

BAB III

PELAKSANAAN KERJA PRAKTEK

3.1 Dasar Teori

3.1.1 Teknologi Cloud Computing

Cloud computing adalah penggunaan teknologi komputer untuk pengembangan berbasis Internet dengan piranti lunak lengkap dan sistem operasional juga tersedia secara online. Menurut Stevan Greve internet bisa dianggap awan besar. Awan berisi komputer yang semuanya saling tersambung. Dari situlah berasal istilah '*cloud*'. Jadi semuanya disambungkan ke '*cloud*', atau awan itu.

Jadi *cloud computing* merupakan sebuah model untuk memberikan kenyamanan, pada akses permintaan (*request*) beberapa pengguna untuk berbagi sumberdaya komputasi yang dikonfigurasi (misalnya, jaringan, server, penyimpanan, aplikasi, dan jasa) yang dapat dengan cepat ditetapkan dan dikembalikan (*response*) kepada pengguna.

3.1.2 Web Service

Sebuah layanan yang tersedia melalui web berisi modul perangkat lunak yang dirancang untuk melakukan serangkaian tugas tertentu, memungkinkan komunikasi antar aplikasi melalui web serta menggunakan protokol / format standar untuk komunikasi. Jenis Webservice ada 2 yaitu SOAP dan REST.

WebService umumnya melakukan pertukaran data menggunakan format XML (*Extensible Markup Language*) dan berkomunikasi menggunakan protokol SOAP (*Simple Object Access Protocol*). Namun saat ini sudah ada protokol RESTful (*Representational State Transfer full*) API dengan format pertukaran data JSON (*JavaScript Object Notation*) yang diklaim lebih efisien dibanding protokol SOAP (SmartBear, n.d).

REST memiliki konsep sederhana karena mengikuti dasar bahasa HTTP 1.1 yang

seluruh web mengerti beberapa *method request* atau layanan yang disediakan:

- **POST** - mengirim data, digunakan untuk mengirim data ke *web service*
- **GET** - membaca data, digunakan untuk mendapatkan data dari *web service*
- **PUT** - memperbaharui data, digunakan untuk menyimpan pembaruan data ke *web service*
- **DELETE** - menghapus data, digunakan untuk menghapus data yang ditentukan

3.1.3 QR Code

QRCode adalah suatu jenis kode matriks atau kode batang dua dimensi yang dikembangkan oleh Denso Wave, sebuah divisi Denso Corporation yang merupakan sebuah perusahaan Jepang dan dipublikasikan pada tahun 1994 dengan fungsionalitas utama yaitu dapat dengan mudah dibaca oleh pemindai QR merupakan singkatan dari *quick response* atau respons cepat, yang sesuai dengan tujuannya adalah untuk menyampaikan informasi dengan cepat dan mendapatkan respons yang cepat pula.

Berbeda dengan kode batang, yang hanya menyimpan informasi secara horizontal, QRCode mampu menyimpan informasi secara horizontal dan vertikal, oleh karena itu secara otomatis QRCode dapat menampung informasi yang lebih banyak daripada kode batang. (soon,2008).

3.1.4 Android

Android adalah sistem operasi bergerak (*mobile operating system*) yang mengadopsi sistem operasi Linux, namun telah dimodifikasi. Android diambil alih oleh Google pada tahun 2005 dari Android, Inc sebagai bagian strategi untuk mengisi pasar sistem operasi bergerak. Google mengambil alih seluruh hasil kerja Android termasuk tim yang mengembangkan Android.

Android bersifat *open source* oleh sebab itu Google secara membagikan secara

gratis *development kit* untuk membuat aplikasi termasuk dengan dokumentasinya sehingga mempermudah dalam membuat aplikasi Android. Google juga menyediakan pasar untuk mendistribusikan aplikasi.

3.2 Analisis dan Perancangan

3.2.1 Tahap analisa kebutuhan sistem

Analisa Kebutuhan dari aplikasi yang akan dikembangkan diuraikan kedalam beberapa sub-bab berikut.

1) Pengguna

Pengguna sistem yang menggunakan aplikasi ini adalah sebagai berikut :

A. Owner / VIP

Merupakan pengguna memiliki hak untuk melihat ringkasan data stok yaitu pembelian dan penggunaan, kunjungan serta monitor inventaris dapur.

B. Admin

Merupakan pengguna memiliki hak untuk melihat ringkasan data stok yaitu pembelian dan penggunaan, kunjungan serta monitor inventaris dapur, dengan tambahan dapat membuka manajemen akun yang berisi hak untuk mengatur kata sandi pengguna lain dan menambahkan pengguna baru.

C. Kepala Dapur

Merupakan pengguna memiliki hak untuk melihat ringkasan data stok yaitu pembelian dan penggunaan, kunjungan serta monitor inventaris dapur, dengan tambahan dapat membuka manajemen akun yang berisi hak untuk mengatur kata sandi pengguna lain dan menambahkan pengguna baru.

D. Juru Masak

Merupakan pengguna memiliki hak untuk melihat ringkasan data stok yaitu pembelian dan penggunaan, kunjungan serta monitor inventaris dapur.

2) Teknologi

Beberapa teknologi yang digunakan dalam pengembangan aplikasi ini berdasarkan kebutuhan dan kegunaannya adalah sebagai berikut :

A. IntelliJ IDEA Community 2018

IDE untuk pengembangan Android dengan developer JetBrains yang mana juga merupakan developer Android Studio. IntelliJ IDEA dipilih karena ukuran lebih kecil dari Android Studio dan sedikit lebih ringan untuk dijalankan, untuk kegunaan dan fitur sama hanya berbeda pada struktur file pada proyek Android saja.

B. Visual Studio Code & Composer

Visual Studio Code merupakan pengembangan *Editor Text* yang bisa ditambahkan banyak kegunaan menggunakan fitur *plugin*. Salah satu *plugin* yang dapat digunakan adalah *composer* yang berguna untuk mengunduh *dependencies* / *library* untuk kebutuhan pengembangan perangkat lunak khususnya yang berbasis web.

Fitur unggulan lain dari Visual Studio Code adalah adanya *Terminal* bawaan, sehingga tidak perlu membuka CMD pada windows perintah dapat langsung dieksekusi melalui *Terminal* Visual Studio Code. Pada pengembangan ini Visual Studio Code berfungsi untuk membuat API yang dikembangkan dari Slim Framework. *Dependencies* Slim Framework dapat diunduh menggunakan perintah *composer* melalui *Terminal* baik CMD maupun Visual Studio Code.

C. Xampp

Xampp mempunyai fungsi sebagai server yang berdiri sendiri (*localhost*), yang terdiri dari program MySQL database, Apache HTTP Server, dan lain lain. Pada pengembangan ini berfungsi untuk pengembangan API secara lokal.

D. Google Chrome & Advanced REST client

Pada pengembangan *Web Service* tentu API perlu diuji untuk memastikan bahwa API memberikan *response* yang benar dari *request* yang dikirimkan. *Extensions Advanced REST client* dipilih karena ukurannya kecil ringan dan dapat untuk menguji API yang telah dihosting maupun uji API secara *localhost*. Penggunaan *Advanced REST client* dapat menghemat *memory* dibandingkan menggunakan API *tester* lain yang ukurannya cukup besar dengan fungsi yang serupa.

3) Fungsi

Fungsi dari sistem ini dibagi berdasarkan penggunaanya sebagai berikut :

A. Owner / VIP & Juru Masak

a. Fitur Login

Fitur *login* berguna untuk menjaga keamanan data serta memberikan informasi pengguna aplikasi tersebut.

b. Fitur Beranda

Berisi ringkasan pembelian dan penggunaan stok selama satu bulan berjalan.

c. Fitur Stok Bahan

Berisi informasi tentang stok bahan dapur pada hari berjalan.

d. Fitur Pemakaian Bahan

Berisi informasi jumlah transaksi pemakaian bahan dalam bentuk grafik *pie* selama satu bulan dan melihat jumlah pemakaian bahan berdasarkan satuan bahan selama satu bulan dalam bentuk grafik garis.

e. Fitur Pembelian Bahan

Berisi informasi jumlah transaksi, dan total harga pembelian bahan dalam bentuk grafik *pie* selama satu bulan dan melihat jumlah pembelian bahan

berdasarkan satuan dan harga bahan selama satu bulan dalam bentuk grafik garis.

f. Fitur Inventaris Dapur

Berisi informasi barang inventaris Divisi Dapur YSI dan dapat melakukan monitor dengan melakukan *Scan QRCode* pada *QRCode* yang ditempel pada inventaris dapur. Hasil scan akan menampilkan informasi dari barang tersebut.

g. Fitur Pengaturan Akun

Berisi fitur untuk mengubah kata sandi akun dan untuk log out akun dari aplikasi.

B. Admin & Kepala Dapur

Untuk admin diberikan fitur – fitur yang sama dengan Admin dan Juru masak namun dengan tambahan sebagai berikut :

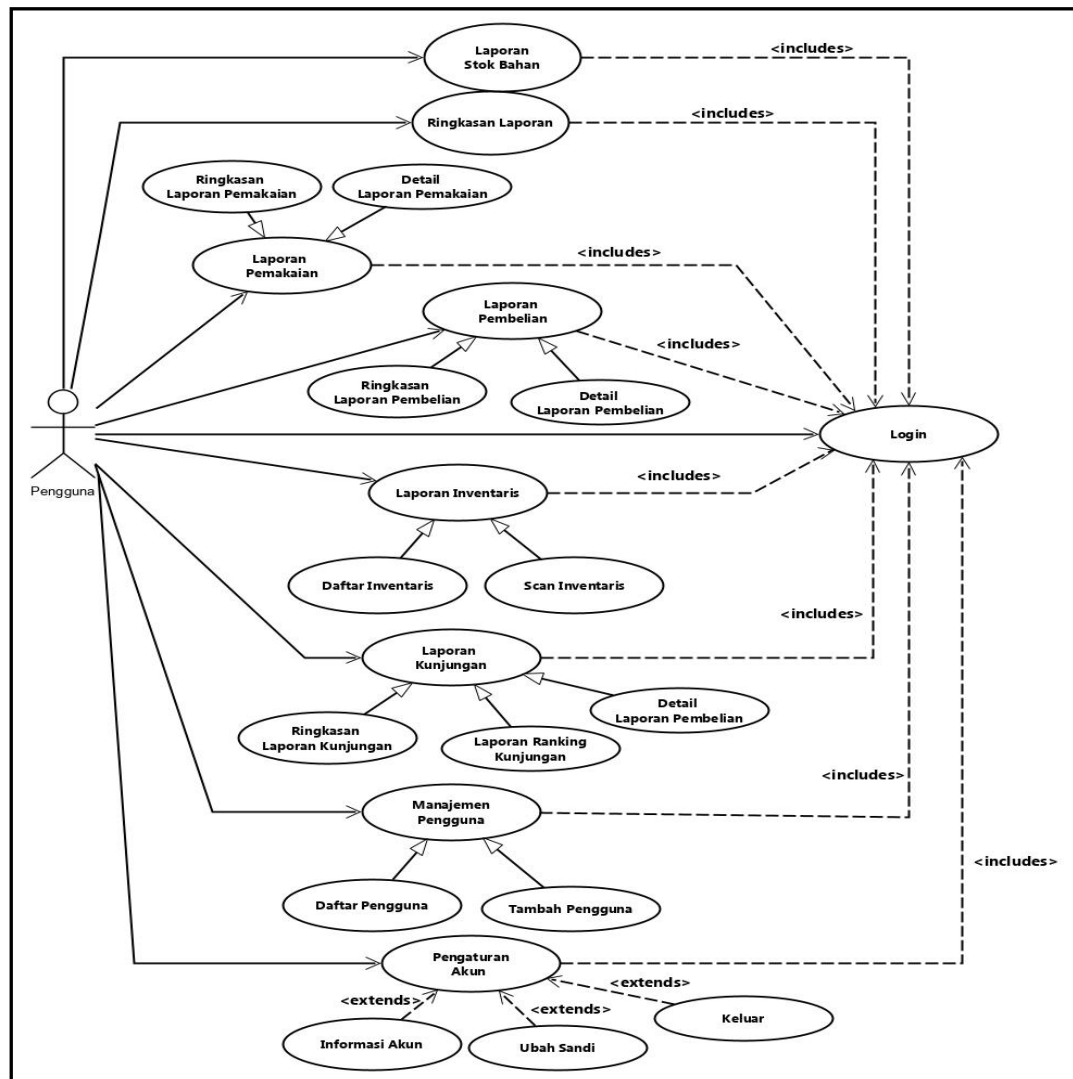
a. Manajemen Akun

Berisi informasi akun yang terdaftar dalam sistem Divisi Dapur YSI dan dapat melakukan aksi atur ulang kata sandi serta menambahkan pengguna baru.

3.2.2 Tahap perancangan UML

Pada tahap perancangan UML ini pengguna disusutkan menjadi 1 buah aktor berdasarkan. Penyusutan ini dapat dilakukan karena sebetulnya semua aktor memiliki fungsi yang sama kecuali fungsi Manajemen Akun yang hanya dimiliki Kepala Dapur dan Admin. Perancangan UML membahas Use Case Diagram, Activity Diagram, Sequence Diagram, Class Diagram.

A. Use Case Diagram



Gambar 3.1 Use Case Diagram

Diagram *Use Case* menggambarkan apa saja aktifitas yang dilakukan oleh suatu sistem dari sudut pandang pengamatan luar. Diagram *Use Case* dekat kaitannya dengan kejadian-kejadian. Kejadian (scenario) merupakan contoh apa yang terjadi ketika seseorang berinteraksi dengan sistem. Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem. *Use case* merupakan sebuah pekerjaan tertentu, misalnya login ke sistem, pengaturan akun, dan sebagainya.

B. Activity Diagram

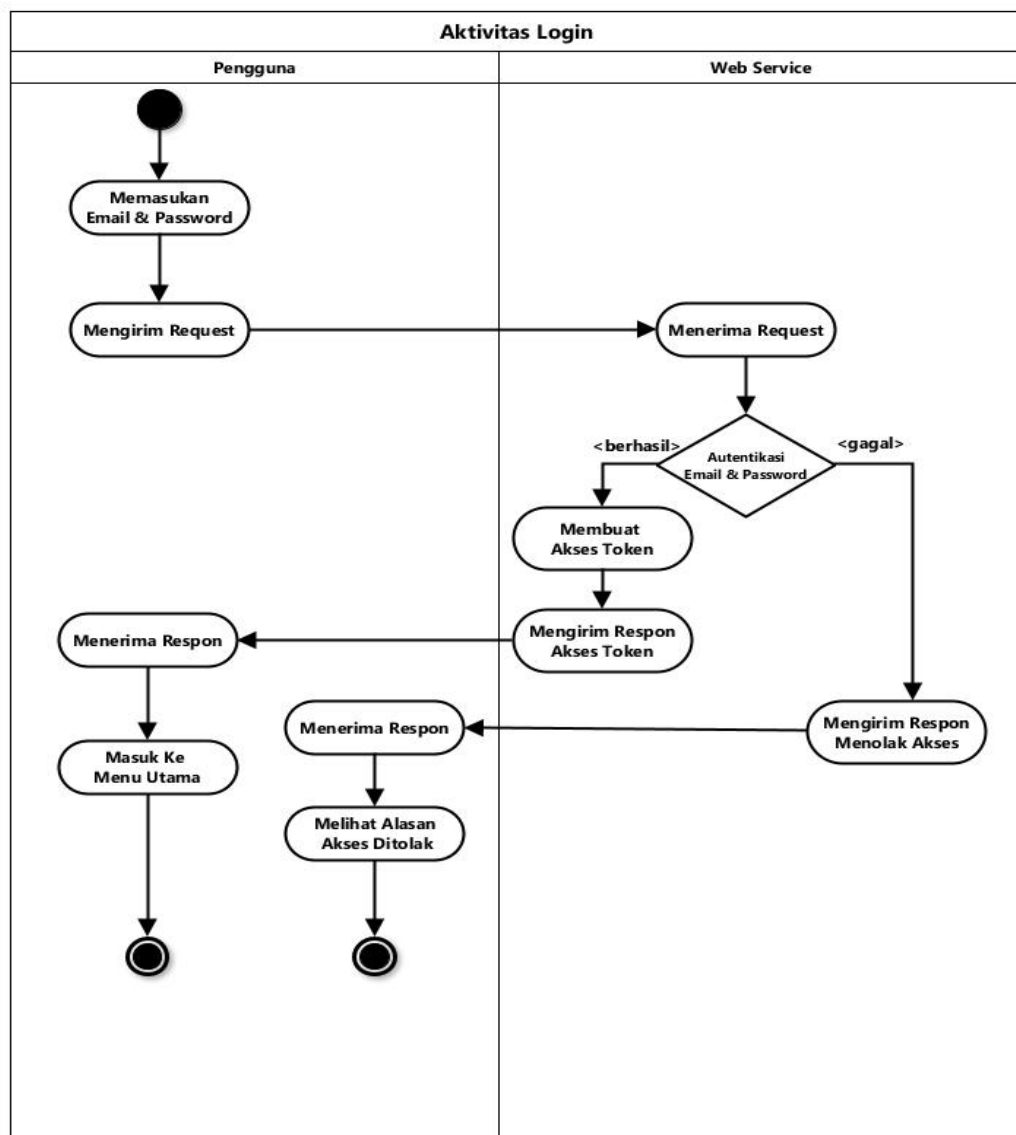
Activity Diagram (Diagram Aktivitas) diperlukan untuk menggambarkan proses bisnis dan urutan aktivitas dalam sebuah proses. Activity Diagram juga dipakai pada *business modeling* untuk memperlihatkan urutan aktifitas proses bisnis. Struktur diagram ini mirip *flowchart* atau *Data Flow Diagram* pada perancangan terstruktur. Activity Diagram sangat bermanfaat apabila kita membuat diagram ini terlebih dahulu dalam memodelkan sebuah proses untuk membantu memahami proses secara keseluruhan.

Activity diagram dibuat berdasarkan sebuah atau beberapa *use case* pada *use case* diagram. Activity Diagram berhubungan dengan diagram *Statechart*. Diagram *Statechart* fokus pada obyek dalam suatu proses (atau proses menjadi suatu obyek), sedangkan Activity Diagram fokus pada aktifitas-aktifitas yang terjadi yang terkait dalam suatu proses tunggal. Activity Diagram menggambarkan aktivitas sistem bukan apa yang dilakukan aktor merupakan suatu pemahaman yang penting.

1. Login

Aktivitas login diawali dengan sistem mengambil masukan email dan password dari pengguna. Kemudian mengirimkan email dan password tersebut dalam sebuah request ke Web Service. WebService menerima request dan melakukan autentikasi email dan password.

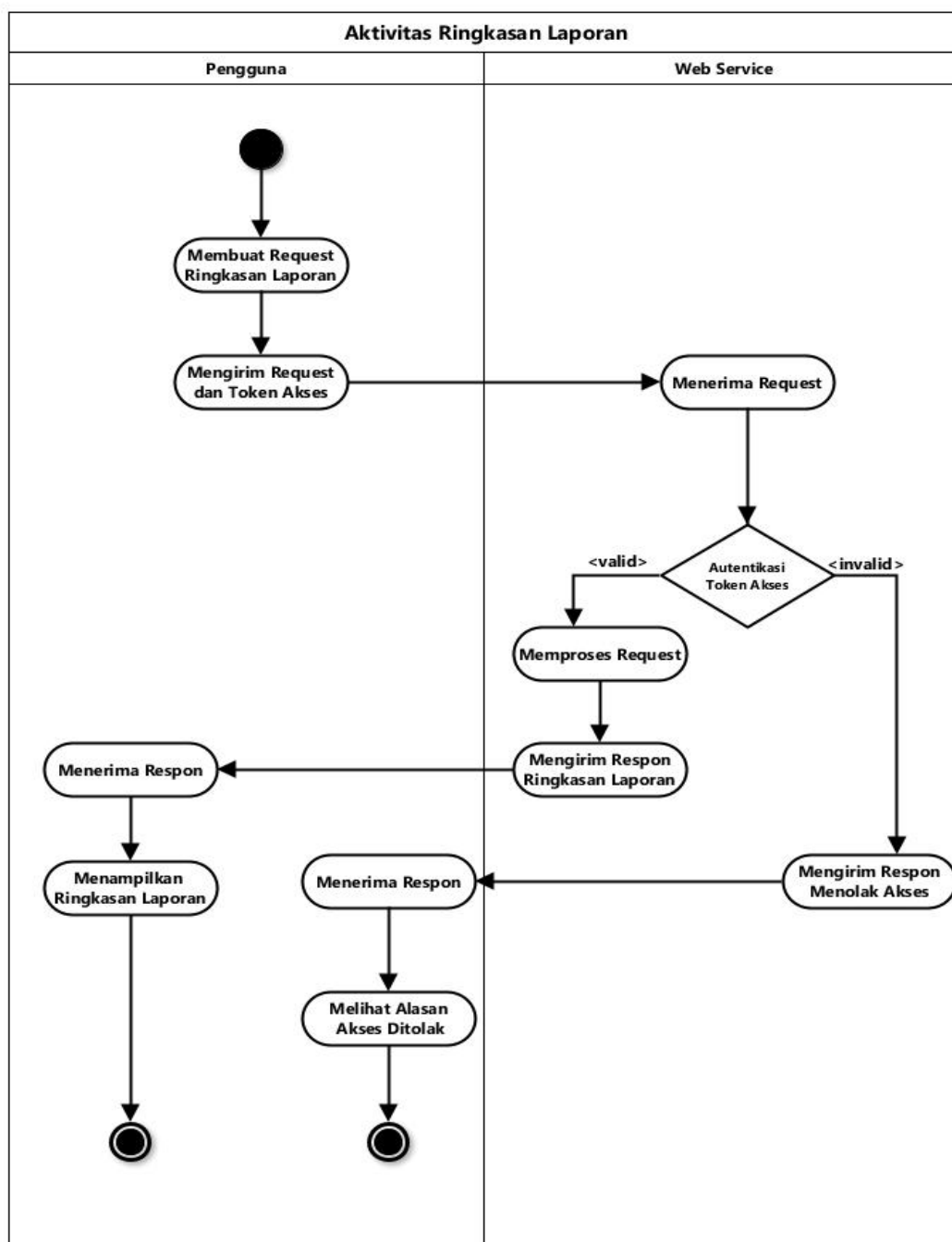
Jika autentikasi gagal maka Web Service akan mengirim respon menolak akses dan diterima sistem pengguna untuk ditampilkan alasan mengapa akses ditolak. Namun jika autentikasi berhasil Web Service akan membuat akses token yang kemudian dikirim ke sistem pengguna. Pengguna menerima respon dan mulai masuk ke Menu utama.



Gambar 3.2 Activity Diagram Login

2. Ringkasan Laporan

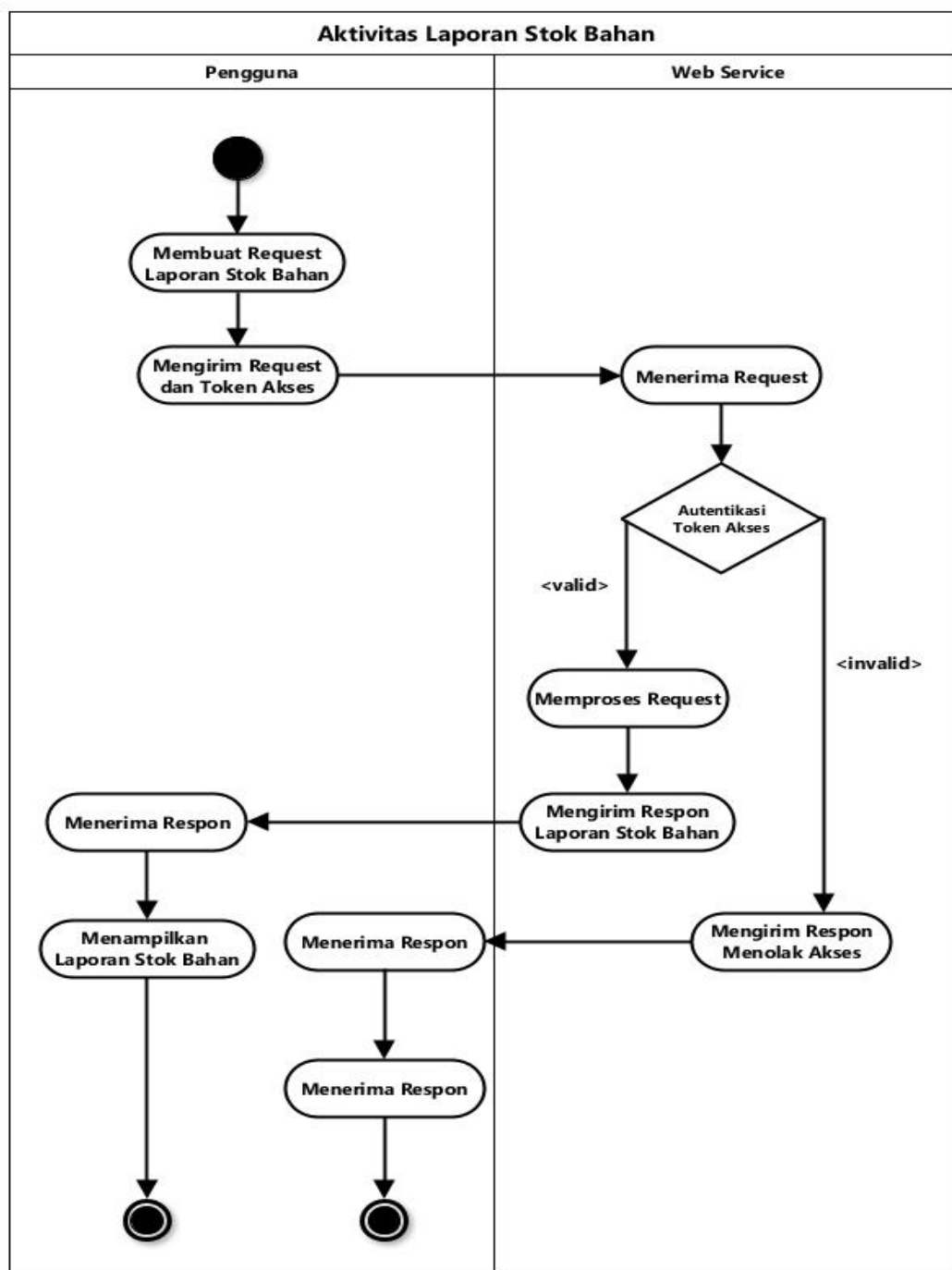
Ringkasan Laporan atau dalam View bernama Beranda merupakan salah satu menu utama dari aplikasi. Dalam aktivitas beranda ini, aplikasi akan membuat *request* yang dibersamai token akses dengan *trigger* dari pengguna memilih beranda. Request dikirim ke Web Service dan dilakukan autentikasi token akses. Jika token akses valid maka web service mengembalikan data ringkasan laporan yang kemudian ditampilkan



Gambar 3.3 Activity Diagram Ringkasan Laporan

3. Laporan Stok Bahan

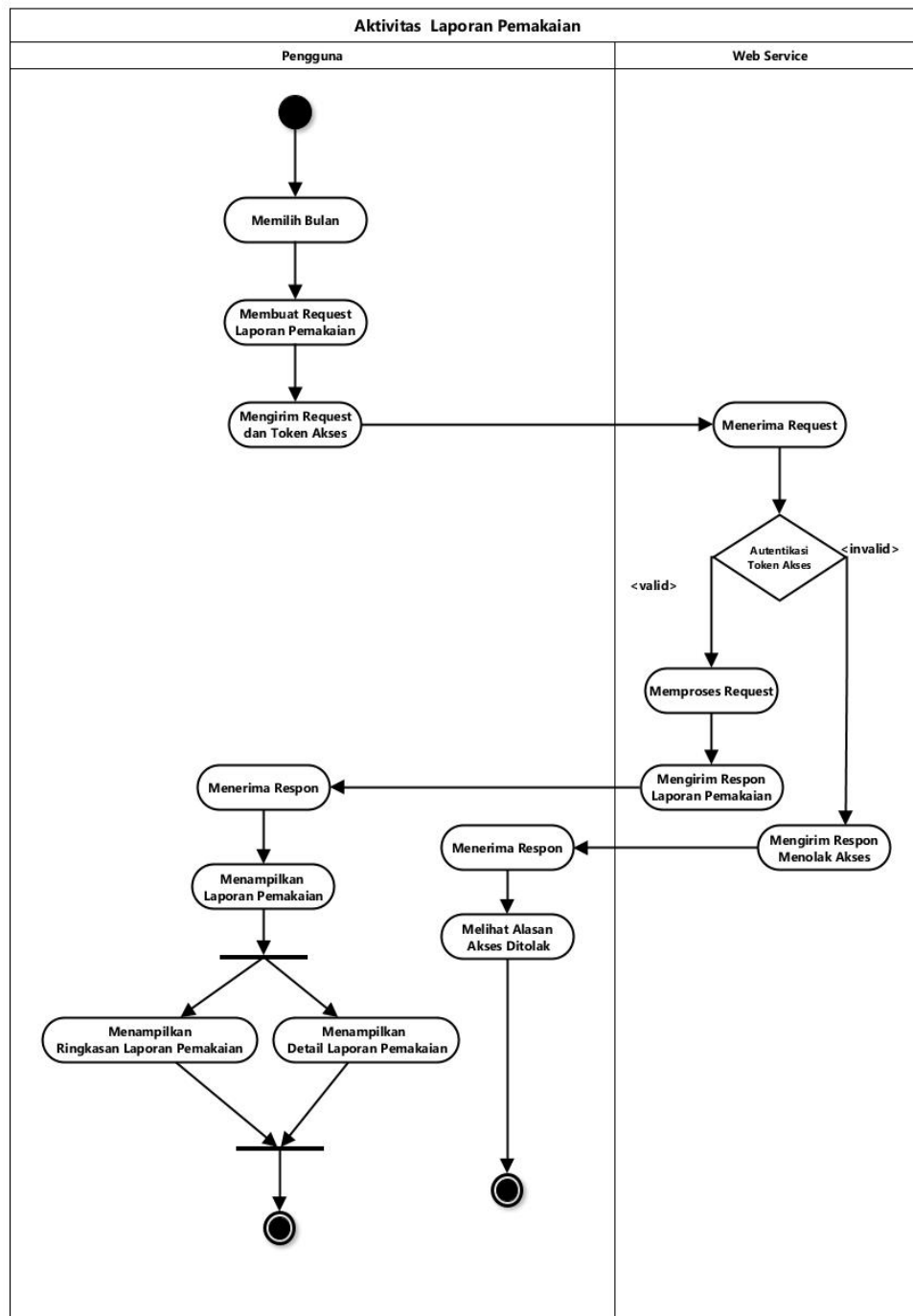
Aktivitas Laporan Stok Bahan ini serupa dengan aktivitas Ringkasan Laporan hanya saja request yang dikirim ke Web Service adalah request Laporan Stok Bahan.



Gambar 3.4 Activity Diagram Laporan Stok Bahan

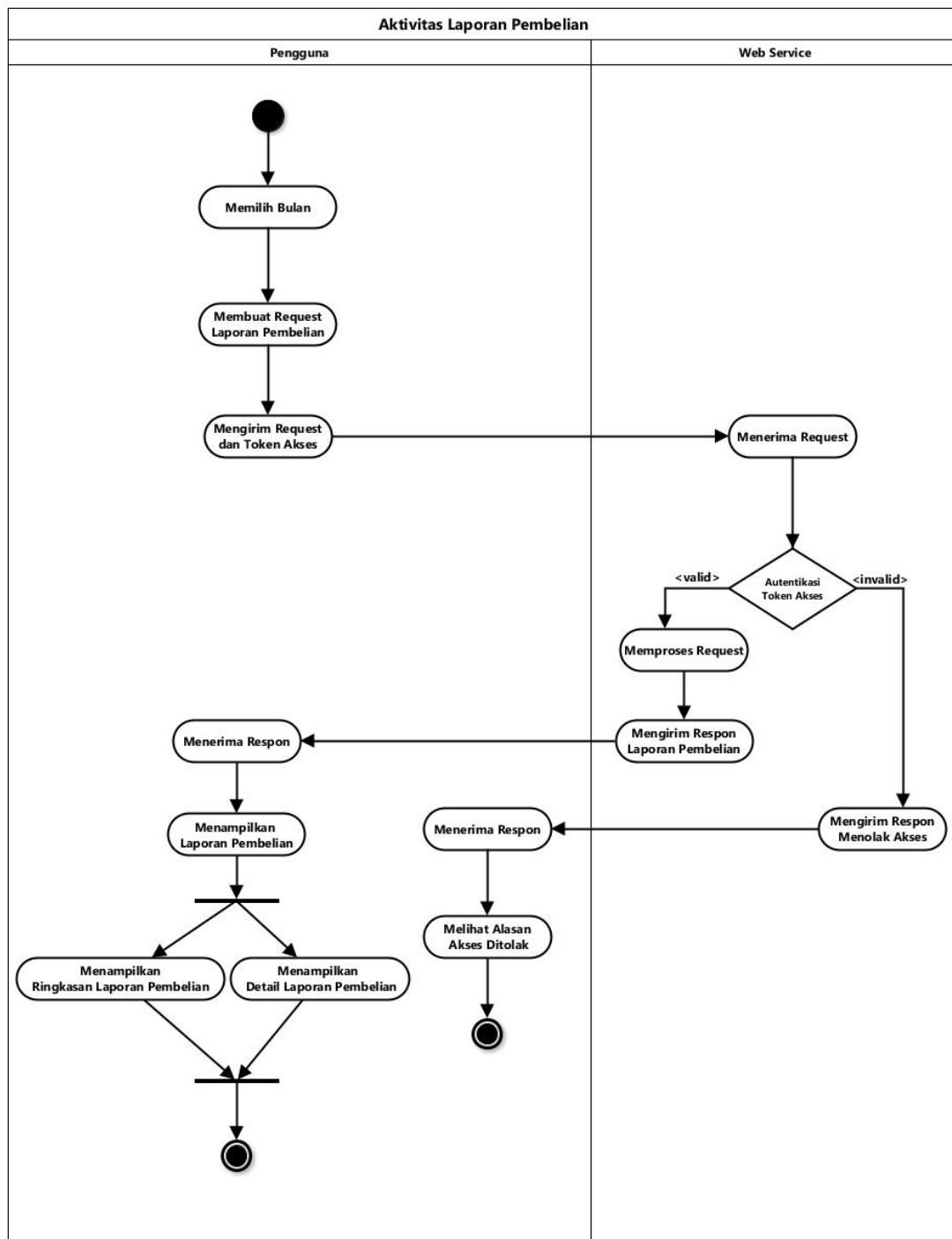
4. Laporan Pemakaian

Aktivitas Laporan Pemakaian ini serupa dengan aktivitas sebelumnya hanya saja request yang dikirim ke Web Service adalah request Laporan Pemakaian dan saat menampilkan dibagi menjadi dua tampilan yaitu Ringkasan dan Detail Laporan Pemakaian.



Gambar 3.5 Activity Diagram Laporan Pemakaian

5. Laporan Pembelian

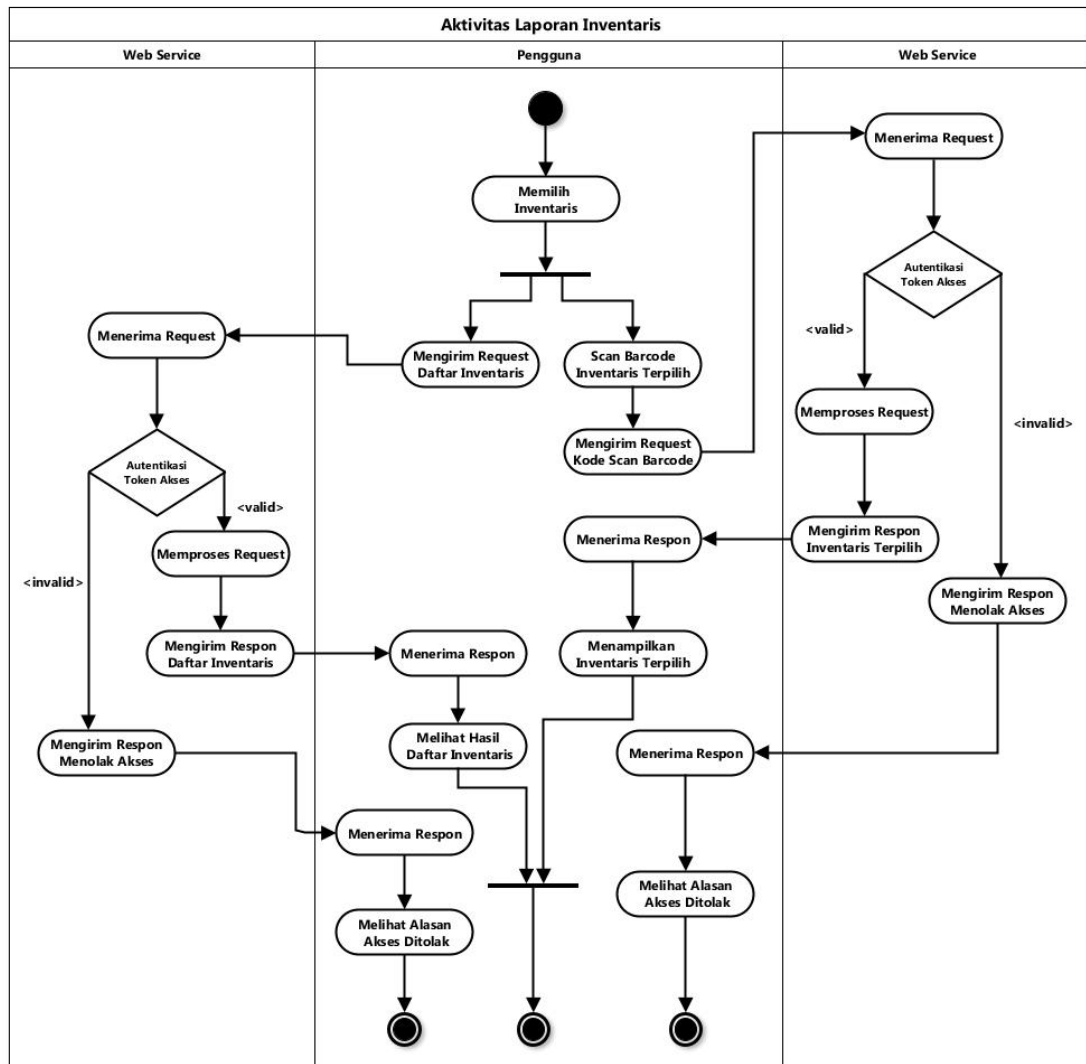


Gambar 3.6 Activity Diagram Laporan Pembelian

Aktivitas Laporan Pembelian ini serupa dengan aktivitas Laporan Pemakaian hanya saja request yang dikirim ke Web Service adalah request Laporan Pembelian dan saat

menampilkan dibagi menjadi dua tampilan yaitu Ringkasan dan Detail Laporan Pembelian .

6. Laporan Inventaris

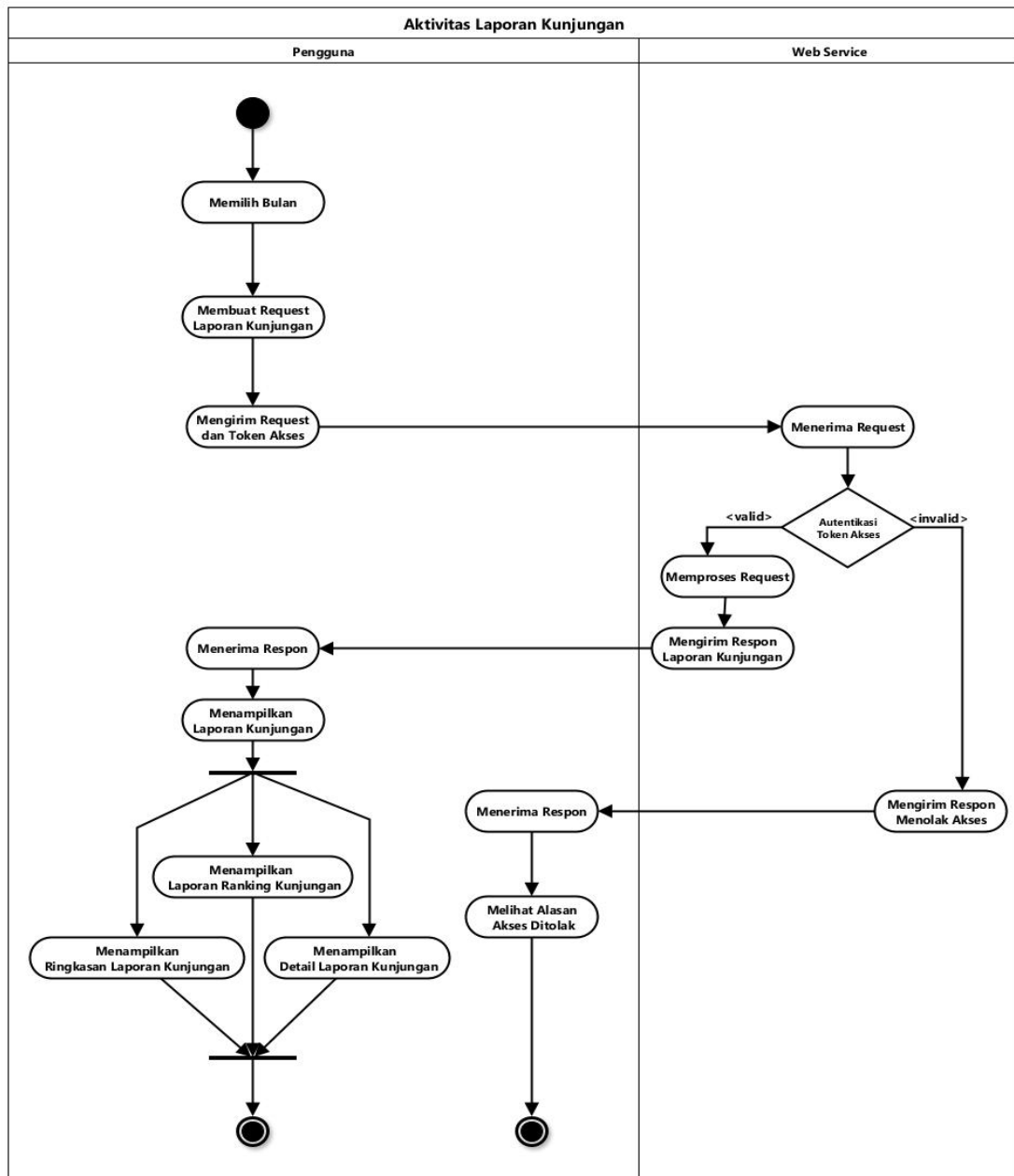


Gambar 3.7 Activity Diagram Laporan Inventaris

Aktivitas Laporan Laporan Inventaris ini aktivitas melihat data serupa dengan aktivitas Ringkasan Laporan hanya saja request yang dikirim ke Web Service adalah request Laporan Inventaris, sedangkan untuk *scan barcode*. Pengguna akan melakukan *scan barcode* pada inventaris yang akan dilihat informasinya. Setelah *scan* kemudian mengirim

request dengan kode *barcode* hasil scan jika ditemukan maka Web Service akan mengembalikan data informasi dari inventaris tersebut.

7. Laporan Kunjungan

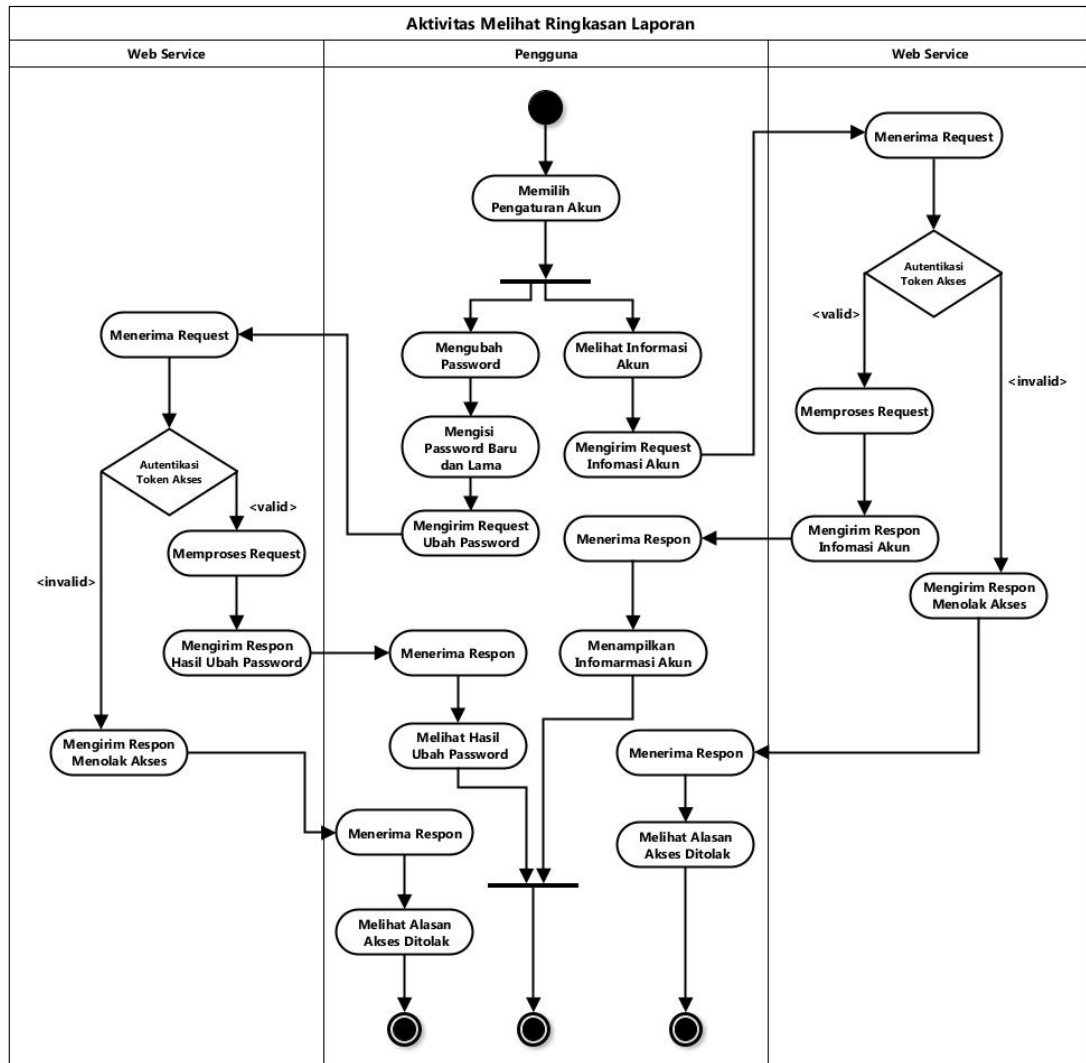


Gambar 3.8 Activity Diagram Laporan Kunjungan

Aktivitas Laporan Laporan Kunjungan ini serupa dengan aktivitas Laporan Pemakaian hanya saja request yang dikirim ke Web Service adalah request Laporan

Pembelian dan saat menampilkan dibagi menjadi tiga tampilan yaitu Ringkasan, Ranking dan Detail Laporan Kunjungan .

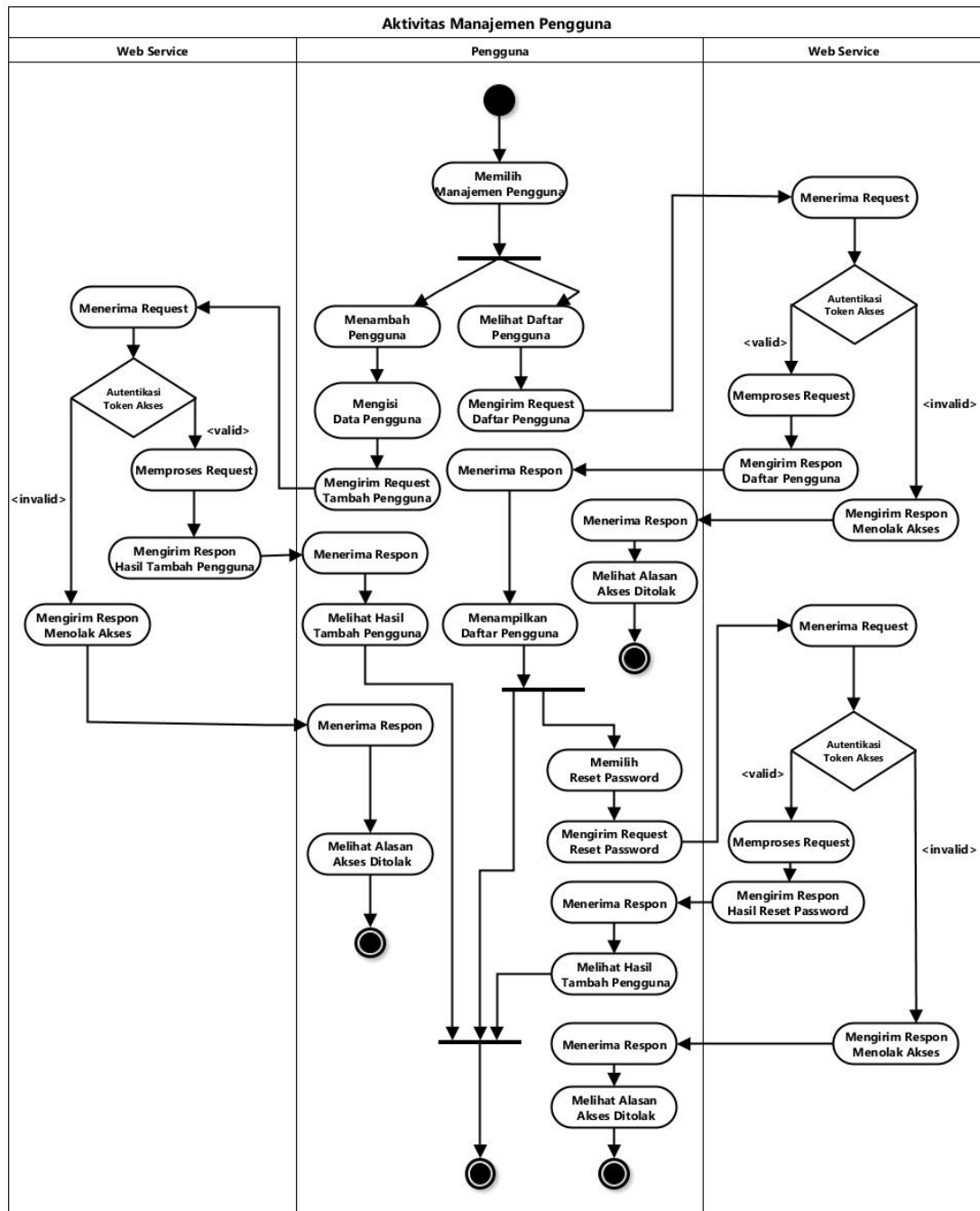
8. Pengaturan Akun



Gambar 3.9 Activity Diagram Pengaturan Akun

Pada aktivitas ini pengguna dapat melihat informasi akunnya dan mengubah kata sandi untuk aktivitas informasi akun ini serupa dengan aktivitas Ringkasan Laporan hanya saja request yang dikirim ke Web Service adalah request Informasi Akun, sedangkan untuk aktivitas ubah kata sandi pengguna mengisi kata sandi lama dan baru kemudian dikirimkan ke Web Service jika valid token akses dan kata sandi lamana maka aksi ubah kata sandi bisa dilanjutkan dan dipastikan berhasil.

9. Manajemen Pengguna



Gambar 3.10 Activity Diagram Manajemen Pengguna

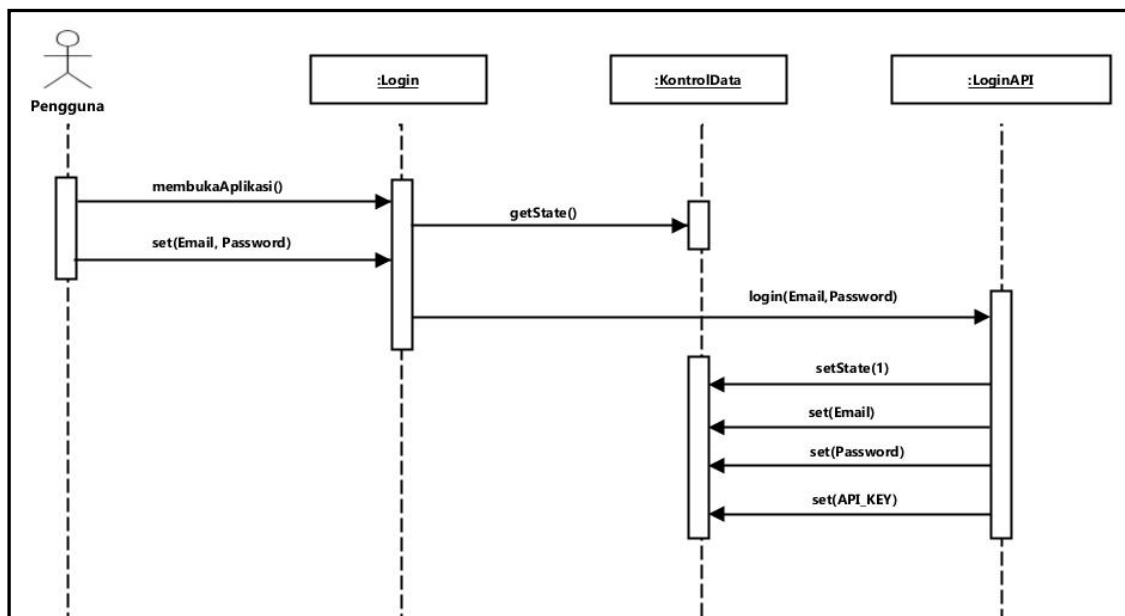
Aktivitas manajemen akun dibagi menjadi tiga yaitu melihat daftar pengguna, reset kata sandi pengguna dan tambahkan pengguna. Aktivitasnya serupa dengan aktivitas - aktivitas sebelumnya

C. SEQUENCE DIAGRAM

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem yang digambarkan terhadap waktu. *Sequence* diagram terdiri atas dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait).

Sequence diagram biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah *event* untuk menghasilkan *output* tertentu. Diagram ini secara khusus berasosiasi dengan *use case* diagram dan memperlihatkan tahap demi tahap apa yang seharusnya terjadi untuk menghasilkan sesuatu di dalam *use case*.

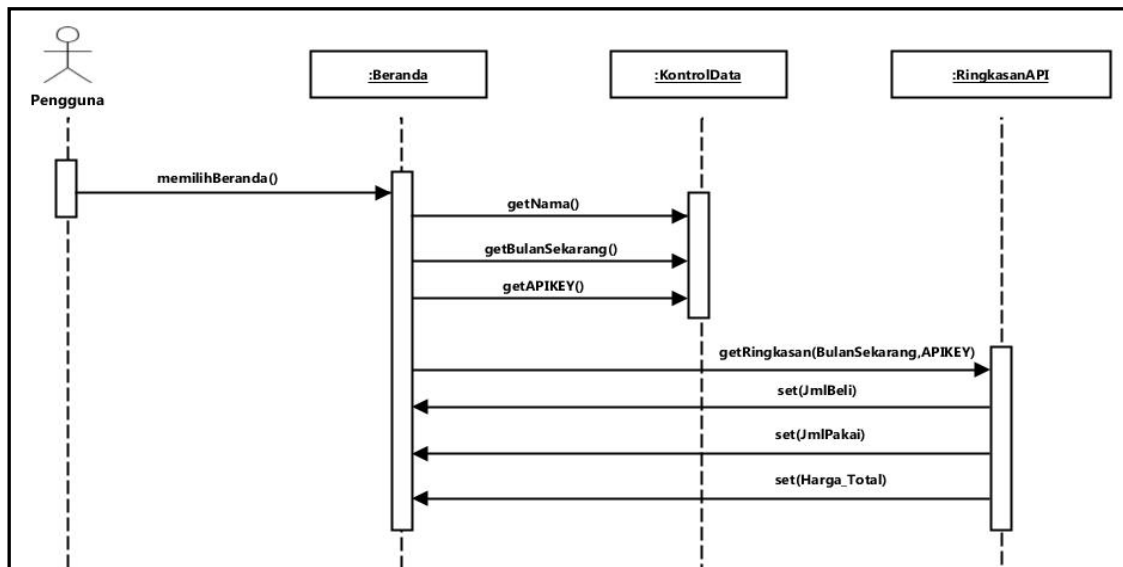
1. Sequence Login



Gambar 3.11 Sequence Login

Tergambar bagaimana objek Login berkomunikasi dengan objek KontrolData sebagai objek yang menyimpan state data dari aplikasi dan juga objek LoginAPI yang menjadi perantara untuk terhubung dengan Web Service.

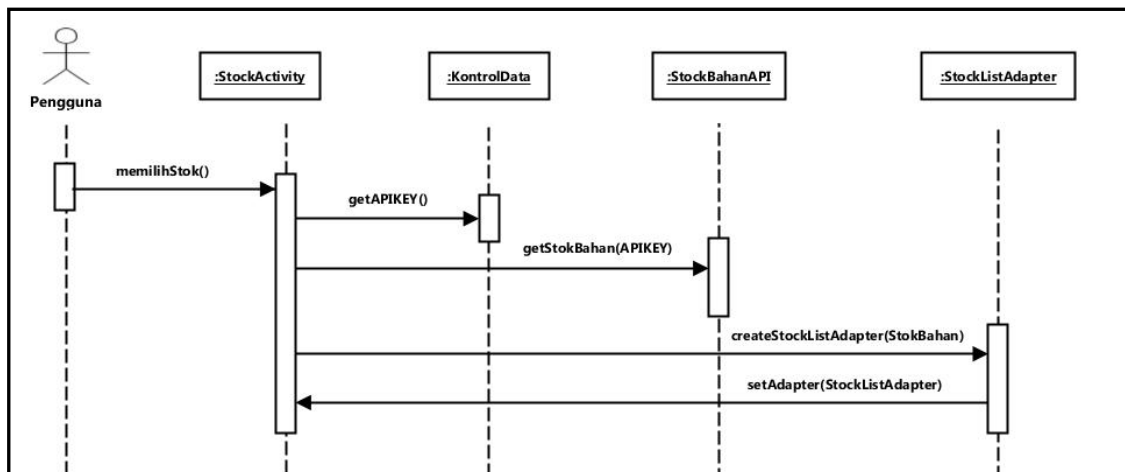
2. Sequence Ringkasan Laporan



Gambar 3.12 Sequence Ringkasan Laporan

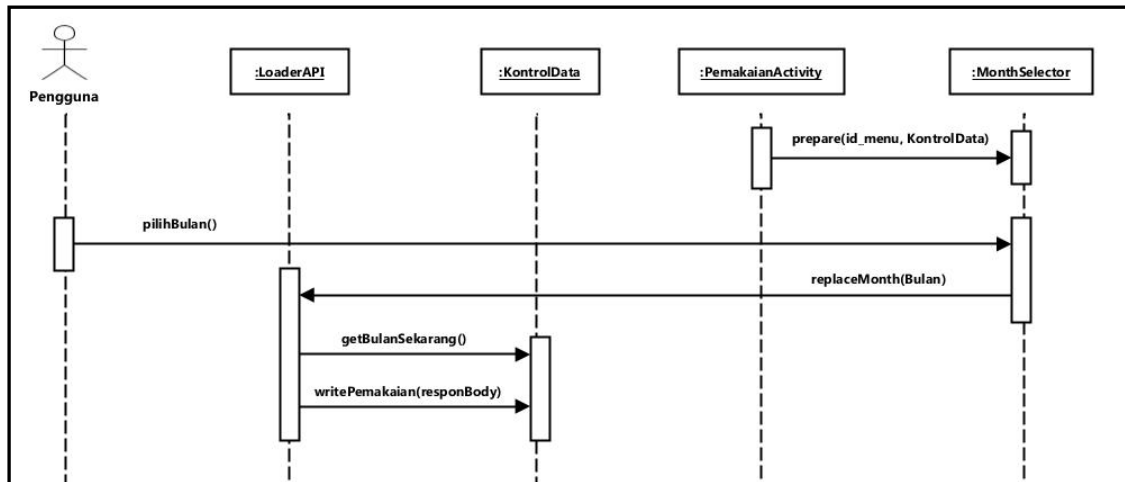
Tergambar bagaimana objek Beranda berhubungan dengan objek KontrolData mengambil data state yang digunakan mengambil data dari Webservice melalui objek RingkasanAPI.

3. Sequence Laporan Stok Bahan



Gambar 3.13 Sequence Laporan Stok Bahan

Serupa dengan Sequence sebelumnya dimana objek berhubungan dengan objek KontrolData untuk mengambil state data yang digunakan mengambil data dari Web Service, tambahannya adalah data yang diterima dari objek RingkasanAPI dikirim ke objek StockListAdapter untuk ditampilkan

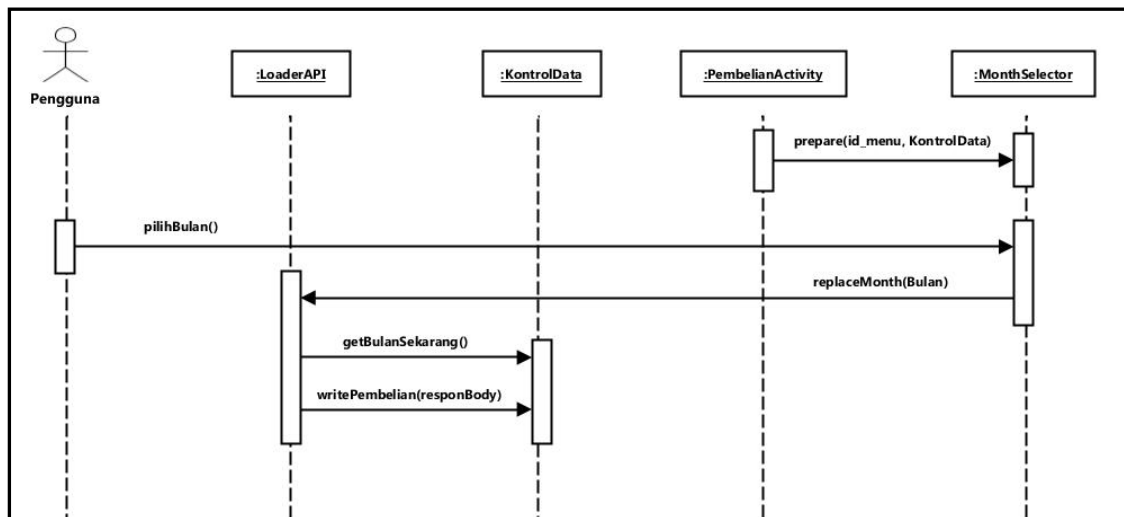


Gambar 3.15 Sequence Laporan Pemakaian MonthSelector

Objek - objek yang berhubungan dalam *case* ini cukup banyak karena pada implementasinya dibagi menjadi beberapa *view/boundary*. Pertama adalah mengirimkan objek *KontrolData* saat membuat objek *LoaderAPI* yang berperan untuk mengambil data dari Web Service.

Setelah data didapat maka data disimpan pada objek *KontrolData* untuk nantinya mempermudah proses distribusikan ke beberapa objek *boundary*. Objek *PemakaianActivity* dibuat sembari membuat objek *PemakaianAdapter* yang berperan sebagai *kontroller* dari beberapa objek *boundary* yaitu *GrafikPemakaian*, *DetailKategoriPemakaian* dan *DetailTransaksiPemakaian*. Objek *boundary* mendapat *trigger* dari pengguna untuk menampilkan datanya yang didapat dari objek *KontrolData*.

Secara *default* data yang diambil *LoaderAPI* dari Web Service merupakan data pada bulan berjalan untuk merubah request berdasarkan bulan lain maka ada objek *MonthSelector*. Objek ini dideklarasikan pada objek *PemakaianActivity*, objek *MonthSelector* baru merubah bulan pada objek *LoaderAPI* setelah mendapat *trigger* dari pengguna yang memilih bulan tertentu, pada aplikasi ini bulan hanya disediakan 6 bulan ke belakang dari bulan berjalan.



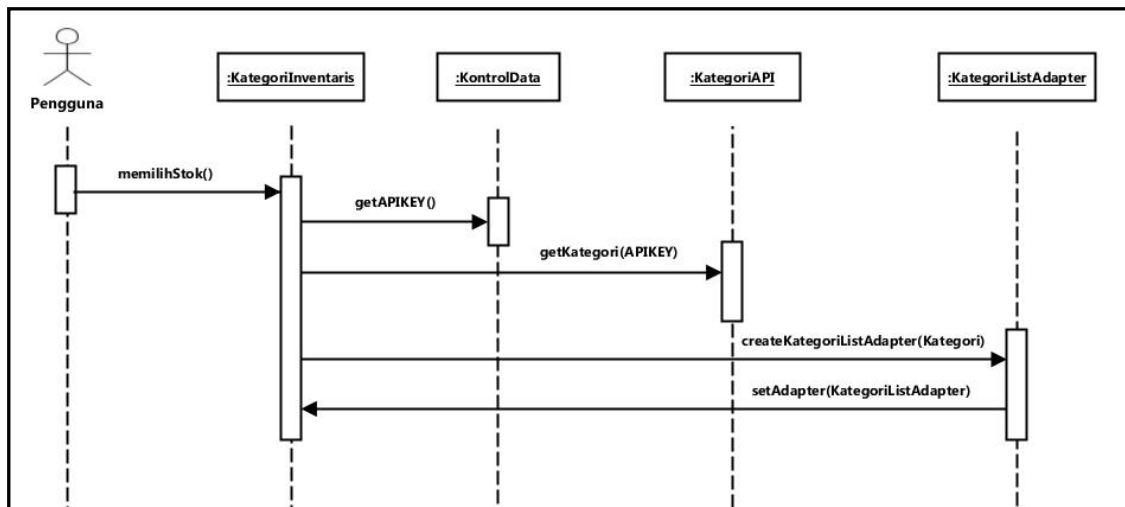
Gambar 3.17 Sequence Laporan Pemakaian MonthSelector

Objek - objek yang berhubungan dalam *case* ini cukup banyak karena pada implementasinya dibagi menjadi beberapa *view/boundary*. Pertama adalah mengirimkan objek *KontrolData* saat membuat objek *LoaderAPI* yang berperan untuk mengambil data dari Web Service.

Setelah data didapat maka data disimpan pada objek *KontrolData* untuk nantinya mempermudah proses distribusikan ke beberapa objek *boundary*. Objek *PemakaianActivity* dibuat sembari membuat objek *PemakaianAdapter* yang berperan sebagai *kontroller* dari beberapa objek *boundary* yaitu *GrafikPemakaian*, *DetailKategoriPemakaian* dan *DetailTransaksiPemakaian*. Objek *boundary* mendapat *trigger* dari pengguna untuk menampilkan datanya yang didapat dari objek *KontrolData*.

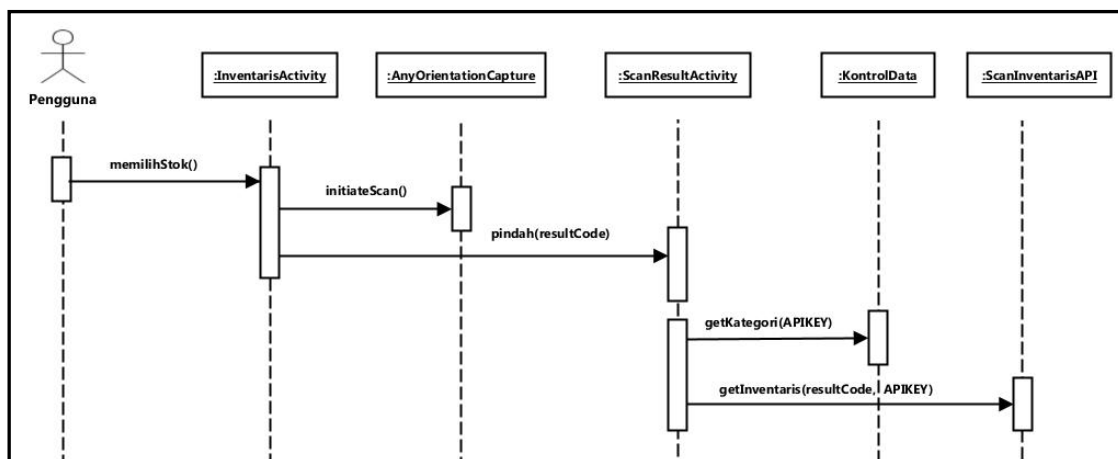
Secara default data yang diambil *LoaderAPI* dari Web Service merupakan data pada bulan berjalan untuk merubah request berdasarkan bulan lain maka ada objek *MonthSelector*. Objek ini dideklarasikan pada objek *PemakaianActivity*, objek *MonthSelector* baru merubah bulan pada objek *LoaderAPI* setelah mendapat *trigger* dari pengguna yang memilih bulan tertentu, pada aplikasi ini bulan hanya disediakan 6 bulan kebelakang dari bulan berjalan.

6. Sequence Laporan Inventaris



Gambar 3.18 Sequence List Kategori Inventaris

Laporan Inventaris bagian ini terdapat 4 objek yang saling terhubung. Objek KategoriInventaris merupakan objek boundary yang mendapat *trigger* dari pengguna, setelah mendapat *trigger* objek mengambil data APIKEY dari objek KontrolData yang digunakan untuk mengirim *request* ke objek KategoriAPI. KategoriAPI membuat objek KategoriListAdapter sembari mengirim data Kategori yang didapat, terakhir objek KategoriListAdapter mengembalikan *view* ke objek *boundary*.



Gambar 3.19 Sequence Inventaris Scan Activity

Sequence ini merupakan bagian dari *activity scan barcode* untuk mendapatkan informasi dari Inventaris terkait. Objek InventarisActivity mengirim permintaan untuk

memulai *scan* kepada objek *AnyOrientationCapture* setelah mendapat *trigger* dari pengguna. Hasil *scan* dikirim ke objek *ScanResultActivity* yang kemudian mengirim permintaan informasi ke objek *ScanInventarisAPI* dengan parameter hasil *scan* dan *APIKEY* yang didapat dari objek *KontrolData*.

D. CLASS DIAGRAM

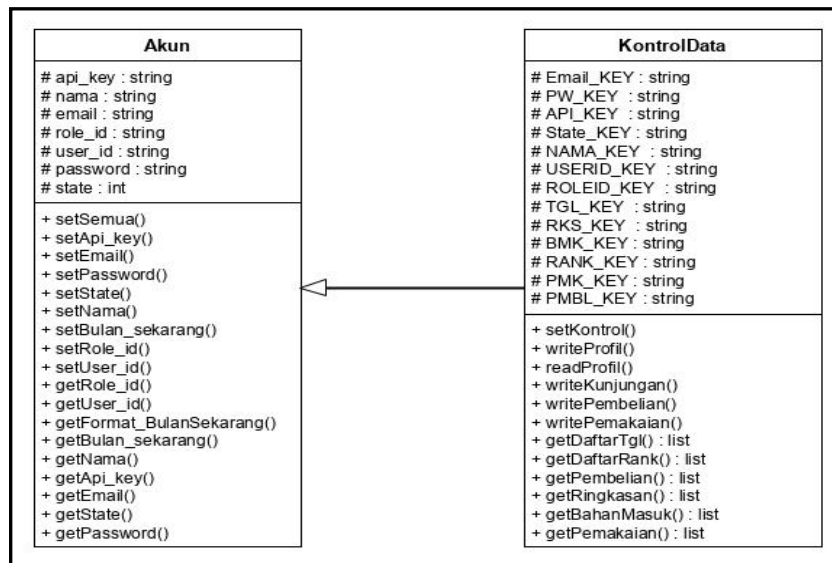
Class diagram dapat membantu dalam memvisualisasikan struktur kelas - kelas dari suatu sistem dan merupakan tipe diagram yang paling ditemui dalam pemodelan system berbasis *object-oriented*. Class Diagram memperlihatkan sekumpulan class, interface, dan collaborations dan relasi yang ada didalamnya.

Karena keterbatasan Class Diagram dibagi menjadi beberapa pembahasan yaitu Class Diagram *Boundary* dan *Control*, *Entity* dan *Control Entity*, serta Class diagram antar *Entity*. *Control Entity* pada pengembangan perangkat lunak ini adalah kelas *KontrolData* yang jika diperhatikan pada pembahasan *Sequence Diagram* objek *KontrolData* ini selalu muncul.

1. Boundary dan Control

Diagram ini menggambarkan bagaimana kelas - kelas *boundary* saling terhubung melalui kelas *control*. Mulai dari login sampai dengan sub dari main menu yang ada. Keunikan dari *object-oriented* android adalah memungkinkan satu kelas *control* untuk mengontrol sub kelas aktivitas lain yang syaratnya adalah aktivitas yang dikontrol berupa kelas *fragment(sub activity)*. Kelas master aktivitas pada perangkat lunak ini adalah *MainActivity*, kelas ini dapat melakukan kontrol aksi navigasi yang diberikan pengguna.

2. Entity dan Control Entity

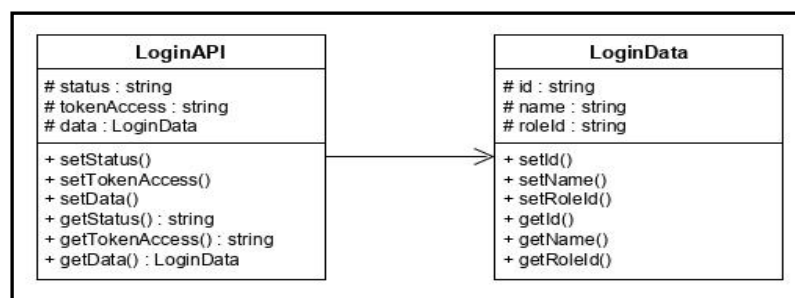


Gambar 3.21 Class Diagram Entity dan Control untuk State

Kegunaan kedua kelas ini adalah sebagai kelas yang menyimpan dan mengontrol berbagai state dalam perangkat lunak. Kelas akun berguna sebagai kelas entitas yang berguna menyimpan data akun pengguna yang nantinya digunakan untuk login otomatis dan menyimpan session login.

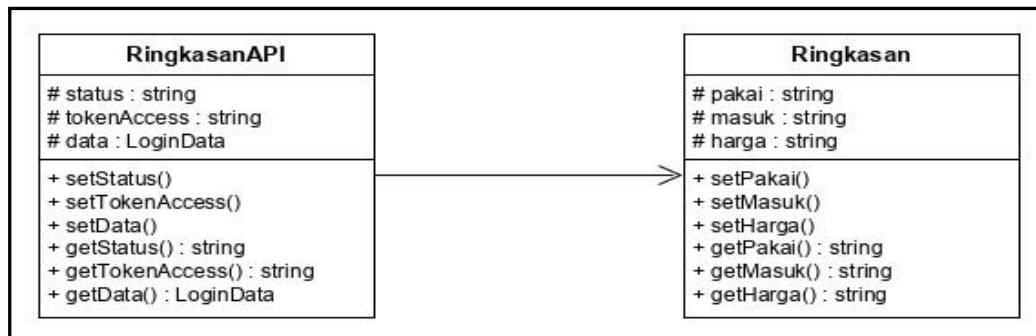
Kelas KontrolData menjadi jembatan (*control class*) perangkat lunak untuk menggunakan entitas Akun. KontrolData juga sebagai kelas yang berguna menyimpan dan mengatur *cache* data yaitu data yang disimpan sementara selama session login pengguna. Pada implementasinya KontrolData *extends* Akun.

3. Entity



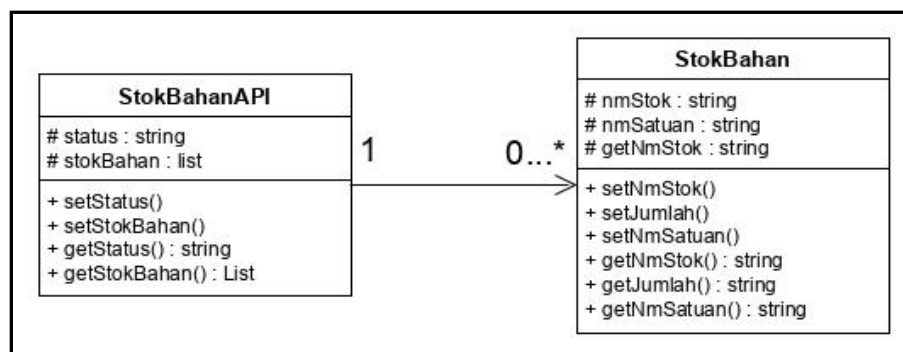
Gambar 3.22 Class Diagram Entity untuk Login

Kelas - kelas ini berguna untuk aktivitas login dan *session* login, LoginAPI diperjelas oleh kelas LoginData.



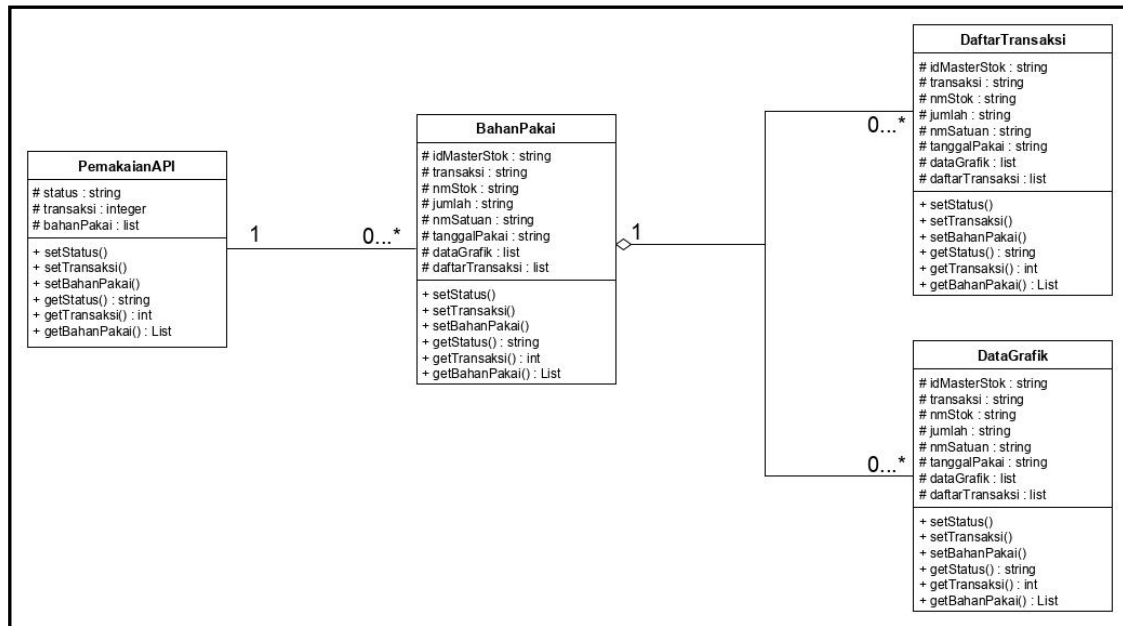
Gambar 3.23 Class Diagram Entity untuk Ringkasan Laporan

Kelas - kelas ini berguna untuk aktivitas Ringkasan laporan, RingkasanAPI diperjelas oleh kelas Ringkasan.



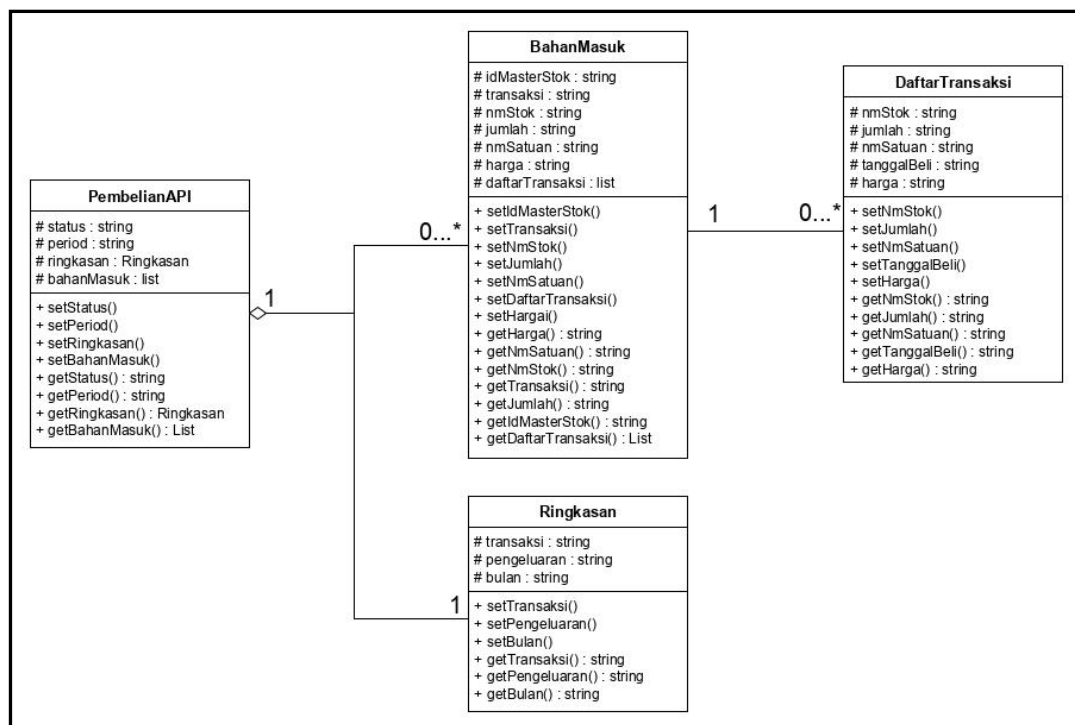
Gambar 3.24 Class Diagram Entity untuk Laporan Stok Bahan

Kelas - kelas ini berguna untuk aktivitas Laporan Stok Bahan, StokBahanAPI diperjelas oleh kelas StokBahan dengan hubungan asosiasi multiply bahwa satu StokBahanAPI akan mempunyai bagian StokBahan dengan jumlah 0 atau lebih.



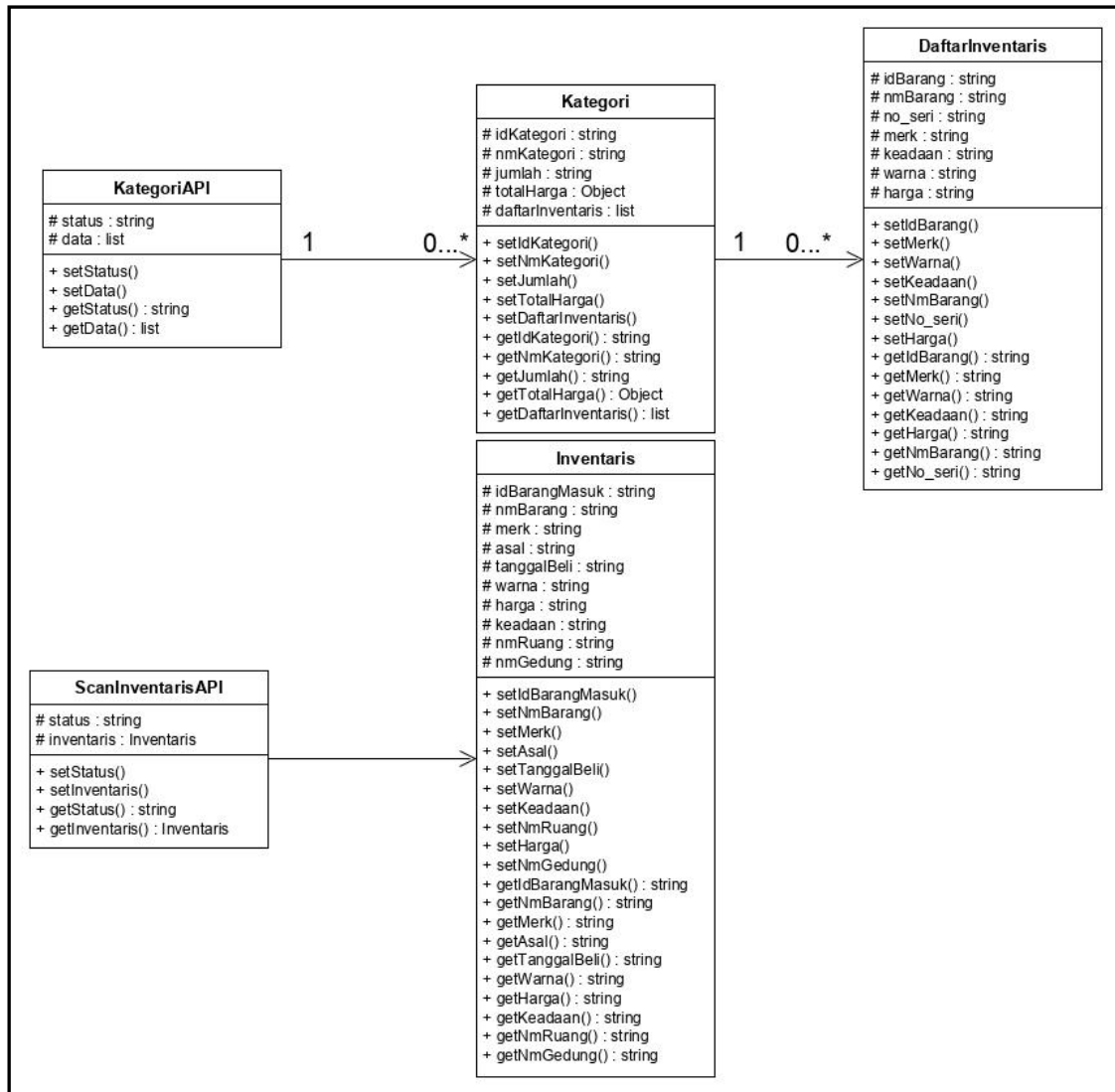
Gambar 3.25 Class Diagram Entity untuk Laporan Pemakaian

Kelas - kelas ini berguna untuk aktivitas Laporan Pemakaian, menggunakan hubungan asosiasi multiply dan agregasi yaitu dimana suatu kelas memiliki beberapa bagian kecil yaitu kelas BahanPakai memiliki 0 atau lebih kelas DaftarTransaksi dan 0 atau lebih kelas DataGrafik.



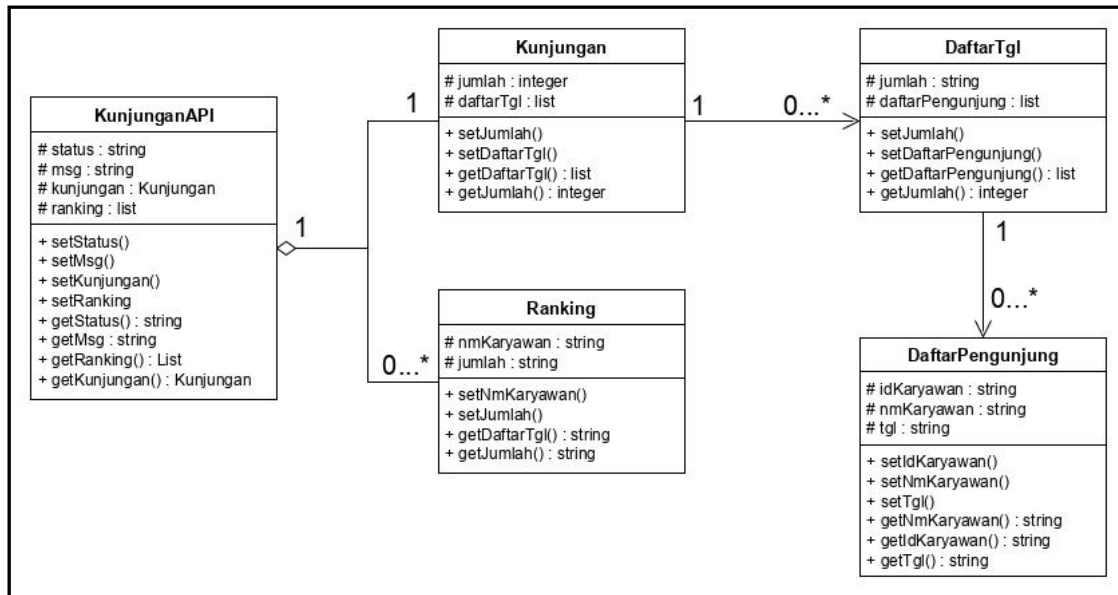
Gambar 3.26 Class Diagram Entity untuk Laporan Pembelian

Kelas - kelas ini berguna untuk aktivitas Laporan Pembelian, hubungan yang digunakan adalah asosiasi multiply dan agregasi dimana PembelianAPI memiliki 0 atau lebih kelas BahanMasuk dan 0 atau lebih kelas Ringkasan.



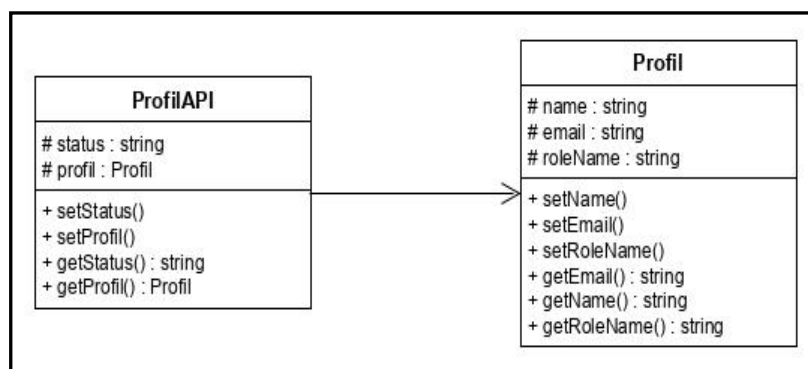
Gambar 3.27 Class Diagram Entity untuk Laporan Inventaris

Kelas - kelas ini berguna untuk aktivitas Laporan Inventaris, hubungan yang digunakan adalah asosiasi dan asosiasi multiply. Satu kelas KategoriAPI memiliki 0 atau lebih kelas Kategori dan satu kelas Kategori memiliki 0 atau lebih kelas DaftarInventaris, sementara kelas ScanInventarisAPI diperjelas oleh kelas Inventaris.



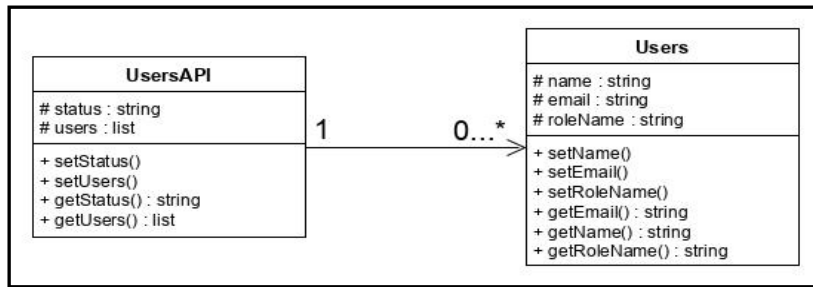
Gambar 3.28 Class Diagram Entity untuk Laporan Kunjungan

Kelas - kelas ini berguna untuk aktivitas Laporan Kunjungan, menggunakan hubungan asosiasi multiply dan agregasi yaitu dimana kelas KunjunganAPI memiliki 0 atau lebih kelas Ranking dan 1 kelas Kunjungan sementara 1 kelas Kunjungan sendiri memiliki 0 atau lebih kelas DaftarTgl dan 1 kelas DaftarTgl memiliki 0 atau lebih kelas DaftarPengunjung.



Gambar 3.29 Class Diagram Entity untuk Pengaturan Akun

Kelas - kelas ini berguna untuk aktivitas Pengaturan Akun dimana kelas ProfilAPI diperjelas oleh kelas Profil.



Gambar 3.30 Class Diagram Entity untuk Pengaturan Akun.

Kelas - kelas ini berguna untuk aktivitas Manajemen Akun dimana kelas **UsersAPI** memiliki 0 atau lebih kelas **Users**.

3.2.3 Tahap perancangan antar muka

Perancangan antarmuka merupakan tahap merancang tampilan antarmuka dari aplikasi ini. Perancangan antarmuka pada pengembangan aplikasi ini antara lain meliputi perancangan Antar Muka *Login*, *Beranda*, *Stok Bahan*, *Pemakaian Bahan*, *Pembelian Bahan*, *Pengaturan Akun*, *Inventaris Dapur*, *Manajemen Akun*.

A. Perancangan antar muka *Login*

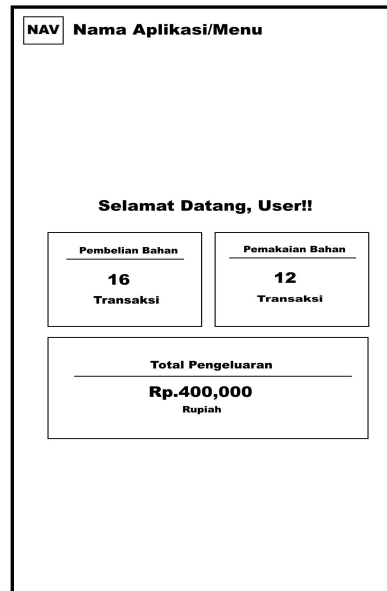
The mockup shows a login interface with the following elements: a central box labeled "LOGO", the text "DIVISI DAPUR" below it, a label "Alamat Email" above an input field, a label "Kata Sandi" above another input field, a "LOGIN" button, and a footer section labeled "INFORMASI :" containing the text "Lupa Password? Hubungi Staf IT Sekolah Teladan".

Gambar 3.31 Desain Antar Muka Login

Pada halaman login tidak terdapat menu pilihan untuk melakukan pendaftaran karena penambahan pengguna baru hanya dapat dilakukan admin dan kepala dapur saja melalui menu manajemen akun.

B. Perancangan antar muka *Beranda*

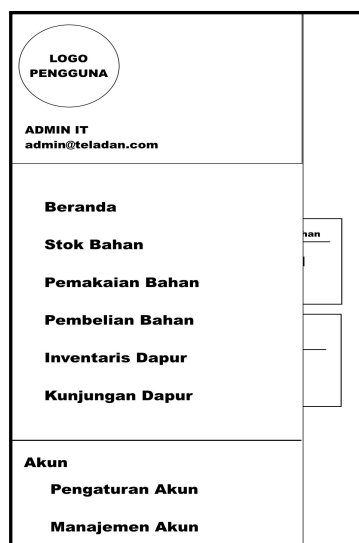
Pada halaman beranda ini ditampilkan ringkasan jumlah transaksi pembelian dan pemakaian serta jumlah pengeluaran untuk pembelian bahan dapur selama satu bulan berjalan.



Gambar 3.32 Desain Antar Muka Beranda

C. Perancangan antar muka *NAV Menu*

Aplikasi ini memiliki menu *side navigation* standar dari Google dan umum digunakan, menu navigasi akan berada pada sisi kiri layar dan dapat dipanggil dengan ketuk ikon menu atau menggeser layar dari kiri ke kanan.



Gambar 3.33 Desain Antar Muka Menu Navigasi

D. Perancangan antar muka Stok Bahan

Menu stok bahan menampilkan jumlah stok bahan pada hari berjalan.

The wireframe shows a mobile application interface for the 'Stock Material' menu. At the top, there is a navigation bar with a 'NAV' button and the text 'Nama Aplikasi/Menu'. Below this, the main header contains 'DAFTAR STOK BAHAN DAPUR' on the left and 'HARI INI' on the right. A search bar is positioned below the header, containing the placeholder text 'Nama Bahan' and 'Stok yang ada'. The main content area is a large, empty rectangular box.

Gambar 3.34 Desain Antar Muka Stok Bahan

E. Perancangan antar muka Pemakaian

Pada pemakaian berisi 2 menu yaitu grafik transaksi pemakaian, detail pemakaian. detail pemakaian memiliki dua sub menu yaitu detail transaksi pemakaian, dan detail grafik pemakaian.

The wireframe shows a mobile application interface for the 'Usage Graph' menu. The top navigation bar includes a 'NAV' button, 'Nama Aplikasi/Menu', and a 'PILIH BULAN' button. Below the navigation bar, there are two tabs: 'GRAFIK PEMAKAIAN' (selected) and 'DAFTAR PEMAKAIAN'. A search bar is located below the tabs, with the placeholder text 'Nama Bulan' and 'Jumlah Transaksi'. The main content area is a large rectangular box labeled 'GRAFIK PIE'.

Gambar 3.35 Antar Muka Grafik Pemakaian

The wireframe shows a mobile application interface for the 'Usage List' menu. The top navigation bar includes a 'NAV' button, 'Nama Aplikasi/Menu', and a 'PILIH BULAN' button. Below the navigation bar, there are two tabs: 'GRAFIK PEMAKAIAN' and 'DAFTAR PEMAKAIAN' (selected). A search bar is located below the tabs, with the placeholder text 'Nama Bulan' and 'Jumlah Transaksi'. The main content area is divided into three sections, each containing a table with the following columns: 'Nama Bahan', 'Jumlah Satuan', 'Transaksi', and 'Jumlah Transaksi'. Each table is followed by a 'Tampilkan Grafik' button.

Gambar 3.36 Desain Antar Muka Daftar Pemakaian

NAV	Nama Aplikasi/Menu	PILIH BULAN
GRAFIK PEMAKAIAN		DAFTAR PEMAKAIAN
Nama Bulan	Nama Bahan	
Jumlah Transaksi	Jumlah Satuan	
DAFTAR DETAIL TRANSAKSI PEMAKAIAN		
Nama Pengguna	Nama Bahan	
Nama Bahan	Jumlah Satuan	
Tanggal Transaksi		
Nama Pengguna	Nama Bahan	
Nama Bahan	Jumlah Satuan	
Tanggal Transaksi		
Nama Pengguna	Nama Bahan	
Nama Bahan	Jumlah Satuan	
Tanggal Transaksi		
Nama Pengguna	Nama Bahan	
Nama Bahan	Jumlah Satuan	
Tanggal Transaksi		

Gambar 3.37 Desain Detail Transaksi Pemakaian

NAV	Nama Aplikasi/Menu	PILIH BULAN
GRAFIK PEMAKAIAN		DAFTAR PEMAKAIAN
Nama Bulan	Nama Bahan	
Jumlah Transaksi	Jumlah Satuan	
GRAFIK DETAIL PEMAKAIAN BAHAN		
GRAFIK GARIS		

Gambar 3.38 Desain Detail Grafik Pemakaian

Detail transaksi pemakaian berisi transaksi bahan terpilih ditampilkan pula grafik pemakaian berisi statistik jumlah pemakaian dalam satu bulan berjalan.

F. Perancangan Antar Muka Pembelian

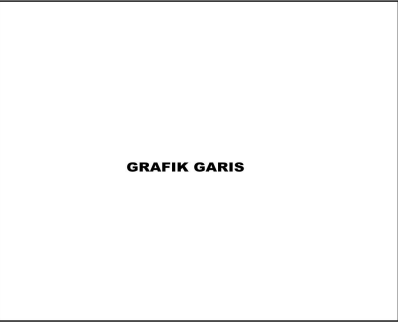
Pada menu/fungsi ini serupa dengan menu pemakaian hanya saja ada tambahan berupa harga yang dikeluarkan untuk membeli bahan tersebut.

NAV	Nama Aplikasi/Menu	PILIH BULAN
GRAFIK PEMBELIAN		DAFTAR PEMBELIAN
Nama Bulan	Total Pengeluaran	
Jumlah Transaksi	IDR	
GRAFIK TRANSAKSI PEMBELIAN		
GRAFIK PIE		
GRAFIK BIAYA PEMBELIAN		

Gambar 3.39 Desain Grafik Pembelian

NAV	Nama Aplikasi/Menu	PILIH BULAN
GRAFIK PEMBELIAN		DAFTAR PEMBELIAN
Nama Bulan	Total Pengeluaran	
Jumlah Transaksi	IDR	
DAFTAR PEMBELIAN BAHAN		
Nama Bahan	Jumlah Transaksi	
Jumlah Satuan	Total Pengeluaran	
Tampilkan Grafik		
Nama Bahan	Jumlah Transaksi	
Jumlah Satuan	Total Pengeluaran	
Tampilkan Grafik		
Nama Bahan	Jumlah Transaksi	
Jumlah Satuan	Total Pengeluaran	
Tampilkan Grafik		

Gambar 3.40 Desain Daftar Pembelian

NAV	Nama Aplikasi/Menu	PILIH BULAN
GRAFIK PEMBELIAN		DAFTAR PEMBELIAN
Nama Bulan	Jumlah Transaksi	
Nama Bahan	Total Pengeluaran	
Jumlah Satuan		
GRAFIK DETAIL PEMBELIAN BAHAN		
 <p>GRAFIK GARIS</p>		

Gambar 3.41 Desain Detail Grafik
Pembelian

NAV	Nama Aplikasi/Menu	PILIH BULAN
GRAFIK PEMBELIAN		DAFTAR PEMBELIAN
Nama Bulan	Jumlah Transaksi	
Nama Bahan	Total Pengeluaran	
Jumlah Satuan		
DAFTAR DETAIL TRANSAKSI PEMBELIAN		
Tanggal Transaksi	Nama Pengguna	
Nama Bahan	Harga	
Jumlah Satuan		
Tanggal Transaksi	Nama Pengguna	
Nama Bahan	Harga	
Jumlah Satuan		
Tanggal Transaksi	Nama Pengguna	
Nama Bahan	Harga	
Jumlah Satuan		

Gambar 3.42 Desain Detail Transaksi
Pembelian

Menu grafik transaksi pembelian menampilkan perbandingan jumlah transaksi bahan dapur menggunakan grafik *pie*(donat). Menu detail pembelian menampilkan detail transaksi pada bulan berjalan yaitu nama bahan, jumlah satuan, dan jumlah transaksi. Detail transaksi pembelian berisi transaksi bahan terpilih ditampilkan pula nama pengguna yang melakukan transaksi serta tanggal saat transaksi dilakukan, sedangkan detail grafik pembelian berisi statistik jumlah pembelian dalam satu bulan berjalan.

G. Perancangan Antar Muka Kunjungan

Kunjungan pegawai ke dapur akan muncul pada menu ini, grafik dipilih 1 minggu terakhir karena konsekuensi dari layar android yang tidak memadai untuk data lebih dari 14 item.

Gambar 3.43 Desain Grafik Kunjungan

Gambar 3.44 Desain Ranking Kunjungan

Selain grafik kunjungan 1 minggu terakhir ada pula *ranking* kunjungan pegawai ke dapur serta rincian kunjungan pada setiap hari.

Gambar 3.45 Desain Hari Kunjungan

Gambar 3.46 Desain Detail Kunjungan

H. Perancangan Antar Muka Inventaris

NAV Nama Aplikasi/Menu SCAN

DAFTAR KATEGORI INVENTARIS

Nama Kategori	Total Harga
Jumlah	Rp. JUMLAH

Gambar 3.47 Desain Kategori Inventaris

NAV Nama Aplikasi/Menu SCAN

DAFTAR INVENTARIS <Nama Kategori>

KODE ID BARANG

NAma | Merek | Warna

Seri

Keadaan

Harga IDR

KODE ID BARANG

NAma | Merek | Warna

Seri

Keadaan

Harga IDR

KODE ID BARANG

NAma | Merek | Warna

Seri

Keadaan

Harga IDR

Gambar 3.48 Desain Detail Inventaris

Terdapat pula fitur untuk scanning QRCode yang tertempel pada aset inventaris untuk melihat detail informasi dari aset inventaris tersebut.

FOKUS QRCode

Arahkan Kamera pada QRCode

Gambar 3.49 Desain Scan QRCode

NAV Nama Aplikasi/Menu SCAN

HOLDER GAMBAR

DETAIL INFORMASI BARANG

ID BARANG

Nama Barang

Merek

Asal

Tanggal Pengadaan

Warna

Gambar 3.50 Desain Hasil Scan QRCode

Menu akan mengarahkan untuk membuka kamera dan mengarahkannya untuk *scan* pada QRCode sehingga mendapatkan hasil informasi dari aset tersebut.

I. Perancangan antar muka Pengaturan Akun

Pada menu ini terdapat fungsi untuk melihat informasi akun, mengubah kata sandi dan keluar (*logout*) dari aplikasi.

Gambar 3.51 Desain Pengaturan Akun

J. Perancangan Antar Muka Manajemen Akun

Gambar 3.52 Desain Manajemen Akun

Gambar 3.53 Desain Tambah Anggota

Menu untuk Admin dan Kepala Dapur, menu ini berfungsi untuk melihat daftar pengguna aplikasi, mengatur ulang kata sandi serta menambahkan pengguna baru.

3.2.4 Tahap pengujian dan implementasi

Implementasi dan pengujian dilakukan setelah perancangan dan desain selesai. Pengkodean mulai dilakukan, dalam pengembangan perangkat lunak ini pengkodean menggunakan bahasa Java, XML, JavaScript, dan PHP.

Pengujian menggunakan metode *black box* yaitu menguji fungsional dari perangkat lunak. Pengujian menggunakan berbagai data yang asli bukan data *dummy* asal sehingga pengujian mendapatkan hasil yang lebih akurat. Pada tahap ini beberapa kali dijumpai fungsi yang gagal karena pada data tertentu.

3.3 Deskripsi Aktivitas

Tabel 3.1 Deskripsi Aktivitas

No	Tanggal	Kegiatan
1	22 Desember 2018	Observasi/Meminta Topik Kerja Praktek
2	24 Desember 2018	Observasi/Meminta Data Inventaris Melakukan Normalisasi Tabel
3	26 Desember 2018	Meminta Masukan tabel normalisasi
4	27 Desember 2018	Mencari dan mencoba referensi QR scan Android
5	28 Desember 2018	Membuat design + prototipe bagian QR Scan inventaris barang
6	29 Desember 2018	Demo, minta pendapat pembimbing, dan memperbaiki desain bagian scan qr
7	31 Desember 2018	Memperbaiki desain bagian QR Code & membuat desain untuk bagian lain
8	2 Januari 2019	Membuat desain bagian lain
9	3 Januari 2019	Mencari dan mencoba referensi membuat Grafik pada android
10	4 Januari 2019	Membuat prototipe bagian kunjungan
11	5 Januari 2019	Demo, minta pendapat pembimbing desain menu kunjungan
12	6 Januari 2019	Mencari penyebab error QR Scan dan cara memperbaiki
13	7 Januari 2019	Memperbaiki error QR Scan
14	8 Januari 2019	Merubah oreintasi QR Scan menjadi Potrait
15	11 Januari 2019	Mendapat tabel untuk proses kunjungan, mempelajari dan mencoba queri
16	12 Januari 2019	Mencari dan mencoba referensi REST API(WebService)
17	14 Januari 2019	Membuat API GET untuk data kunjungan dan test pada localhost
18	15 Januari 2019	Membuat API GET untuk data inventaris dan test pada localhost
19	16 Januari 2019	Mencari dan mencoba referensi handler koneksi webservice Android
20	17 Januari 2019	Mengunggah Web Service ke hosting : GET kunjungan & inventaris Mencoba menghubungkan android dengan webservice

Tabel 3.2 Deskripsi Aktivitas Lanjutan

21	18 Januari 2019	Membuat method/fungsi token untuk API login, aplikasi mobile membutuhkan token untuk menggantikan session
22	19 Januari 2019	Membuat API untuk beranda dan stok bahan dan menghubungkan dengan Android
23	21 Januari 2019	Membuat API pembelian dan pemakaian
24	22 Januari 2019	Menghubungkan dengan Android
25	23 Januari 2019	Demo, meminta pendapat tentang yang sudah dikerjakan khususnya bagian ketidak sesuaian dengan desain.
26	24 Januari 2019	Memperbaiki, mengganti interface grafik kemudian didemokan lagi
27	25 Januari 2019	Memperbaiki query Membuat API untuk profil dan ubah password
28	26 Januari 2019	Menghubungkan dengan Android Membuat API untuk manajemen akun.
29	28 Januari 2019	Menghubungkan dengan Android Finishing menambahkan beberapa error handling
30	29 Januari 2019	Merger hosting baru dengan projek versi Web