Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

BAB III

METODOLOGI PENELITIAN

3.1 Metode Penelitian

Pada bab sebelumnya sudah dijelaskan tentang teori-teori dan algoritma yang digunakan sebagai dasar untuk membuat aplikasi ini. Selanjutnya aplikasi ini dibuat berdasarkan tahapan-tahapan sebagai berikut.

1. Studi Literatur

   Pengumpulan data-data dan mempelajari konsep-konsep yang dibutuhkan dalam pelaksanaan penelitian. Proses pengumpulan data dilakukan dengan survei serta mempelajari jurnal-jurnal yang berhubungan dengan penelitian yang dilakukan.

2. Desain Fungsi

   a. Merancang *dependency diagram* yang merupakan *flow*/aliran proses dari program yang dibuat.

   b. Merancang fungsi-fungsi didalam aplikasi yang dituliskan dalam sebuah pseudocode.

   c. Merancang GUI (*Graphic User Interface*). Tampilan antarmuka yang merupakan komponen penting dalam sebuah aplikasi. Dimana melalui tampilan antarmuka ini *user* dapat berinteraksi dengan aplikasi yang kita buat sehingga tampilan antarmuka dari aplikasi harus dibuat *user-friendly* dan tidak membuat *user* yang mengalami kesulitan pada saat menggunakan aplikasi.

Implementasi Metode Backward..., Rizka Hayuning Santoso Putri, FTI UMN, 2012

3. Coding

   Proses menulis *script code* program sesuai dengan yang sudah dibuat dalam pseudocode. Pada aplikasi ini *code* dibuat dengan menggunakan bahasa Java.

4. Pengujian

   Pada tahap pengujian ini dilakukan dengan:

   a. Meninjau ulang aplikasi yang dibuat, apakah sudah sesuai denga *user requirement* yang semestinya serta apakah aplikasi dapat berjalan dengan baik.

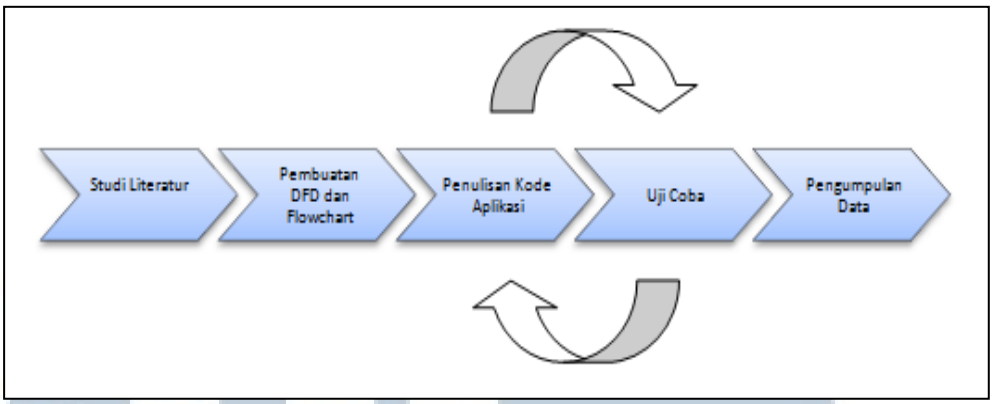
   b. Mencari kekurangan program dan *bug*. Jika ditemukan *bug*, maka apliakasi akan diperbaiki dalam proses maintenance hingga *bug* tidak ditemukan lagi.

5. Analisa Data

   Setelah aplikasi dapat berjalan dengan baik dan tidak ditemukan *bug*, maka dilakukan pengumpulan dan pencatatan data-data hasil aplikasi. Dimana data-data yang dihasilkan tersebut digunakan sebagai bahan untuk penarikan kesimpulan.

Pembuatan serta pengembangan aplikasi ini menggunakan metode *prototyping.* Dimana proses pembuatan aplikasi diawali dengan pengumpulan data-data yang dibutuhkan. Setelah seluruh data yang dibutuhkan sudah lengkap maka dibuatlah diagram-diagram perancangan aplikasi.

Setelah itu mulai dilakukan proses pembangunan aplikasi yang sesuai dengan rancangan yang sudah dibuat. Kemudian dilakukan uji coba terhadap aplikasi untuk mengetahui kekurangan dari aplikasi.



Gambar 3.1 Proses pembuatan aplikasi

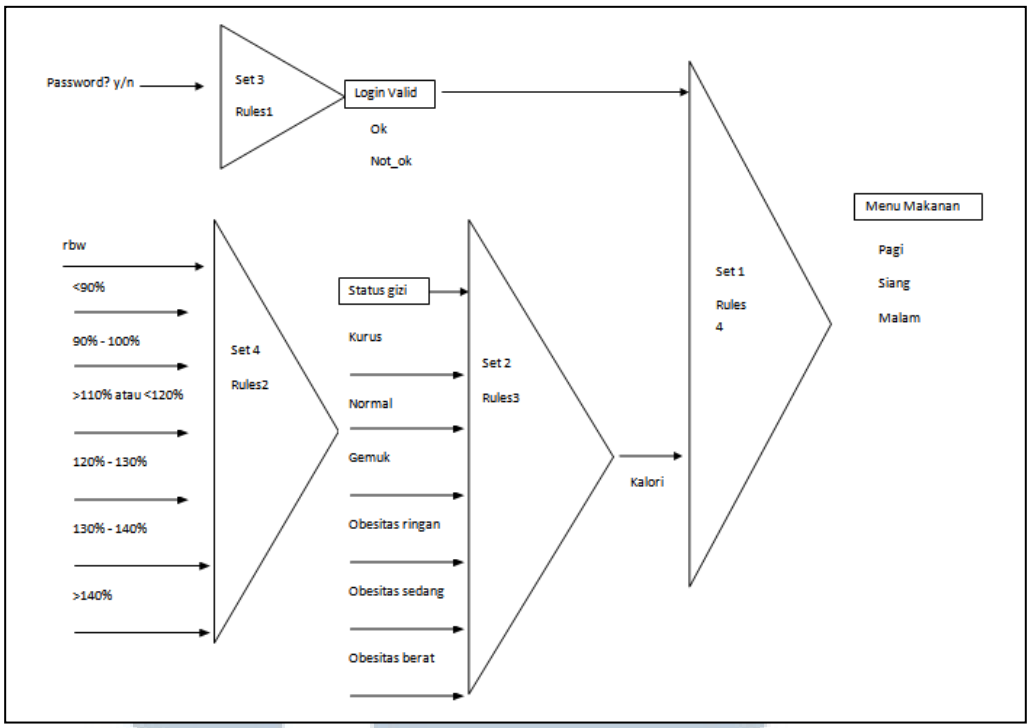
3.2 Diagram Sistem

Pada bagian diagram sistem ini terdapat beberapa gambar diagram yang akan menjelaskan proses-proses yang dilakukan oleh aplikasi. Adapun diagram-diagramnya sebagai berikut.

3.2.1 Dependency Diagram

Dependency Diagram dibuat untuk menentukan faktor-faktor yang mempengaruhi dalam penentuan menu makanan. *Dependency diagram* untuk aplikasi sistem pemilihan *food combining* dapat dilihat pada gambar 3.2

UNIVERSITAS MULTIMEDIA NUSANTARA

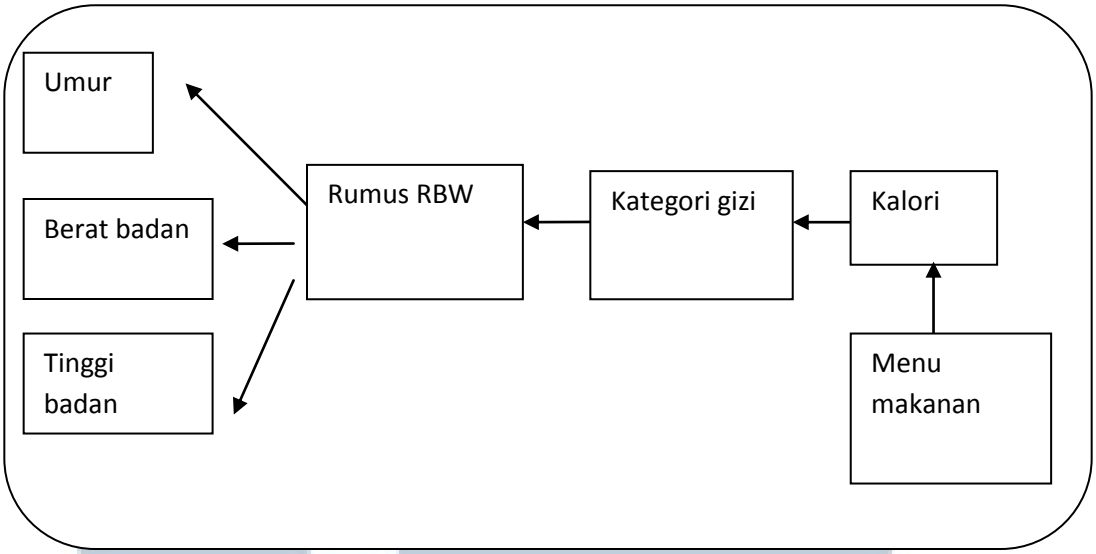


Gambar 3.2 Dependency diagram aplikasi *food combining*

3.2.2 Proses Backward Chaining

Proses metode *backward chaining* dibawah ini menjelaskan tentang hubungan antar proses bagaimana menu makanan didapatkan. Dimana program menerima input dari *user* berupa umur, berat badan, dan tinggi yang kemudian data tersebut diolah untuk dengan menggunakan rule rumus 1 untuk mendapatkan kategori gizi. Kategori gizi yang sudah didapatkan digunakan untuk menghitung kalori yang dibutuhkan oleh *user*. Setelah kalori didapatkan maka menu makanan dicari dan ditentukan dimana dari kombinasi tersebut jumlah kalori totalnya sama dengan kalori yang seharusnya dibutuhkan oleh *user*.

UNIVERSITAS MULTIMEDIA NUSANTARA

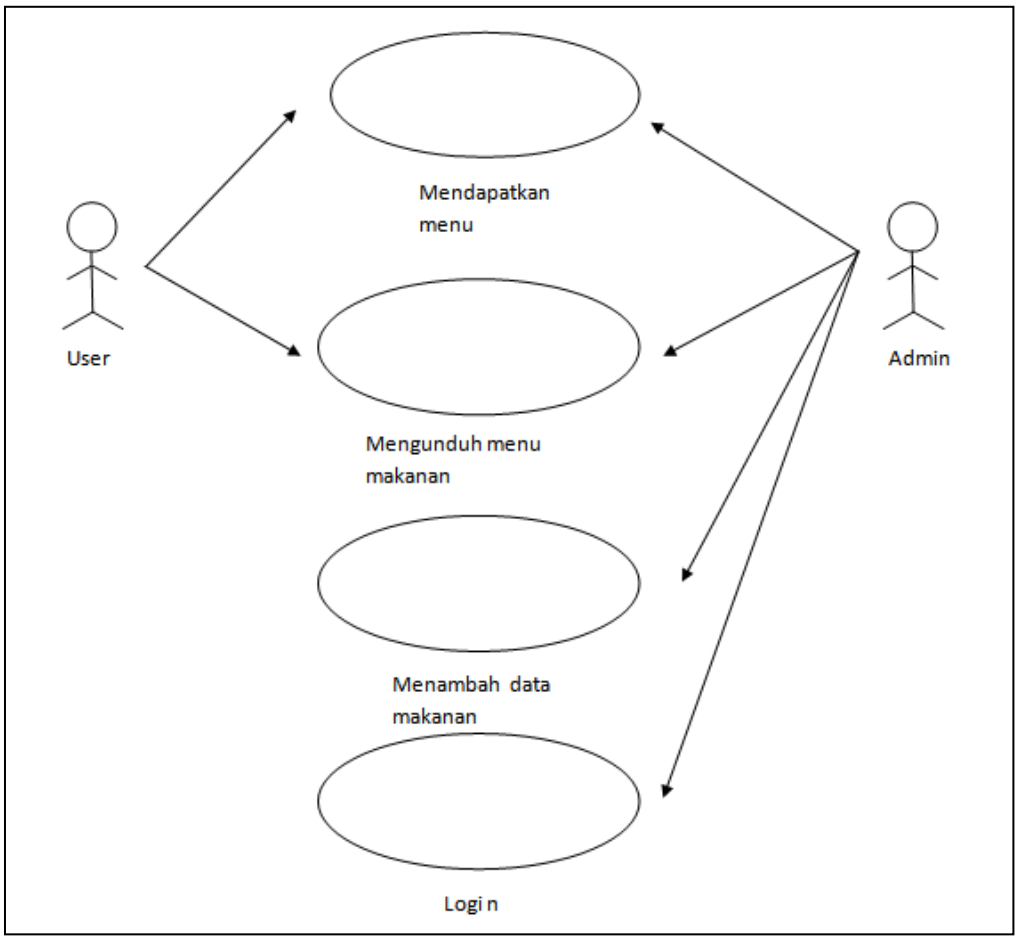


Gambar 3.3 Proses Metode *Backward Chaining*

3.2.3 Use Case Diagram

Use case diagram merupakan interaksi antara sistem dengan *actor* yang ada pada aplikasi. Berikut ini adalah gambaran dari *use case* diagram aplikasi pemilihan *food combining*.

UMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA



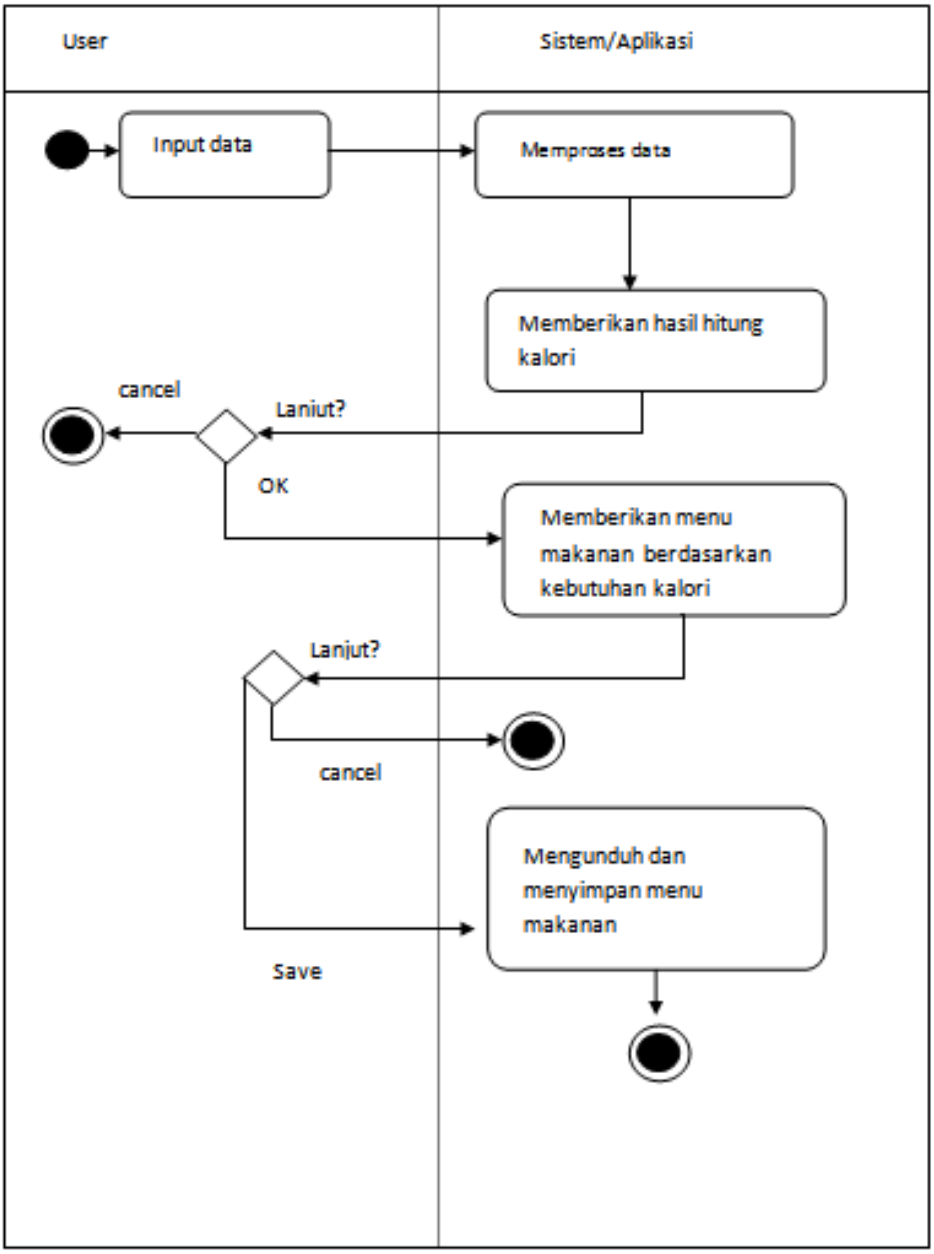
Gambar 3.4 *Use Case Diagram* Aplikasi Pemilihan Food Combining

Use case diagram aplikasi *food combining* pada gambar 3.4 di atas terdiri dari dua aktor yaitu *user* dan *admin*. *User* berinteraksi dengan dua sistem: mendapatkan menu, dan mengunduh menu makanan. Sedangkan *admin* berinteraksi dengan empat sistem, yaitu *login*, mendapatkan menu, mengunduh menu makanan, dan menambah data makanan.

UNIVERSITAS MULTIMEDIA NUSANTARA

3.2.4 Activity Diagram

Activity diagram menggambarkan proses bisnis dan urutan proses dalam aplikasi *food combining*.



Gambar 3.5 *Activity Diagram* Aplikasi *Food Combining*

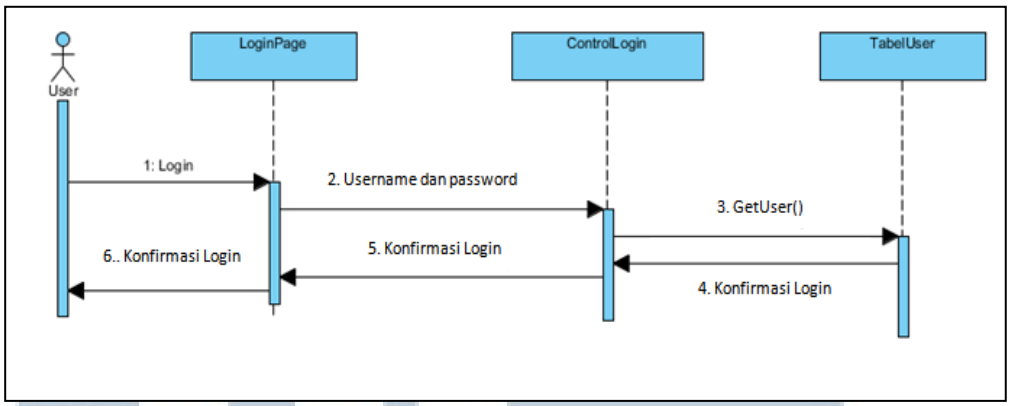
NUSANTARA

Proses yang terdapat pada aplikasi *food combining* diawali oleh *user*, dengan menginputkan data-data yang diminta lalu data-data tersebut diproses oleh sistem untuk mendapatkan hasil kalori yang dibutuhkan oleh *user*. Setelah itu *user* membuat pilihan, jika tidak maka aplikasi selesai, jika ya maka sistem akan men-*generate* menu makanan sesuai dengan kebutuhan kalori dari *user*. Setelah menu muncul, *user* akan menentukan untuk menyimpan menu tersebut atau tidak. Jika *user* memutuskan untuk menyimpan menu makanan tersebut maka sistem akan mengunduh menu serta menyimpannya dan aplikasi selesai.

3.2.5 Sequence Diagram

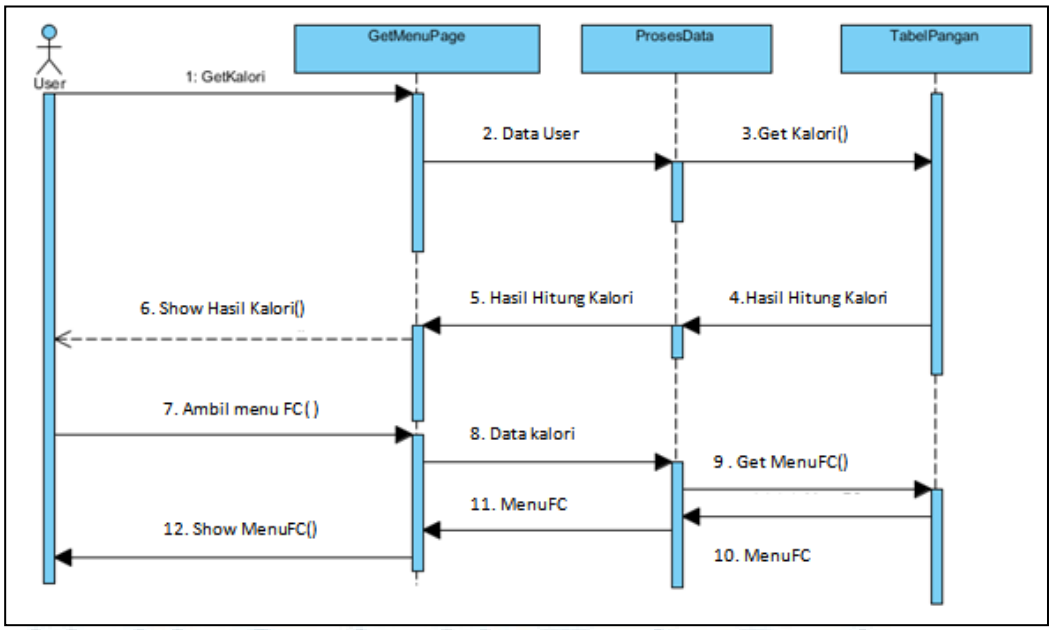
Sequence diagram adalah suatu diagram yang menggambarkan interaksi antar obyek dan mengindikasikan komunikasi diantara obyek-obyek tersebut. Diagram ini juga menunjukkan serangkaian pesan yang dipertukarkan oleh obyek-obyek yang melakukan suatu tugas atau aksi tertentu. Di dalam aplikasi *food combining* ini terdapat empat *Sequence* diagram, dimana keempat diagram tersebut ditunjukkan oleh gambar 3.6.

UNIVERSITAS MULTIMEDIA NUSANTARA



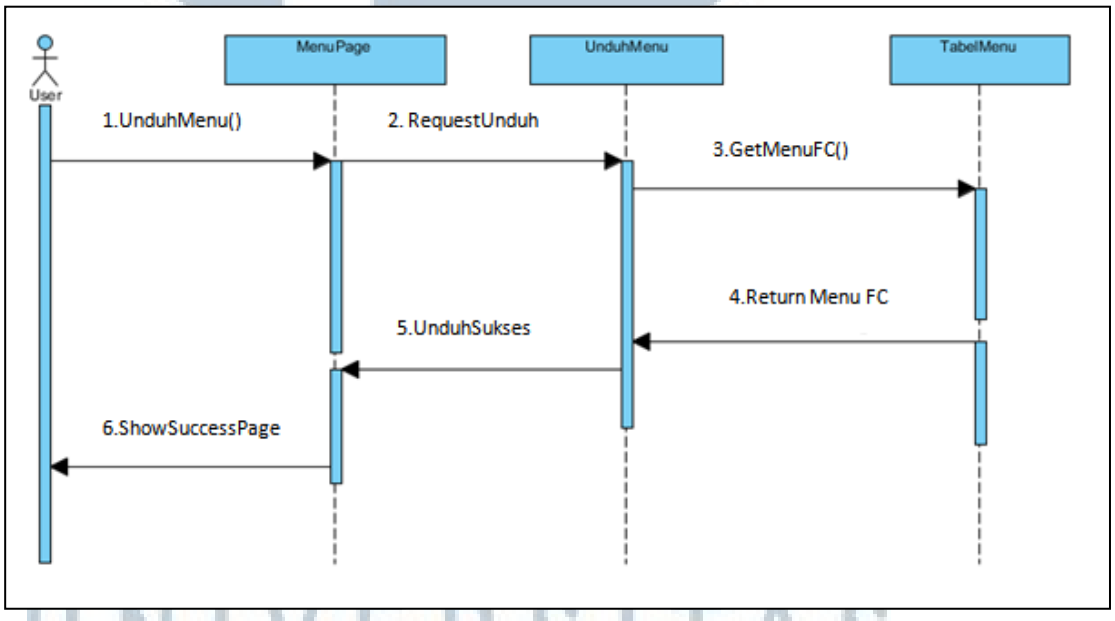
Gambar 3.6 *Sequence Diagram Login*

Sequence diagram *login* pada gambar 3.6 menunjukkan proses *login user* untuk masuk ke dalam aplikasi/sistem. *User* akan men-*trigger method login*, dengan memasukkan *password.* Setelah itu kontrol akan melakukan *query* ke tabel User . Jika *password* sesuai kontrol akan memberikan konfirmasi *login* sukses kepada *user* melalui LoginPage.



Gambar 3.7 *Sequence Diagram* Mendapatkan Menu

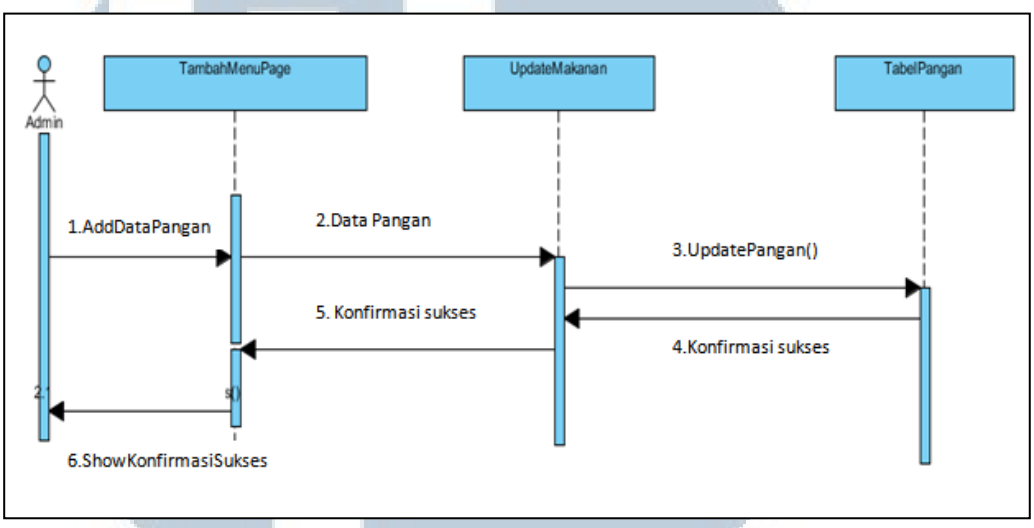
Sequence diagram mendapatkan menu pada gambar 3.7 di atas menggambarkan proses *user* mendapatkan kombinasi menu makanan pada aplikasi pemilihan *food combining*. Saat sistem menampilkan getmenupage, maka *method* GetKalori akan dijalankan dan akan mengirimkan data pada kontrol, yang kemudian berdarakan perhitungan *method* tersebut didapatkan hasil kalori. Hasil kalori ini akan dipakai saat *user* menggunakan *method* AmbilMenuFC. Dimana data kalori diproses oleh kontrol dan kontrol akan melakukan *query* untuk mendapatkan kombinasi menu makanan yang kemudian hasil dari *query* tersebut ditampilkan kepada *user*.



Gambar 3.8 *Sequence Diagram* Mengunduh Menu

Sequence diagram mengunduh menu pada gambar 3.8 di atas menjelaskan tentang proses pengunduhan menu *food combining*. Proses ini

diawali dengan pemanggilan *method* UnduhMenu, dari *method* tersebut maka kontrol akan melakukan *query* GetMenuFC ke TabelMenu. Proses ini akan mengembalikan konfirmasi pengunduhan kepada *user* melalui MenuFCPage.



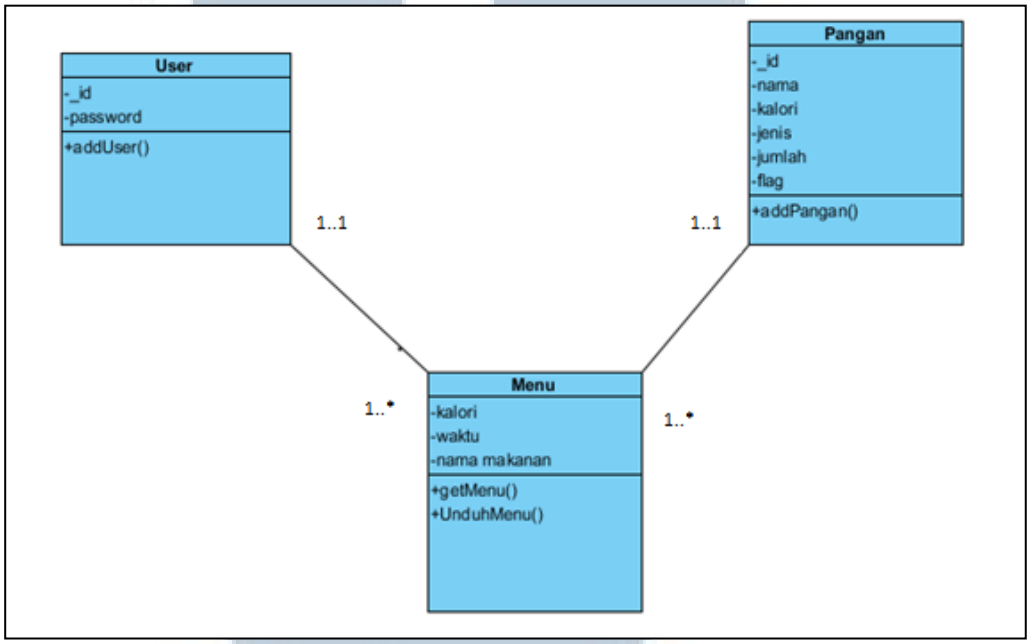
Gambar 3.9 *Sequence Diagram* Tambah Pangan

Sequence diagram tambah pangan mendeskripsikan tentang proses penambahan pangan. Dimana proses penambahan itu diawali dengan dijalankannya method AddDataPangan(), kemudian berdasarkan data-data yang didapat kontrol melakukan *query* UpdatePangan ke tabel Pangan. Proses akan diakhiri dengan dimunculkannya konfirmasi penambahan pangan melalui TambahMenuPage.

3.2.6 Class Diagram

Class diagram adalah salah satu diagram UML yang merupakan deskripsi *group* dari obyek-obyek dengan *properties*, *behaviour*, *relationship* dengan

obyek lain dan juga semantik. Class diagram dari sistem pemilihan *food combining* ditunjukkan oleh gambar 3.10 berikut ini.



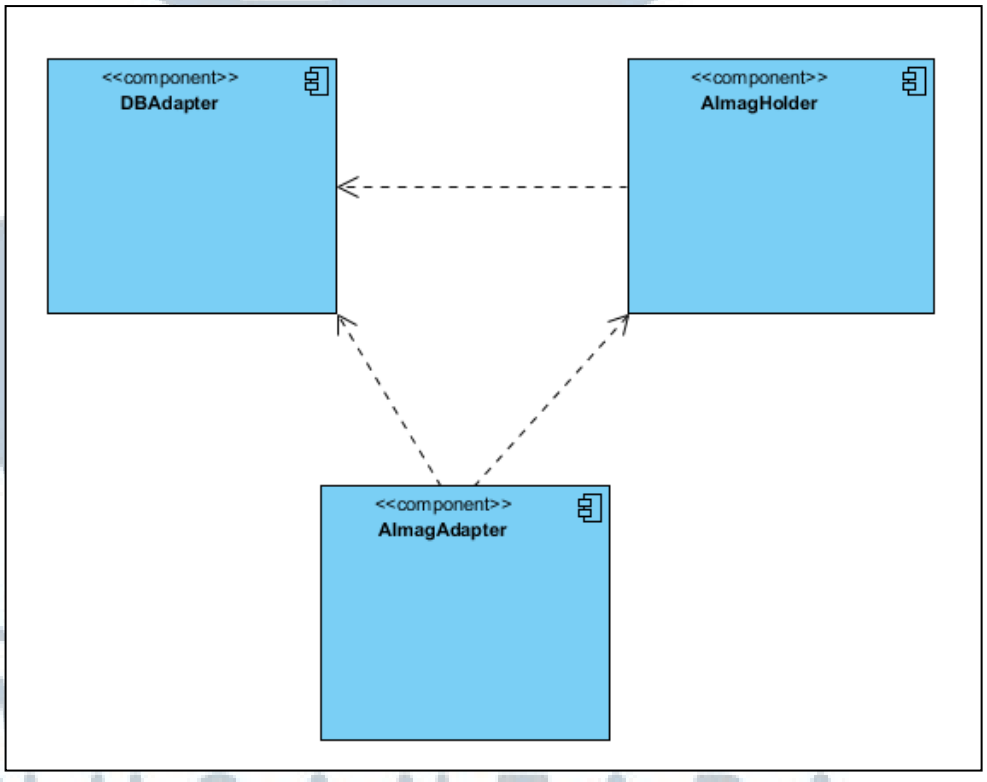
Gambar 3.10 *Class Diagram* aplikasi pemilihan food combining

Gambar *class* diagram pada gambar 3.10 di atas menunjukkan terdapat tiga *class* dalam sistem pemilihan *food combining*. *Class* User dengan *attribute* _id dan password, dimana *class* ini memiliki *method* addUser(). Class Menu dengan *attribute* kalori,waktu,nama makanan, *class* Menu ini memiliki dua *method* yaitu getMenu() dan UnduhMenu(). *Class* Pangan memiliki beberapa *attribute* yaitu _id, nama, kalori, jenis, jumlah, flag. *Class* pangan memiliki *method* addPangan(). Hubungan antar class tersebut adalah sebagai berikut. *Class User* memiliki relasi *one to many* terhadap *class* Menu, yang memiliki arti satu *user* dapat memiliki lebih dari satu menu. *Class* Pangan memiliki relasi *one to*

many terhadap *class* Menu, dimana satu pangan dapat masuk ke dalam banyak menu.

3.2.7 Component Diagram

Component diagram terdiri dari satu atau beberapa *diagram component*, sebuah diagram *component* menggambarkan bagaimana relasi antara *component* yang satu dengan yang lainnya. *Component* menggambarkan sebagian potongan secara fisik dari implementasi suatu sistem, mencakup *software code* (*source*, *binary*, atau *executable*). *Component diagram* aplikasi pemilihan *food combining* dapat dilihat pada gambar 3.11



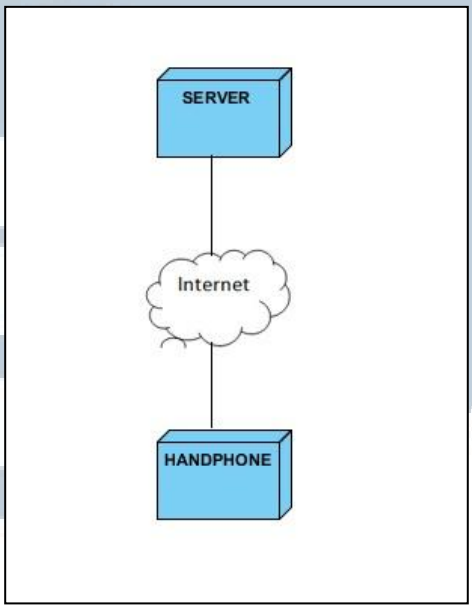
Gambar 3.11 *Component diagram* aplikasi pemilihan *food combining*

Gambar *component diagram* menjelaskan hubungan atau relasi antar *component* dalam aplikasi pemilihan *food combining*. Dari gambar di atas terdapat tiga *component* pada aplikasi pemilihan *food combining*, yaitu DBAdapter, AlmagHolder, dan AlmagAdapter. DBAdapter adalah *component* yang berisi tentang semua yang berkaitan dengan *database*, mulai dari koneksi ke *database* dan *query-query* yang dilakukan untuk mendapatkan data dari *database*. AlmagAdapter adalah *component* yang bertugas menjadi *layout inflater* dari *cursor* ke dalam *list view*. Sedangkan AlmagHolder adalah *component* yang berupa *adapter* untuk menampilkan data yang sudah diambil dari *database* ke dalam sebuah *list view*. Dari gambar di atas dapat dijelaskan bahwa *component* AlmagHolder dan *component* AlmagAdapter bergantung pada *component* DBAdapter. Dimana AlmagAdapter dan AlmagHolder tidak dapat dikompilasi sebelum DBAdapter selesai dikompilasi. Selain itu component AlmagAdapter juga bergantung pada AlmagHolder, dapat diartikan bahwa component AlmagAdapter ini bergantung pada dua *component*, dimana AlmagAdapter tidak dapat dikompilasi sebelum DBAdapter dan AlmagHolder selesai dikompilasi.

3.2.8 Deployment Diagram

Deployment diagram merupakan diagram yang menggambarkan segala macam hal yang berkaitan dengan penyebaran fisik aplikasi. Dalam aplikasi

pemilihan *food combining* terdapat beberapa *hardware* yang digunakan unutk proses implementasi dari aplikasi ini seperti *server*, jaringan internet, dan *client*. *Deployment* diagram sistem pemilihan menu *food combining* ditunjukkan oleh gambar 3.12

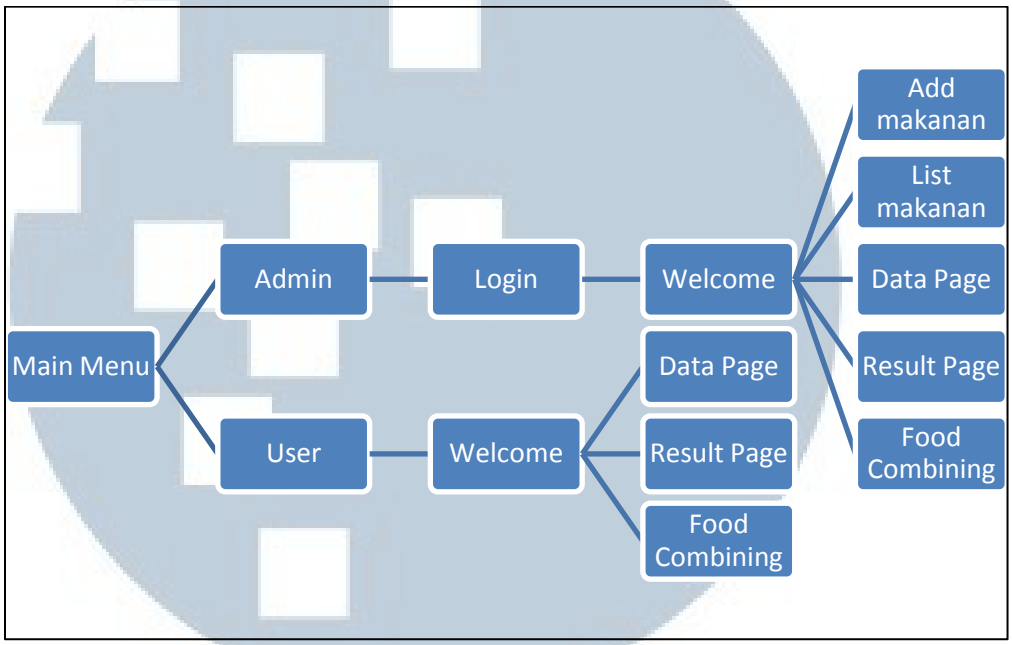


Gambar 3. 12 *Deployment diagram* aplikasi *food combining*

3.2.9 Rancangan Navigasi

Berikut adalah rancangan navigasi dari aplikasi yang akan dibangun untuk penelitian. Dimana *admin* memerlukan *login* sebelum masuk ke aplikasi, lalu ke *welcome page* kemudian masuk ke menu utama yaitu : *add* makanan, *list* makanan, data *page, result page, food combining*. Sedankan user biasa akan

langsung menuju ke menu utama : data *page*, *result page*, *food combining*, setelah melewati *welcome page*.



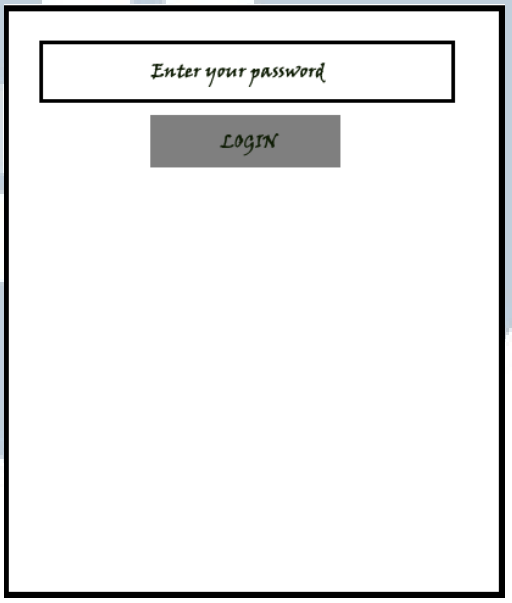
Gambar 3.13 Rancangan navigasi aplikasi *food combining*

3.3 Design Graphical User Interface Aplikasi

Dalam sebuah pembangunan aplikasi terdapat salah satu tahap yang dinamakan desain *user interface*. Pada tahap ini dilakukan desain *user interface* aplikasi yang nantinya akan menjadi penghubung antara *user* dengan sistem. Berikut adalah desain *graphical user interface* dari aplikasi *food combining*.

UNIVERSITAS MULTIMEDIA NUSANTARA

3.3.1 Sketsa halaman *login*

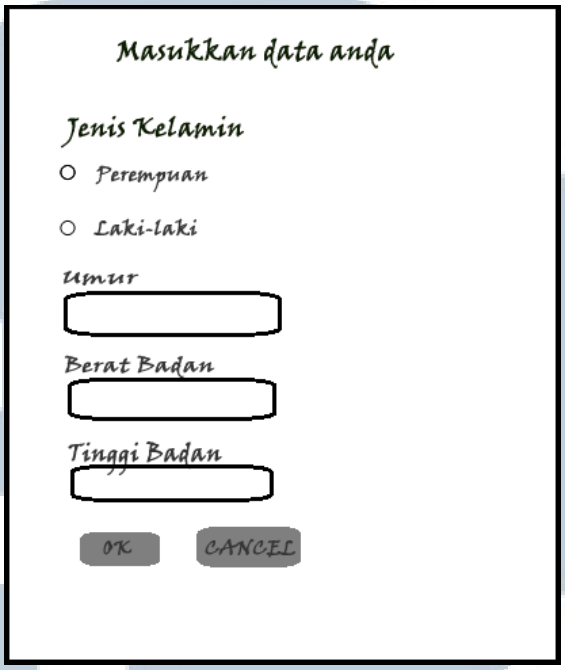


Gambar 3.14 Sketsa halaman *login*

Halaman ini adalah halaman yang pertama kali muncul saat *user* membuka aplikasi. Terdapat satu *text box* untuk memasukkan *password user,* serta terdapat satu tombol *login* yang akan menghubungkan *user* ke halaman utama jika *password* yang dimasukkan benar. Namun jika *password* yang dimasukkan salah maka aplikasi akan meminta *password* sampai *password* yang dimasukkan *valid.*

UNIVERSITAS MULTIMEDIA NUSANTARA

3.3.2 Sketsa halaman utama

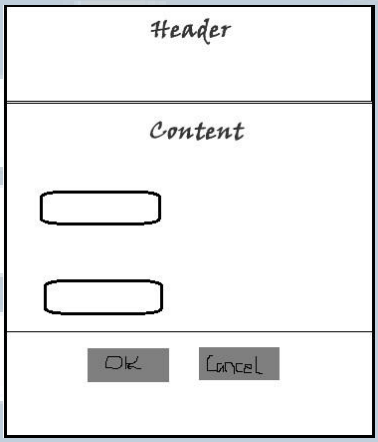


Gambar 3.15 Sketsa halaman utama

Halaman utama merupakan halaman yang akan muncul ketika *admin* sukses memasukkan *password*. Sedangkan bagi *user* biasa halaman ini akan langsung muncul ketika *user* membuka aplikasi. Halaman ini berfungsi untuk mendapatkan data-data dari *user* yang kemudian data-data tersebut akan dihitung untuk mendapatkan kadar kalori serta mendapatkan nilai *relative body weight* dari *user*.

UNIVERSITAS MULTIMEDIA NUSANTARA

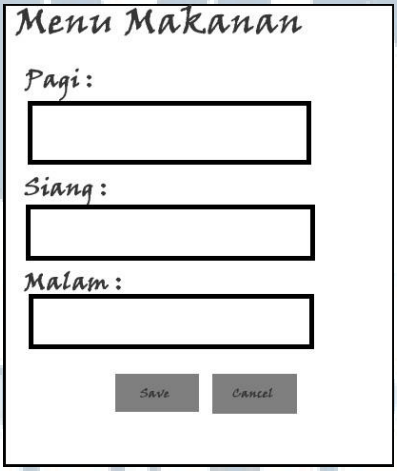
3.3.3 Sketsa halaman hasil hitung



Gambar 3.16 Sketsa halaman hasil hitung

Halaman hasil hitung ini menampilkan hasil perhitungan data-data yang sudah diinputkan oleh *user*. Halaman ini menampilkan *relative body weight*, status gizi, kadar kalori yang dibutuhkan oleh *user*.

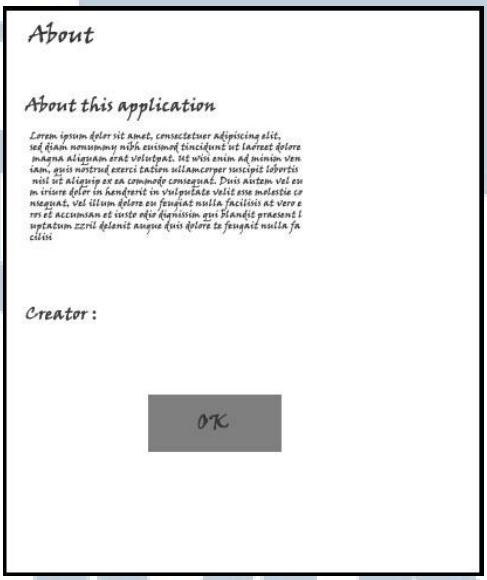
3.3.4 Sketsa halaman menu makanan



Gambar 3.17 Sketsa halaman menu makanan

Pada halaman ini akan ditampilkan daftar menu makanan yang sudah diakumulasikan sesuai dengan kalori yang dibutuhkan oleh *user*. Pada halaman ini terdapat dua *button*, yaitu *button save* dan *cancel*. *Button save* berfungsi untuk mengunduh menu makanan sedangkan *button cancel* akan mengarahkan *user* kembali ke menu utama.

3.3.5 Sketsa halaman *about*



Gambar 3.18 Sketsa halaman *about*

Pada halaman ini akan ditampilkan sedikit informasi tentang aplikasi dan tentang *creator* aplikasi ini.

UNIVERSITAS MULTIMEDIA NUSANTARA