



### **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 III

### METODOLOGI PENELITIAN DAN PERANCANGAN SISTEM

#### 3.1 Metodologi Penelitian

Metode Penelitian yang digunakan dalam penelitian ini antara lain:

a. Studi Fisibilitas

Studi fisibilitas digunakan untuk mengukur efektivitas dari faktor yang memengaruhi pola makan. Pengukuran ini dilakukan dengan menggunakan kuisisioner digital yang disebar secara *online* maupun melalui media sosial. Target responden dari kuisisioner studi fisibilitas ini adalah orang penderita diabetes.

b. Studi Literatur

Dalam studi literatur, pembelajaran terhadap berbagai teori-teori yang berhubungan dengan perancangan dan pembangun aplikasi *android* yang menggunakan metode gamifikasi. Salah satu teori yang diambil yaitu *Framework* dari Werbach dan Hunter.

c. Perancangan dan Pembuatan Sistem

Tahap ini dimulai dengan perancangan arsitektur dari aplikasi android sebagai dasar dari pembuatan sistem. Sistem yang dibangun berupa aplikasi *mobile* berbasis android dan menggunakan metode gamifikasi. Setelah rancangan arsitektur sistem dibuat, tahap ini dilanjutkan dengan perancangan dan pemrograman sistem aplikasi android. Pembuatan sistem juga mencakup perancangan sistem antarmuka pengguna dan penentuan fitur-fitur tambahan yang dibutuhkan pada sistem.

d. Pengujian sistem

Proses pengujian sistem dimulai dengan menggunakan metode Hedonic-motivation System Adoption Model (HMSAM), dimana pengguna mencoba menggunakan aplikasi secara beta. Metode dengan HMSAM dilakukan agar dapat mengukur tingkat penerimaan pengguna terhadap sistem. Setelah itu akan dibagikan kuisisioner dan terdapat delapan bagian utama dari kuisisioner yaitu *joy, control, focused immersion, temporal dissociation, curiosity, perceived ease of use, perceived usefulness*, dan *behavioral intention to use*.

e. Evaluasi

Evaluasi sistem dilakukan dengan menganalisis kuisisioner yang dihasilkan dari studi lapangan pengujian sistem. Hasil evaluasi akan disimpulkan melalui penggunaan Skala Likert agar dapat mengukur kegunaan dari penerapan sistem.

### **3.2. Teknik Pengumpulan Data**

Pada tahap pengembangan sistem, pengumpulan data akan dilakukan dengan mempelajari dokumentasi dari buku, jurnal, dan sumber-sumber terpercaya lainnya. Data tersebut berperan sebagai acuan dalam menentukan elemen-elemen gamifikasi yang akan digunakan dalam sistem. Saat tahap pengujian, dilakukan studi lapangan dengan cara meminta pengguna untuk menggunakan aplikasi dalam periode waktu tertentu. Setelah itu pengguna diminta mengisi kuisisioner dalam bentuk Skala Likert. Skala Likert dipilih karena data yang bersifat kualitatif dapat dikuantisasi untuk kepentingan analisis selanjutnya.

### **3.3. Teknik Pengambilan Sampel**

Teknik pengambilan sampel yang digunakan pada penelitian ini, baik dalam studi fisibilitas maupun pengujian sistem, adalah Proportionate Stratified Random Sampling (Dawson, 2009). Teknik pengambilan sampel ini digunakan karena Menurut Roscoe dalam bukunya *Research Methods for Business* (Sugiyono, 2012), ukuran sampel yang diperlukan adalah paling sedikit berjumlah 30. Oleh karena itu, pengumpulan data dari sampel dilakukan terhadap minimal 30 responden.

### **3.4. Analisis Gamifikasi**

Implementasi gamifikasi paling baik dilakukan dalam enam tahapan yang dikenal dengan sebutan *Six Steps to Gamification* (Werbach dan Hunter, 2012).

#### *a. Define business Objectives*

Tujuan utama dibangunnya aplikasi ini adalah meningkatkan ketaan dalam menjaga pola makan bagi pengguna.

#### *b. Delineate target behaviors*

Aksi-aksi yang harus dilakukan pengguna pada aplikasi ini akan dijelaskan sebagai berikut.

- Pengguna melakukan aksi *register* untuk mendaftarkan diri agar dapat menggunakan fitur-fitur yang telah disediakan. Banyaknya orang yang mendaftar dapat diukur dengan melihat jumlah pengguna di *database*.
- Pengguna melakukan aksi memilih makanan yang ada pada daftar makanan. Makanan yang terpilih dapat dilihat pada daftar makanan yang telah dikonsumsi di *database*.

- Pengguna menjalankan misi tertentu untuk mendapatkan *reward* yang telah ditentukan. Hal ini dapat diukur dengan melihat jumlah misi yang telah diselesaikan.
- Pengguna mengumpulkan lencana sebagai tanda pencapaian suatu usaha tertentu. Pencapaian tersebut dapat dilihat pada jumlah *badges* yang telah didapatkan pengguna pada halaman *profile*. Pengguna dapat mengumpulkan lencana apabila telah menyelesaikan misi utama yang berkaitan dengan keaktifan pengguna dalam menggunakan aplikasi.
- Pengguna membandingkan peringkat dirinya dengan orang lain melalui fitur *leaderboard*. Peringkat dapat diukur dengan jumlah poin yang dimiliki setiap pengguna. Melalui fitur ini rasa kompetitif pengguna dapat meningkat sehingga pengguna akan berusaha mengumpulkan poin secara terus menerus.

c. *Describe your players*

Aplikasi menyediakan fitur-fitur untuk tiga tipe pemain yaitu *quest* untuk pemain tipe *explorers*. *Badges* untuk pemain tipe *achievers*, *leaderboard* untuk pemain tipe *killers*.

d. *Devise activity cycles*

Aktivitas pada aplikasi ini dibagi menjadi dua siklus yaitu engagement loops dan progression stairs.

- *Engagement Loops*

Siklus ini dibagi menjadi tiga tahapan, pada tahapan *motivation* pengguna dapat melihat *leaderboard* untuk melihat peringkat pengguna, dan mendapatkan *badges*. Pada tahapan *action* pengguna dapat memilih makanan yang sesuai dengan

jadwal makan dan menyelesaikan *quest* yang diberikan. Pada tahapan *feedback* pengguna menerima *rewards* dalam bentuk poin yang digunakan untuk menaikkan *level*.

- *Progression Stairs*

Tingkatan dari setiap pengguna digambarkan menggunakan *level*. Pengguna dapat menaikkan *level* apabila berhasil mengumpulkan poin dalam jumlah tertentu. Semakin tinggi *level*, poin yang harus dikumpulkan pengguna semakin banyak.

e. *Don't forget the fun*

Pengguna harus merasa tertarik dan senang saat menggunakan sistem sehingga pengguna termotivasi untuk menggunakan sistem kembali. Aspek *game* yang memotivasi pengguna dijabarkan sebagai berikut.

- *Achievement*, tipe pemain *achiever* harus memiliki tujuan tertentu agar termotivasi menggunakan sistem.

- *Badges*, tipe pemain *achiever* termotivasi untuk mengumpulkan semua *badge* yang ada.

- *Leaderboard*, tipe pemain *killer* ingin mengalahkan pemain lainnya, dengan adanya *leaderboards* pemain merasa senang apabila telah berhasil mengalahkan pemain lainnya dan menduduki peringkat yang tinggi.

- *Quest*, tipe pemain *achievers* termotivasi untuk menyelesaikan semua *quest* yang ada dan pemain *explorer* akan termotivasi ketika ada *quest* baru yang muncul dalam sistem.

f. *Deploy the appropriate tools*

Sistem yang akan dibuat adalah aplikasi berbasis *mobile* menggunakan metode gamifikasi. Pengguna harus melakukan proses login agar dapat mengakses seluruh fitur yang ada, apabila pengguna belum memiliki akun maka pengguna harus melakukan proses *register* terlebih dahulu. Fitur utama yang disediakan untuk pengguna adalah fitur untuk memilih makanan dengan jumlah kalori yang kurang dari maksimal kalori harian yang dikonsumsi.

### **3.5. Perancangan Sistem**

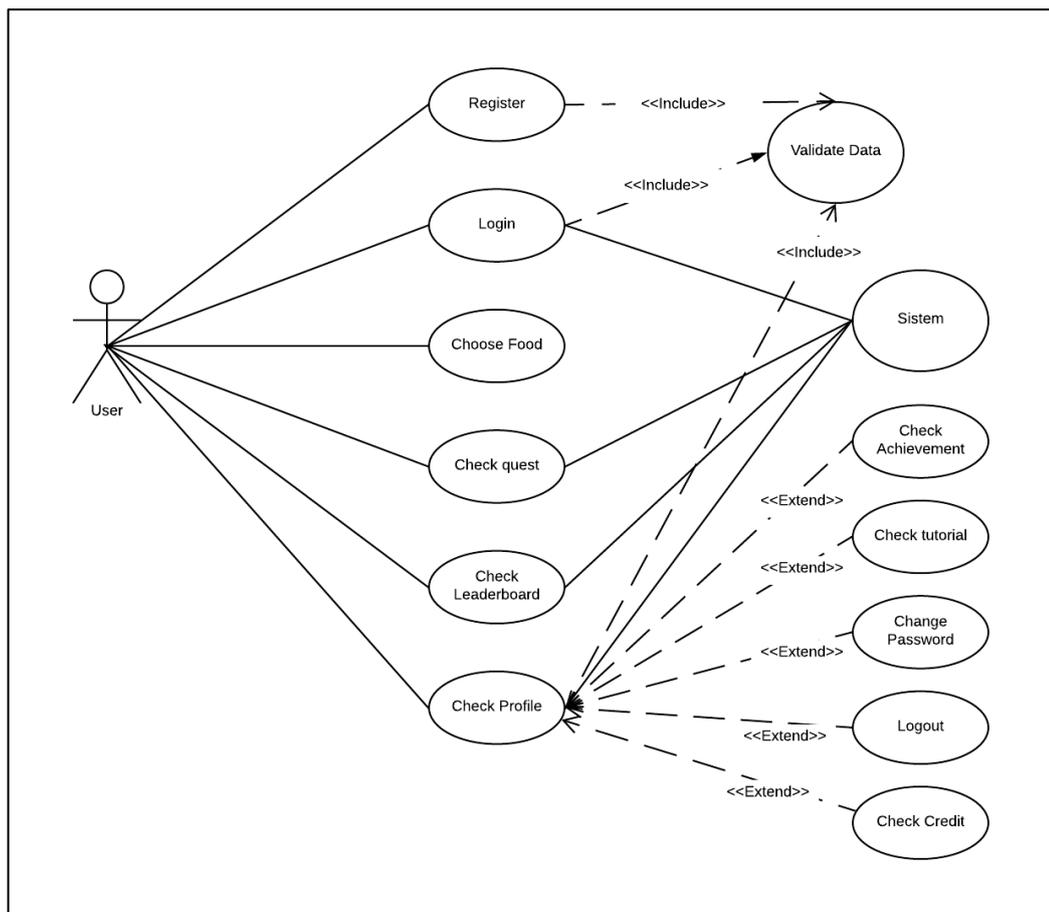
Adapun penelitian ini dibuat menggunakan basis *android* dengan android studio dan menggunakan MySQL untuk membangun *database* dari *website*. Untuk membuat aplikasi *android* dengan metode gamifikasi dilakukan beberapa tahapan perancangan dimulai dari perancangan *unified modeling language* (UML) yang terdiri dari beberapa bagian meliputi *usecase*, *activity diagram*, *sequence diagram*, dan *class diagram*. Setelah itu tahap perancangan dilakukan dengan pembuatan *entity relationship diagram*, perancangan tampilan antarmuka, dan perancangan aset-aset yang digunakan.

#### **3.5.1 Use Case Diagram**

*Use Case Diagram* untuk aplikasi ini memiliki 2 aktor yaitu *user* dan sistem. *User* terhubung dengan 9 *use case* meliputi *register*, *create personal info*, *login*, *logout*, *choose food*, *give points for choose food*, *check quest*, *check leaderboard*, dan *check profile*. Sedangkan sistem terhubung dengan enam *use case* meliputi

login, logout, give points for choose food, check quest, check leaderboard, dan check profile.

Use case validate data terhubung include dengan use case register, create personal info, login, dan check profile. Hubungan include di use case validate data menunjukkan bahwa usecase validate data merupakan bagian yang tidak terpisahkan dari keempat use case lainnya. Sedangkan use case check reward yang terhubung exclude dengan give points for choose food merupakan fungsionalitas tambahan. Use case diagram check reward Use case diagram secara lengkap dapat dilihat pada Gambar 3.1.

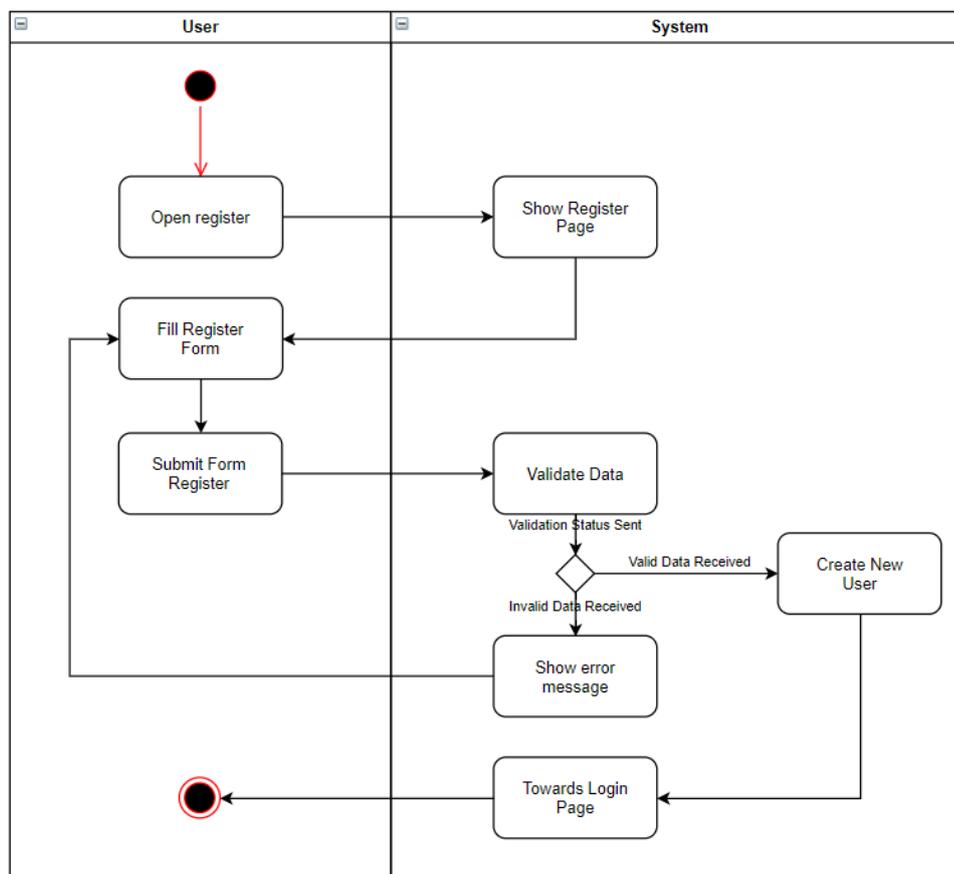


Gambar 3.1 Use Case Diagram

### 3.5.2 Activity Diagram

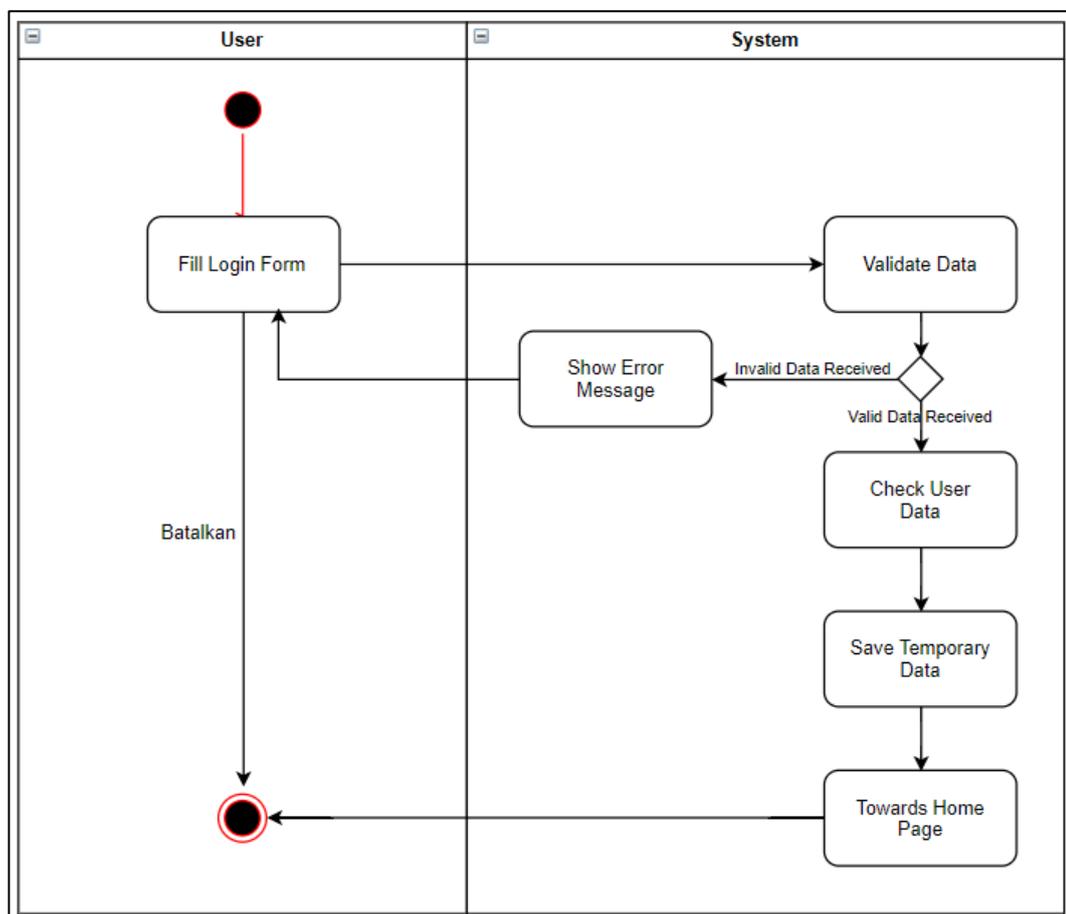
Terdapat 11 *activity diagram* dalam perancangan aplikasi untuk mengatur pola makan menggunakan metode gamifikasi. *Activity diagram* berguna untuk menjelaskan lebih lanjut alur yang terjadi di dalam *use case*. Berikut penjelasan untuk masing-masing diagram.

Proses *register* dimulai ketika pengguna melakukan pengisian *form* pendaftaran, kemudian dilakukan pengecekan data pengguna. Jika data pengguna tidak sesuai maka dibuat *error messages*. Jika data sesuai kriteria maka data pengguna disimpan di database. *Activity diagram* untuk *register* ditampilkan pada Gambar 3.2.



Gambar 3.2 Activity Diagram Register

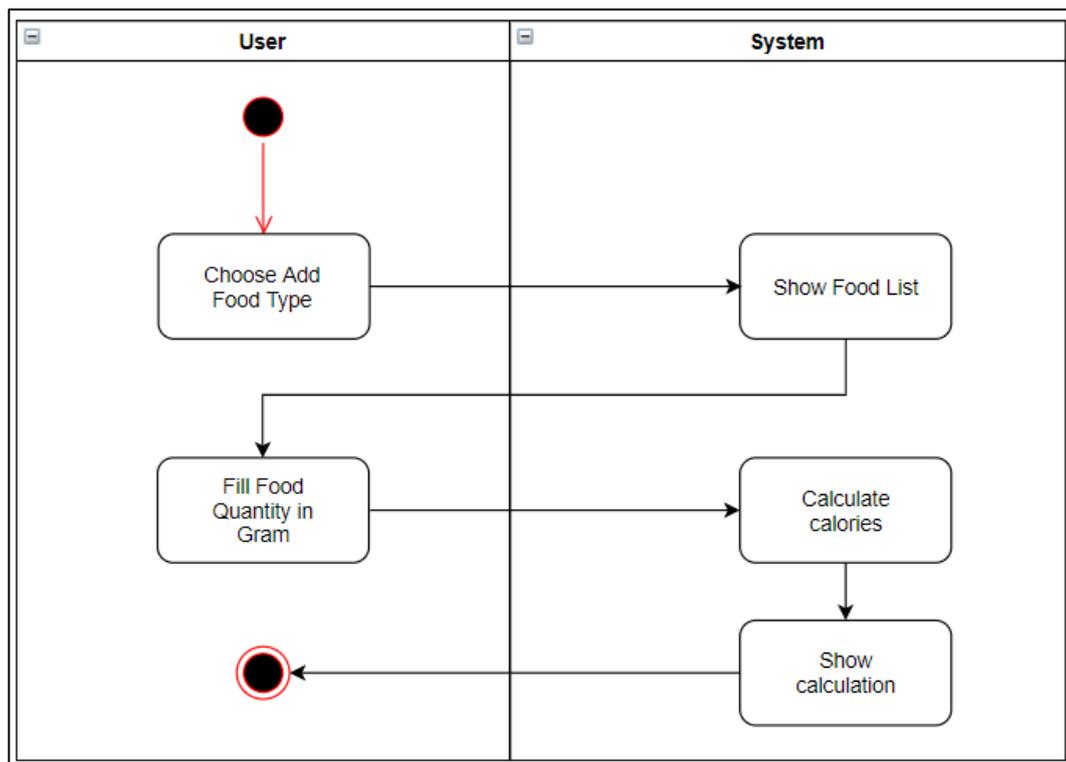
Setelah melakukan *register*, pengguna dapat melakukan *login* dengan memasukkan *email* dan *password*. Data tersebut akan divalidasi lalu *email* dan *password* akan dicocokkan dengan data di database. Jika data tidak sesuai dengan data yang terdaftar, maka akan dibuat *error message*. Jika data sesuai, maka status pengguna akan diubah menjadi aktif. Gambar 3.3 menampilkan gambar *activity diagram login*.



Gambar 3.3 *Activity Diagram Login*

Fitur utama yang dapat dilakukan pengguna adalah *Choose food* atau memilih dan menambahkan makanan kedalam daftar makanan sehari-hari di jam tertentu. Dibagian *Choose food* terdiri dari 5 bagian utama yang dipisahkan

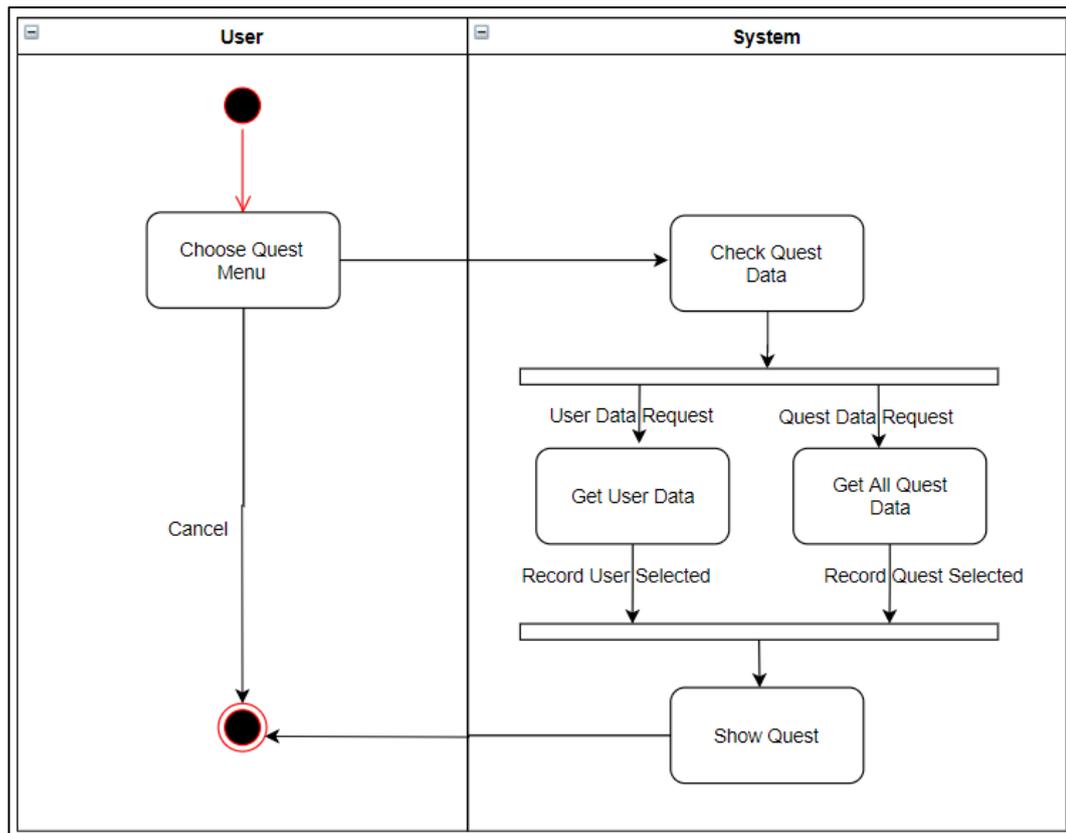
berdasarkan jadwal makanan harian yaitu *breakfast*, *breakfast snack*, *lunch*, *lunch snack*, dan *dinner*. Dimana secara umum, penderita diabetes dianjurkan untuk disiplin dalam hal mengatur jadwal makanan. Ketika pengguna akan memilih makanan, sistem akan menampilkan daftar makanan. Lalu pengguna diminta untuk mengisi kuantiti dalam satuan gram. Setelah pengguna memilih beberapa makanan dan kuantitinya, sistem akan menghitung kandungan kalori berdasarkan kuantiti yang telah dimasukkan oleh pengguna. Lalu sistem akan menampilkan hasil perhitungan tersebut kehalaman utama. Diagram 3.4 menampilkan gambar *activity diagram Choose food*.



Gambar 3.4 Activity Diagram Choose Food

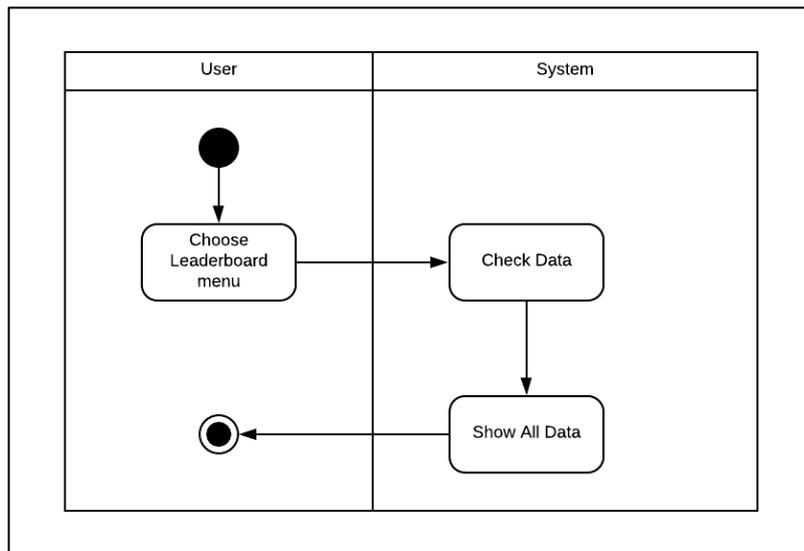
Pada halaman *quest*, terdapat daftar *quest* yang telah ditentukan dari sistem. Pengguna dapat melihat *quest* yang telah selesai atau belum dari pengecekan data dari sistem. Ketika pengguna memilih menu *quest*, system akan mengecek semua

data *quest*. Lalu sistem akan melakukan pengambilan data pengguna dan data *quest*. Selanjutnya, sistem akan menampilkan data *quest* yang telah diselesaikan pengguna maupun *quest* yang belum diselesaikan pengguna. *Activity diagram* untuk *check quest* dapat dilihat di gambar 3.5



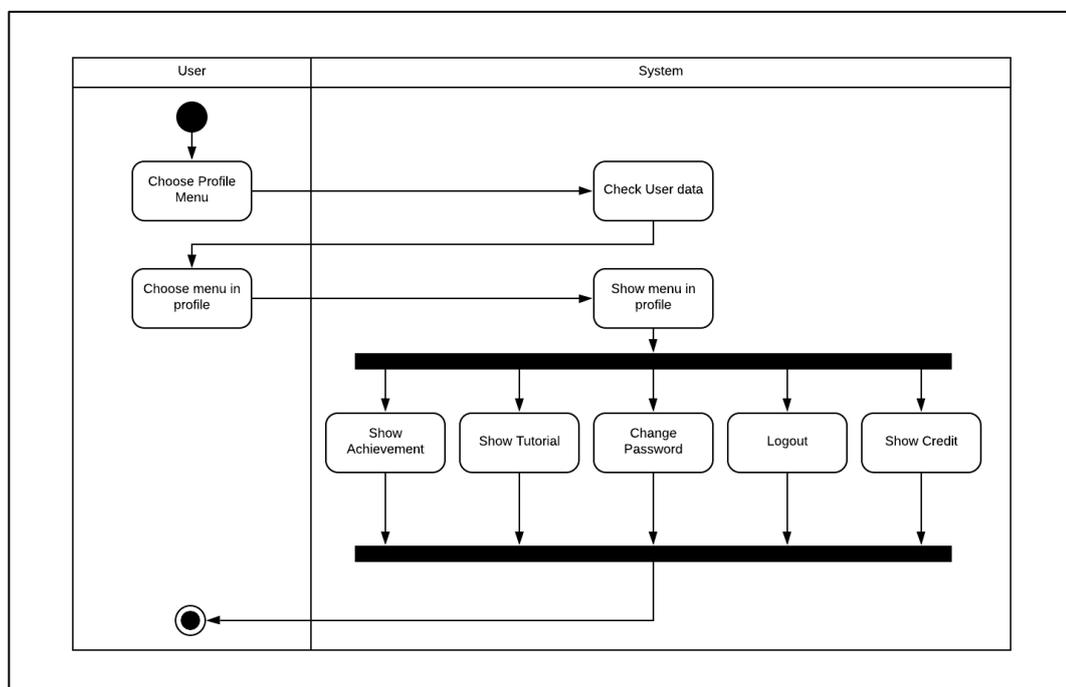
Gambar 3.5 *Activity Diagram Check Quest*

Selanjutnya pada menu *leaderboard*, sistem akan menampilkan daftar pengguna yang telah mencapai peringkat 3 tertinggi dari semua pengguna yang terdaftar. Gambar 3.6 merupakan *activity diagram* dari *check leaderboard*.



Gambar 3.6 Activity Diagram Check Leaderboard

Ketika pilih halaman *profile*, akan diberikan pilihan banyak menu antara lain *achievement*, *tutorial*, *change password*, *logout*, dan *credit*. Activity diagram untuk *check profile* bisa dilihat pada Gambar 3.7.

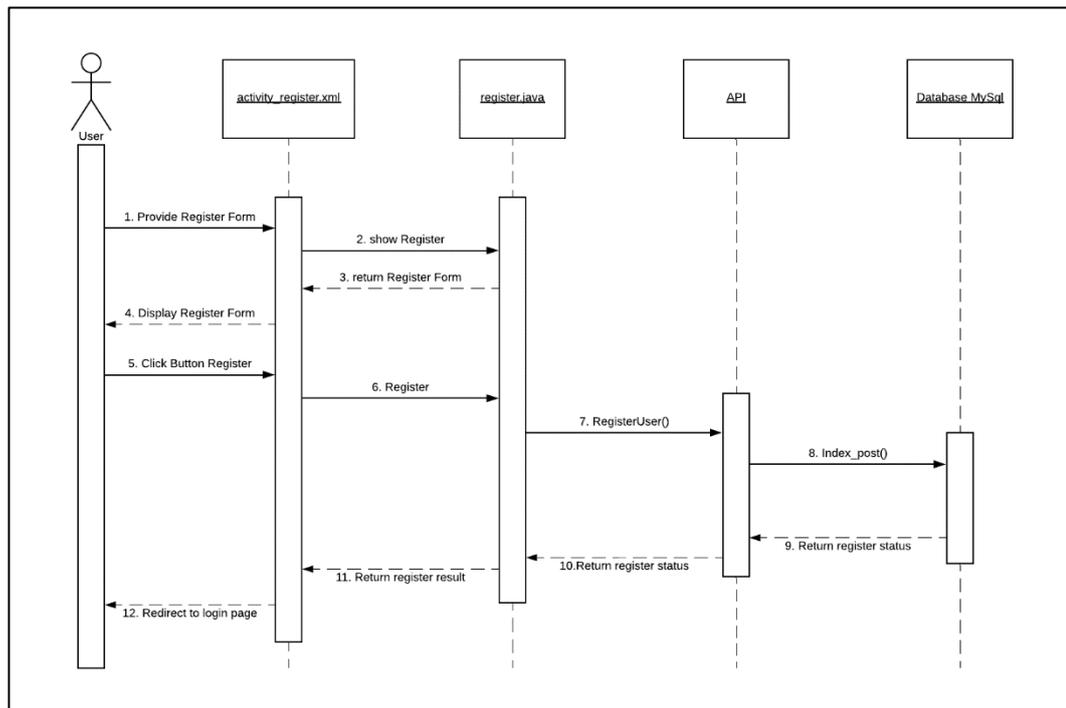


Gambar 3.7 Activity diagram check profile

### 3.5.3 Sequence Diagram

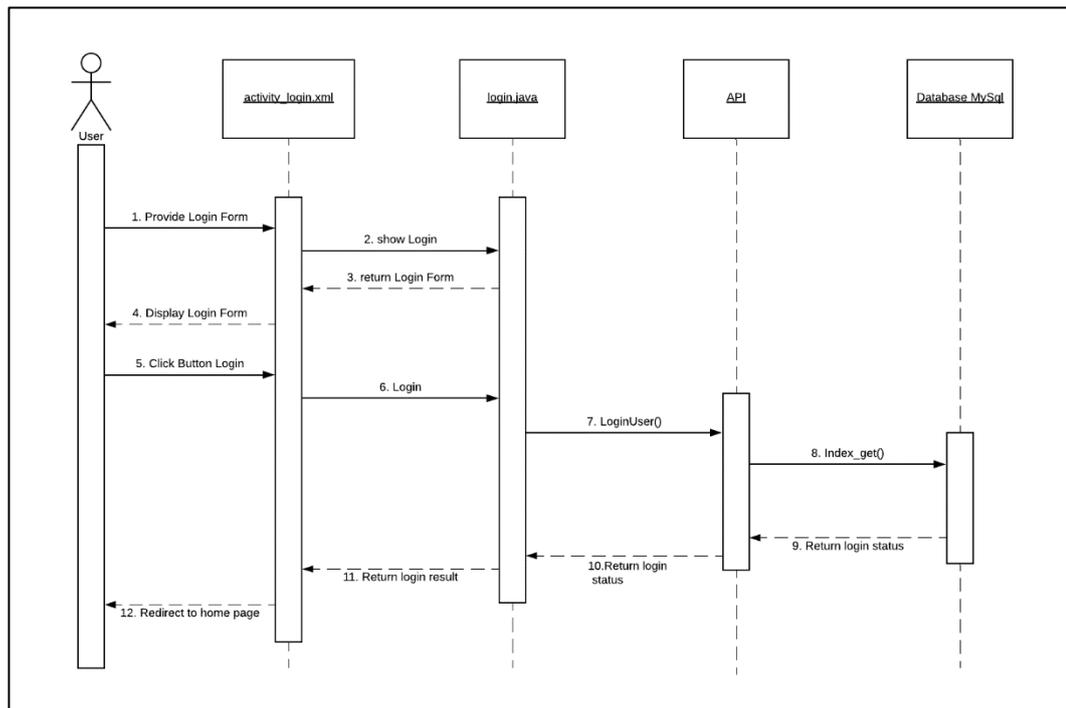
*Sequence Diagram* digunakan untuk menggambarkan pemanggilan *function* beserta dengan *class* yang digunakan. Terdapat 8 diagram dalam perancangan aplikasi ini. Berikut penjelasan lebih lanjut untuk masing-masing *sequence diagram*.

Saat pengguna menekan *register* akan terjadi *request* untuk membuka halaman baru ke *register class*, setelah memproses data yang diperlukan *register class* akan mengarahkan pengguna ke halaman *register*. Saat pengguna selesai mengisi data dan menyelesaikan tahapan *register* maka terjadi pemanggilan *function Register User* ke API, lalu di API akan melakukan *request via retrofit* ke database dengan menggunakan *post*. Setelah proses berhasil, database akan mengeluarkan status berhasil atau tidak ke API dan menuju *register class*. *Register class* akan mengembalikan hasil apabila semua proses telah berhasil. Lalu pengguna akan diarahkan ke halaman *login*. Gambar 3.8 menampilkan *sequence diagram register*.



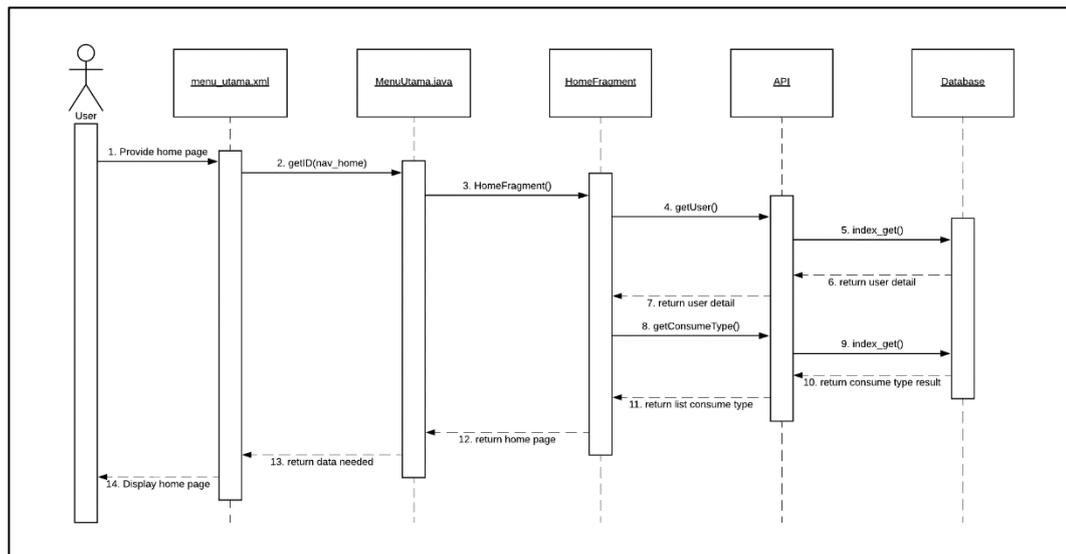
Gambar 3.8 *Sequence Diagram Register*

Ketika menu *login* ditekan, *login class* akan mengarahkan pengguna ke halaman *login*. Saat pengguna selesai mengisi data berupa *email* dan *password* maka terjadi pemanggilan *function LoginUser* ke API, lalu di API akan melakukan *request via retrofit* ke database dengan menggunakan *get*. Setelah proses berhasil, database akan mengeluarkan status berhasil atau tidak ke API dan menuju *login class*. *Login class* akan mengembalikan hasil apabila semua proses telah berhasil. Lalu pengguna akan diarahkan ke halaman utama. Gambar 3.9 menampilkan *sequence diagram login*.



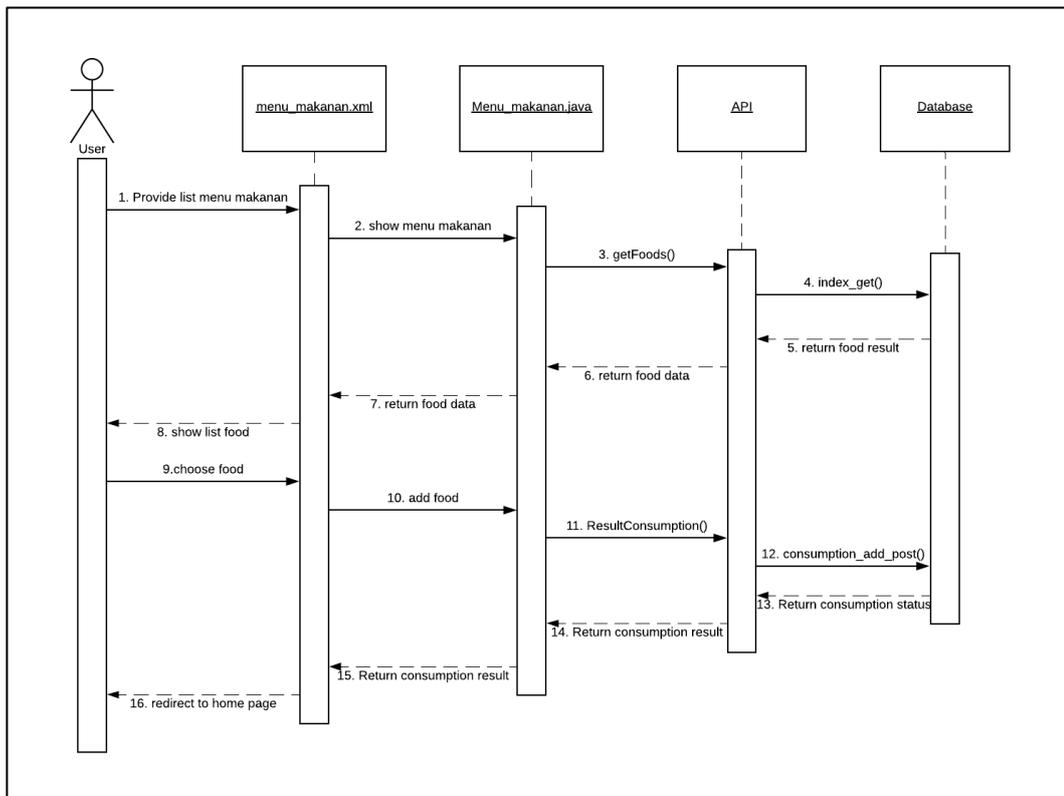
Gambar 3.9 *Sequence Diagram Login*

Setelah pengguna berhasil masuk ke halaman utama, menu utama *class* akan *get ID* berupa *nav\_home* yang merupakan *id* dari navigasi untuk membuka halaman utama. Lalu menu utama *class* akan mencari *fragment* yang cocok dengan *id nav\_home* berupa *home fragment*. *Home fragment* akan mengambil data pengguna dari API yang akan diteruskan ke *database* agar bisa ditampilkan nama dan dihitung jumlah minimum kalori harian berdasarkan data *register* yang terdaftar. Setelah *home fragment* mendapatkan *data user*, *home fragment* akan mengambil *consume type* ke API dan diteruskan ke *database*. Lalu *database* akan merespon dengan menampilkan daftar *consume type* ke API dan diteruskan ke *home fragment*, disimpan didalam menu utama *class* dan ditampilkan kepada pengguna berupa *recycle view*. Gambar 3.10 menampilkan *sequence diagram* menu utama.



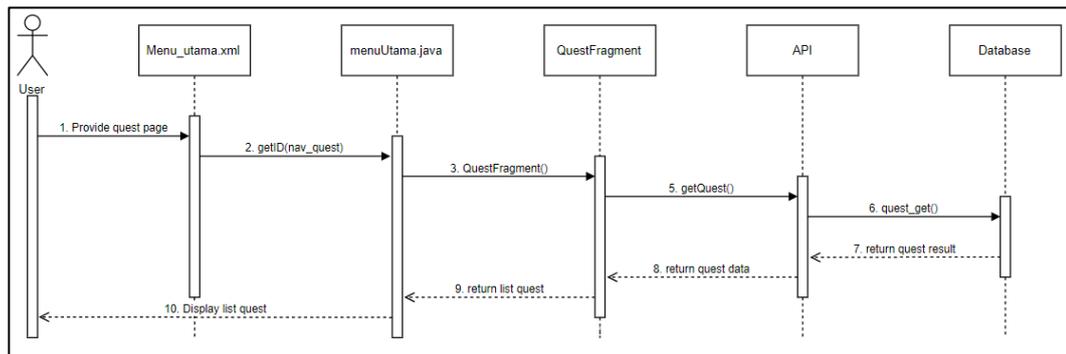
Gambar 3.10 *Sequence Diagram Menu Utama*

Ketika pengguna memilih tombol *add food* dari halaman utama, menu makanan *class* akan memanggil *function getFoods* ke API, lalu di API akan melakukan *request via retrofit* ke database dengan menggunakan *get*. Setelah proses berhasil, database akan mengeluarkan status berhasil atau tidak ke API dan menuju menu makanan *class*. Menu makanan *class* akan mengembalikan hasil berupa daftar makanan. Setelah pengguna menerima daftar menu makanan, pengguna bisa memilih makanan tersebut, direkam oleh menu makanan *class*, dan diteruskan ke database dengan *function consumption\_add\_post*. Lalu database akan merekam data makanan yang telah ditambahkan oleh pengguna. Dari database, API akan mengembalikan data yang terekam menuju halaman utama. Gambar 3.11 menampilkan *sequence diagram choose food*.



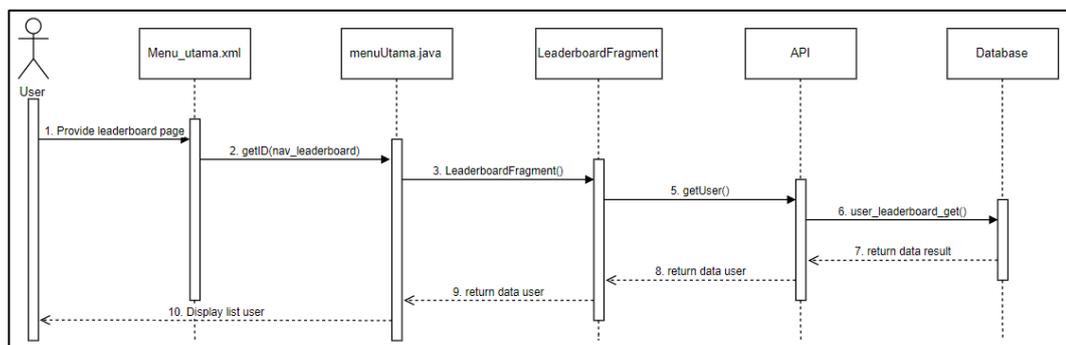
Gambar 3.11 *Sequence Diagram Choose Food*

Halaman *quest* dimulai dari menu utama dengan mencari *id nav\_quest* dan memanggil *function QuestFragment*. Lalu *questFragment* akan menuju API dengan menggunakan *function getQuest* dan diteruskan ke database. Lalu dari database akan merespon dengan mengembalikan hasil *quest* untuk menampilkan daftar *quest* melalui API, *QuestFragment*, dan menampilkan daftar *quest* kepada pengguna. Gambar 3.12 menampilkan *sequence diagram check quest*.



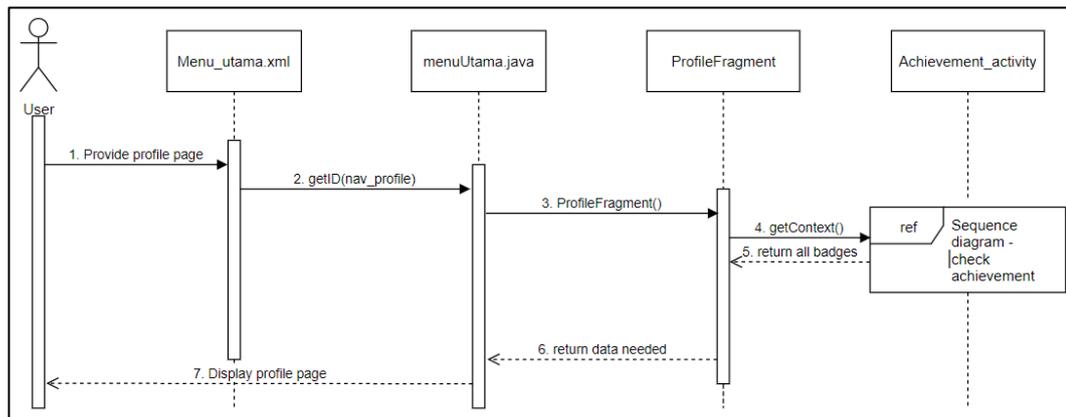
Gambar 3.12 *Sequence Diagram Check Quest*

Halaman *leaderboard* dimulai dari mencari *id nav\_leaderboard* dan memanggil *function leaderboard fragment*. Lalu *leaderboard fragment* akan *request getUser* ke API dan diteruskan ke database menggunakan *function user\_leaderboard\_get()*. Lalu dikembalikan hasilnya dari database menuju *leaderboard fragment* dan ditampilkan ke pengguna. Gambar 3.13 menampilkan *sequence diagram check leaderboard*.



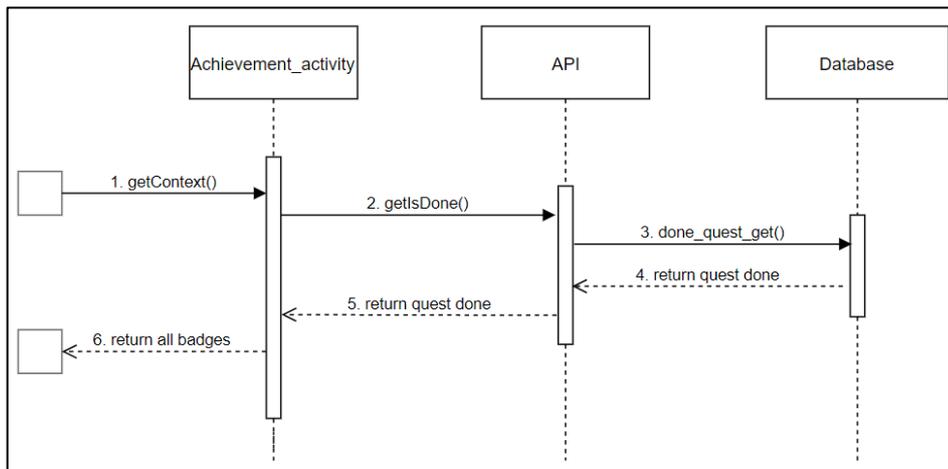
Gambar 3.13 *Sequence Diagram Check Leaderboard*

Halaman *profile* dari menu utama akan mencari *id* berupa *nav profile* dan memanggil *function profile fragment*. Lalu *profile fragment* akan memanggil *get context* untuk menuju halaman menu *achievement* dan menampilkan semua daftar *badge* kepada pengguna. Gambar 3.14 menampilkan *sequence diagram check profile*.



Gambar 3.14 *Sequence Diagram Check Profile*

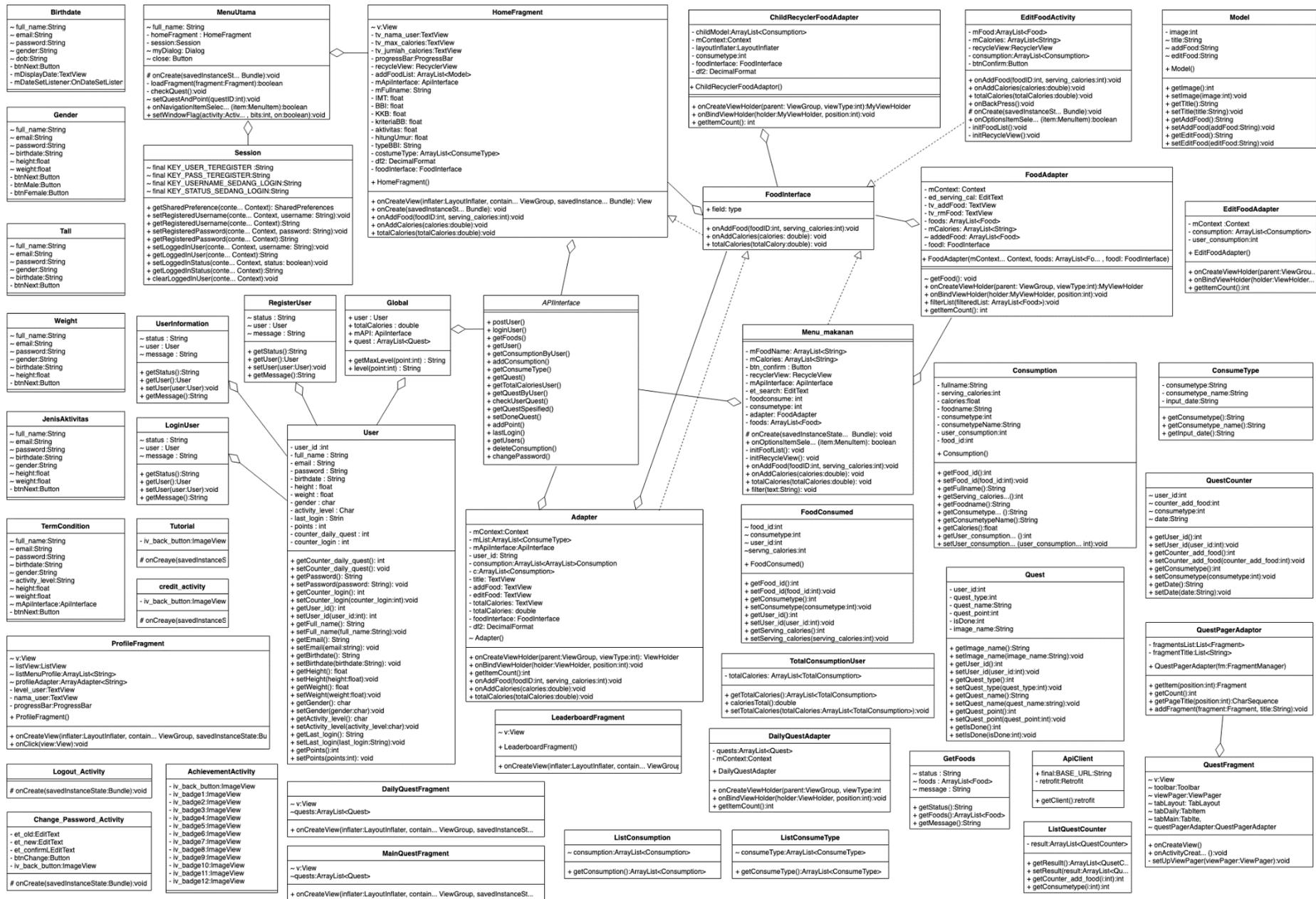
*Sequence diagram check achievement* bertujuan untuk melihat daftar *badge* dari *quest* yang telah selesai. Maka saat *function get context* dipanggil oleh *achievement class*, *achievement class* akan menggunakan *function get is done* ke API yang merupakan *function* untuk mengambil data *quest* yang telah selesai dan diteruskan ke database menggunakan *function done quest get*. Lalu database akan merespon dengan mengembalikan tanda *quest* yang telah selesai ke API dan dikembalikan ke *achievement class*. *Achievement class* akan menampilkan data *quest* yang telah selesai dalam bentuk *badge*. Gambar 3.15 menampilkan *sequence diagram check achievement*.



Gambar 3.15 *Sequence Diagram Check Achievement*

### 3.5.4 Class Diagram

*Class diagram* untuk aplikasi ini dapat dilihat pada Gambar 3.16. Terdapat beberapa bagian *class* yang dibuat pada aplikasi ini seperti *class activity*, *class fragment*, *class model*, dan *class adapter*. *Class activity* digunakan untuk interaksi dengan pengguna, *class fragment* untuk mengimplement beberapa fitur sebuah *activity*, *class model* digunakan untuk membantu menjalankan fungsi CRUD terhadap suatu database dan *class adapter* untuk mengatur *item* pada *list*. Gambar 3.16 menampilkan *class diagram*



Gambar 3.16 Class Diagram

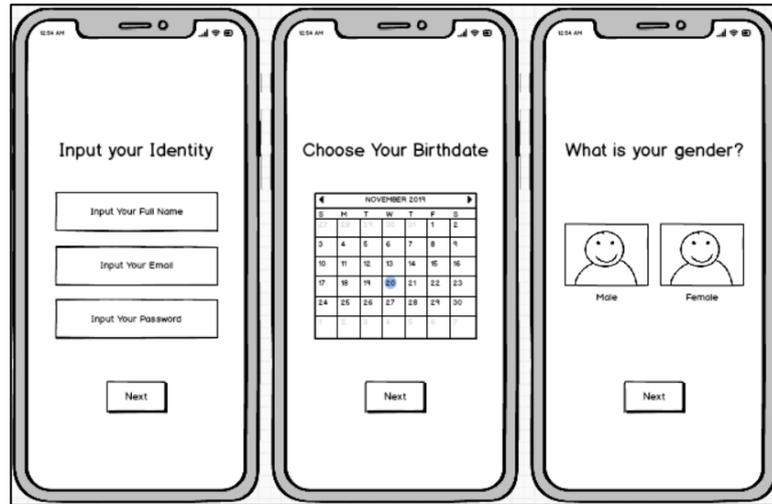
### 3.5.5 Perancangan antarmuka

Saat pertama kali pengguna membuka aplikasi, pengguna akan diarahkan ke halaman *login*, pada tampilan ini terdapat *form* yang terdiri dari dua kolom yaitu kolom *email* dan *password*. Terdapat tombol untuk melakukan *login*, tombol untuk melakukan pendaftaran, dan tombol untuk *forgot password*. Gambar 3.17 menunjukkan rancangan antarmuka untuk tampilan *login*.



Gambar 3.17 Rancangan Tampilan *Login*

Pengguna baru yang belum memiliki akun harus mendaftar terlebih dahulu. Terdapat beberapa data diri yang harus diisi oleh pengguna yaitu nama lengkap pengguna, *email*, *password*, tanggal lahir, jenis kelamin, berat badan, tinggi badan, dan jenis aktivitas. Pada sebelah kanan bawah ada tulisan *register* yang bisa ditekan untuk menuju ke tampilan *register*. Tampilan register dibagi-bagi menjadi beberapa kategori seperti pada Gambar 3.18.



Gambar 3.18 Rancangan tampilan form *Identity*, tanggal lahir, dan jenis kelamin



Gambar 3.19 Rancangan Tampilan tinggi badan, berat badan, dan jenis aktivitas

Setelah pengguna berhasil melakukan *login*, maka pengguna akan diarahkan ke tampilan halaman utama. Pada bagian atas halaman terdapat informasi nama dan tanggal pada hari itu serta maksimal jumlah kalori yang telah ditetapkan oleh sistem berdasarkan perhitungan dari data yang telah dimasukkan. Pada bagian bawah informasi, terdapat bagian untuk menampilkan daftar makanan yang telah dipilih. Selain itu, pada bagian bawah tampilan terdapat *navigation bar* untuk berpindah

halaman yang terdiri dari halaman utama, halaman *quest*, halaman *leaderboard*, dan halaman *profile*. Gambar 3.20 menunjukkan tampilan dari rancangan antar muka pada halaman utama.



Gambar 3.20 Rancangan tampilan halaman utama

Ketika ingin menambah makanan, pengguna bisa menekan tombol *add food* pada bagian bawah. Setelah itu akan muncul tampilan daftar makanan beserta jumlah kalornya, serta bisa di ubah jumlah satuan sajiannya. Pengguna akan mendapat konfirmasi berupa tulisan “Your Food has Been Added” pada bagian bawah ketika makanan sudah ditambahkan. Makanan yang sudah dipilih bisa dilihat pada halaman utama berupa daftar dengan total kalornya. Gambar 3.21 merupakan rancangan tampilan untuk daftar makanan beserta total kalori yang sudah dipilih.



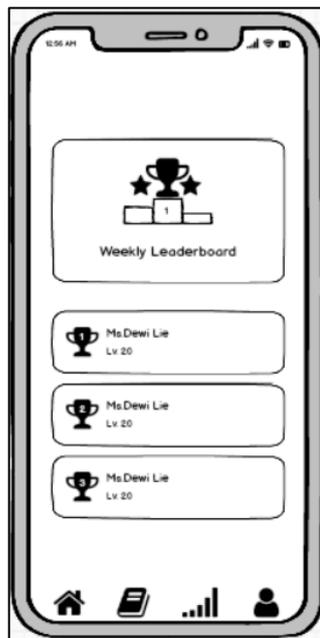
Gambar 3.21 Rancangan tampilan daftar makanan, konfirmasi makanan yang dipilih, dan daftar makanan yang telah dipilih.

Pada halaman *quest* yang terdiri dari 2 panel utama yaitu *Daily quest* dan *Main quest*. *Daily quest* berisi beberapa *quest* harian yang harus diselesaikan oleh pengguna. Selanjutnya ada *Main quest* yang berisi beberapa *quest* utama yang dapat dilakukan oleh pengguna secara berkala satu per satu. Gambar 3.22 Rancangan tampilan daftar *quest*.



Gambar 3.22 Rancangan tampilan daftar *quest*

Tampilan halaman *leaderboard* menampilkan daftar pengguna terbaik setiap minggunya. Halaman *leaderboard* menampilkan daftar nama pengguna berikut dengan *level* yang sudah dicapai pada Gambar 3.23.



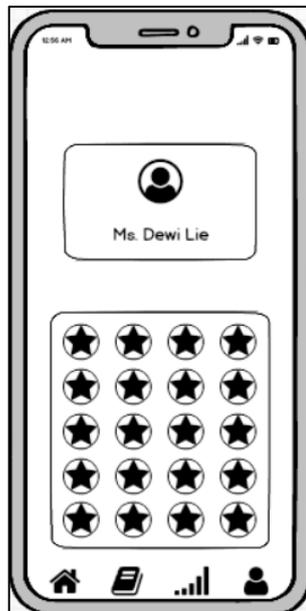
Gambar 3.23 Rancangan tampilan *leaderboard*

Tampilan pada halaman *profile* pada halaman bagian atas terdapat informasi jenis kelamin yang diwakili dengan gambar dan nama dari pengguna. Di halaman bawah terdapat menu lainnya seperti koleksi lencana, pengaturan, pertanyaan, dan *credit*. Gambar 3.24. merupakan Rancangan tampilan *profile*



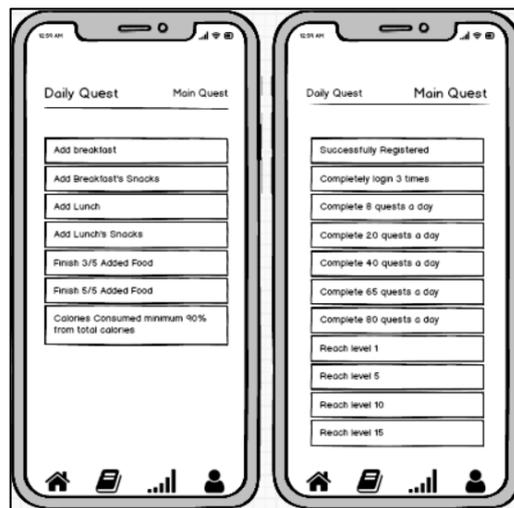
Gambar 3.24 Rancangan tampilan *profile*

Halaman *Achievement* berisi kumpulan lencana yang telah didapat maupun belum didapat. Uang menjadi pembeda ketika lencana sudah didapat adalah lencana tersebut berwarna terang. Sedangkan lencana yang belum di dapat berwarna lebih gelap. Gambar 3.25 merupakan rancangan tampilan dari menu *Achievement*.



Gambar 3.25 Rancangan Tampilan *Achievement*

Halaman *quest* berisi daftar *quest* yang terdiri dari 2 panel utama yaitu *daily quest* dan *main quest*. *Daily quest* berisi *quest* harian yang seharusnya diselesaikan oleh pengguna untuk mendapatkan banyak poin. Sedangkan *main quest* berisi *quest* utama yang bisa diselesaikan pengguna untuk mendapatkan lencana. Gambar 3.26 merupakan rancangan tampilan *quest*.



Gambar 3.26 Rancangan tampilan *quest*

### 3.5.6 Daftar Aset

Rancangan aplikasi gamifikasi untuk mengatur pola makan menggunakan aset-aset yang ada. Berikut adalah daftar penggunaan aset beserta dengan referensinya.

#### A. Aset Fitur Badges

Terdapat 13 *badge* yang bisa didapatkan oleh user pada aplikasi gamifikasi untuk mengatur pola makan. Aset untuk *badges* dibuat oleh Nicholas Chandra dan dapat dilihat pada Tabel 3.1

Tabel 3.1 Aset Fitur *Badge*

No	Aset	Deskripsi <i>Badges</i>
1		Pengguna berhasil daftar
2		Pengguna berhasil <i>login</i> selama 3 hari berturut-turut
3		Pengguna berhasil menyelesaikan semua <i>daily quest</i> dalam 1 hari
4		Pengguna berhasil menyelesaikan 20 <i>daily quest</i>
5		Pengguna berhasil menyelesaikan 40 <i>daily quest</i>
6		Pengguna berhasil menyelesaikan 50 <i>daily quest</i>
7		Pengguna berhasil menyelesaikan 80 <i>daily quest</i>
8		Pengguna mencapai <i>level 2</i>
9		Pengguna mencapai <i>level 3</i>
10		Pengguna mencapai <i>level 5</i>
11		Pengguna mencapai <i>level 7</i>
12		Pengguna menempati peringkat 1 <i>leaderboard</i>

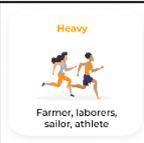
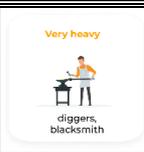
13		Pengguna menempati peringkat 2 <i>leaderboard</i>
14		Pengguna menempati peringkat 3 <i>leaderboard</i>

## B. Aset-Aset Tambahan

Terdapat beberapa aset tambahan berupa lambang *daily quest* yang sudah diselesaikan dan belum diselesaikan, gambar pemilihan *gender*, dan gambar *activity level* pada *register*. Daftar aset tambahan dapat dilihat pada Tabel 3.2

Tabel 3.2 Aset Tambahan

No.	Aset	Deskripsi	Sumber
1		Lambang <i>daily quest</i> yang belum diselesaikan	<a href="https://www.freepik.com/">https://www.freepik.com/</a>
2		Lambang <i>daily quest</i> yang sudah diselesaikan	<a href="https://www.freepik.com/">https://www.freepik.com/</a>
3		Gambar <i>gender</i> perempuan	<a href="https://www.freepik.com/">https://www.freepik.com/</a>
4		Gambar <i>gender</i> laki-laki	<a href="https://www.freepik.com/">https://www.freepik.com/</a>
5		Gambar <i>level</i> aktivitas kategori istirahat	<a href="https://www.freepik.com/">https://www.freepik.com/</a>

6		Gambar <i>level</i> aktivitas kategori ringan	<a href="https://www.freepik.com/">https://www.freepik.com/</a>
7		Gambar <i>level</i> aktivitas kategori sedang	<a href="https://www.freepik.com/">https://www.freepik.com/</a>
8		Gambar <i>level</i> aktivitas kategori berat	<a href="https://www.freepik.com/">https://www.freepik.com/</a>
9		Gambar <i>level</i> aktivitas kategori sangat berat	<a href="https://www.freepik.com/">https://www.freepik.com/</a>