



### **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

### METODE DAN PERANCANGAN APLIKASI

#### 3.1 Metode Penelitian

Metode-metode yang digunakan dalam penelitian ini adalah sebagai berikut.

##### 1. Analisis Literatur

Analisis literature adalah mencari teori dari buku, jurnal, artikel, maupun referensi lainya yang tersedia secara *online* maupun *offline* yang dapat membantu dalam penelitian.

##### 2. Perancangan sistem

Perancangan sistem akan dilakukan dengan merancang tampilan untuk pengguna serta fitur-fitur apa saja yang perlu disediakan untuk mempermudah atau membuat nyaman pengguna saat menggunakan aplikasi.

##### 3. Pemrograman Sistem

Pemrograman sistem adalah penggabungan antara fitur-fitur yang telah ditentukan dengan tampilan yang telah dibuat, sehingga ketika pengguna melakukan sesuatu terhadap tampilan aplikasi, maka aplikasi akan melakukan suatu pekerjaan tertentu.

##### 4. Testing dan Debugging Aplikasi

*Testing* dan *debugging* aplikasi dilakukan seiring dengan pembangunan aplikasi sehingga dapat mengurangi kesalahan-kesalahan yang tidak diinginkan.

##### 5. Survei

Survei dilakukan sesudah aplikasi berhasil dibuat dengan tujuan mengumpulkan sampel data. Pengumpulan sampel data dilakukan dengan

meminta sejumlah responden untuk menggunakan aplikasinya yang telah dibuat dan mengisi kuisioner.

#### 6. Analisis Sampel Data

Setelah sampel data terkumpul, maka dapat dilakukan analisis sampel data untuk mengetahui apakah aplikasi yang dibuat sudah memenuhi manfaat penelitian.

#### 7. Penulisan Laporan

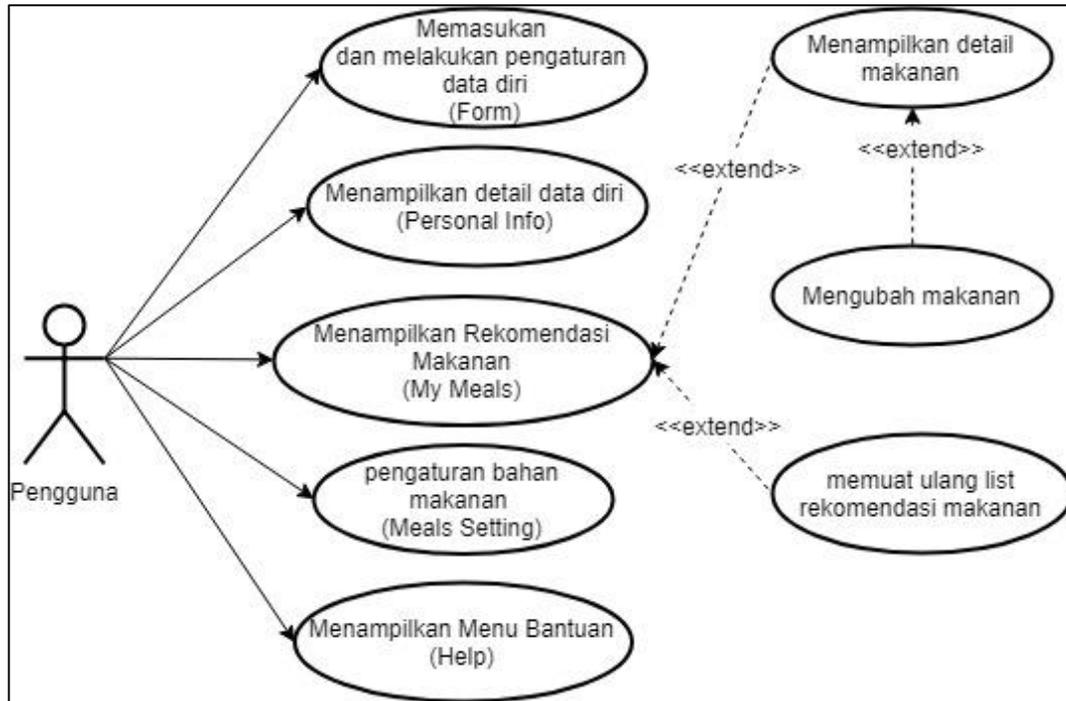
Penulisan laporan berguna untuk membuat dokumentasi dari suatu penelitian serta memberikan informasi terhadap peneliti selanjutnya dalam penelitian sejenis.

### 3.2 Rancangan Aplikasi

Sistem yang dirancang berorientasi *object* sehingga pemodelan yang digunakan *Unified Modeling Language* (UML). UML merupakan metode yang menggunakan grafis serta merupakan bahasa untuk visualisasi, spesifikasi, konstruksi serta dokumentasi dari sebuah sistem pengembangan perangkat lunak berbasis objek (Danuri, 2009).

#### 3.2.1 Use Case Diagram

*Use Case Diagram* menggambarkan fungsi yang ada di dalam sistem terkait dengan apa yang dibuat di dalam sistem. *Use case* juga menggambarkan urutan transaksi yang berhubungan dan dilakukan oleh satu *actor*



Gambar 3.1 Use Case Diagram

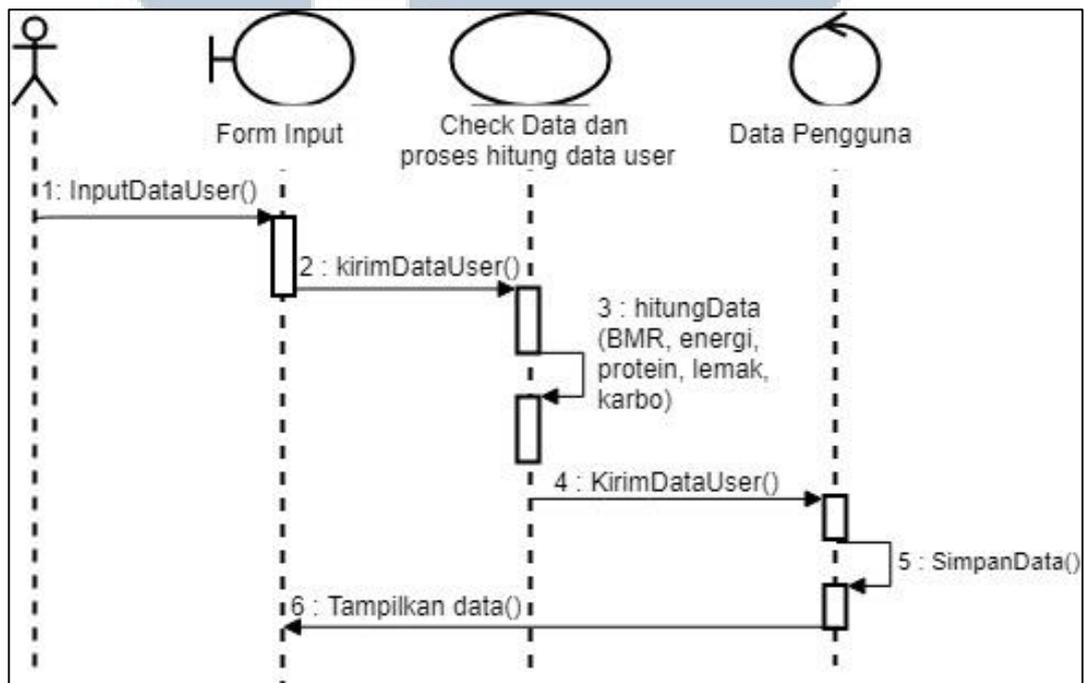
Gambar 3.1 menunjukkan apa yang dikerjakan pengguna dan sistem. Aplikasi diawali dengan memasukkan data pribadi seperti nama, umur, jenis kelamin, berat badan, tinggi badan, tingkat aktivitas yang dilakukan oleh pengguna dalam keseharian. Setelah pengguna memasukkan data pribadinya, aplikasi akan melakukan perhitungan kebutuhan-kebutuhan pengguna seperti jumlah kalori, karbohidrat, protein, lemak, BMR (*Basal Metabolic Rate*) dan IMT (Indeks Masa Tubuh), aplikasi telah siap untuk menghitung jumlah bahan makanan yang diperlukan dan menampilkannya sesuai dengan jadwal makan pengguna (makan pagi, makan siang, makan malam, dan selingan), jika pengguna ingin melihat detail dari salah satu menu makanan (jumlah protein, karbohidrat, lemak, dan kalori makanan) maka pengguna dapat menekan menu makanan tersebut dan melihat detailnya, di dalam *activity* detail ini pengguna juga dapat mengubah menu makanan jika tidak menyukai menu makanan yang ada. Pengguna juga dapat

melihat semua menu makanan dan dapat memilih menu makanan yang tidak disukai atau alergi terhadap menu makanan tersebut. Pengguna juga dapat melihat cara pemakaian aplikasi ini di menu *Help*.

### 3.2.2 Sequence Diagram

*Sequence* Diagram adalah diagram yang digunakan untuk menggambarkan perilaku pada sebuah *scenario* dalam sistem yang dibangun. Diagram ini menunjukkan rangkaian pesan yang dikirim antara *object* dan hubungan antara *object* serta apa yang dilakukan oleh satu *object* dalam sistem (Herdian, 2014).

#### A. Sequence Diagram Memasukan dan Melakukan Pengaturan Data Diri

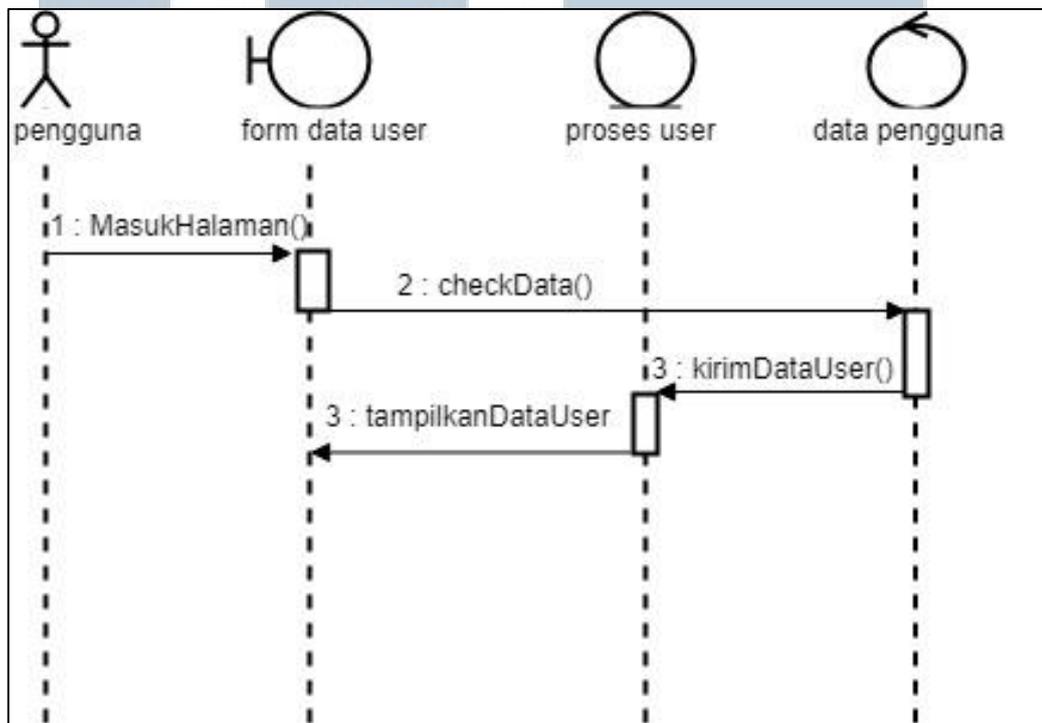


Gambar 3.2 *Sequence* Diagram Memasukan dan Melakukan Pengaturan Data Diri

Gambar 3.2 merupakan gambar *sequence* diagram untuk fungsi *input* data pengguna. Pertama kali pengguna memasukan data ke dalam *Form Input Data User*. Di dalam tahap ini pengguna memasukan data data diri berupa nama, jenis

kelamin, umur, tinggi badan, berat badan, tujuan berolahraga, dan tingkat aktivitas dalam sehari. Selanjutnya data akan diperiksa di dalam fungsi *checkDataDanHitungData*, setelah mendapatkan data yang valid maka data akan di simpan di dalam database.

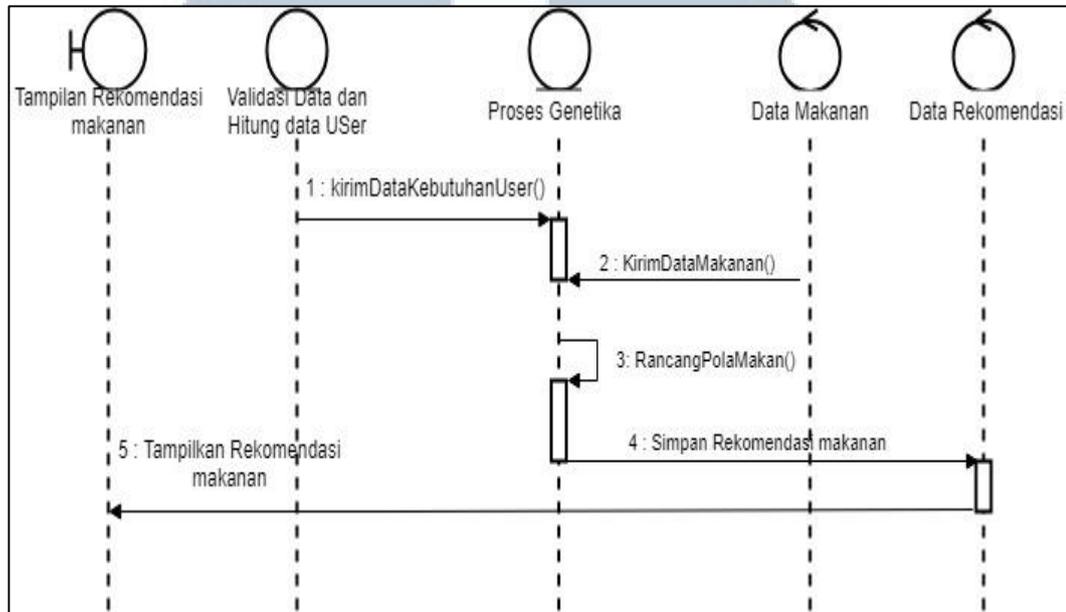
## B. Sequence Diagram Menampilkan Detail Data Diri



Gambar 3.3 *Sequence Diagram* Menampilkan Detail Data Diri

Gambar 3.3 merupakan gambar *sequence diagram* detail data diri. Pertama kali *form data user* akan memeriksa apakah data pengguna sudah terisi di dalam database atau belum. Jika data pengguna sudah ada maka data akan dimasukkan ke dalam *user class*, di dalam kelas ini terdapat aktivitas *setter* dan *getter* yang nantinya akan dipakai untuk menampilkan data pengguna.

### C. Sequence Diagram Menampilkan Rekomendasi Makanan

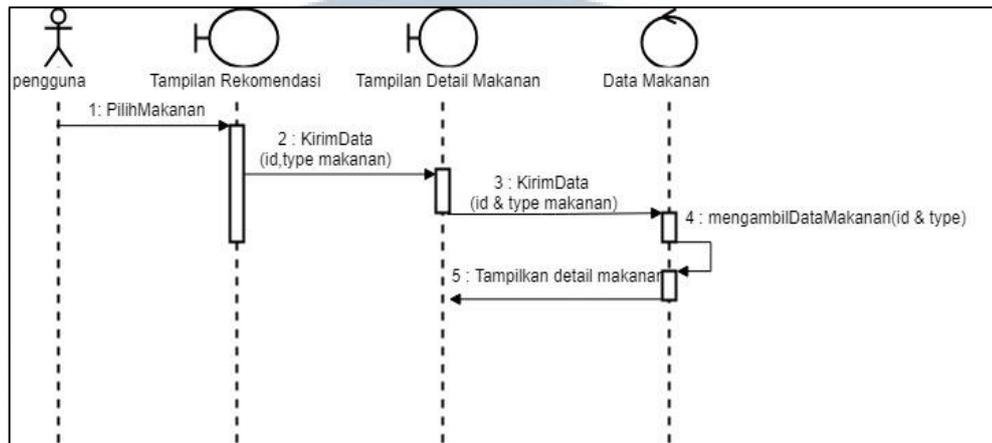


Gambar 3.4 *Sequence Diagram* Menampilkan Rekomendasi Bahan Makanan

Gambar 3.4 adalah *sequence diagram* rekomendasi bahan makanan. Pertama kali pengguna memasukkan data ke dalam *form input*, setelah itu data akan melalui proses validasi, jika data valid maka data akan disimpan di dalam database. selanjutnya data pengguna dan data makanan akan masuk ke dalam kelas genetika. Di dalam kelas genetika ini memproses data makanan dan data pengguna menjadi data rekomendasi yang lalu akan disimpan di dalam database rekomendasi. Setelah data rekomendasi terbentuk rekomendasi makanan dapat langsung ditampilkan.

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

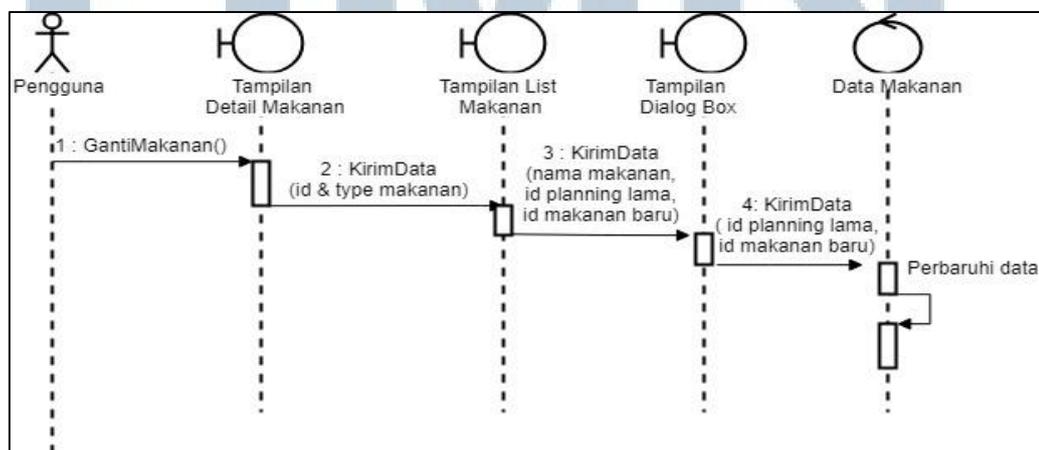
#### D. Sequence Diagram Menampilkan Detail Bahan Makanan



Gambar 3.5 Sequence Diagram Menampilkan Detail Bahan Makanan

Gambar 3.5 adalah *sequence* diagram detail bahan makanan. Pertama kali pengguna memilih makanan yang ingin ditampilkan detailnya, maka tampilan rekomendasi makanan mengirimkan data berupa *id* dan *type* makanan yang dipilih ke dalam tampilan detail makanan, tampilan detail makanan akan mengirim data *id* dan *type* makanan ke database dan memeriksa *id* dan *type* makanan yang ingin ditampilkan

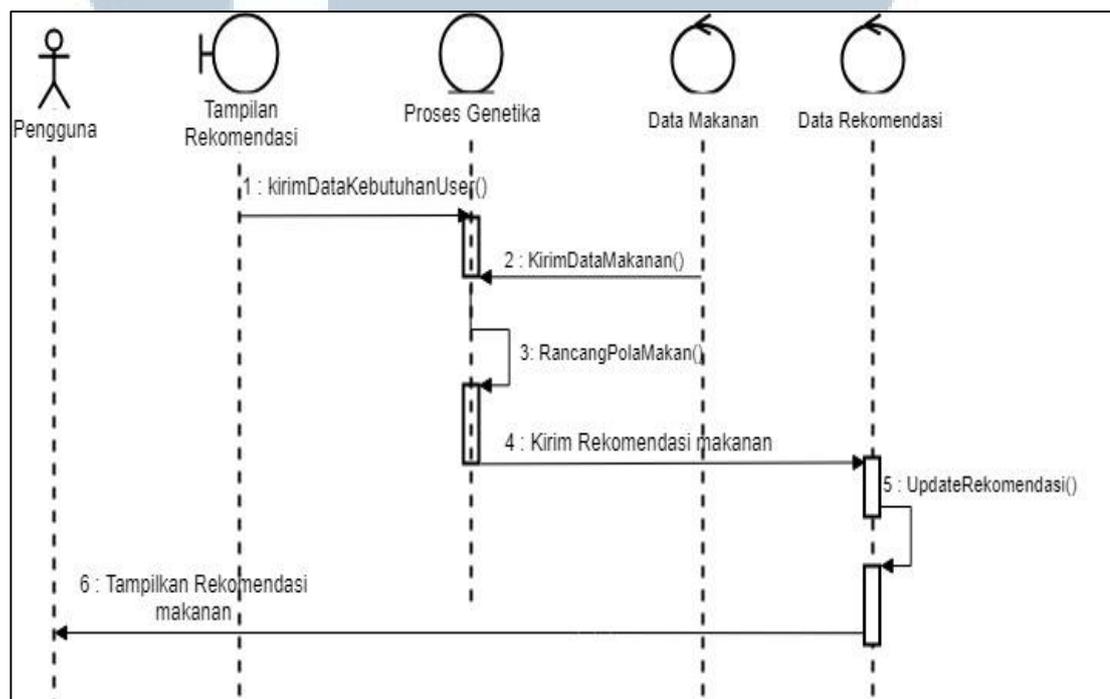
#### E. Sequence Diagram Mengubah Baham Makanan



Gambar 3.6 Sequence Diagram Mengubah Baham Makanan

Gambar 3.6 adalah *sequence* diagram pengubahan bahan makanan. Pertama kali pengguna memilih makanan yang ingin diganti, setelah itu tampilan detail makanan akan mengirim *id* dan *type* makanan ke dalam tampilan daftar makanan. Di dalam tampilan daftar makanan data berupa nama, *id planning* lama, dan *id planning* baru akan dikirim ke dalam dialog *update*, di dalam dialog *update* data akan dikirim ke database makanan dan akan dilakukan perubahan data rekomendasi makanan.

#### F. Sequence Diagram Memuat Ulang List Rekomendasi Makanan

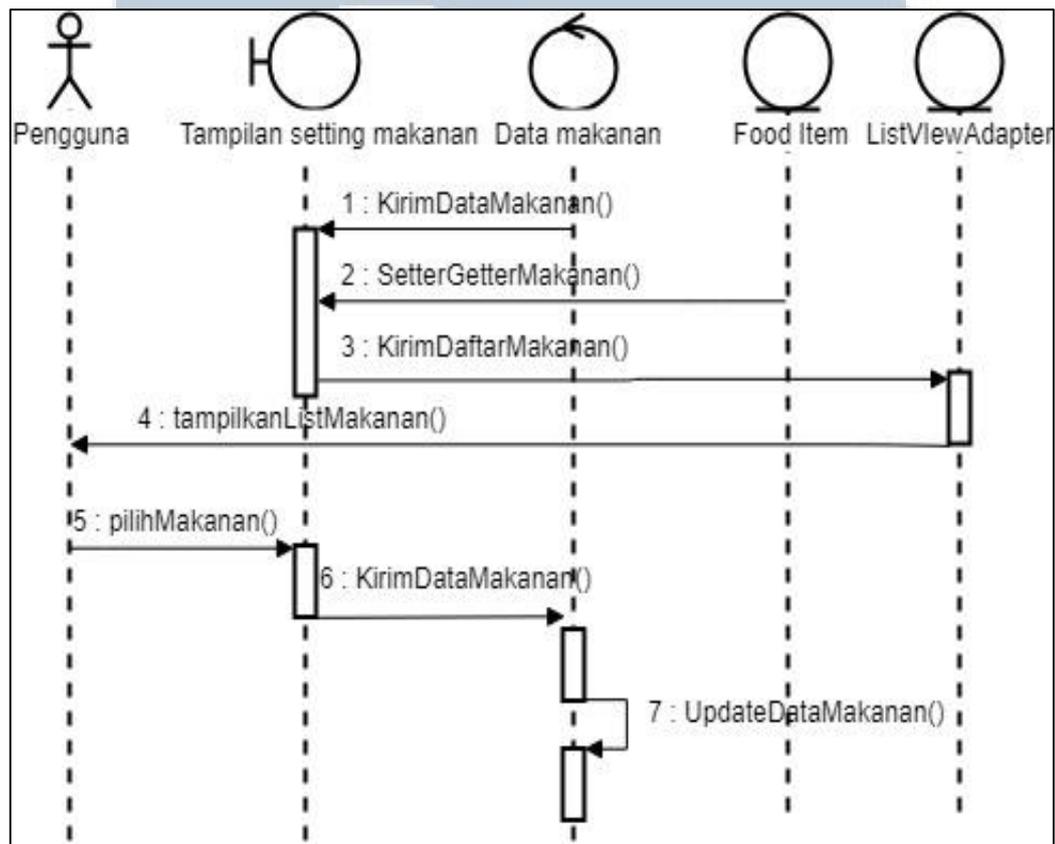


Gambar 3.7 *Sequence* Diagram Memuat Ulang List Rekomendasi Makanan

Gambar 3.7 adalah *sequence* diagram memuat ulang seluruh rekomendasi. Proses pertama tampilan rekomendasi mengirimkan data kebutuhan pengguna ke dalam kelas genetika dan kelas data makanan mengirimkan data makanan ke dalam kelas genetika. Proses selanjutnya adalah kelas genetika memproses data pengguna

dan data makanan menjadi rekomendasi makanan baru lalu menyimpannya ke dalam database rekomendasi dan menampilkannya.

### G. Sequence Diagram Pengaturan Bahan Makanan

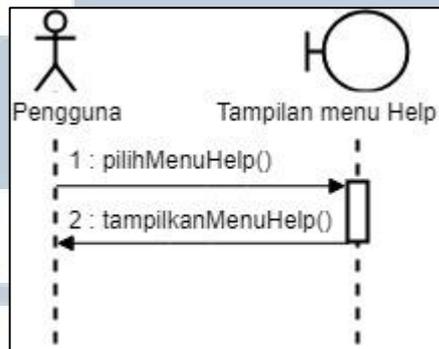


Gambar 3.8 *Sequence* Diagram Pengaturan Bahan Makanan

Gambar 3.8 adalah *sequence* diagram pengaturan bahan makanan. Pertama kali database akan mengirimkan data makanan ke dalam tampilan *list* makanan, lalu kelas *foodItem* akan melakukan *setter* dan *getter* terhadap data makanan, proses selanjutnya data akan dimasukan ke dalam kelas *listview adapter* yang berfungsi untuk menampilkan data makanan dalam bentuk daftar. Jika pengguna memilih

makanan yang tidak ingin ditampilkan ke dalam rekomendasi makanan maka tampilan *list* makanan akan mengirim data ke dalam database untuk di *update*.

#### H. Sequence Diagram Menampilkan Menu Bantuan



Gambar 3.9 *Sequence* Diagram Menampilkan Menu Bantuan

Gambar 3.9 adalah *sequence* diagram menu bantuan. Jika pengguna menekan tombol menu bantuan, maka tampilan menu bantuan akan muncul dan memberikan beberapa penjelasan mengenai menu-menu yang terdapat dalam aplikasi

### 3.2.3 Activity Diagram

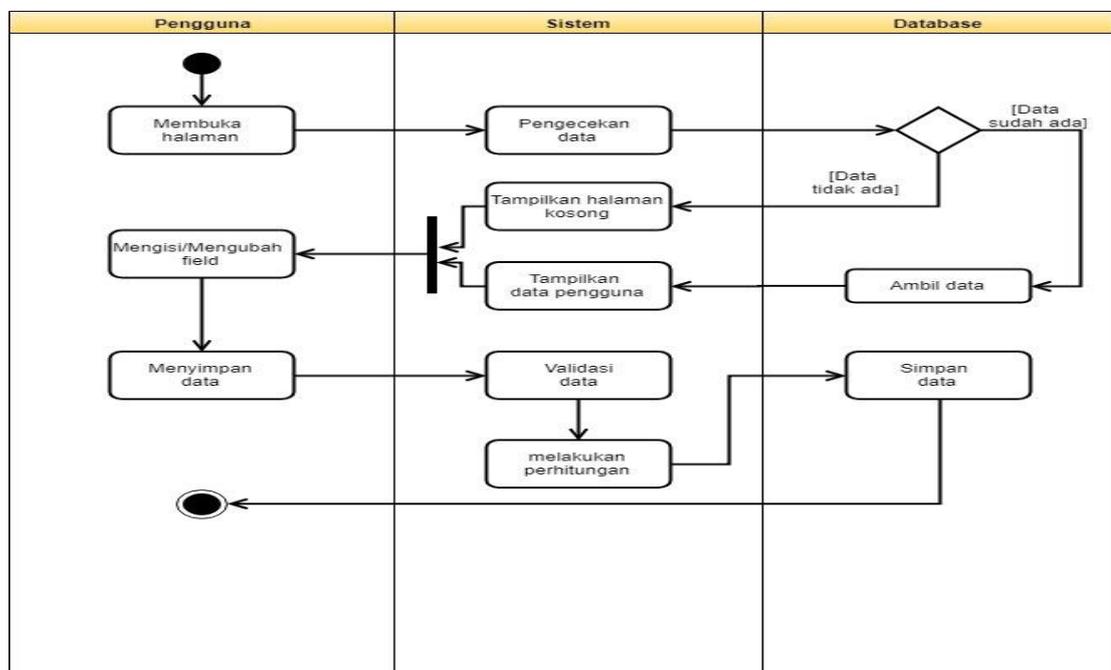
*Activity* diagram adalah diagram yang menggambarkan rangkaian aliran dari aktivitas dan berfungsi untuk menggambarkan interaksi yang terbentuk di dalam sistem. Berikut *activity* diagram yang tergambar dalam sistem yang dibangun.

#### A. Aktivitas Memasukan dan Melakukan Pengaturan Data Diri

Aktivitas data diri, merupakan aktivitas yang digunakan untuk mengisi data pengguna baru, melihat data pengguna yang sudah ada dan dapat mengubah data pengguna yang sudah ada. Di dalam aktivitas ini pengguna dapat mengisi data diri

seperti nama pengguna, jenis kelamin pengguna, umur pengguna, tinggi pengguna dalam satuan *centimeter*, berat pengguna dalam satuan kilogram, tujuan pengguna menggunakan meal *fitness planning* ini (*fat loss, maintain physique, gain muscle, bulking*), dan terakhir pengguna wajib memilih tingkat aktivitas yang dilakukan apakah ringan, agak berat, berat, atau *extream*.

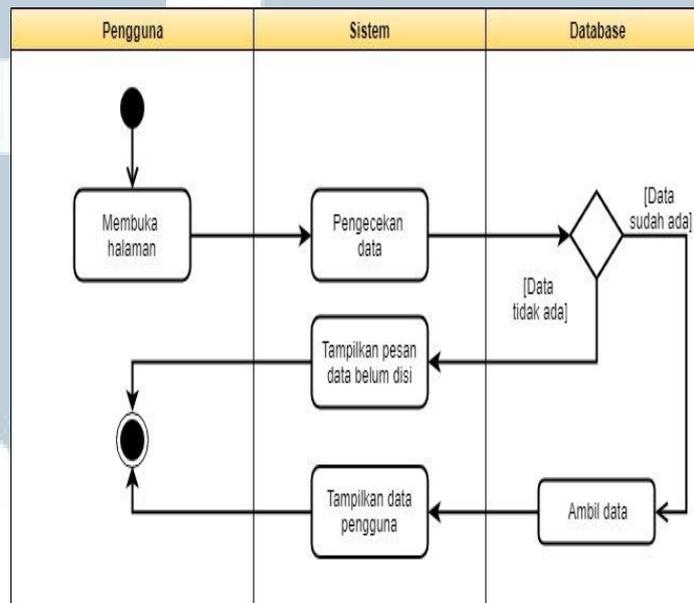
Setelah mendapatkan data-data dari pengguna sistem akan memulai melakukan perhitungan BMR (*Bassal Metabolic Rate*) menggunakan rumus yang telah dijelaskan pada landasan teori. Selanjutnya sistem melakukan perhitungan kebutuhan kalori dengan menggunakan hasil perhitungan BMR sebelumnya. Selanjutnya setelah mendapatkan kebutuhan kalori, sistem dapat melakukan perhitungan kebutuhan karbohidrat, lemak, dan protein dengan menggunakan hasil perhitungan kalori sebelumnya dan rumus yang telah dijelaskan di dalam landasan teori, selanjutnya data-data tersebut disimpan ke dalam database.



Gambar 3.10 Activity Diagram Memasukan dan Pengaturan Data Diri

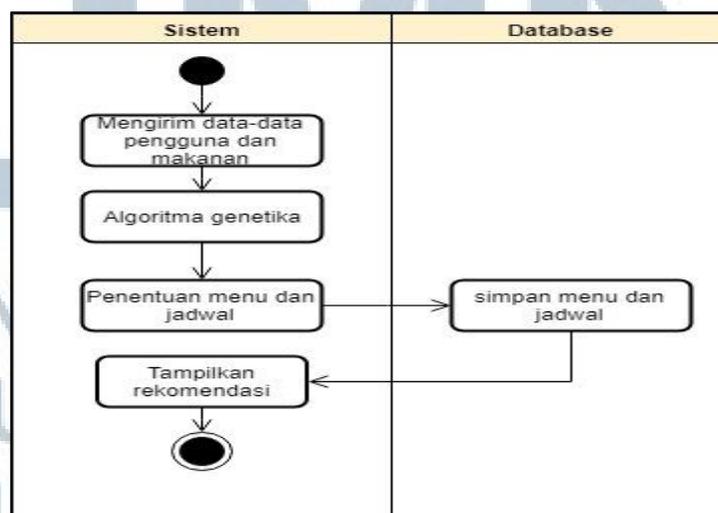
## B. Aktivitas Menampilkan Detail Data Diri

Aktivitas ini digunakan untuk menampilkan detail dari data pengguna. dimana pengguna dapat melihat lebih jelas lagi detail dari protein, karbohidrat, lemak dan kalori yang dibutuhkan dalam sehari.



Gambar 3.11 Activity Diagram Detail Data Diri

## C. Aktivitas Menampilkan Rekomendasi Makanan

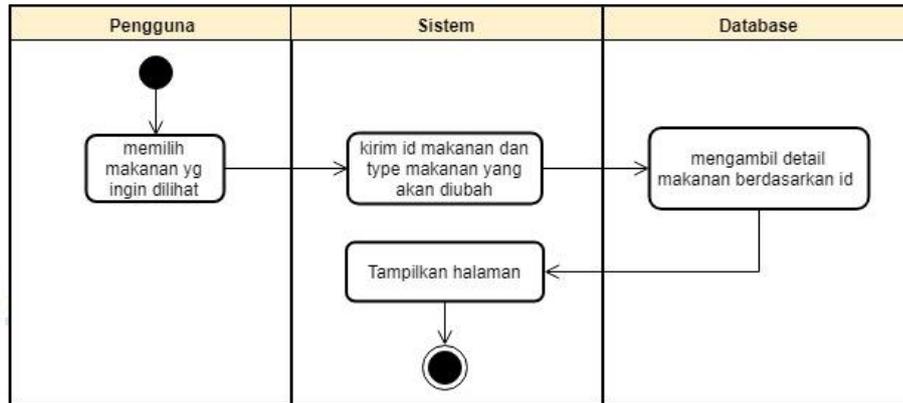


Gambar 3.12 Activity Diagram Rekomendasi Makanan

Di dalam aktivitas ini yang pertama kali dilakukan sistem adalah menjalankan algoritma genetika dengan melakukan pengecekan terhadap data-data dari pengguna. Setelah melakukan perhitungan sebelumnya terhadap data-data pengguna seperti perhitungan kebutuhan kalori, karbohidrat, protein dan lemak pengguna, sistem lalu akan menghitung komposisi yang tepat untuk jadwal makanan pengguna yakni makan pagi, selingan 1, makan siang, selingan 2, dan makan malam. Lalu pengguna dapat melihat rekomendasi bahan makanan, melihat detail dari menu makanan yang ditampilkan, mengubah menu makanan yang sesuai dengan keinginan pengguna atau pengguna dapat *me-refresh* kembali menu rekomendasi yang telah ada. Kemudian semua data makanan rekomendasi akan disimpan ke dalam database, sehingga jika pengguna ingin melihat kembali menu rekomendasi cukup dengan menekan menu “*My Meals*”.

#### **D. Aktivitas Menampilkan Detail Bahan Makanan**

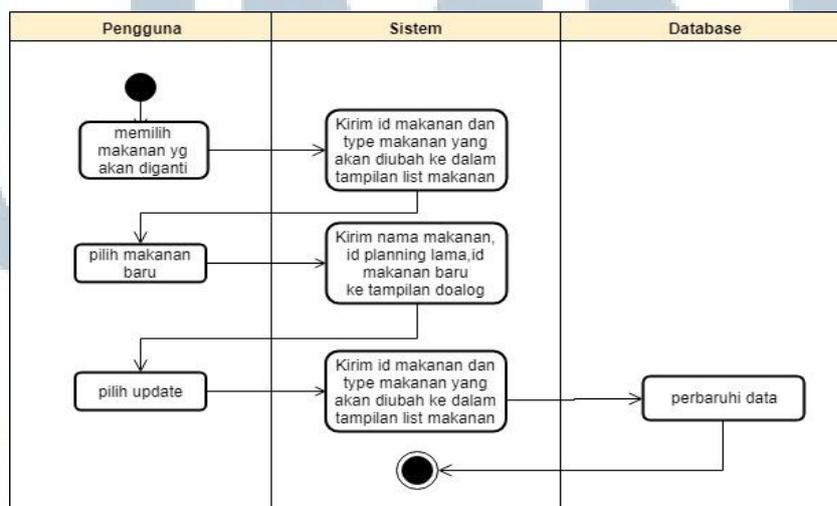
Setelah proses rekomendasi makanan ditampilkan, pengguna dapat melihat detail dari menu makanan tersebut dari kandungan protein, karbohidrat, lemak, dan energi. Proses ini diawali dengan mengambil *id* menu makanan dari daftar rekomendasi dan kategori makanan tersebut, selanjutnya database akan mencari menu makanan berdasarkan *id* menu makanan dari daftar rekomendasi dan menampilkan detail menu makanan tersebut.



Gambar 3.13 Activity Diagram Detail Makanan

### E. Aktivitas Mengubah Baham Makanan

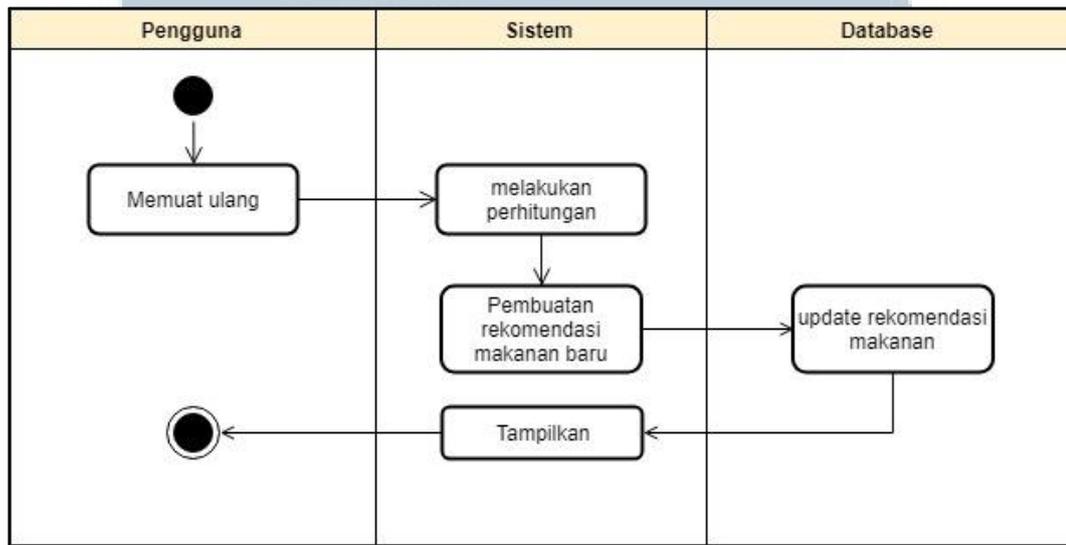
Setelah mendapatkan tampilan detail dari menu, pengguna dapat melakukan perubahan menu makanan jika tidak sesuai dengan keinginan pengguna. Proses ini diawali dengan mengambil *id* menu makanan dari daftar rekomendasi dan kategori makanan yang akan diubah, setelah itu sistem akan menampilkan daftar seluruh menu makanan berdasarkan kategori yang didapat. Pengguna dapat memilih satu menu dari daftar. Selanjutnya database rekomendasi akan diperbahruhi dengan mengganti *id* menu makanan lama dengan *id* menu makanan baru.



Gambar 3.14 Activity Diagram Perubahan Bahan Makanan

## F. Aktivitas Memuat Ulang List Rekomendasi Makanan

Di dalam aktivitas ini jika pengguna ingin mengganti seluruh menu makanan yang sudah ada, maka pengguna cukup menekan tombol muat ulang maka sistem akan kembali mengulang proses pembuatan rekomendasi menu makanan.

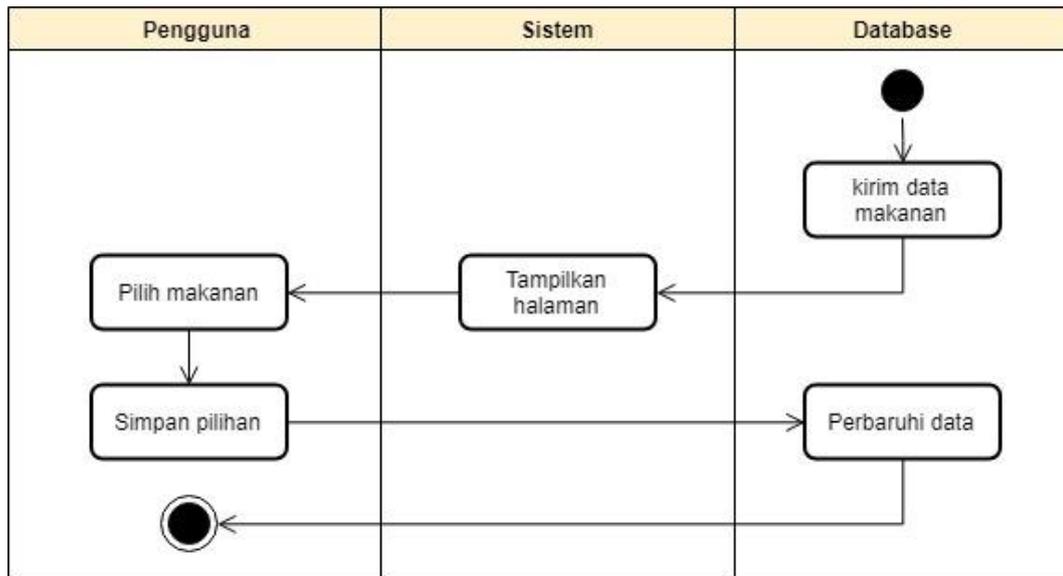


Gambar 3.15 Activity Diagram Memuat Ulang Seluruh Rekomendasi

## G. Aktivitas Pengaturan Bahan Makanan

Aktivitas ini berisikan proses penyesuaian bahan makanan yang akan digunakan oleh pengguna. Pengaturan bahan makanan dilakukan dengan mengubah kolom status di dalam database. Jika bernilai aktif maka bahan makanan tersebut digunakan sebagai daftar rekomendasi, jika bernilai pasif maka bahan makanan tersebut tidak digunakan sebagai daftar rekomendasi.

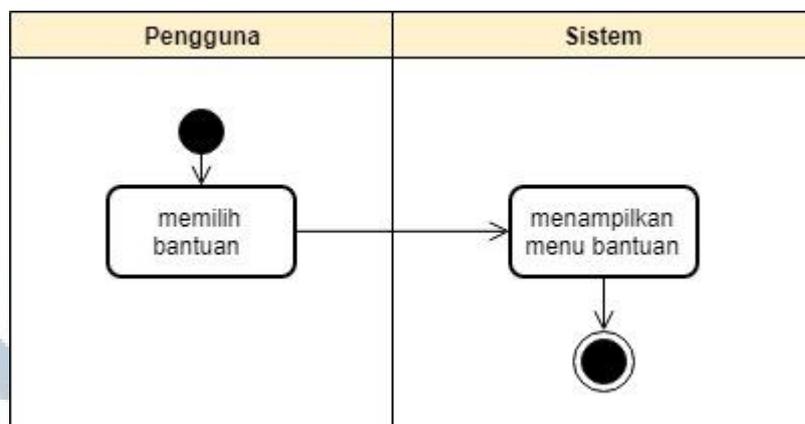
UNIVERSITAS  
MULTIMEDIA  
NUSANTARA



Gambar 3.16 Activity Diagram Pengaturan Bahan Makanan

#### H. Aktivitas Menampilkan Menu Bantuan

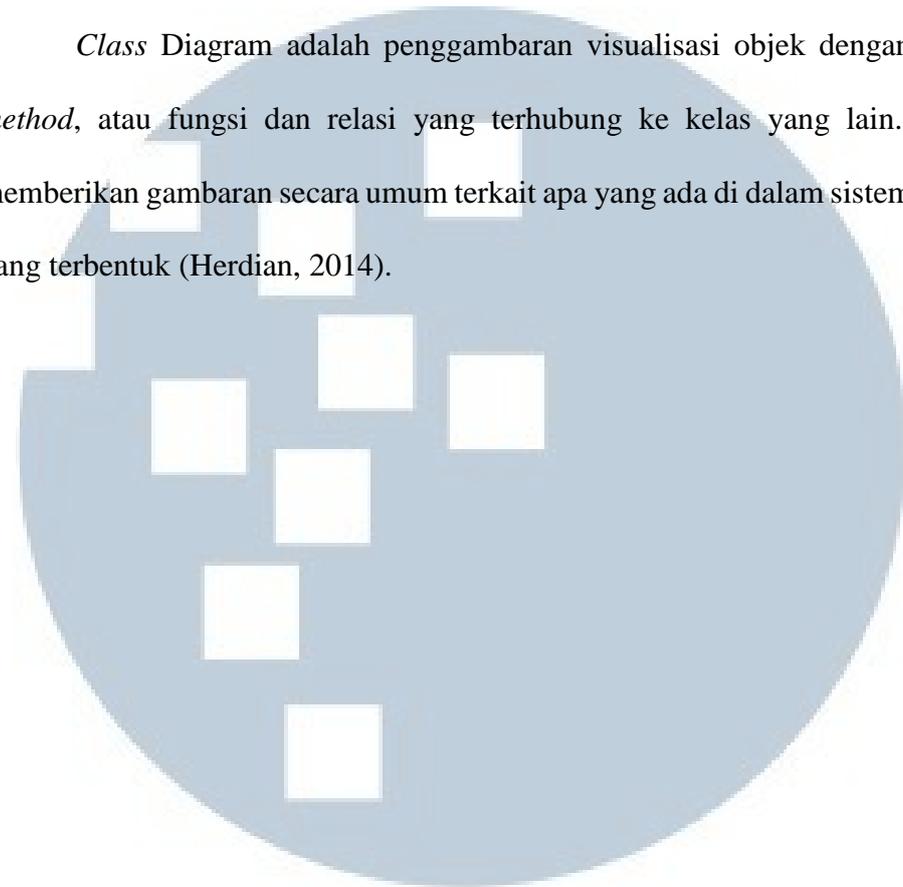
Aktivitas ini digunakan untuk menampilkan bantuan berupa cara pemakaian aplikasi dan hal-hal terkait dengan aplikasi. Pengguna cukup memilih menu yang ingin dilihat penjelasan lebih lanjutnya.



Gambar 3.17 Activity Diagram Menu Bantuan

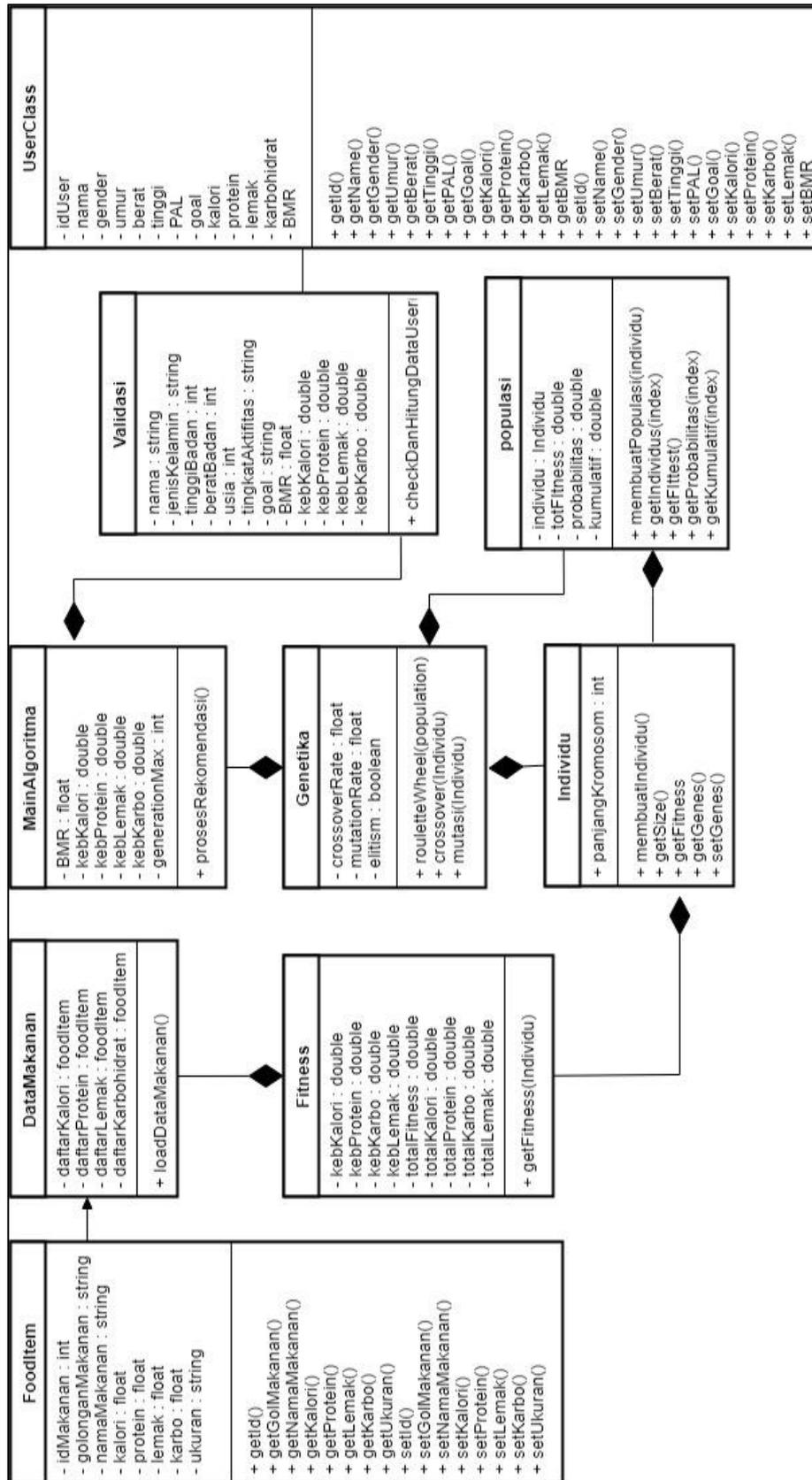
### 3.2.4 Class Diagram

*Class Diagram* adalah penggambaran visualisasi objek dengan property, *method*, atau fungsi dan relasi yang terhubung ke kelas yang lain. Kelas ini memberikan gambaran secara umum terkait apa yang ada di dalam sistem dan objek yang terbentuk (Herdian, 2014).



UMMN

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA



Gambar 3.18 Class Diagram

### 3.2.5 Flowchart Algoritma Genetika

*Flowchart* adalah agan-bagan yang mempunyai arus yang menggambarkan langkah-langkah penyelesaian suatu masalah. *Flowchart* merupakan cara penyajian dari suatu algoritma.

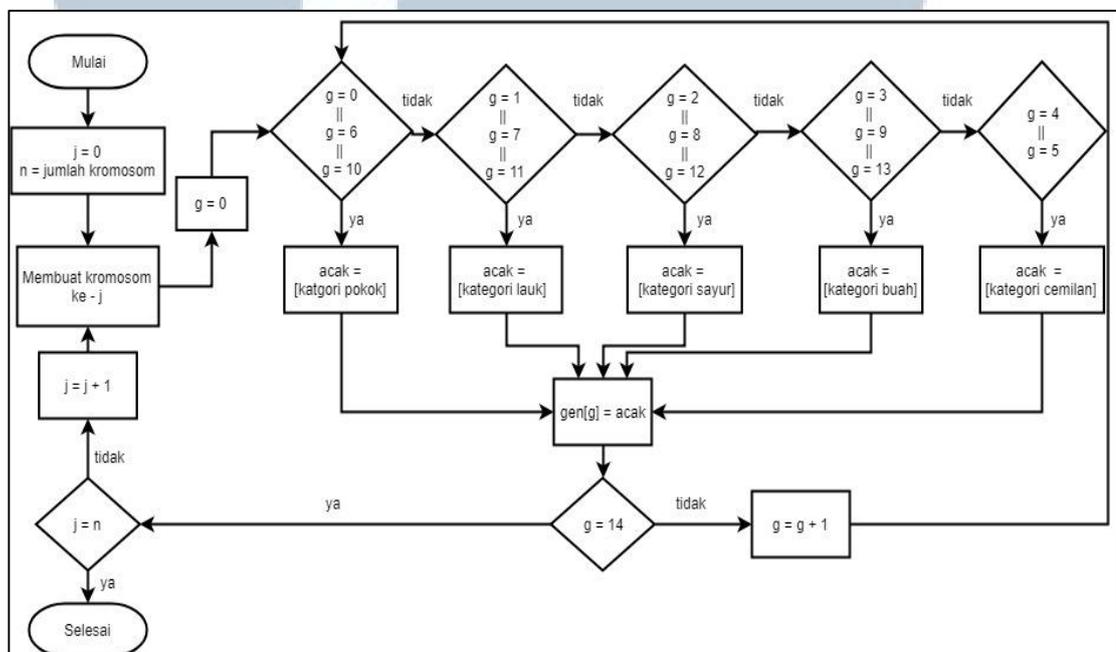
#### 3.2.5.1 Aktivitas Algoritma Genetika

Pada saat menjalankan algoritma genetika, hal pertama yang dilakukan adalah mendapatkan jumlah kebutuhan kalori, karbohidrat, lemak, dan protein sebagai batasan penentu solusi. Solusi dikatakan baik jika jumlah kalori, karbohidrat, lemak dan protein yang ditawarkan mendekati kebutuhan pengguna. Selanjutnya menetapkan nilai peluang *crossover*, peluang mutasi, jumlah individu dalam satu populasi dan jumlah generasi. Setelah parameter-parameter algoritma sudah ditetapkan maka tahap selanjutnya adalah membangkitkan populasi awal, menghitung nilai *fitness* dari masing-masing kromosom, melakukan pengurutan populasi dari nilai *fitness* tertinggi untuk menjaga agar suatu kromosom tidak rusak, melakukan *roulette wheel*, *crossover*, *mutasi*, menghitung nilai *fitness* dari masing-masing kromosom, melakukan pengurutan populasi dari nilai *fitness* tertinggi dari generasi sebelumnya, kembali lagi melakukan *roulette wheel*, *crossover*, dan mutasi sampai jumlah generasi yang telah ditetapkan diawal.

Berikut adalah penjelasan dari masing-masing tahap algoritma genetika dari proses inialisasi populasi awal, proses hitung nilai *fitness*, menjaga agar suatu kromosom yang baik agar tidak rusak dan bisa digunakan untuk generasi berikutnya, proses *roulette wheel*, proses *crossover*, dan proses *mutasi*.

## A. Proses Inisialisasi Populasi Awal

Pembangkitkan populasi awal dilakukan dengan memasukan bilangan *real* yang diacak ke dalam 14 gen pertama atau kromosom pertama. Dimana gen-gen tersebut mewakili bahan makanan berdasarkan kategori, yang mana setiap kategori memiliki nilai kalori, karbohidrat, protein, dan lemak tersendiri. Proses pengisian gen akan berlangsung sebanyak jumlah kromosom dalam populasi. Setelah gen-gen tersebut terisi, maka proses akan dilanjutkan dengan menghitung nilai *fitness*.



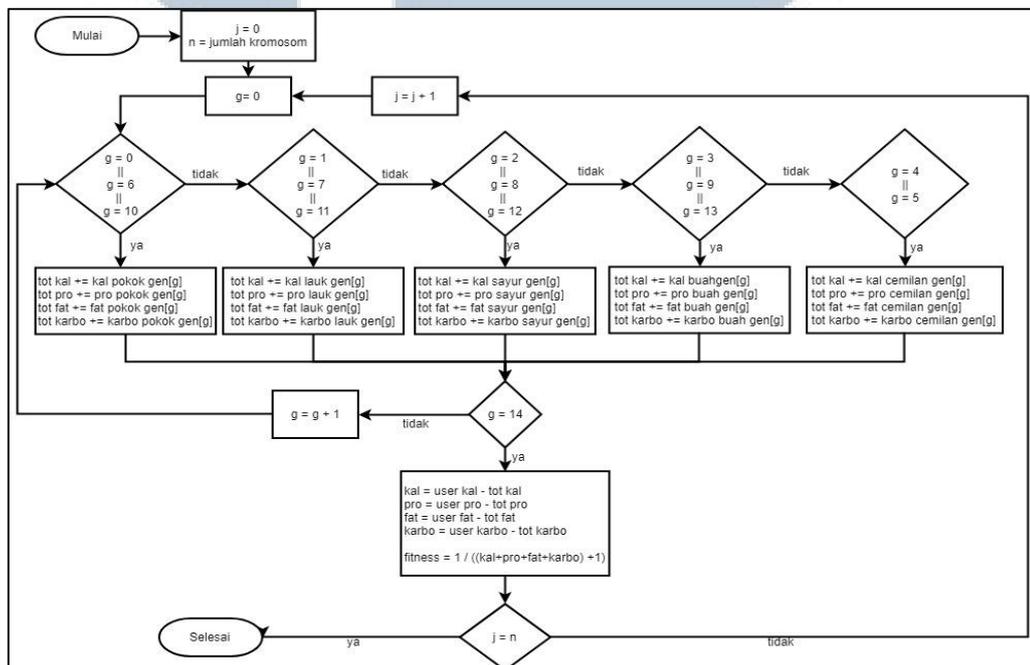
Gambar 3.19 Flowchart Proses Inisialisasi Awal

## B. Proses Hitung Nilai Fitness

Pada proses ini masing kromosom akan dihitung jumlah *fitness*nya. Pertama akan dilakukan perhitungan jumlah total kalori, karbohidrat, lemak, dan protein dari bahan makan dalam satu kromosom. Nilai-nilai kebutuhan yang diperoleh akan diproses dengan nilai kebutuhan yang diperlukan oleh pengguna dengan

mengurangi hasil kebutuhan pengguna dengan total kebutuhan dari perhitungan di dalam algoritma genetika (jumlah total kalori, karbohidrat, lemak, dan protein). Berikut adalah rumus untuk menghitung nilai *fitness*.

Nilai *fitness* yang diperoleh akan memiliki interval [0,1] dimana nilai *fitness* yang dihasilkan adalah bilangan real positif. Rumus di atas akan memberikan nilai *fitness* yang mendekati 1 apabila nilai kebutuhan yang didapatkan dari perhitungan mendekati nilai kebutuhan yang dibutuhkan oleh pengguna. Jika nilai *fitness* semakin mendekati nilai 1, maka kromosom tersebut semakin kuat. Akan tetapi jika nilai *fitness* semakin mendekati nilai 0, maka kromosom tersebut akan semakin lemah dan dengan demikian peluang untuk dibuang lebih tinggi.



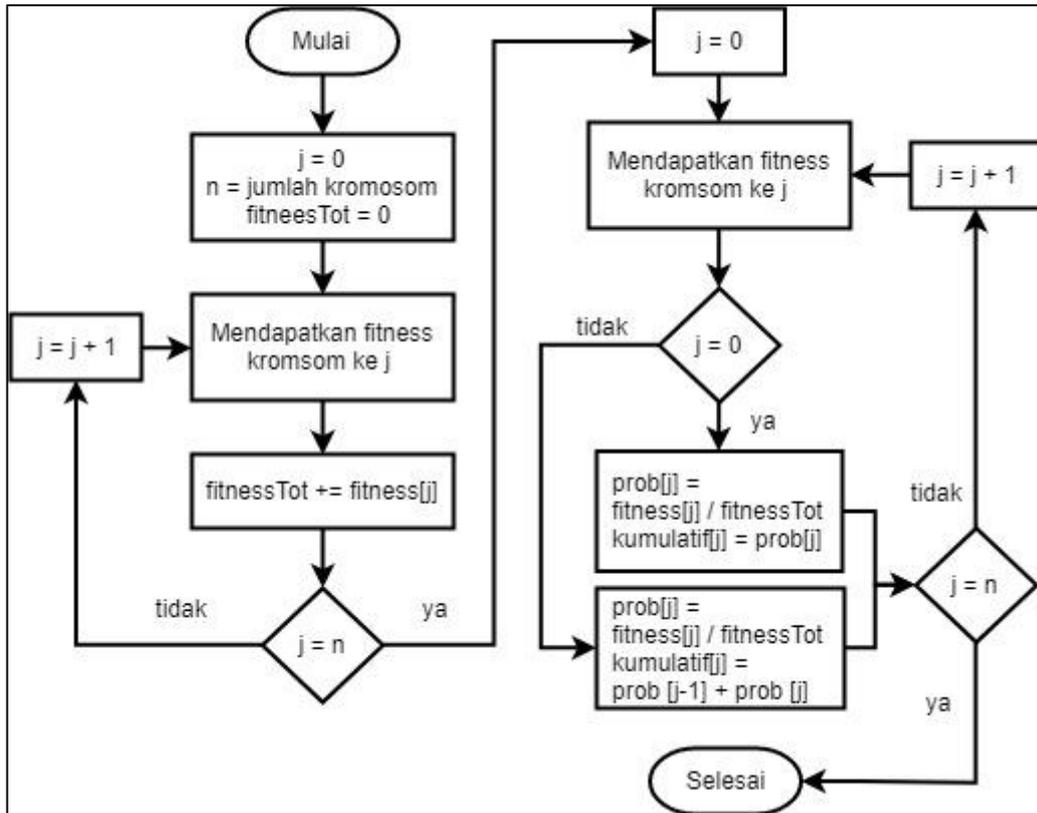
Gambar 3.20 Flowchart Proses Hitung Nilai *Fitness*

### C. Proses Perhitungan Nilai Kumulatif dan Nilai Relatif Fitness

Di dalam proses ini masing-masing kromosom dalam satu populasi akan dihitung jumlah nilai kumulatif dan relatifnya, dimana nilai kumulatif digunakan untuk mempermudah dalam pemilihan kromosom dalam suatu populasi yang akan dibuat selanjutnya. Jika dalam suatu kromosom memiliki nilai *fitness* kumulatif yang memiliki selisih lebih besar dengan nilai *fitness* kumulatif pada kromosom sebelumnya akan memiliki peluang lebih besar untuk menjadi anggota populasi pada generasi berikutnya. Sedangkan nilai *relative* akan menunjukkan seberapa dominan suatu kromosom di dalam populasinya. Dimana suatu kromosom akan memiliki nilai relatif yang tinggi jika memiliki nilai *fitness* yang tinggi di antara kromosom-kromosom lainnya dan semakin tinggi nilai *fitness* yang dimiliki akan meningkatkan peluang untuk menjadi anggota kromosom di dalam populasi berikutnya.

Prosedure ini dimulai dengan menghitung total jumlah *fitness* semua kromosom yang terdapat di dalam suatu populasi. Selanjutnya dari masing-masing kromosom akan dihitung nilai *fitness*nya dibagi dengan nilai *fitness* secara keseluruhan yang telah dihitung sebelumnya. Kemudian akan dicari nilai kumulatifnya dengan cara menambah nilai relatif dari kromosom tersebut dengan nilai-nilai kromosom sebelumnya, dimana jika kromosom tersebut adalah kromosom pertama akan memiliki nilai kumulatif dan relatif yang sama.

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

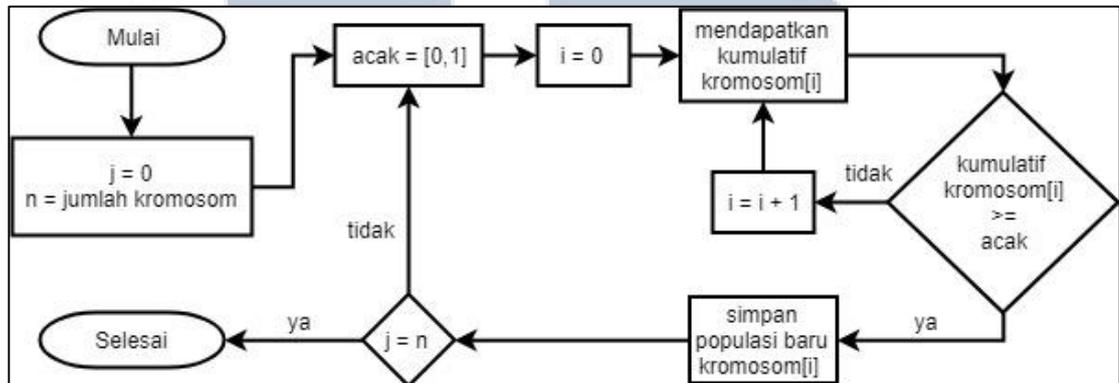


Gambar 3.21 Flowchart Proses Hitung Nilai Kumulatif dan Relatif

#### D. Proses Roulette Wheel

Setelah melakukan proses perhitungan kumulatif dan relatif maka proses berikutnya adalah melakukan seleksi terhadap kromosom-kromosom di dalam populasi. Proses seleksi ini menggunakan metode *roulette wheel*. Seperti namanya, prosedur ini menirukan permainan *roulette*. Dalam permainan tersebut pilihan yang semakin luas akan memiliki peluang terpilih semakin besar. prosedur ini diawali dengan membuat bilangan acak dengan interval dari 0 sampai 1 yang merupakan bilangan acak *real* positif. Banyaknya bilangan acak mengikuti jumlah kromosom dalam suatu populasi, sehingga masing-masing kromosom memiliki satu bilangan acak. setelah itu bilangan acak akan dibandingkan dengan nilai kumulatif dari masing-masing kromosom. Contoh

jika bilangan acak lebih besar dari kumulatif kromosom pertama dan lebih kecil dari kumulatif kromosom kedua maka pilih kromosom ke dua untuk menjadi induk. Proses ini dilakukan sebanyak jumlah kromosom di dalam populasi.

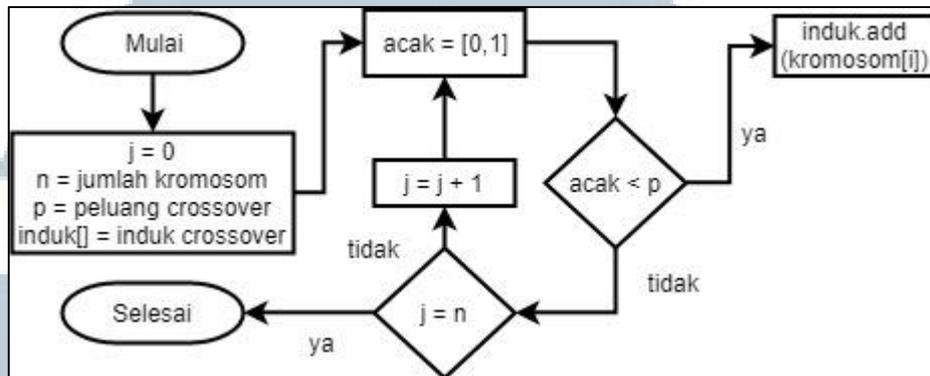


Gambar 3.22 Flowchart Proses Roulette

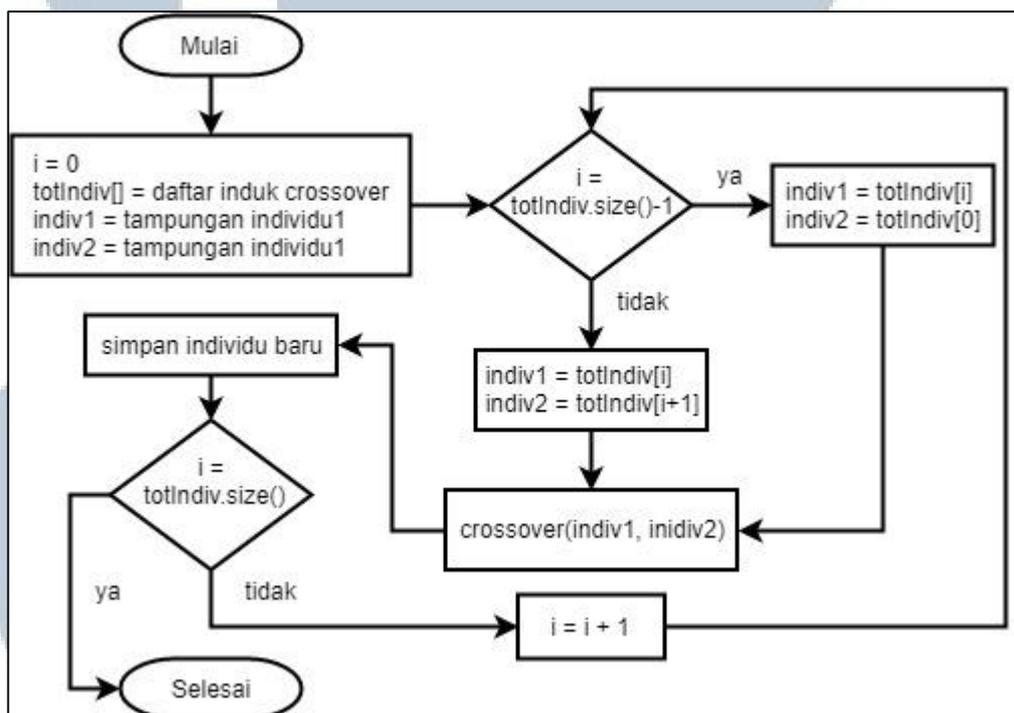
### E. Proses Crossover

Setelah melakukan proses seleksi menggunakan *roulette wheel* proses selanjutnya adalah proses *crossover*. Metode yang digunakan adalah metode *one-cut point*, yaitu memilih secara acak satu posisi gen dalam suatu kromosom induk kemudian saling menukar gen. kromosom yang dijadikan induk dipilih secara acak dan jumlah kromosom yang mengalami *crossover* dipengaruhi oleh parameter *crossover\_rate*. Proses ini diawali dengan membangkitkan bilangan acak sebanyak jumlah kromosom dalam sebuah populasi. Bilangan acak memiliki interval  $[0,1]$  dimana masing-masing bilangan acak yang dibangkitkan mewakili masing-masing kromosom. Jika bilangan acak memiliki nilai lebih besar dari *crossover\_rate* maka yang dijadikan induk adalah kromosom yang diwakili oleh bilangan acak tersebutlah yang dijadikan induk. Proses selanjutnya adalah menentukan posisi *crossover*, proses ini dilakukan dengan cara membangkitkan bilangan acak dengan interval 1 sampai panjang

kromosom dikurang 1. Setelah mendapatkan anak hasil persilangan tersebut akan menggantikan posisi induknya masing-masing.



Gambar 3.23 Flowchart Proses Pemilihan Induk Crossover

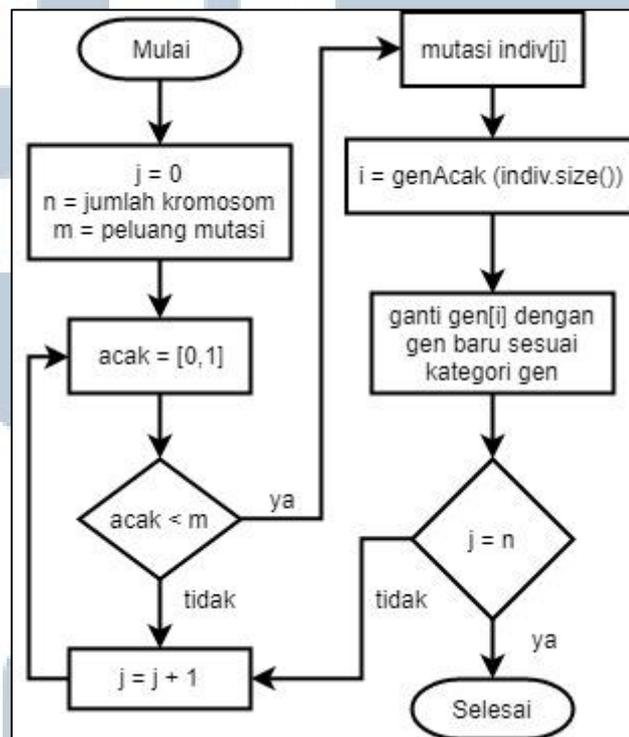


Gambar 3.24 Flowchart Proses Crossover

## F. Proses Mutasi

Proses ini bertujuan untuk memunculkan gen-gen baru untuk membuat kromosom lebih variatif. Proses ini diawal dengan membangkitkan bilangan

acak dengan interval  $[0,1]$  sebanyak jumlah kromosom di dalam satu populasi, sehingga bilangan acak tersebut akan mewakili masing-masing kromosom di dalam populasi tersebut. Selanjutnya memilih posisi gen yang akan mengalami mutasi dengan cara membangkitkan bilangan acak sebanyak jumlah kromosom di dalam populasi, selanjutnya mengganti gen tersebut dengan gen baru sesuai posisi gen lama.

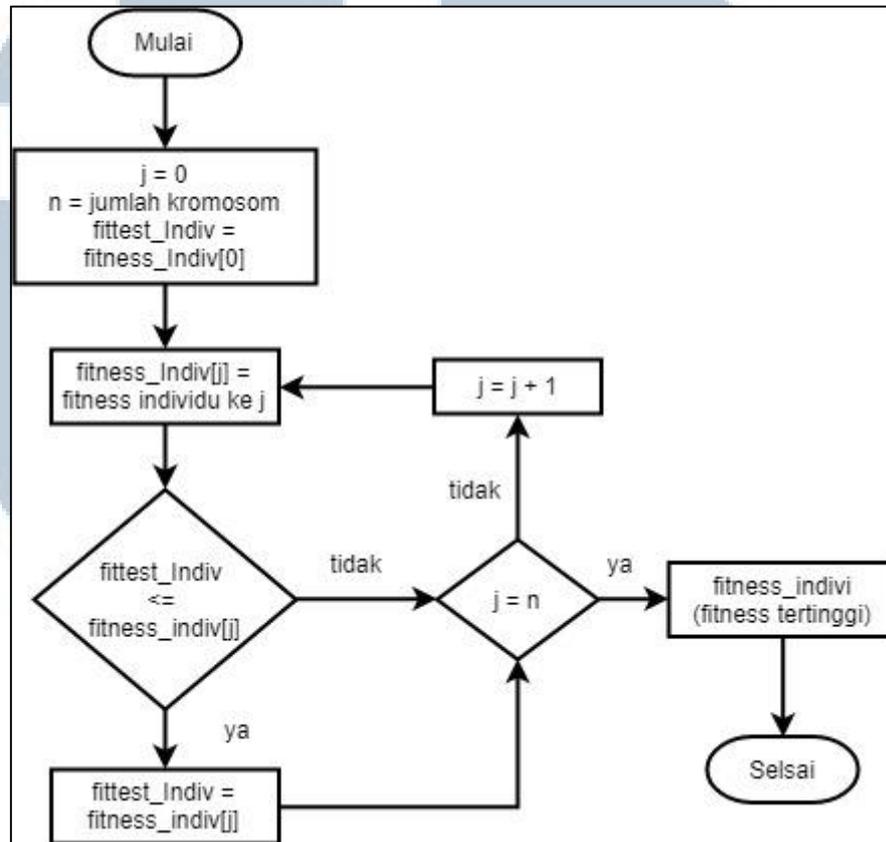


Gambar 3.25 Flowchart Proses Mutasi

### G. Proses Elitisme

Di dalam proses ini akan menyimpan atau menjaga agar suatu kromosom yang memiliki nilai *fitness* yang sangat tinggi agar tidak rusak. Proses seleksi yang dilakukan secara acak tidak ada jaminan bahwa suatu individu yang bernilai *fitness* tertinggi akan selalu terpilih. Walaupun individu bernilai *fitness* tertinggi terpilih, mungkin saja individu tersebut akan rusak (nilai *fitness*nya

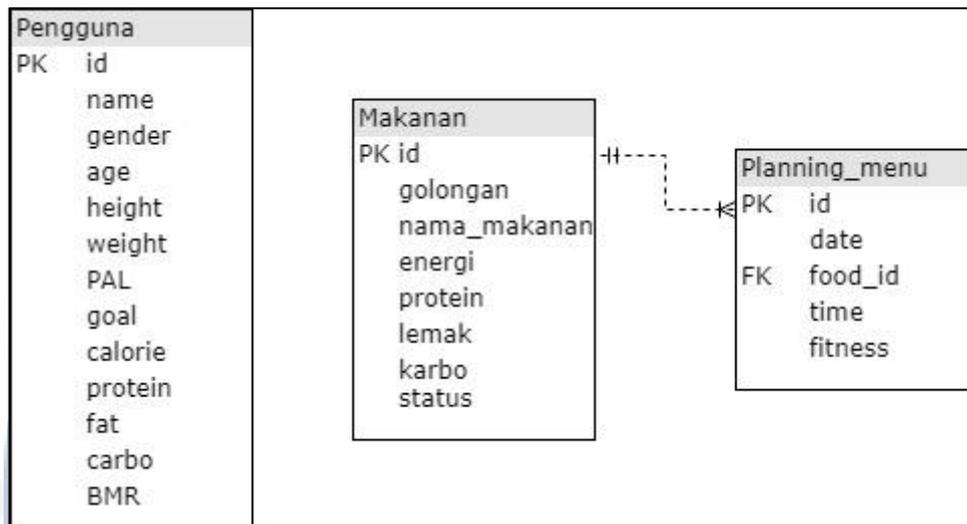
menurun) karena proses *crossover*. Oleh karena itu, untuk menjaga agar individu bernilai *fitness* tinggi tersebut tidak hilang selama evolusi, maka perlu dibuat *copy*-nya.



Gambar 3.26 Flowchart Proses Elitisme

### 3.2.6 Entity Relationship Diagram

*Entity Relationship Diagram* adalah sebuah konsep untuk mendeskripsikan hubungan antara penyimpanan dan didasarkan pada persepsi dari sebuah dunia nyata yang terdiri dari sekumpulan objek yaitu disebut sebagai *entity* dan hubungan atau relasi antar objek-objek tersebut. Dalam proses perancangannya, aplikasi ini menggunakan basis data dengan bantuan *SQLite* di dalam *Android*. Berikut adalah rancangan basis data yang digunakan.



Gambar 3.27 Entity Relationship Diagram

Berikut adalah perincian dari tabel-tabel di dalam struktur basis data di atas.

1. Nama tabel : Pengguna

Kegunaan : Untuk menyimpan data pengguna.

Tabel 3.1 Tabel Pengguna

Nama Kolom	Tipe Data	Keterangan
id	INTEGER	<i>primary_key</i>
name	TEXT	Nama pengguna
gender	TEXT	Jenis kelamin pengguna
age	INTEGER	Umur pengguna
height	INTEGER	Tinggi pengguna
weight	INTEGER	Berat pengguna
PAL	REAL	<i>Physical Activity Level</i> pengguna
goal	TEXT	Tujuan pengguna

Tabel 3.2 Tabel Pengguna (lanjutan)

Nama Kolom	Tipe Data	Keterangan
calorie	REAL	Kalori yang dibutuhkan pengguna
protein	REAL	Protein yang dibutuhkan pengguna
fat	REAL	Lemak yang dibutuhkan pengguna
carbo	REAL	Karbo yang dibutuhkan pengguna
BMR	REAL	<i>Basal Metabolic Rate</i> yang dimiliki pengguna

2. Nama tabel : Makanan

Kegunaan : Menyimpan detail dari makanan yang akan dijadikan bahan rekomendasi

Tabel 3.3 Tabel Makanan

Nama Kolom	Tipe Data	Keterangan
id	INTEGER	<i>primary_key</i>
golongan	TEXT	Golongan makanan
nama_makanan	TEXT	Nama makanan
energi	REAL	Energi makanan
protein	REAL	Protein makanan
lemak	REAL	Lemak makanan
karbo	REAL	Karbohidrat makanan
status	TEXT	Status makanan dipakai atau tidak

3. Nama tabel : Plannin\_menu

Kegunaan : Menyimpan daftar menu yang akan digunakan untuk rekomendasi.

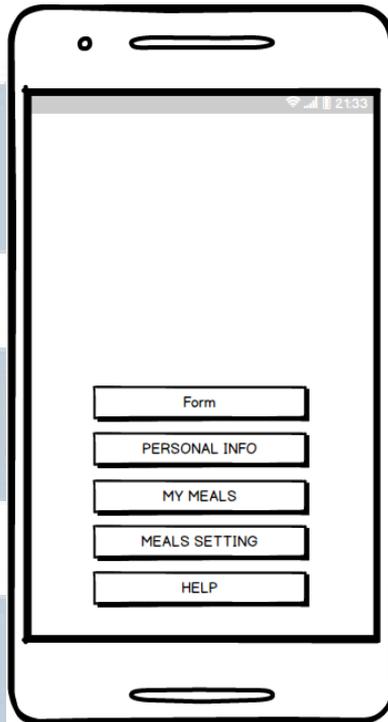
Tabel 3.4 Tabel Planning\_menu

Nama Kolom	Tipe Data	Keterangan
id	INTEGER	<i>primary_key</i>
date	TEXT	Hari rekomendasi makanan
Food_id	INTEGER	Id makanan
Time	TEXT	waktu makanan
<i>fitness</i>	REAL	<i>Fitness</i> makanan

### 3.2.7 Rancangan Antarmuka Aplikasi

Berikut rancangan aplikasi *meals fitness planning* dengan algoritma genetika berbasis android. Di dalam aplikasi ini terdapat delapan tampilan yang terdiri dari tampilan utama, tampilan pengisian data diri, tampilan rekomendasi makanana, tampilan detail makanan, tampilan daftar ganti makanan, tampilan data diri, tampilan pengaturan makanan, dan tampilan bantuan pengguna. Rancangan tampilan menu utama dapat dilihat pada gambar di bawah ini

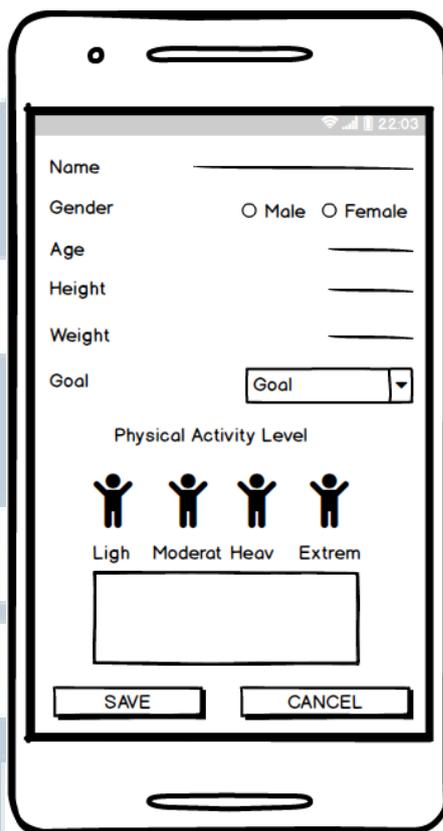
U N I V E R S I T A S  
M U L T I M E D I A  
N U S A N T A R A



Gambar 3.28 Menu Utama

Pada tampilan halaman utama, di dalam halaman ini terdapat lima tombol, tombol tersebut adalah tombol data diri, tombol *My Meals*, tombol *Personal Info*, tombol *Meals Setting*, dan tombol Bantuan.

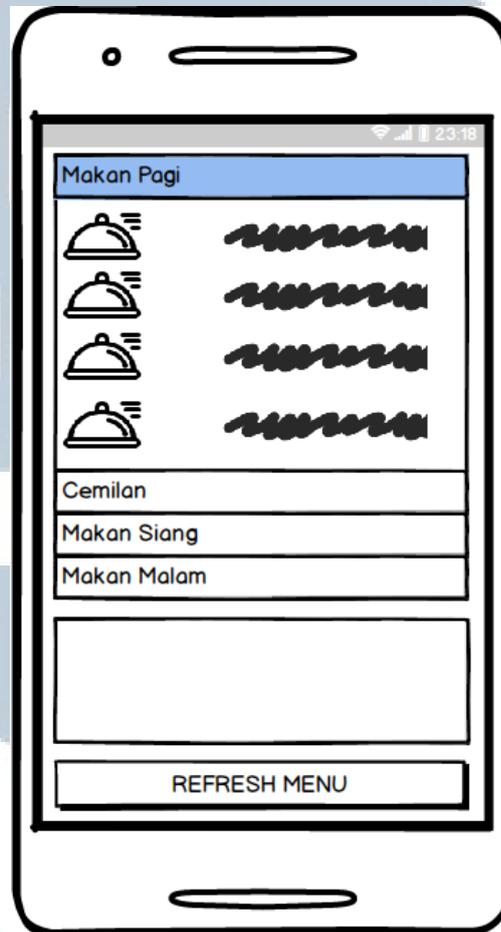
Tombol data diri digunakan untuk mengubah tampilan yang berfungsi untuk mengisi dan mengubah data diri pengguna aplikasi. Tombol *My Meals* digunakan untuk mengubah tampilan menjadi tampilan yang menampilkan daftar menu rekomendasi berdasarkan jadwal. Tombol *Personal Info* digunakan untuk mengubah tampilan yang menampilkan detail data diri pengguna aplikasi. Tombol *Meals Setting* digunakan untuk mengubah tampilan yang berfungsi untuk menampilkan daftar menu makanan berdasarkan kategori yang bertujuan agar pengguna dapat memilih makanan yang tidak diinginkan di dalam rekomendasi. Tombol bantuan digunakan untuk mengubah tampilan yang berfungsi untuk memberikan penjelasan bagi pengguna bagaimana cara menggunakan aplikasi.



Gambar 3.29 *Form Insert Data User*

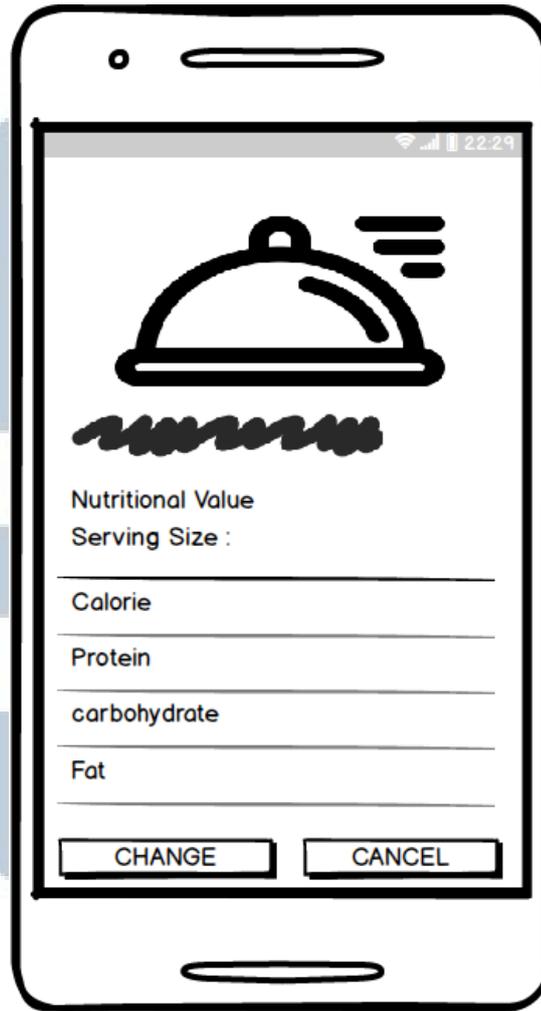
Selanjutnya tampilan pengisian data pribadi pengguna yang dapat dilihat di gambar. Tampilan ini menyediakan komponen *edit text* yang digunakan untuk mengisi nama pengguna, umur pengguna, tinggi pengguna, dan berat pengguna. Sedangkan untuk memilih jenis *gender* terdapat komponen *radio button*. Untuk memilih jenis *goal* yang diinginkan pengguna terdapat komponen *spinner*. Sedangkan untuk memilih tingkat *Physical Activity Level* pengguna terdapat empat *image view* dengan masing-masing tingkatan, untuk mengetahui penjelasan dari *Physical Activity Level* yang dipilih pengguna terdapat *text view* yang akan memberikan penjelasan dari *Physical Activity Level* yang dipilih pengguna. Terdapat dua *button*, *button* pertama yang berada di sebelah kiri digunakan untuk menyimpan data baru pengguna atau perubahan data pengguna dan *button* yang

berada di sebelah kiri digunakan untuk membatalkan dan mengembalikan tampilan ke menu utama.



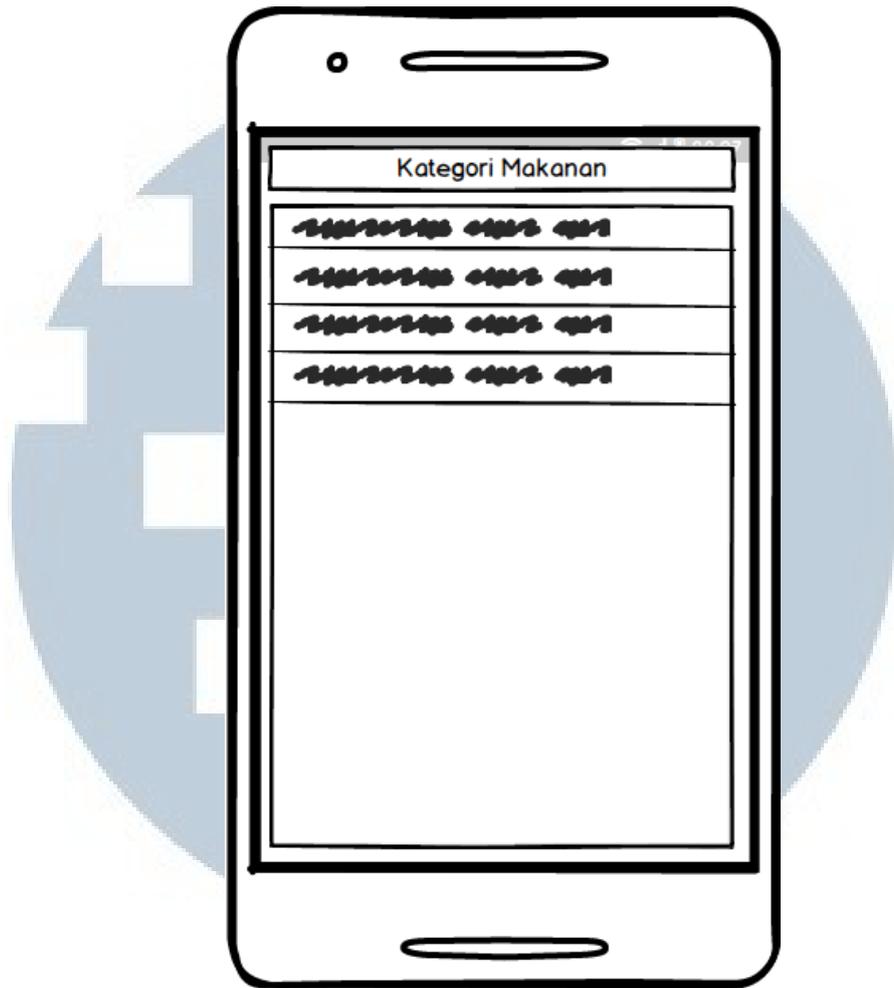
Gambar 3.30 Tampilan Rekomendasi Makanan

Tampilan selanjutnya adalah tampilan rekomendasi makanan yang dapat dilihat pada gambar di atas. Tampilan ini berguna untuk menunjukkan daftar rekomendasi makanan yang dapat digunakan pengguna untuk memenuhi kebutuhan pada saat itu. Terdapat komponen *expandable list view* yang berguna untuk menunjukkan daftar rekomendasi makanan berdasarkan waktunya. Selanjutnya terdapat *text view* yang berfungsi untuk menunjukan jumlah total kalori, protein, karbohidrat, dan lemak pada daftar rekomendasi makanan tersebut. Selanjutnya terdapat *button* yang berguna untuk memuat ulang daftar makanan.



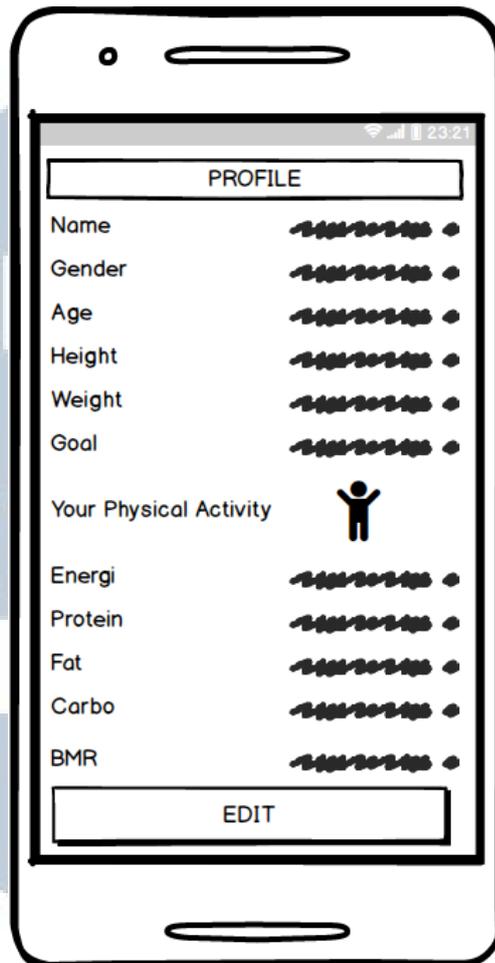
Gambar 3.31 Tampilan Detail Menu Makanan

Tampilan selanjutnya adalah tampilan detail dari makanan yang dipilih yang dapat dilihat seperti gambar di atas. Tampilan ini menampilkan detail dari makanan yang dipilih dari tampilan daftar rekomendasi makanan sebelumnya. Terdapat *image view* komponen yang berguna untuk menampilkan gambar makanan dari makanan yang dipilih. Terdapat *text view* untuk menunjukkan nama makanan, *nutrition value* seperti jumlah kalori, protein, karbohidrat, dan lemak. Terdapat dua *button*, *button* pertama untuk mengubah makanan dan *button* kedua untuk membatalkan dan mengembalikan ketampilan sebelumnya.



Gambar 3.32 Tampilan Daftar Pengganti Makanan

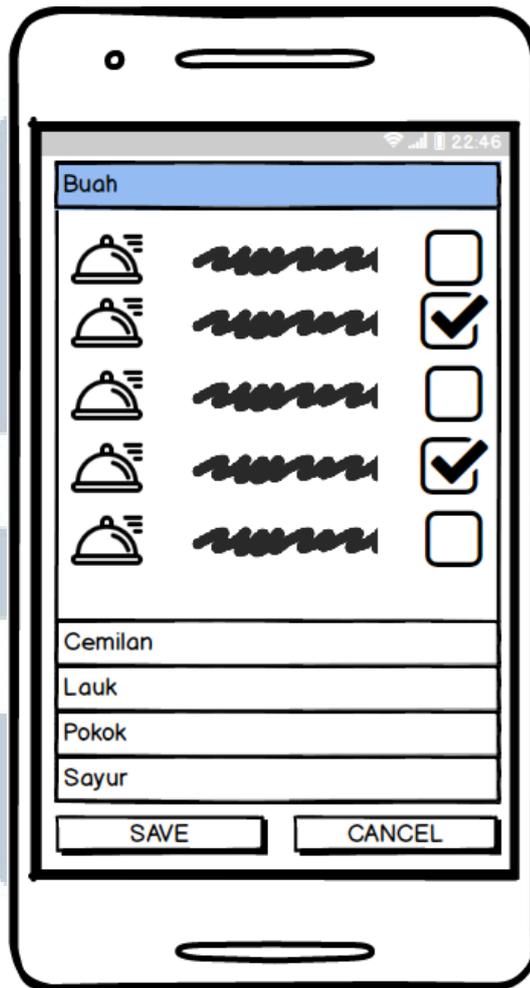
Tampilan selanjutnya adalah tampilan yang menampilkan daftar pengganti makan yang dapat dilihat pada gambar di atas. Di dalam tampilan pengguna dapat melihat daftar ganti makanan yang ingin diganti berdasarkan kategorinya. Di dalam tampilan ini terdapat komponen *text view* untuk menunjukkan kategori dari makanan yang ingin diganti. Komponen selanjutnya adalah *list view* yang berguna untuk menunjukkan daftar makanan pengganti.



Gambar 3.33 Tampilan Detail Data Pengguna

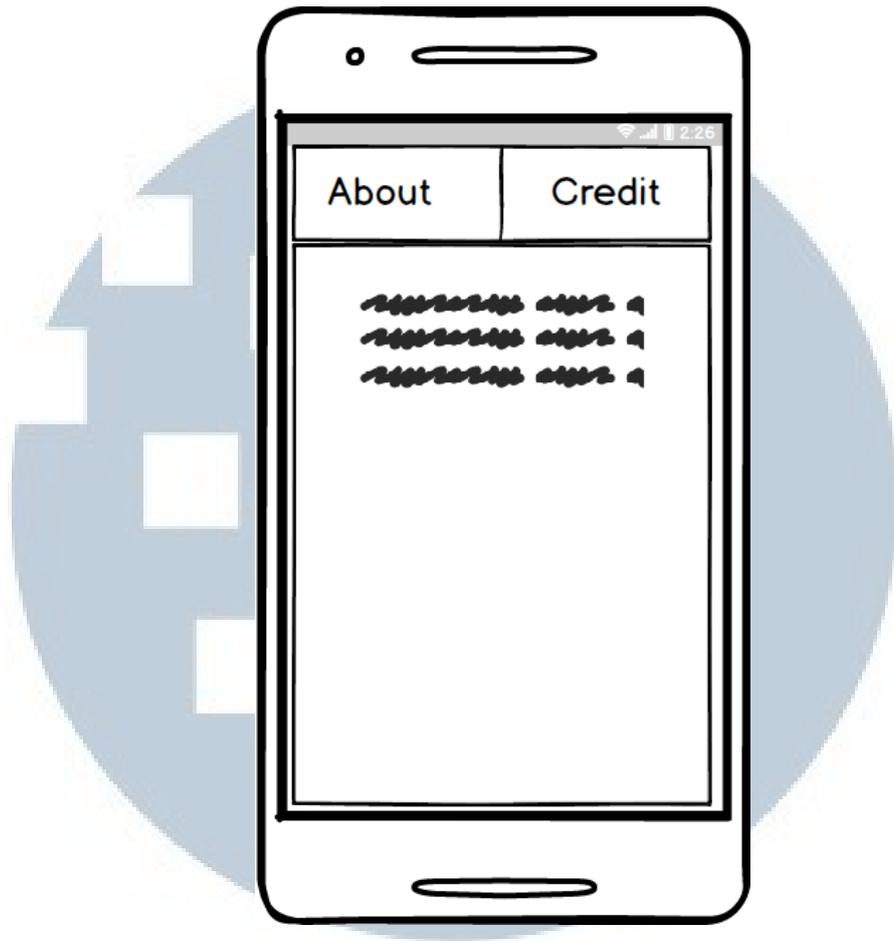
Tampilan selanjutnya adalah tampilan *profile* data diri pengguna yang dapat dilihat seperti gambar di atas. Pada tampilan ini pengguna dapat melihat detail dari data diri pengguna. Komponen yang terdapat dalam tampilan ini adalah *text view* yang menunjukkan judul tampilan dan menunjukkan detail data diri pengguna di dalam *table layout*. Terdapat satu buah komponen *button* untuk mengubah data diri pengguna.

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA



Gambar 3.34 Tampilan Pengaturan Bahan Makanan

Selanjutnya adalah tampilan pengaturan makanan seperti pada gambar di atas. Dimana tampilan ini berfungsi untuk memilih makanan yang tidak ingin dimunculkan di dalam daftar rekomendasi makanan. Terdapat komponen *list view* yang berisikan komponen *image view* untuk menunjukkan gambar dari makanan, *text view* untuk menunjukkan nama makanan, dan *checkbox* untuk menandai makanan yang tidak ingin dimunculkan ke dalam rekomendasi makanan. Daftar makanan yang dimunculkan dibagi berdasarkan kategori makanan. Terdapat juga komponen *button* dimana berfungsi untuk menyimpan hasil pilihan pengguna.



Gambar 3.35 Tampilan Menu Bantuan

Selanjutnya adalah tampilan dari menu *Help*. Dimana tampilan ini berfungsi untuk menampilkan informasi mengenai cara pemakaian dari menu-menu yang terdapat di dalam aplikasi.

UMN  
UNIVERSITAS  
MULTIMEDIA  
NUSANTARA