localhost*.
9. FSSearch: kelas ini merupakan pembangunan koneksi dengan *FatSecret* API untuk pencarian data makanan. Terdapat variabel *appMethod*, *appKey*, *appSecret*, dan *appURL* yang bertipe *String*.



Kelas ini memiliki metode `getRecipe`, `generateOauthParams`, `sign`, `paramify`, `join`, dan `nonce`.

10. `FSGet`: kelas ini merupakan pembangunan koneksi dengan `FatSecret` API untuk pengambilan data makanan. Terdapat variabel `appMethod`, `appKey`, `appSecret`, dan `appURL` yang bertipe `String`. Kelas ini memiliki metode `getRecipe`, `generateOauthParams`, `sign`, `paramify`, `join`, dan `nonce`.

11. `FoodRecipe`: kelas ini berfungsi untuk menyimpan struktur data makanan yang akan diambil. Terdapat variabel `recipe_name`, `recipe_desc`, `recipe_id`, dan `recipe_cal` yang bertipe `String`. Kelas ini memiliki metode `FoodRecipe`, `getID`, `getName`, `getCalorie`, dan `getFoodDesc`.

12. `SearchAdapter`: kelas ini berfungsi sebagai *adapter* khusus untuk menampilkan data pada `ListView` aplikasi ini. Terdapat variabel `context` dengan tipe `Context` dan `mItem` dengan tipe `ArrayList<FoodRecipe>`. Kelas ini memiliki metode `SearchAdapter` dan `getView`.

13. `FoodFragment`: memiliki variabel `rootView` dengan tipe `View`, `tvRecipe` dan `tvCal` dengan tipe `TextView`, `backBtn` dan `btnRec` dengan tipe `Button`, `listView` dengan tipe `ListView`, `foodIDs` dengan tipe `String[]`, `mFSGet` dengan tipe `FSGet`, `linkGet` dengan tipe `String`, `adapter` dengan tipe `SearchAdapter`, `currentPage` dengan tipe `int`, dan `bundle` dengan tipe `Bundle`. Kelas ini memiliki metode



onCreateView untuk menampilkan tampilan kelas, metode onClickBack untuk menjalankan transaksi ke *fragment* HomeFragment jika backBtn ditekan, metode onClickRec untuk melakukan rekomendasi jika btnRec ditekan, dan metode GetDataRec untuk menjalankan rekomendasi.





### 3. 2. 6. Sequence Diagram

Dalam aplikasi ini, interaksi utama dimulai ketika pengguna membuka aplikasi pada pertama kalinya. *Class MainActivity* akan memulai transaksi ke *WelcomeFragment* yang dilanjutkan ke *GenderFragment* di mana pengguna harus mengisi data jenis kelamin. Setelah data diisi, pengguna masuk ke beberapa *fragment* serupa, yaitu *WeightFragment*, *HeightFragment*, dan *AgeFragment* yang diharuskan mengisi data ketika diminta pada tampilan halaman kelas tersebut.

Setelah data pengguna didapatkan, *HomeFragment* akan menampilkan kebutuhan kalori pengguna dan pengguna dapat mencari makanan. Pencarian makanan terjadi ketika *HomeFragment* menjalankan metode *onClickSearch* yang mengarah ke kelas *SearchFragment*. Pada kelas ini, pengguna dapat memasukkan data makanan yang dicari. *Query* dari pengguna akan dikirim ke *FSSearch* untuk membangun koneksi pencarian dengan *FatSecret API*. *SearchFragment* juga akan lanjut dengan mengambil data berdasarkan struktur dari kelas *FoodRecipe* sebelum data-data makanan dimasukkan dalam daftar makanan dengan *SearchAdapter*.

Dari daftar makanan hasil pencarian, pengguna dapat memilih satu makanan yang paling sesuai untuk dilakukan rekomendasi pada kelas *FoodFragment* ketika menekan tombol rekomendasi. Kelas tersebut akan memanggil kelas *FSGet* yang membangun koneksi pengambilan deskripsi data makanan dengan *FatSecret API*, yang akan dimasukkan ke *item FoodRecipe*. *FoodFragment* menjalankan *SearchAdapter* untuk masukkan data ke *ListView* untuk menampilkannya pada pengguna. Proses tersebut digambarkan pada Gambar 3. 7.



### 3. 2. 7. Desain Basis Data Makanan

Basis data makanan dengan nama food ditujukan untuk dapat menampung semua data makanan yang ada setelah diambil dari FatSecret. Basis data ini akan menyimpan data ID makanan (*recipe\_id*) sebagai *primary key*, nama makanan (*recipe\_name*), ID bahan makanan (*food\_id*), nama bahan makanan (*food\_name*), kalori makanan (*calories*), berat makanan (*units*), dan deskripsi berat makanan (*measurement*). Desain dari basis data makanan dapat dilihat pada Gambar 3. 8.

food						
<u>recipe_id</u>	recipe_name	food_id	food_name	calories	units	measurement

Gambar 3. 8 Desain Basis Data Makanan

### 3. 2. 8. Desain User Interface

Desain *user interface* yang ingin ditampilkan adalah desain yang sederhana tetapi jelas, dengan landasan dari hasil analisis kebutuhan yang telah dilakukan sebelumnya. Untuk rancangan aplikasi ini, ada empat desain utama, yaitu desain halaman *input user data*, *home*, *search*, dan *recommendation*.

Desain halaman *input user data* akan menggunakan desain serupa dengan aplikasi FatSecret untuk memudahkan tanggapan yang akan diberi pengguna yang fokus dengan pertanyaan untuk mendapatkan variabel perhitungan TEE (jenis kelamin, berat badan, tinggi badan, dan tanggal lahir). Pada halaman ini akan ada *TextView* untuk menampilkan pertanyaan, *EditText* sebagai kotak jawaban atau *RadioButton* sebagai pilihan jawaban untuk *input* pengguna, dan *Button* untuk menampilkan halaman selanjutnya. Gambar halaman *input user* ditampilkan pada Gambar 3. 9.



Desain halaman *home* akan menampilkan jumlah kalori yang dibutuhkan pengguna berdasarkan perhitungan TEE serta memberi akses langsung untuk mengubah data pengguna dan mencari makanan untuk mendapatkan rekomendasi. Pada halaman ini, akan ditampilkan gambar jenis kelamin sebagai deskripsi utama data pengguna melalui `ImageView`, serta hasil TEE dan keterangan banyak kalori yang dapat dikonsumsi tiap makan melalui `TextView`. Dua fitur aplikasi juga dimunculkan dalam bentuk `Button` untuk memudahkan akses pengguna. Gambar desain halaman *home* ditampilkan pada Gambar 3. 10.



Gambar 3. 9 Desain Halaman Input User    Gambar 3. 10 Desain Halaman Home

Desain halaman *search* fokus untuk menampilkan halaman pencarian makanan, dengan tampilan daftar hasil makanan yang mengandung nama makanan, deskripsi makanan, dan kalori makanan. Untuk memberi tampilan yang jelas, daftar makanan akan ditampilkan dengan `ListView`. Gambar desain halaman *search* ditampilkan pada Gambar 3. 11.



Desain halaman *recommendation* fokus untuk menampilkan makanan pilihan yang dicari pengguna serta hasil rekomendasi berupa daftar makanan serupa. Pada halaman ini akan fokus pada rekomendasi sehingga akan ada *TextView* untuk menampilkan data makanan yang dipilih dan *ListView* untuk menampilkan hasil rekomendasi secara langsung. *Button* dengan nama *recommendations* berfungsi untuk menampilkan hasil rekomendasi ke *ListView* dan *Button Go back to Home* berfungsi untuk kembali ke halaman *home*. Gambar halaman *recommendation* ditampilkan pada Gambar 3. 12.



**Gambar 3. 11 Desain Halaman Search – Gambar 3. 12 Desain Halaman Recommendation**

### **3. 3. Nilai Entrepreneurship**

Dalam pengembangan aplikasi tugas akhir ini, ada beberapa faktor yang harus dipertimbangkan agar aplikasi yang dirancang dapat berjalan menjadi suatu bisnis yang baik dan berkelanjutan.

#### **3. 3. 1. Opportunity Creation**

Dengan perkembangan teknologi sekarang yang memudahkan masyarakat dalam menjaga pola hidup sehat, aplikasi yang fokus pada kepuasan rasa makanan

sekaligus memberi alternatif lebih sehat masih jarang terlihat. Banyak aplikasi mengenai makanan sehat lebih fokus akan pencatatan makanan yang dimakan dan pencarian resep atau makanan sehat itu sendiri untuk menjaga kesehatan tubuh. Akan tetapi, belum ada aplikasi yang memberikan pilihan alternatif ketika seseorang ingin menikmati makanan yang seharusnya tidak baik buat tubuh seperti makanan cepat saji.

Untuk itu, aplikasi yang dapat memberikan rekomendasi makanan alternatif berkalori lebih rendah dengan bahan yang serupa dapat memuaskan orang yang masih ingin menikmati makanan kesukaannya tetapi juga mengarah ke pola hidup yang lebih baik.

### **3. 3. 2. Creativity and Innovation**

Aplikasi ini memberikan pilihan makanan alternatif dengan kalori lebih rendah dan bahan serupa bagi orang yang memiliki gaya hidup dengan tingkat aktivitas minim dan ingin mengonsumsi makanan enak yang biasanya memiliki kalori tinggi. Selain dapat mencari berbagai makanan dari FatSecret, aplikasi ini juga dapat memberikan rekomendasi berdasarkan masukkan makanan yang dicari pengguna. Ada kesempatan terbuka bagi orang untuk lebih mudah menjalankan pola hidup sehat tanpa membatasi makanan tertentu dalam faktor rasa. Kemudahan akses informasi tersebut juga disajikan dalam aplikasi *mobile* sehingga pengguna dapat langsung menerima informasi yang ada dengan jelas.

### **3. 3. 1. Market Sensitivity**

Sebagian besar orang cenderung untuk tidak mempertahankan hidup yang sehat karena banyaknya variasi makanan yang ingin dikonsumsi atau anggapan

bahwa sehat berarti tidak enak atau hambar. Maka dari itu, aplikasi sistem rekomendasi ini difokuskan untuk segmen pengguna dengan aktivitas minim dan yang peduli akan jumlah kalori yang dikonsumsi oleh tubuh agar dapat membantu pengguna untuk memudahkan hidup sehat.

Untuk penyebaran aplikasi, aplikasi ini akan di-*upload* ke Google Play Store dan dilakukan *marketing* di media sosial, khususnya Instagram dan Facebook untuk menarik perhatian orang dan kemudahan penyampaian. Aplikasi ini juga direncanakan untuk mendapat penghasilan dari *in-app advertising* atau langsung menjual aplikasi itu sendiri (*paid app*).

