



### **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**

### **PELAKSANAAN KERJA MAGANG**

#### **3.1 Kedudukan dan Koordinasi**

Praktek kerja magang dijalani di bawah divisi Sentra Layanan Perbankan Elektronik (SLPE) dalam bagian Urusan Pemantauan ATM dengan kedudukan *Android Developer Intern*. Proyek yang dikerjakan adalah aplikasi Peduli ATM BCA yang meliputi aplikasi Android dan aplikasi Web. Aplikasi Android berguna sebagai sarana karyawan internal BCA untuk melaporkan permasalahan yang ditemui terkait ATM BCA. Sedangkan aplikasi Web akan dipantau *realtime* oleh admin sehingga tanggapan mengenai laporan permasalahan dapat langsung dikategorikan dan diteruskan ke tim terkait.

Pengerjaan aplikasi dikoordinasikan langsung oleh Bapak Antonius Andre dan Bapak Aqua Vitawan. Bapak Aqua Vitawan menjabarkan kebutuhan aplikasi Peduli ATM BCA dan mengontrol hasil pekerjaan. Bapak Antonius Andre memantau dan menerima laporan harian perkembangan pengerjaan aplikasi, serta memberikan arahan dalam pengerjaan.

Tanggung jawab selama praktek kerja magang adalah pengerjaan aplikasi Android. Sedangkan pengerjaan aplikasi Web diserahkan pada seorang rekan kerja sesama peserta magang.

#### **3.2 Tugas yang dilakukan**

Aplikasi Peduli ATM BCA berbasis Android dibangun menggunakan aplikasi Android Studio dengan bahasa pemrograman Java. Sistem basis data

terletak di *server* dan menggunakan bahasa MySQL. Aplikasi Android dapat melakukan komunikasi dengan sistem basis data di *server* dengan menggunakan *web service* (API) yang dibangun menggunakan bahasa pemrograman PHP.

Terdapat 4 fitur utama dari aplikasi Peduli ATM BCA berbasis Android yaitu.

### 1. Sign in

Sebelum dapat membuat laporan, *user* diharuskan untuk *sign in* terlebih dahulu. *User* dapat *sign in* hanya menggunakan Google Account melalui fitur Google Sign In. Setelah sukses *sign in*, *user* diarahkan ke menu utama dan dapat memilih untuk membuat laporan, melihat *report history*, melihat informasi mengenai aplikasi, ataupun *sign out*.

### 2. Membuat Laporan

Laporan dapat dibuat dengan mengambil foto dari mesin ATM yang bermasalah dan mengisi *form*. *User* harus mengambil foto terlebih dahulu sebelum dapat mengisi *form*. Dalam *form* terdapat *field-field*, yaitu WSID (ID dari ATM), masalah, dan deskripsi masalah dan sebuah peta yang menggunakan Google Maps. Setelah mengisi *field* WSID, *user* diharuskan mengklik *button* “Check” yang akan mengecek keberadaan ATM melalui API dan menampilkan lokasinya pada peta jika WSID tersebut terdaftar. Setelah mengisi semua *field*, *user* dapat mengklik *button* “Submit” untuk mengirim laporan ke *server* melalui *web service* API.

### 3. Lihat Report History

Laporan-laporan yang pernah dikirim oleh *user* ditampilkan pada *Report History*. Laporan-laporan ini diambil dari *server* melalui *web service* API. Dengan menggunakan *Card View* dan *PullToRefreshList*, *user* dapat me-

*refresh* dengan menarik bagian atas aplikasi. Dalam *list*, tiap laporan ditampilkan dalam *Card View* yang berisi tanggal laporan, WSID, masalah, dan status. Jika *user* mengklik salah satu laporan, ditampilkan detail laporan yang meliputi tanggal laporan, WSID, masalah, deskripsi, status, foto, dan peta lokasi ATM (menggunakan Google Maps).

#### 4. *Push Notification*

Saat masalah yang telah dilaporkan *user* selesai ditangani dan status laporan sudah *closed*, *user* akan mendapat *push notification* bahwa laporannya telah selesai. *Push notification* diimplementasikan menggunakan OneSignal.

Rincian jadwal kegiatan per-minggu yang dilakukan selama proses perancangan dan pembangunan aplikasi Peduli ATM BCA berbasis Android dijelaskan pada Tabel 3.1 dan Tabel 3.2.

Tabel 3.1 Uraian pekerjaan per-minggu

Minggu Ke -	Uraian Pekerjaan
1	Perancangan navigasi antar <i>activity</i> dan <i>fragment</i> Implementasi fitur Google Sign In Implementasi fitur kamera dan upload foto Implementasi fitur Google Map
2	Menghubungkan aplikasi Android dengan API Memperbaiki fitur Google Map Implementasi <i>download</i> dan menampilkan gambar dari <i>server</i>
3	Implementasi <i>push notification</i> menggunakan OneSignal Mempelajari dan mengimplementasikan <i>PullToRefreshList</i> , <i>Card View</i> , dan <i>Custom Adapter</i> <i>Debugging</i>
4	Memastikan aplikasi berjalan di Android API di atas 23 Memperbaiki navigasi Desain <i>user interface</i> Presentasi bagian Urusan Pemantauan ATM

Tabel 3.2 Uraian pekerjaan per-minggu (lanjutan)

Minggu Ke -	Uraian Pekerjaan
5	<i>Handling Exception</i> <i>Debugging</i> Revisi desain <i>user interface</i> dan navigasi
6	Presentasi biro Layanan ATM <i>Testing</i> bagian Urusan Pemantauan ATM
7	<i>Testing</i> bagian Urusan Pemantauan ATM <i>Debugging</i>
8	Presentasi divisi Operasi <i>Debugging</i>
9	Penulisan laporan hasil kerja magang untuk BCA
10	Presentasi divisi IT dan Manajemen Risiko <i>Debugging</i>
11	Melengkapi berkas-berkas Serah terima aplikasi

### 3.3 Uraian Pelaksanaan Kerja Magang

Pelaksanaan kerja magang terbagi menjadi beberapa bagian pengerjaan yakni proses pelaksanaan, kendala yang ditemukan, serta solusi atas kendala yang ada. Bagian proses pelaksanaan menjelaskan alur kerja, fungsionalitas, dan antarmuka sistem.

#### 3.3.1 Proses Pelaksanaan

Proses perancangan dan pembangunan aplikasi Peduli ATM BCA berbasis Android dikerjakan dengan spesifikasi sistem sebagai berikut.

##### a. *Software*

- *Operating System* Windows 10 64-bit
- IDE Android Studio
- Sublime Text 3

- Restlet Client 2.13.2

b. *Hardware*

- Processor Intel Core i7-4500U CPU @ 1.80GHz 2.40 GHz
- RAM 8.00 GB
- Hard Disk Drive 1 TB
- NVIDIA Graphic Card GeForce GT740M
- *Smartphone* Sony Xperia C3 (Android OS 5.1.1)
- *Smartphone* Samsung Galaxy S8 (Android OS 7.0)

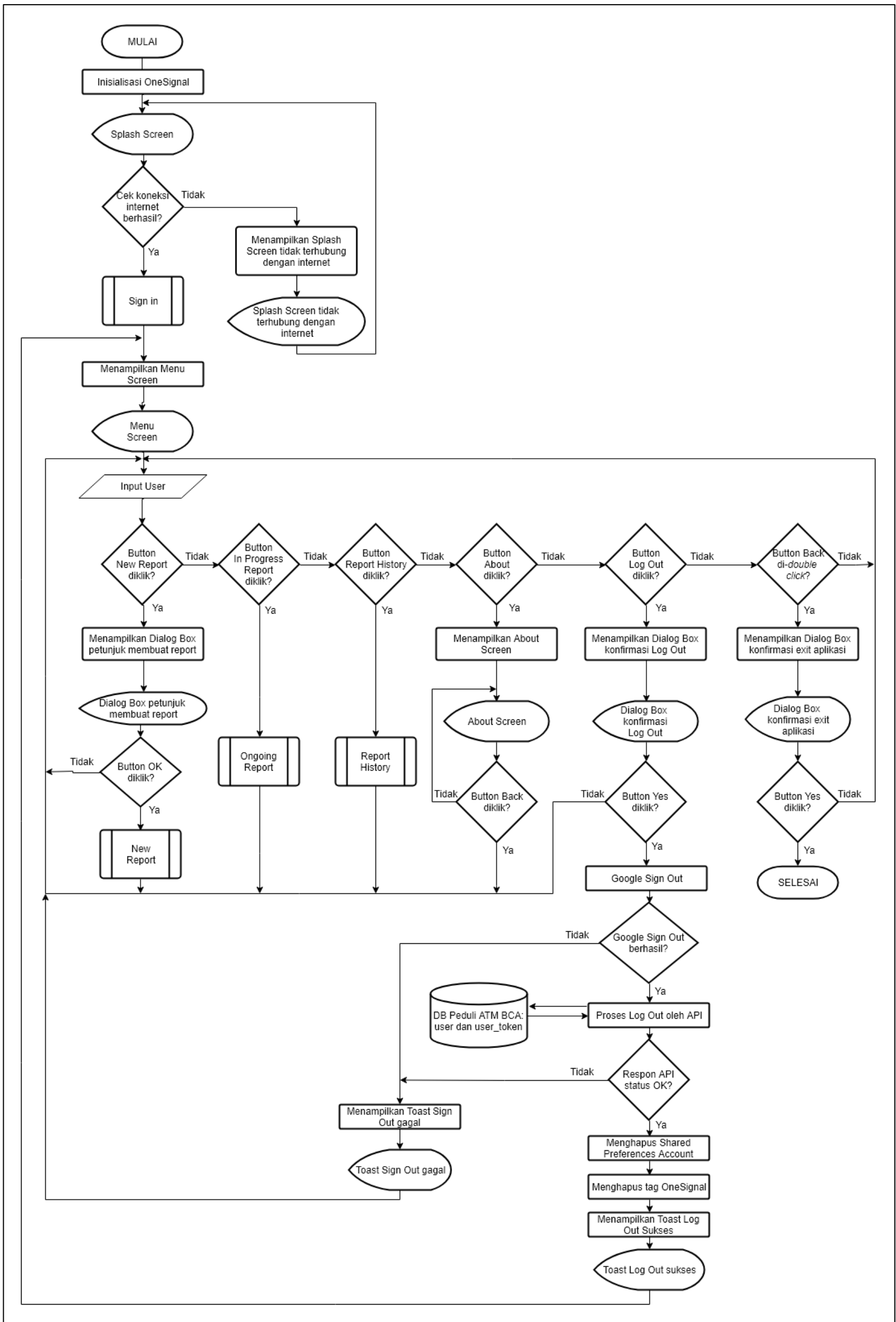
Dalam pelaksanaannya, alur kerja aplikasi *mobile* Peduli ATM BCA berbasis Android akan dijabarkan dalam *flowchart*, *Data Flow Diagram*, struktur tabel, *Entity Relationship Diagram* (ERD), *Web Service* (API), dan rancangan antarmuka. Implementasi pengembangan akan dijabarkan dalam implementasi sistem.

**A. Flowchart**

**A.1 Flowchart Umum**

Alur proses aplikasi Peduli ATM BCA berbasis Android dapat dijabarkan dengan *flowchart* seperti pada Gambar 3.1. Pada saat aplikasi dijalankan, OneSignal akan diinisialisasi lalu ditampilkan *Splash Screen*. Pada *Splash Screen* akan diperiksa apakah perangkat terhubung dengan koneksi internet. Jika perangkat terhubung dengan koneksi internet, dilanjutkan ke proses *Sign in*.

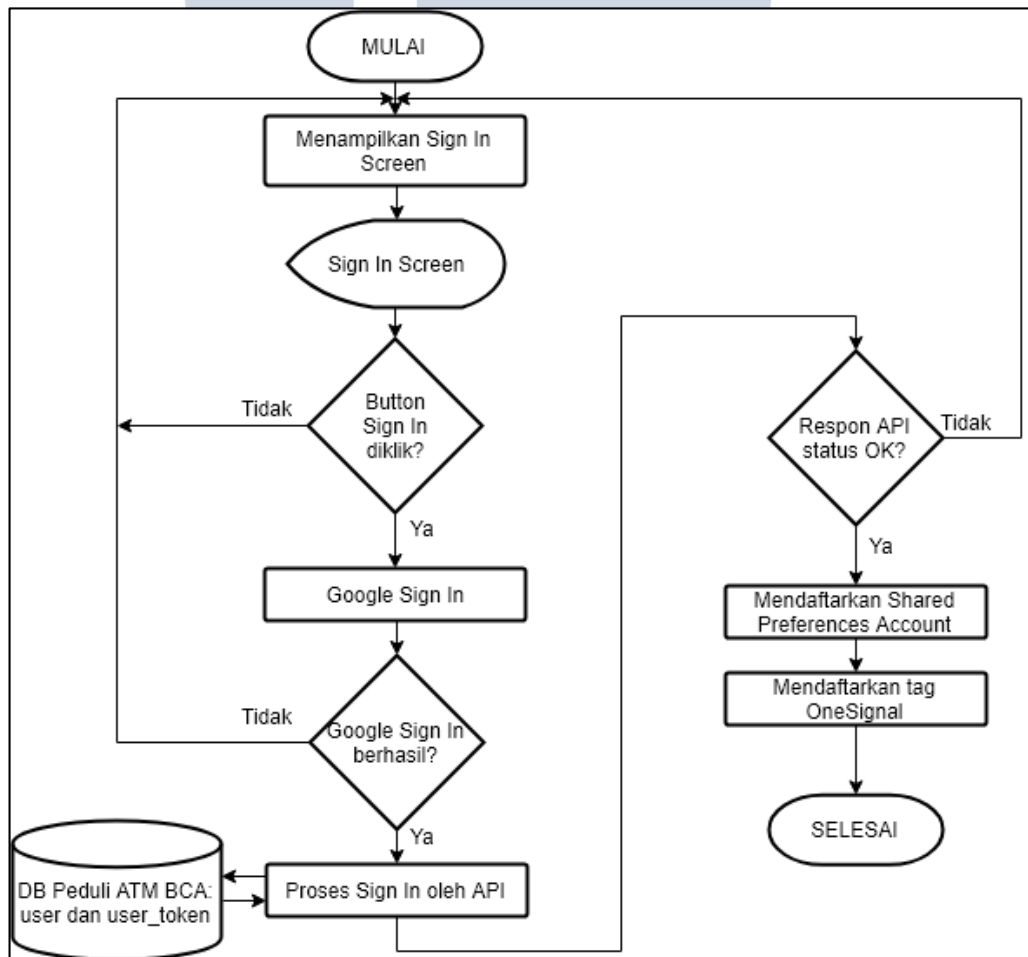
Setelah *sign in* berhasil, ditampilkan *Menu Screen* yang memiliki 5 buah *button* yaitu *button New Report* untuk membuat laporan baru, *button In Progress Report* untuk melihat *report* dengan status *open*, *button Report History* untuk melihat seluruh *report* yang pernah dibuat, *button About* untuk melihat informasi aplikasi, dan *button Log Out* untuk *log out*.



Gambar 3.1 Flowchart Aplikasi Peduli ATM BCA

## A.2 Flowchart Sign In

Gambar 3.2 menjelaskan alur proses *sign in* pada aplikasi Peduli ATM BCA berbasis Android. Proses Sign In dilakukan menggunakan Google Account dengan bantuan API Google Sign-In. Setelah proses *sign in* melalui Google Sign-In berhasil, selanjutnya adalah proses *sign in* dengan API Peduli ATM BCA untuk merekam informasi akun bagi *user* yang baru pertama menggunakan dan mencatat Google *token*, IP *address*, waktu *sign in* dari *user*. Jika respon status dari API adalah *OK*, dilanjutkan dengan mendaftarkan Shared Preferences Account dan tag OneSignal agar notifikasi dapat diterima di perangkat tersebut.

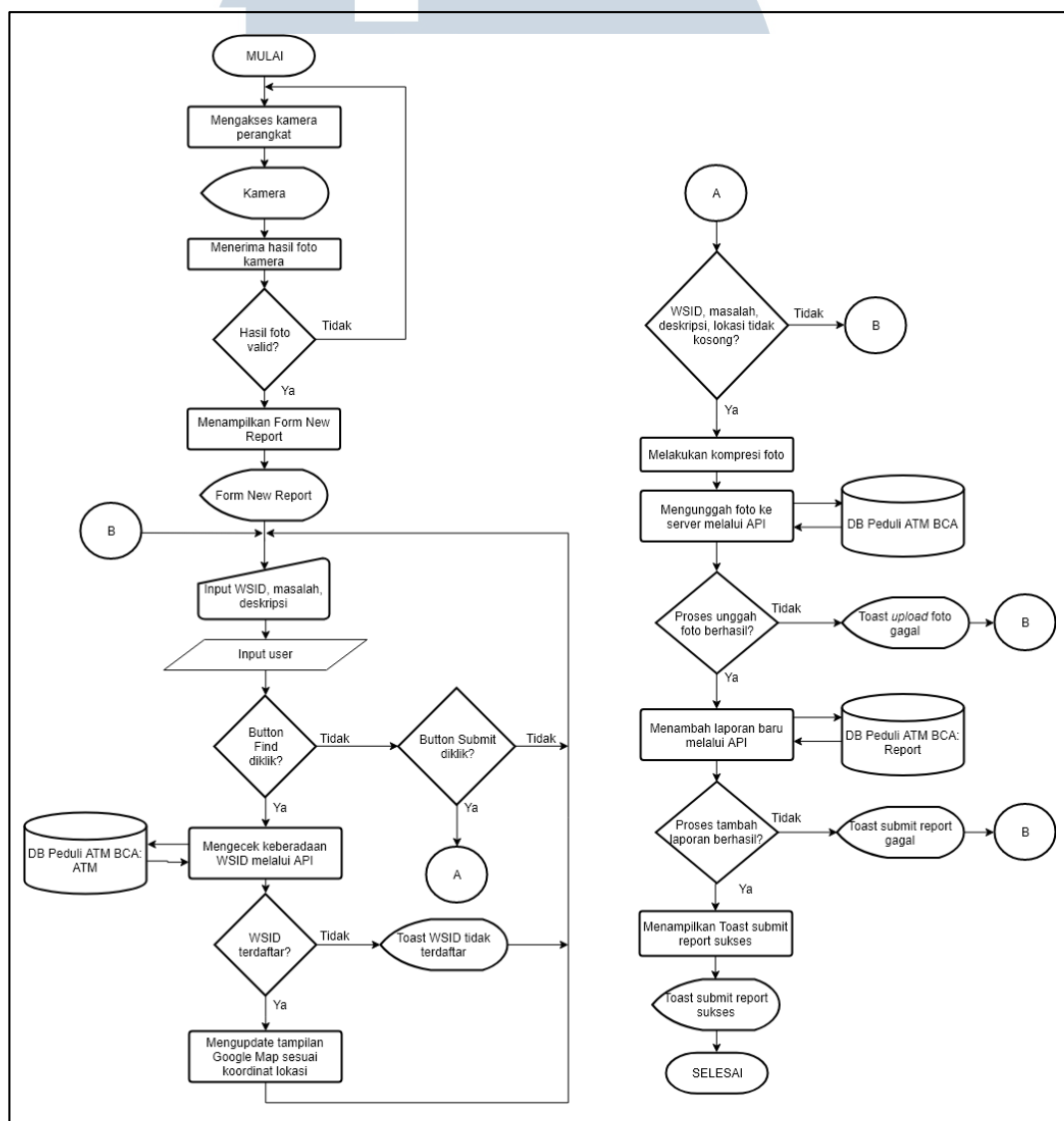


Gambar 3.2 Flowchart Sign In



### A.3 Flowchart New Report

Alur proses membuat laporan baru dijelaskan pada Gambar 3.3. *User* diarahkan ke kamera untuk mengambil foto. Setelah foto berhasil diambil, maka akan diteruskan ke tampilan *form*. Aplikasi akan menerima *input* WSID, masalah, dan deskripsi dari *user*. Jika *button Find* ditekan, WSID yang di-*input* akan diperiksa melalui API ke *database*. Jika WSID terdaftar dalam *database*, Google Map akan menampilkan lokasi ATM dengan WSID tersebut pada peta dunia.



Gambar 3.3 Flowchart New Report

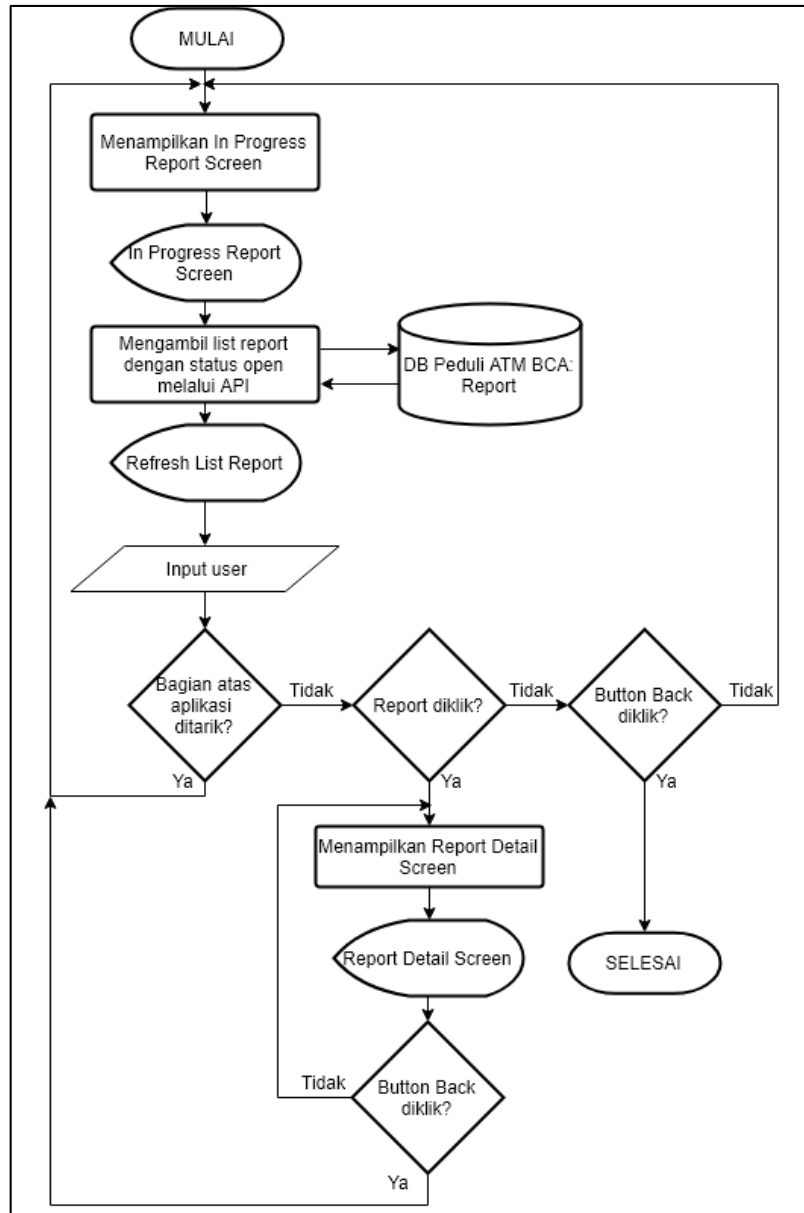
*Field* WSID, masalah, deskripsi, dan lokasi dicek sudah terisi atau tidak saat *user* menekan *button Submit*. Jika semua *field* telah terisi dan WSID *valid*, proses kompresi foto dan pengunggahan foto ke *server* melalui API dilakukan. Jika foto berhasil diunggah, dimulai proses submit laporan melalui API untuk meng-*update* basis data. Jika proses *submit* berhasil, ditampilkan Toast bahwa laporan sukses dikirim. Jika proses *submit* gagal, ditampilkan Toast bahwa laporan gagal dikirim.

#### **A.4 Flowchart In Progress Report**

Gambar 3.4 menjelaskan alur proses melihat In Progress Report. Dalam menu In Progress Report, *user* dapat melihat sebuah *list* yang berisi laporan-laporan yang pernah dibuat dan masih dalam status open.

Dengan bantuan API, data laporan yang telah dikirim dan masih dalam status diproses akan ditampilkan dalam bentuk *list* yang menampilkan field WSID, lokasi, masalah, waktu laporan dibuat, dan status laporan dalam bentuk *icon* jam pasir.

Menggunakan *library PullToRefreshList*, *list* pada In Progress Report dapat menerima *gesture* tarik dari *user*. Jika melakukan *gesture* tarik pada bagian atas aplikasi, *list* di-*refresh* dan mengulang pengambilan data melalui API. Jika salah satu laporan dalam *list* di-*tap*, diarahkan ke tampilan Report Detail yang berisi informasi lebih rinci. *Field* waktu laporan dibuat, WSID, lokasi, masalah, dan deskripsi ditampilkan. Foto mesin ATM yang diunggah juga ditampilkan dalam *ImageView*. Lokasi WSID sesuai koordinat dalam *database* ditampilkan dalam peta dengan bantuan Google Map.

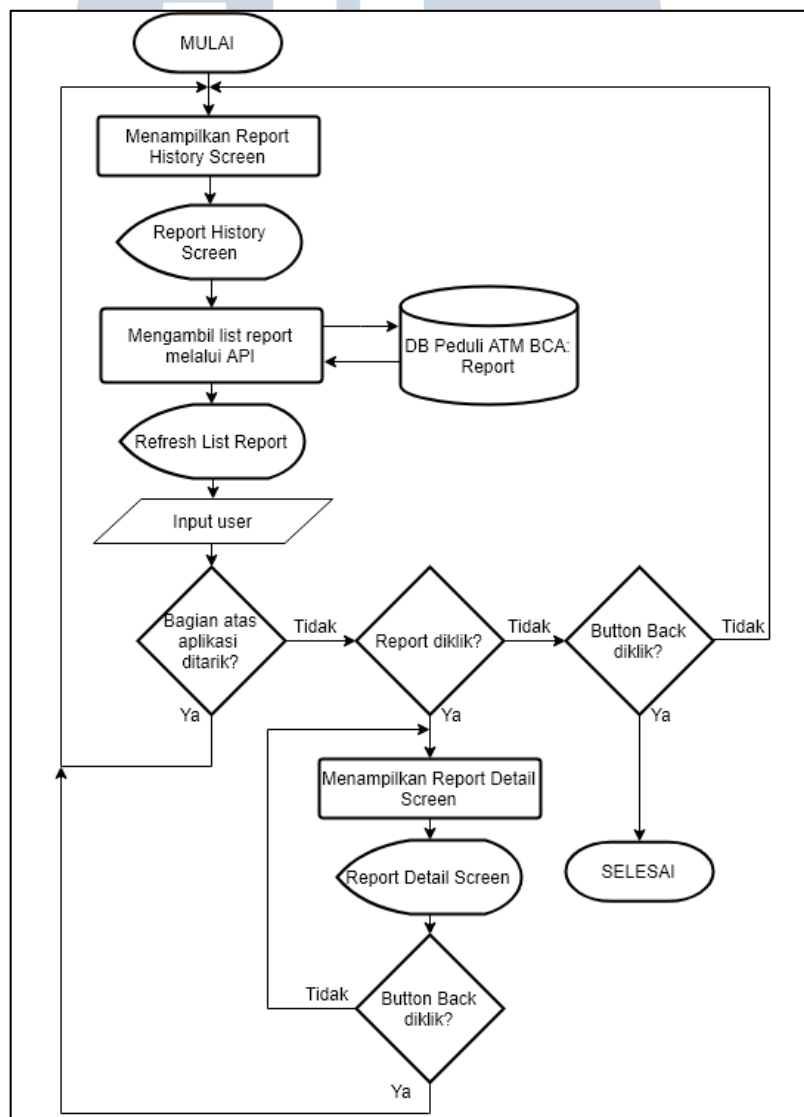


Gambar 3.4 Flowchart In Progress Report

### A.5 Flowchart Report History

Gambar 3.5 menjelaskan alur proses melihat Report History. Report History berisi sebuah *list* yang berisi semua laporan yang pernah dibuat. Dengan bantuan API, data laporan yang telah dikirim dan masih dalam status diproses akan ditampilkan dalam bentuk *list* yang menampilkan field WSID, lokasi, masalah, waktu laporan dibuat, dan status laporan dalam bentuk *icon* jam pasir.

Menggunakan *library PullToRefreshList*, *list* pada In Progress Report dapat menerima *gesture* tarik dari *user*. Jika melakukan *gesture* tarik pada bagian atas aplikasi, *list* di-*refresh* dan mengulang pengambilan data melalui API. Jika salah satu laporan dalam *list* di-*tap*, diarahkan ke tampilan Report Detail yang berisi informasi lebih rinci. *Field* waktu laporan dibuat, WSID, lokasi, masalah, dan deskripsi ditampilkan. Foto mesin ATM yang diunggah juga ditampilkan dalam *ImageView*. Lokasi WSID sesuai koordinat dalam *database* ditampilkan dalam peta dengan bantuan Google Map.

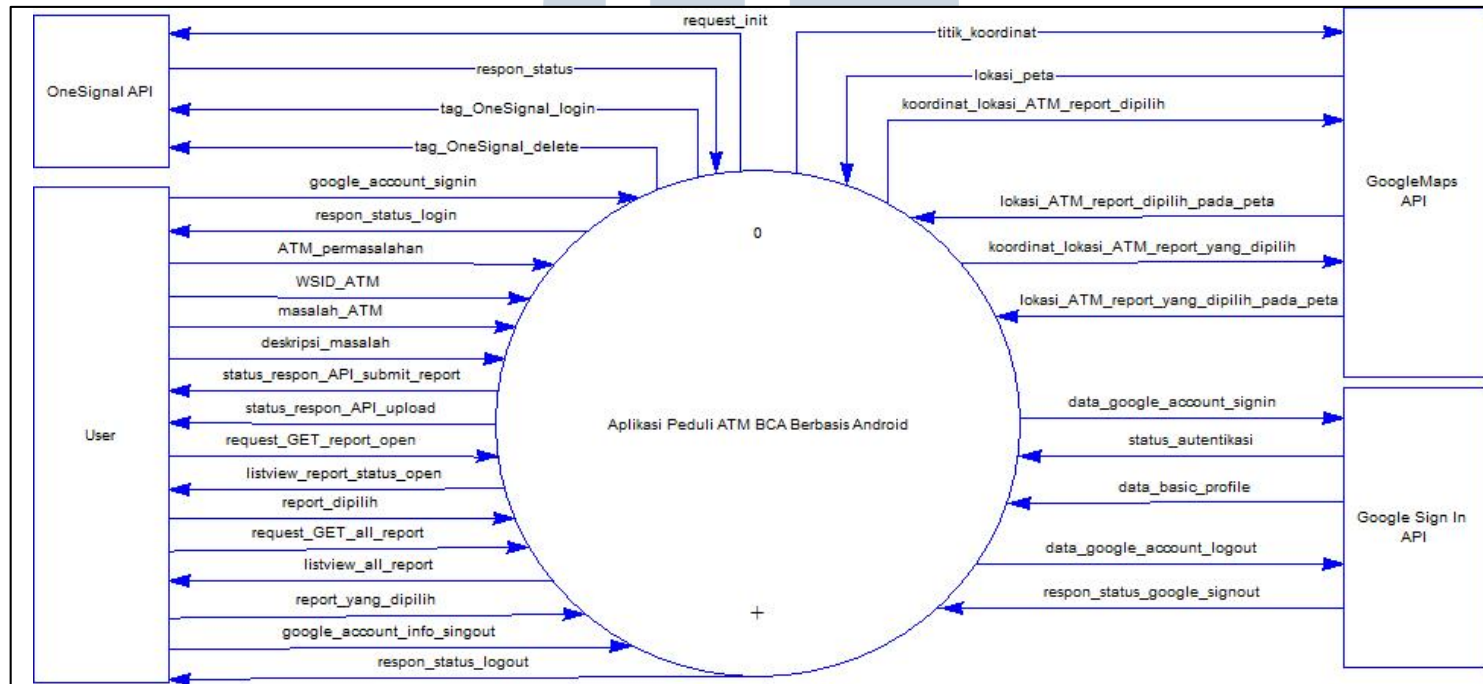


Gambar 3.5 Flowchart Report History

## B. Data Flow Diagram (DFD)

### B.1 Context Diagram

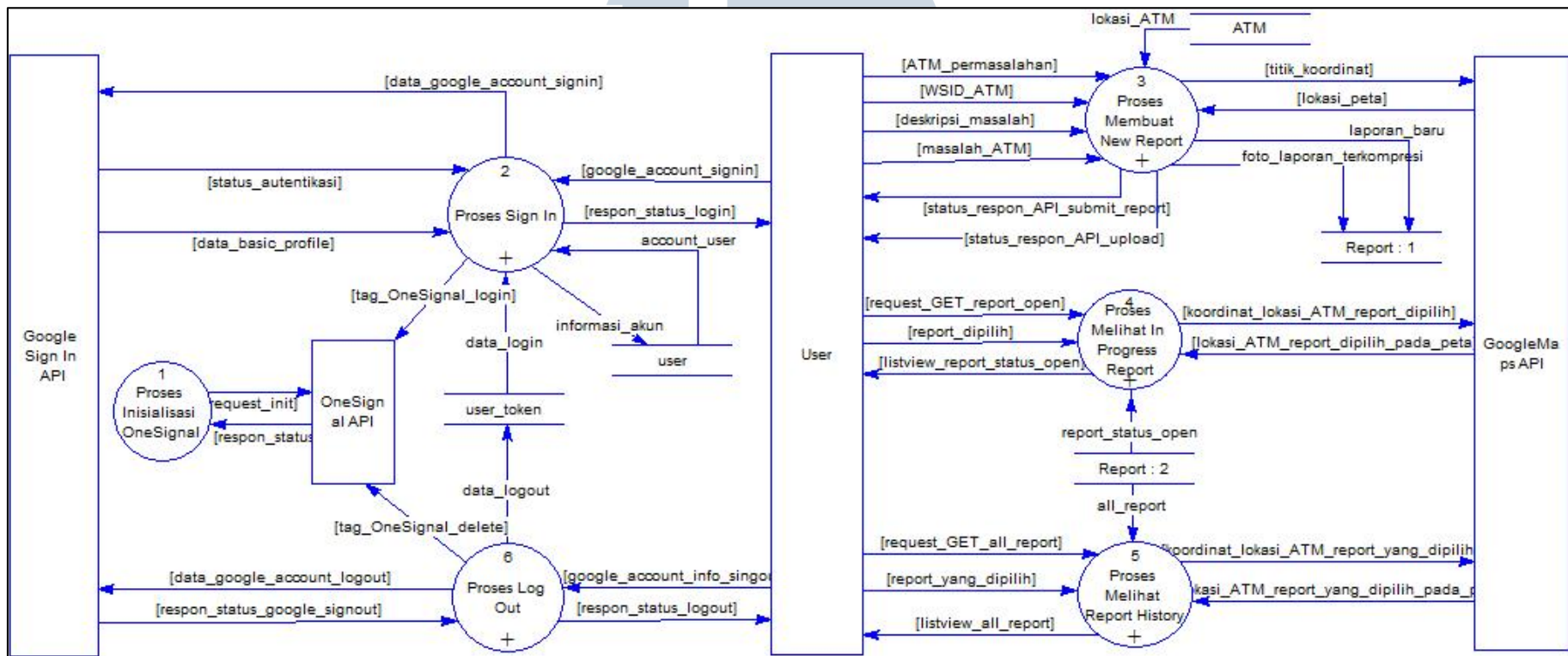
Gambar 3.6 merupakan Context Diagram dari Aplikasi Peduli ATM BCA Berbasis Android. Context Diagram ini menjelaskan aliran data dari keempat entitas pada sistem yaitu *User*, OneSignal API, Google Sign In API, dan GoogleMaps API.



Gambar 3.6 DFD Context Diagram

## B.2 Diagram Level 1

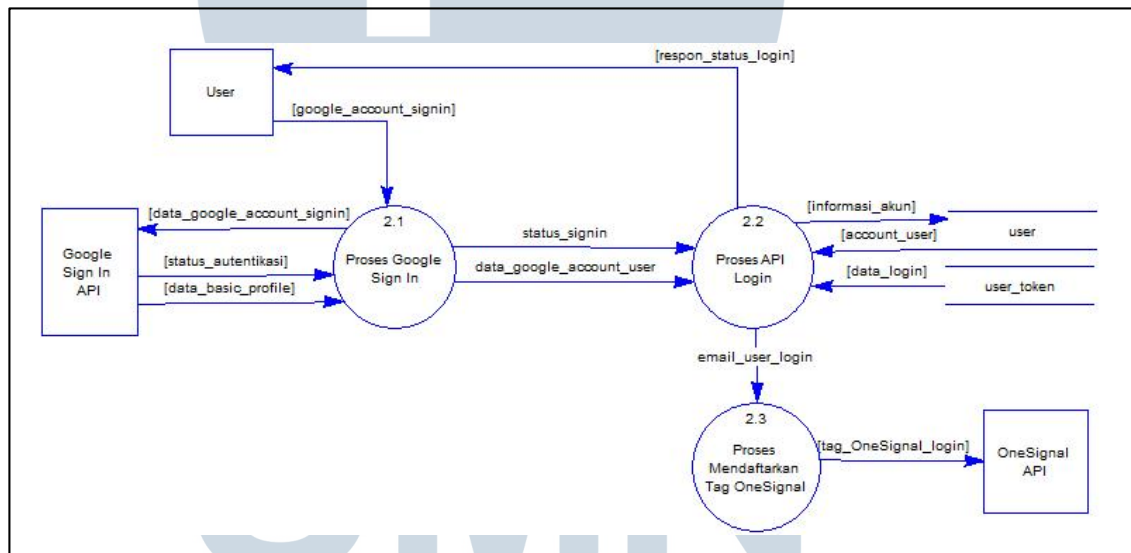
Gambar 3.7 menjabarkan DFD Level 1 yang terdiri dari 6 proses, yaitu proses inialisasi OneSignal, proses *sign in*, proses membuat *report* baru, proses melihat *in progress report*, proses melihat *report history*, dan proses *logout*.



Gambar 3.7 DFD Level 1

### B.3 Diagram Level 2

Proses Sign In terbagi menjadi 3 sub-proses seperti pada Gambar 3.8. Proses Google Sign In menerima Google Account *user* untuk diautentikasi melalui Google Sign In API. Google Sign In API mengembalikan *data\_basic\_profile user* yang berisi *firstname, lastname, email, dan gtoken*. Setelah proses Google Sign In berhasil, dilakukan proses *login* melalui API Peduli ATM BCA untuk meng-*update database* tabel *user* dan *user\_token*. Kemudian *email user* akan didaftarkan sebagai *tag* OneSignal melalui OneSignal API sehingga *user* dapat menerima *push notification* mengenai *report* yang pernah dikirim.



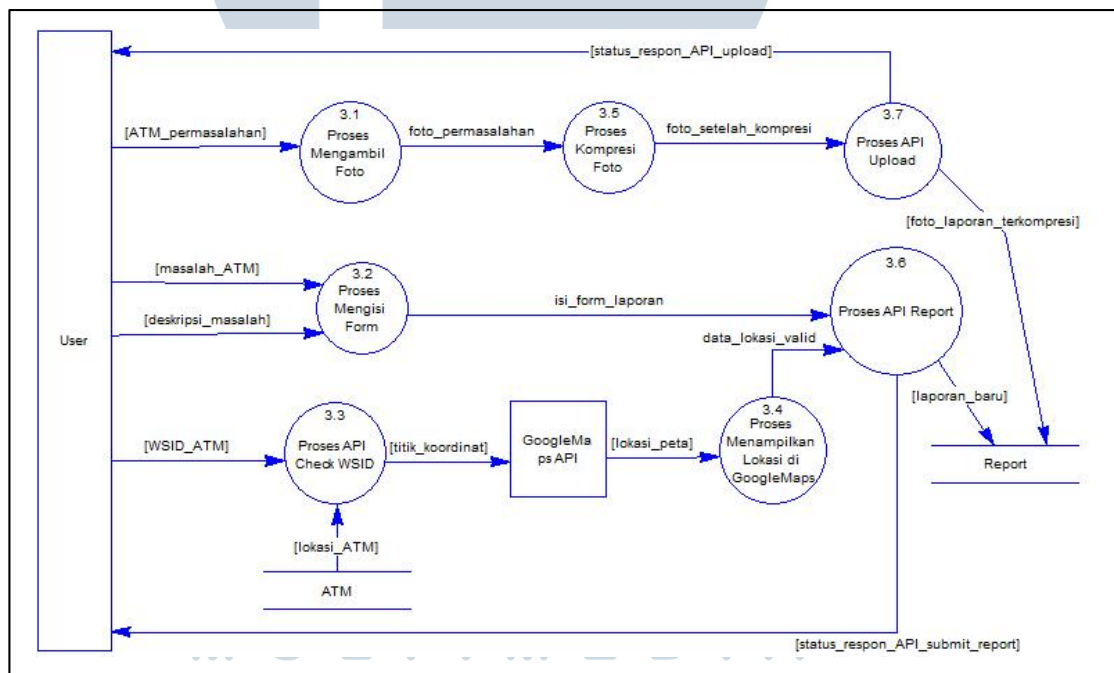
Gambar 3.8 DFD Level 2 Proses Sign In

Proses membuat *new report* dijabarkan pada Gambar 3.9. Proses membuat *new report* memiliki 7 sub-proses, yaitu proses mengambil foto, proses mengisi *form*, proses mengecek WSID, proses menampilkan lokasi di GoogleMaps, proses kompresi foto, proses API *report*, dan proses API *upload*.

Proses mengambil foto dilakukan oleh *user* menggunakan kamera. Foto tersebut melalui proses kompresi foto untuk mengurangi ukurannya menggunakan

*library* Compressor. Setelah itu dilakukan proses *upload* melalui API dan disimpan dalam *database*.

Proses mengisi *form* diisi oleh *user* dengan data permasalahan yang ditemui. Proses API Check WSID menerima input WSID\_ATM dari *user* untuk diperiksa apakah WSID tersebut ada dan lokasinya dimana. API Check WSID akan *query database* untuk mendapatkan lokasi ATM berupa titik koordinat *longitude* dan *latitude*. Titik koordinat ini dikirim melalui GoogleMaps API sehingga dapat ditampilkan pada peta dunia di proses menampilkan lokasi di GoogleMaps. Aliran data isi\_form\_laporan dan data\_lokasi\_valid dikirim melalui proses API Report untuk di-*insert* ke dalam *database*.

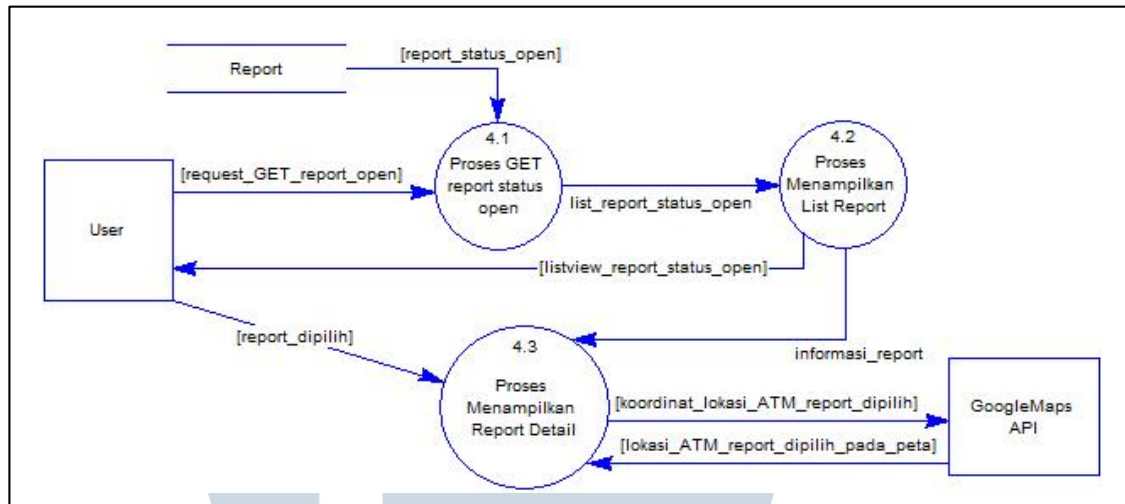


Gambar 3.9 DFD Level 2 Proses Membuat New Report

Proses melihat *in progress report* diuraikan pada Gambar 3.10 dengan 3 sub-proses. Proses GET *report status open* menerima request\_GET\_report\_open dari *user* saat *user* menekan *menu* In Progress Report. Melalui API, semua *report* dengan status *open* diambil dari *database* dan ditampilkan ke *user* dalam bentuk

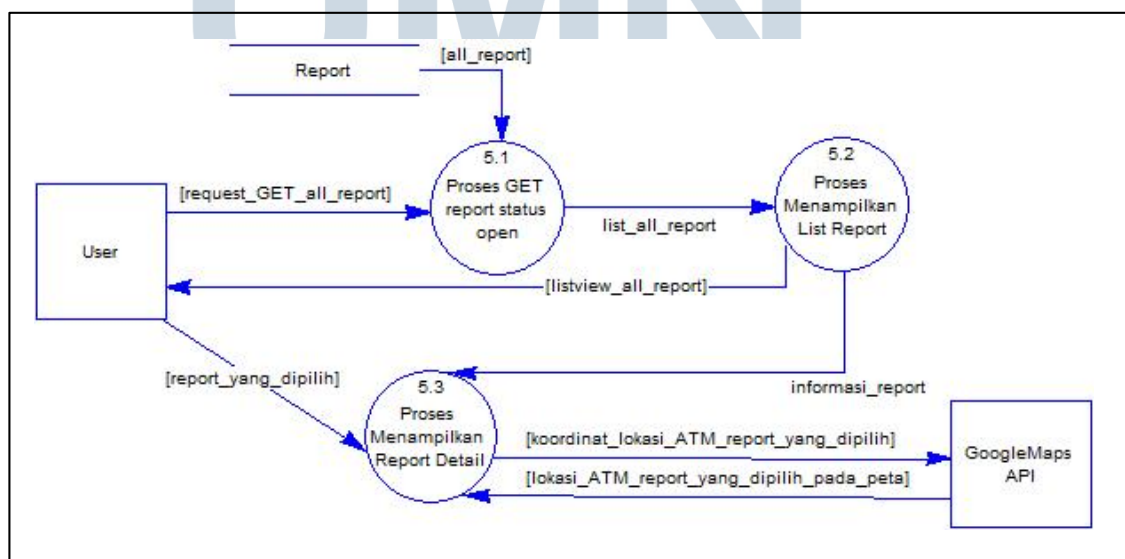


*list*. Proses menampilkan *report detail* menerima *input* *report\_dipilih* dan menampilkan Report Detail Screen yang berisi GoogleMaps yang akan menyesuaikan dengan koordinat lokasi ATM dari *report* yang dipilih.



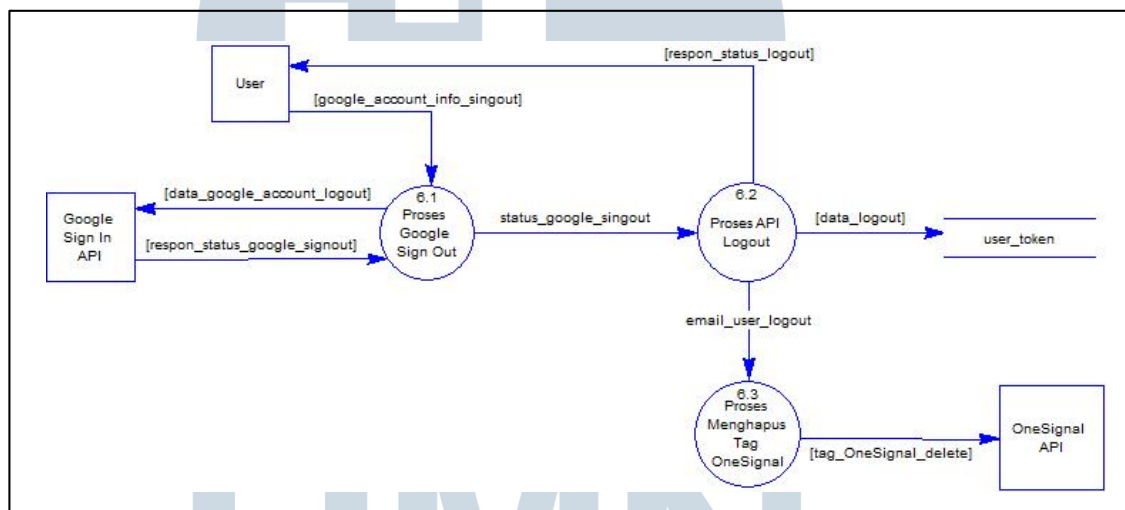
Gambar 3.10 DFD Level 2 Proses Melihat In Progress Report

Proses melihat Report History dijabarkan pada Gambar 3.11 dengan 3 sub-proses. Proses melihat Report History sama dengan proses melihat *in progress report* pada Gambar 3.10 dengan perbedaan *report* yang diambil melalui API tidak hanya yang berstatus *open*, tetapi semua *report* yang pernah dibuat oleh *user*.



Gambar 3.11 DFD Level 2 Proses Melihat Report History

Gambar 3.12 menjabarkan DFD Level 2 Proses Log Out. Proses ini memiliki 3 sub-proses, yaitu proses Google Sign Out, proses API Logout, dan proses menghapus *tag* OneSignal. Proses Google Sign Out dilakukan oleh Google Sign In API. Jika Google Sign Out berhasil, proses API Logout dilakukan untuk meng-update *data\_logout* pada tabel *user\_token*. Pada proses menghapus *tag* OneSignal, *email user* yang merupakan *tag* OneSignal akan dikirim ke OneSignal API untuk dihapus sehingga *user* tidak akan menerima *push notification* untuk *account* tersebut.



Gambar 3.12 DFD Level 2 Proses Log Out

## C. Struktur Tabel

### C.1 Tabel User

Tabel User merupakan tabel yang menyimpan informasi *account user*. Informasi *account user* disimpan pada saat pertama kali *user* melakukan Google Sign In pada Aplikasi Peduli ATM BCA berbasis Android. Struktur tabel User dijelaskan pada Tabel 3.3.

Tabel 3.3 Tabel *User*

Nama Kolom	Tipe Data	Keterangan
id	int(11)	<i>Primary key</i> tabel <i>User</i>
ip	char(32)	<i>IP Address user</i> saat registrasi
register	timestamp	Waktu saat registrasi
username	char(100)	Alamat <i>email user</i>
firstname	char(30)	Nama depan <i>user</i>
lastname	char(30)	Nama belakang <i>user</i>
password	char(32)	MD5 dari <i>password</i> digabung dengan <i>salt</i>
salt	char(30)	Nama belakang <i>user</i>
type	varchar(50)	Tipe <i>login user</i>

### C.2 Tabel *User\_Token*

Tabel *User\_Token* menyimpan *token* yang diperoleh saat *user* melakukan *login*. Tabel *User\_Token* dijabarkan pada Tabel 3.4.

Tabel 3.4 Tabel *User\_Token*

Nama Kolom	Tipe Data	Keterangan
id	int(11)	<i>Primary key</i> tabel <i>User_Token</i>
token	char(32)	<i>Token</i> yang diperoleh saat <i>login</i>
user_id	int(11)	<i>Foreign key</i> yang merujuk pada tabel <i>user</i>
ip	char(15)	<i>IP address</i> yang digunakan <i>user</i>
login	timestamp	Waktu saat <i>login</i>
logout	timestamp	Waktu saat <i>logout</i>

### C.3 Tabel *Report*

Tabel *Report* berisi data laporan-laporan yang dibuat oleh *user*. Struktur tabel *Report* dijabarkan pada Tabel 3.5.

Tabel 3.5 Tabel Report

Nama Kolom	Tipe Data	Keterangan
id	int(11)	<i>Primary key</i> tabel Report
ip	char(15)	IP <i>address user</i> saat mengirim laporan
user_id	int(4)	<i>Foreign key</i> yang merujuk ke tabel <i>User</i>
wsid	char(4)	WSID ATM
date	timestamp	Waktu saat laporan dikirim
close	timestamp	Waktu saat laporan menjadi status <i>closed</i>
problem	varchar(100)	Masalah yang dilaporkan <i>user</i>
category_id	int(4)	id kategori masalah
image	varchar(22)	Nama <i>file</i> foto yang di- <i>upload</i>
description	text	Deskripsi masalah
status	tinyint(1)	Status laporan

#### C.4 Tabel ATM

Tabel ATM berisi data ATM. Tabel 3.6 menjelaskan struktur tabel ATM.

Tabel 3.6 Tabel ATM

Nama Kolom	Tipe Data	Keterangan
wsid	char(4)	<i>Primary Key</i> tabel ATM
location	varchar(1000)	Lokasi ATM
kww_id	int(4)	<i>Foreign key</i> yang merujuk ke tabel KWW
vendor_id	int(4)	<i>Foreign key</i> yang merujuk ke tabel Vendor
latitude	char(18)	Nilai <i>latitude</i> lokasi ATM
longitude	char(18)	Nilai <i>longitude</i> lokasi ATM

#### C.5 Tabel Status

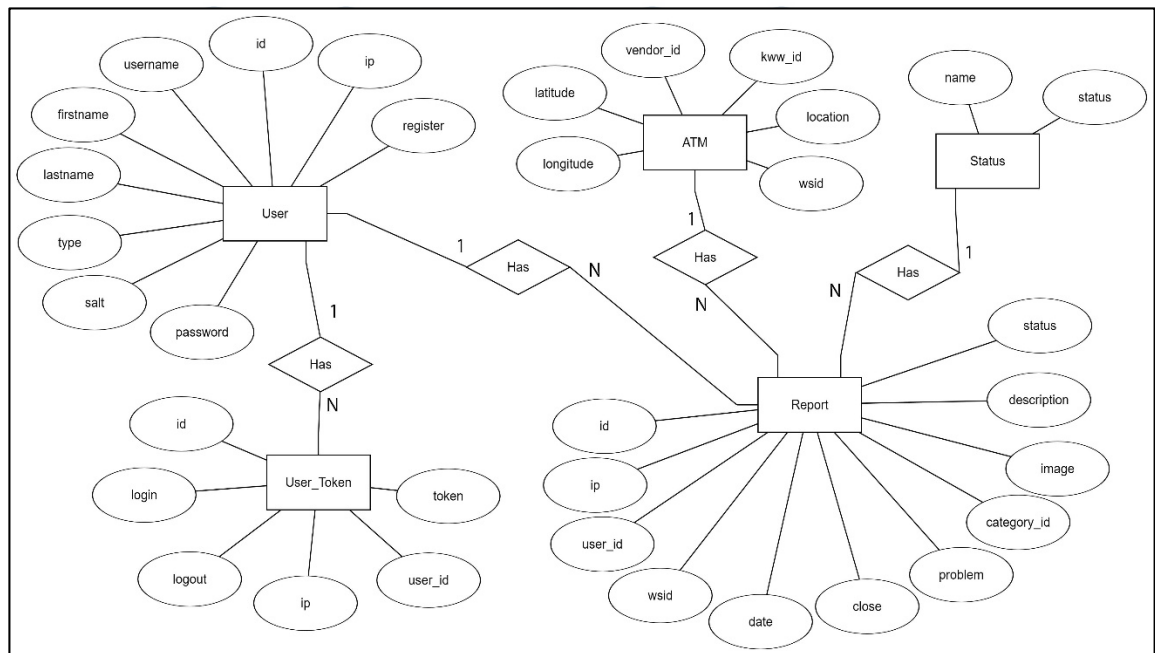
Tabel Status menyimpan informasi status laporan. Struktur tabel Status dijelaskan pada Tabel 3.7.

Tabel 3.7 Tabel Status

Nama Kolom	Tipe Data	Keterangan
status	tinyint(1)	Primary key tabel Status
name	char(100)	Deskripsi status

#### D. Entity Relationship Diagram (ERD)

Gambar 3.13 menjabarkan Entity Relationship Diagram (ERD) yang menjelaskan relasi antar tabel dalam *database* Peduli ATM BCA. Tabel User memiliki relasi *one-to-many* dengan tabel User\_Token karena tabel User\_Token dapat menampung beberapa informasi *login* dan *logout user*. Tabel User juga memiliki relasi *one-to-many* dengan tabel Report karena seorang *user* dapat membuat lebih dari satu *report*. Tabel ATM memiliki relasi *one-to-many* dengan tabel Report karena suatu ATM dapat berada di beberapa *report*. Tabel status dan tabel Report juga memiliki relasi *one-to-many* karena masing-masing *report* memiliki status masing-masing.

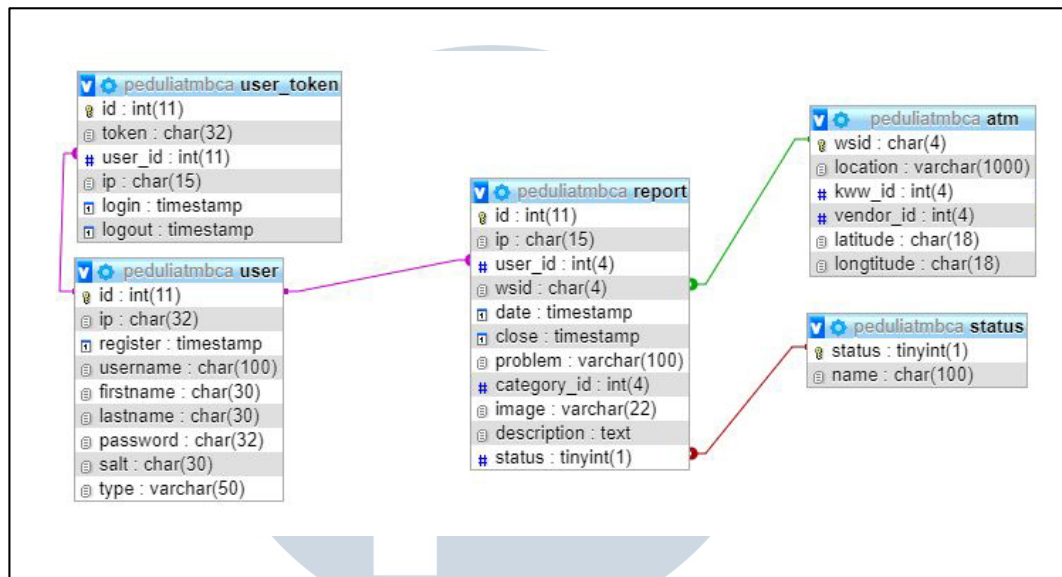


Gambar 3.13 Entity Relationship Diagram

## E. Database Schema

Skema *database* aplikasi Peduli ATM BCA dijelaskan pada Gambar 3.14.

Tabel Report berelasi dengan tabel User, tabel ATM, dan tabel Status. Tabel User berelasi dengan tabel User\_Token.



Gambar 3.14 Database Schema

## F. Web Service (API)

Aplikasi Peduli ATM BCA menggunakan 7 API yaitu Login, Logout, Report, Upload, Check WSID, Load Open, dan Load History. Setiap API dapat diakses melalui URI [https://\[host\]/\[nama\\_API\]](https://[host]/[nama_API]).

### F.1 Login

API Login digunakan baik untuk meregistrasi *user* baru maupun *login user* yang telah registrasi. API ini menerima parameter seperti diuraikan pada Tabel 3.8. API ini diakses melalui [https://\[host\]/login](https://[host]/login).

Tabel 3.8 Tabel Parameter API Login

Parameter	Keterangan
<i>username</i>	Email Google <i>account user</i>
<i>firstname</i>	Nama depan Google <i>account</i>
<i>lastname</i>	Nama belakang Google <i>account</i>
<i>gplus</i>	Bernilai "true" atau 1
<i>gtoken</i>	<i>Token Google account</i>

API Login menghasilkan *output* berupa JSON seperti pada Gambar 3.15. *Output* yang dihasilkan berbeda pada saat registrasi atau *login* sukses, registrasi gagal, atau *login* gagal karena verifikasi *gtoken* gagal.

```

//registrasi atau login sukses
{
  "success" : "true",
  "data" :
  {
    "status" : "OK",
    "uid" : "uid",
    "token" : "token"
  }
}

//registrasi gagal
{
  "success" : "false",
  "data" :
  {
    "status" : "FAIL",
    "message": "unable to register user"
  }
}

//login gagal, verifikasi gtoken gagal
{
  "success" : "false",
  "data" :
  {
    "status" : "FAIL",
    "message": "invalid gtoken"
  }
}

```

Gambar 3.15 Hasil Proses API Login

## F.2 Logout

API Logout menerima parameter seperti diuraikan pada Tabel 3.9. API ini diakses melalui [https://\[host\]/logout](https://[host]/logout).

Tabel 3.9 Tabel Parameter API Logout

Parameter	Keterangan
uid	<i>User id</i>
token	<i>Token yang diperoleh saat sign in</i>

API Logout menghasilkan *output* berupa JSON seperti pada Gambar 3.16.

*Output* yang dihasilkan berbeda pada saat *logout* sukses dan *logout* gagal.

```
//logout sukses
{
  "success" : "true",
  "data" :
  {
    "status" : "OK",
    "message": "logout success"
  }
}

//logout gagal
{
  "success" : "false",
  "data" :
  {
    "status" : "FAIL",
    "message": "unable to logout"
  }
}
```

Gambar 3.16 Hasil Proses API Logout

## F.3 Report

API Report digunakan untuk mengirimkan informasi *report* yang telah diisi oleh *user*. API ini menerima parameter seperti diuraikan pada Tabel 3.10. API ini diakses melalui [https://\[host\]/report](https://[host]/report).



Tabel 3.10 Tabel Parameter API Report

Parameter	Keterangan
uid	<i>User id</i>
token	<i>Token yang diperoleh saat sign in</i>
wsid	WSID ATM yang dilaporkan
problem	Masalah yang dilaporkan
uploaded_file	Nama file foto yang diunggah
description	Deskripsi masalah lebih rinci

API Report menghasilkan *output* berupa JSON seperti pada Gambar 3.17.

*Output* yang dihasilkan berbeda pada saat pengiriman *report* sukses, pengiriman *report* gagal, dan autentikasi uid dan *token* gagal.

```

//report sukses
{
  "success" : "true",
  "data" :
  {
    "status" : "OK",
    "message": "reported succesfully"
  }
}

//report gagal
{
  "success" : "false",
  "data" :
  {
    "status" : "FAIL",
    "message": "unable to report"
  }
}

//autentikasi uid dan token gagal
{
  "success" : "false",
  "data" :
  {
    "status" : "FAIL",
    "message": "invalid uid or token, please logout and re-login"
  }
}

```

Gambar 3.17 Hasil Proses API Report

## F.4 Upload

API Upload digunakan untuk mengunggah foto dengan *format* gif atau jpg atau png yang telah diambil *user* ke server. API Upload menerima parameter seperti diuraikan pada Tabel 3.11. API ini diakses melalui [https://\[host\]/upload](https://[host]/upload).

Tabel 3.11 Tabel Parameter API Upload

Parameter	Keterangan
uid	<i>User id</i>
token	<i>Token yang diperoleh saat sign in</i>
uploaded_file	Nama file foto yang diunggah

API Upload menghasilkan *output* berupa JSON seperti pada Gambar 3.18. *Output* yang dihasilkan berbeda pada saat proses *upload* sukses, *upload* gagal, dan autentikasi uid dan *token* gagal.

```
//upload sukses
{
  "success" : "true",
  "data" :
  {
    "status" : "OK",
    "file": "filename"
  }
}

//upload gagal
{
  "success" : "false",
  "data" :
  {
    "status" : "FAIL",
    "message": "unable to upload file"
  }
}

//autentikasi uid dan token gagal
{
  "success" : "false",
  "data" :
  {
    "status" : "FAIL",
    "message": "invalid uid or token, please logout and re-login"
  }
}
```

Gambar 3.18 Hasil Proses API Upload

## F.5 Check WSID

API Check WSID digunakan untuk mengecek keberadaan suatu WSID dan jika ada, mengembalikan lokasinya. API Check WSID menerima parameter seperti diuraikan pada Tabel 3.12. API ini diakses melalui [https://\[host\]/check\\_wsid](https://[host]/check_wsid).

Tabel 3.12 Tabel Parameter API Check WSID

Parameter	Keterangan
uid	User id
token	Token yang diperoleh saat <i>sign in</i>
wsid	WSID ATM yang akan dicek

API Check WSID menghasilkan *output* berupa JSON seperti pada Gambar 3.19. *Output* yang dihasilkan berbeda pada saat WSID terdaftar dalam *database*, WSID tidak terdaftar dalam *database*, dan autentikasi uid dan *token* gagal.

```
//WSID ada dalam database
{
  "success" : "true",
  "data" :
  {
    "status" : "OK",
    "atm": {
      "location" : "location",
      "latitude" : "latitude",
      "longitude" : "longitude"
    }
  }
}

//WSID tidak ada dalam database
{
  "success" : "false",
  "data" :
  {
    "status" : "FAIL",
    "message": "WSID tidak ditemukan"
  }
}

//autentikasi uid dan token gagal
{
  "success" : "false",
  "data" :
  {
    "status" : "FAIL",
    "message": "invalid uid or token, please logout and re-login"
  }
}
```

Gambar 3.19 Hasil Proses API Check WSID

## F.6 Load Open

API Load Open digunakan untuk mendapatkan data dari *report-report* dengan status *open* yang di-submit oleh *user* bersangkutan. API ini menerima parameter seperti diuraikan pada Tabel 3.13. API ini diakses melalui [https://\[host\]/load\\_open](https://[host]/load_open).

Tabel 3.13 Tabel Parameter API Load Open

Parameter	Keterangan
uid	<i>User id</i>
token	<i>Token yang diperoleh saat sign in</i>

API Check WSID menghasilkan *output* berupa JSON seperti pada Gambar 3.20. *Output* tersebut berbeda pada saat autentikasi uid dan *token* berhasil atau gagal

```
//Load Report dengan status open
{
  "success" : "true",
  "data" :
  {
    "status" : "OK",
    "count" : "count",
    "history": {
      "rid" : "rid",
      "wsid" : "wsid",
      "date" : "date",
      "problem" : "problem",
      "location" : "location",
      "latitude" : "latitude",
      "longitude" : "longitude",
      "image" : "tiny/image",
      "description" : "description",
      "status" : "status"
    }
  }
}

//autentikasi uid dan token gagal
{
  "success" : "false",
  "data" :
  {
    "status" : "FAIL",
    "message": "invalid uid or token, please logout and re-login"
  }
}
```

Gambar 3.20 Hasil Proses API Load Open

## F.7 Load History

API Load History digunakan untuk mendapatkan data semua *report* yang telah di-*submit* oleh *user* bersangkutan. API Load History menerima parameter seperti diuraikan pada Tabel 3.14. API ini diakses melalui [https://\[host\]/load\\_history](https://[host]/load_history).

Tabel 3.14 Tabel Parameter API Load History

Parameter	Keterangan
uid	<i>User id</i>
token	<i>Token yang diperoleh saat sign in</i>

API Check WSID menghasilkan *output* berupa JSON seperti pada Gambar 3.21. *Output* tersebut berbeda pada saat autentikasi uid dan *token* berhasil atau gagal

```
//Load semua Report
{
  "success" : "true",
  "data" :
  {
    "status" : "OK",
    "count" : "count",
    "history":
    [
      {
        "rid" : "rid",
        "wsid" : "wsid",
        "date" : "date",
        "problem" : "problem",
        "location" : "location",
        "latitude" : "latitude",
        "longitude" : "longitude",
        "image" : "tiny/image",
        "description" : "description",
        "status" : "status"
      }
    ]
  }
}

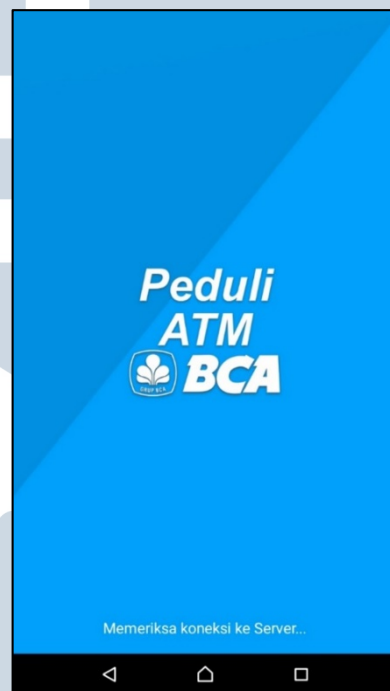
//autentikasi uid dan token gagal
{
  "success" : "false",
  "data" :
  {
    "status" : "FAIL",
    "message": "invalid uid or token, please logout and re-login"
  }
}
```

Gambar 3.21 Hasil Proses API Load History

## G. Antarmuka Sistem

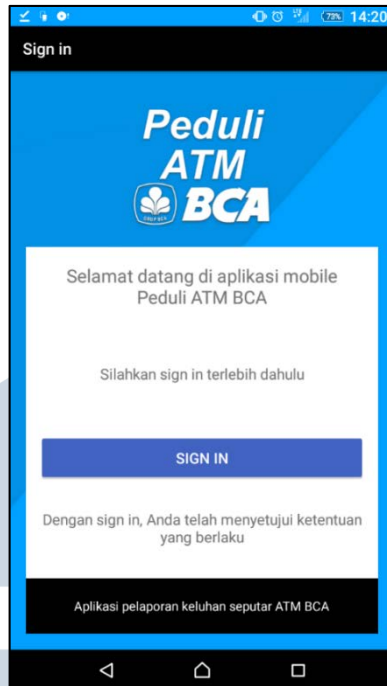
Pada bagian ini adalah penjelasan tampilan antarmuka aplikasi Peduli ATM BCA berbasis Android.

Gambar 3.22 adalah Splash Screen yang merupakan layar pertama yang dilihat oleh *user* setelah menjalankan aplikasi. Pada halaman ini, koneksi internet akan diperiksa apakah terhubung atau tidak. Jika terhubung, layar akan berpindah ke layar selanjutnya yaitu Sign In Screen.



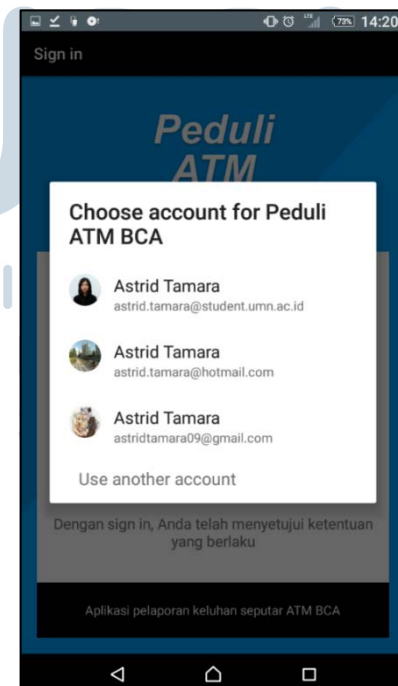
Gambar 3.22 Splash Screen

Sign In Screen merupakan halaman untuk autentikasi *account user*. Pada Sign In Screen, terdapat *button* untuk melakukan Google Sign In seperti terlihat pada Gambar 3.23.



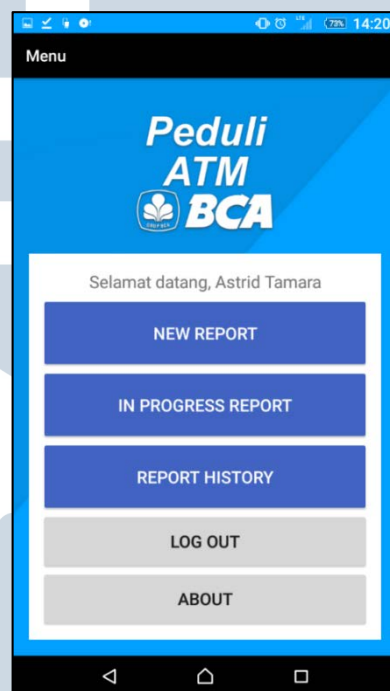
Gambar 3.23 Sign In Screen

Setelah button di-*tap*, Google Sign In API meminta *user* memilih salah satu *account* dari semua *account* yang terdaftar seperti pada Gambar 3.24. Kemudian dilanjutkan dengan proses Google Sign In dan proses *login* oleh API.



Gambar 3.24 Tampilan Pemilihan Google Account

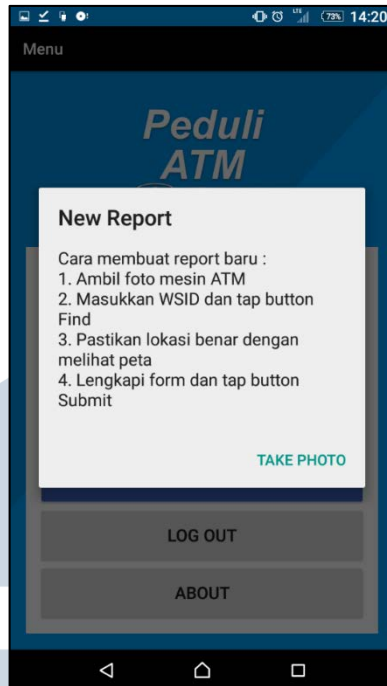
Setelah sukses Sign In, *user* akan dihadapkan dengan Menu Screen seperti terlihat pada Gambar 3.25. Menu Screen terdiri dari 5 *button*, yaitu *button* New Report untuk membuat laporan permasalahan baru, *button* In Progress Report untuk melihat *list report* yang pernah dikirim dan masih berstatus *open*, *button* Report History untuk melihat *list* semua *report* yang pernah dibuat, *button* Logout untuk melakukan proses *logout*, dan *button* About untuk melihat informasi mengenai aplikasi.



Gambar 3.25 Menu Screen

Ketika *button* New Report diklik, sebuah *dialog box* akan tampil. *Dialog box* tersebut berisi langkah-langkah pembuatan *report* baru dan sebuah *button* “Take Photo” seperti terlihat pada Gambar 3.26.





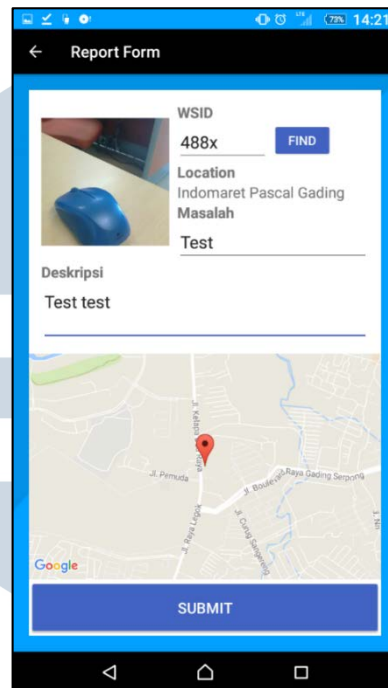
Gambar 3.26 Dialog Box New Report

Saat *user* menekan *button* “Take Photo” pada *dialog box* pada Gambar 3.26, *user* akan diarahkan ke kamera seperti pada Gambar 3.27.



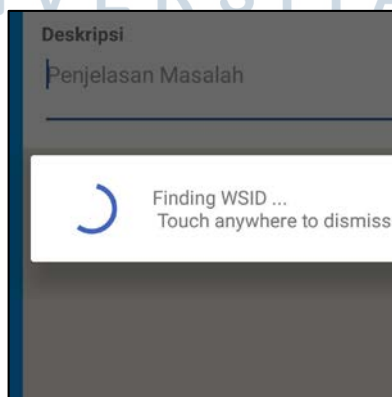
Gambar 3.27 Camera Screen

Setelah melewati proses pengambilan foto, *user* akan diarahkan ke Report Form Screen seperti pada Gambar 3.28. Pada halaman ini, *user* mengisi WSID, masalah, dan deskripsi masalah yang ditemukan. Setelah mengisi WSID, *user* harus menekan *button* “Find” untuk melakukan pengecekan WSID.



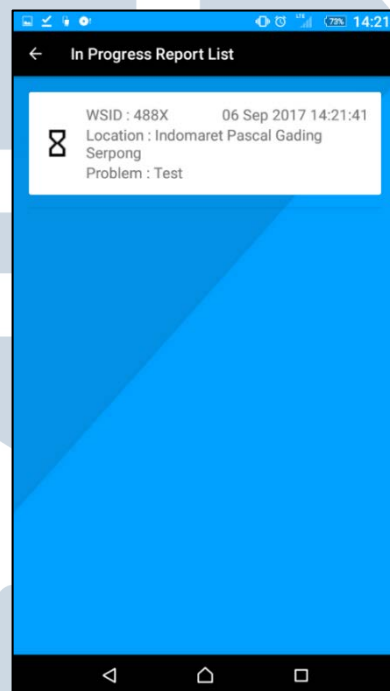
Gambar 3.28 Report Form Screen

Setelah *button* “Find” ditekan, aplikasi menampilkan *progress dialog* seperti pada Gambar 3.29 dan dilakukan pengecekan dengan API. Jika WSID tersebut *valid*, lokasinya ditampilkan pada *TextView location* dan peta.



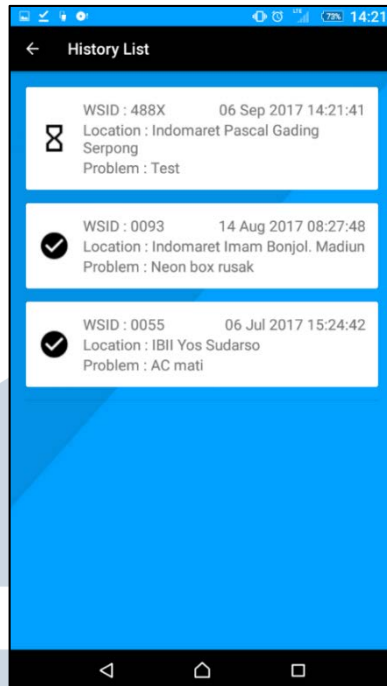
Gambar 3.29 Progress Dialog Finding WSID

Gambar 3.30 memperlihatkan In Progress Report Screen. Pada halaman ini, *user* dapat melihat *list report* yang masih berstatus *open*. *Report* ditampilkan dalam *CardView* yang berisi *icon* jam pasir, *WSID*, *location*, *problem*, dan waktu *report* dibuat. *List report* diurutkan sesuai kronologis pembuatan *report*. *Report* yang lebih baru akan tampil di atas. Jika salah satu *report* diklik, *user* akan diarahkan ke Report Details Screen seperti pada Gambar 3.32.



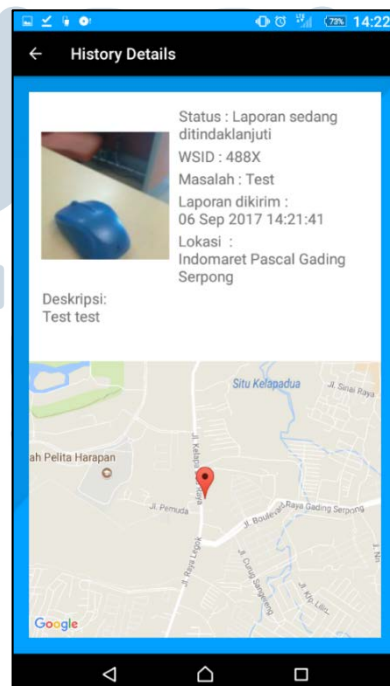
Gambar 3.30 In Progress Report Screen

Report History Screen dapat dilihat pada Gambar 3.31. Pada halaman ini, *user* dapat melihat *list* semua *report* yang pernah dikirimnya. *Report* ditampilkan dalam *CardView* yang berisi *icon* jam pasir (status *open*) atau *check* (status *closed*), *WSID*, *location*, *problem*, dan waktu *report* dibuat. *List report* diurutkan sesuai kronologis pembuatan *report*. *Report* yang lebih baru akan tampil di atas. Jika salah satu *report* diklik, *user* akan diarahkan ke Report Details Screen seperti pada Gambar 3.32.



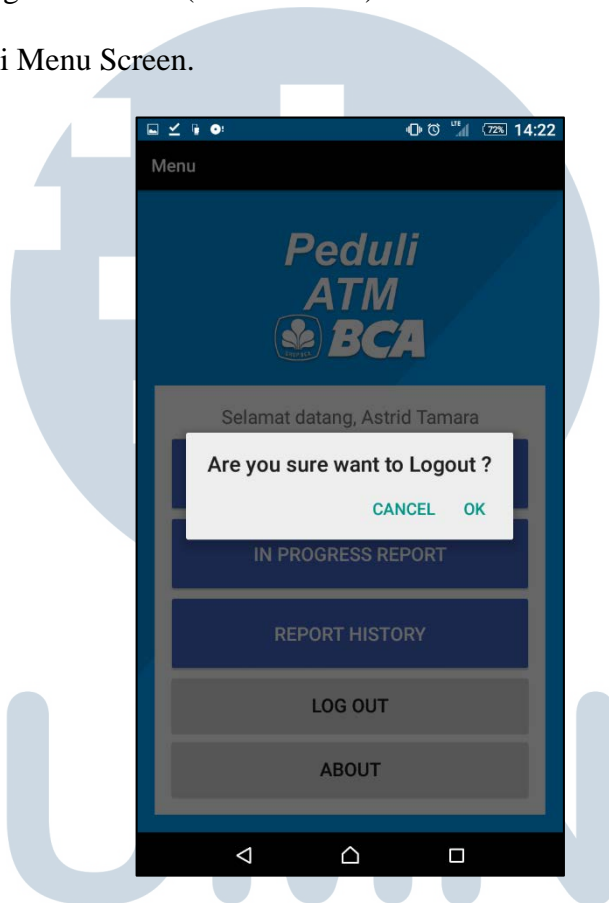
Gambar 3.31 Report History Screen

Tampilan History Details Screen ditunjukkan pada Gambar 3.32. History Details Screen menampilkan status, WSID, masalah, waktu pengiriman laporan, lokasi, deskripsi, dan peta koordinat lokasi dari *report* yang dipilih.



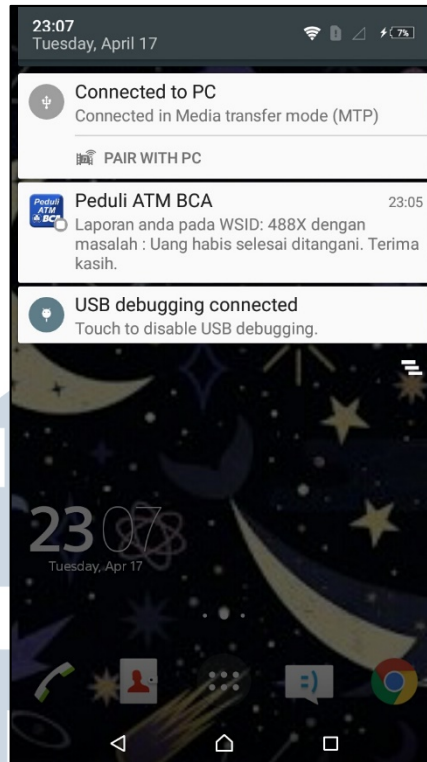
Gambar 3.32 History Details Screen

Jika *user* memilih *button* Log Out pada Menu Screen, sebuah *confirmation dialog box* ditampilkan seperti pada Gambar 3.33. Terdapat dua buah *button* pada *dialog box* tersebut, *button OK* dan *button cancel*. Jika *user* memilih *button OK*, proses Google Sign Out dan proses *logout* oleh API dikerjakan lalu tampilan kembali ke Sign In Screen (Gambar 3.23). Jika *button cancel* yang dipilih, *user* tetap berada di Menu Screen.



Gambar 3.33 Tampilan Confirmation Log Out

Jika laporan yang pernah dikirim oleh *user* telah selesai ditindaklanjuti dan statusnya menjadi *closed*, sebuah *push notification* akan dikirimkan ke perangkat *user*. *Push notification* tersebut berisi informasi WSID ATM dan masalah yang terjadi dari laporan yang statusnya telah menjadi *closed*. Penerimaan *push notification* di perangkat *user* dapat dilihat pada Gambar 3.34.



Gambar 3.34 Tampilan Push Notification

### 3.3.2 Testing

Setelah aplikasi selesai dirancang dan dibangun, dilakukan *testing* untuk evaluasi. *Testing* dilakukan secara internal dalam divisi Urusan Pemantauan ATM dengan menyebarkan hasil *build* aplikasi berupa *file* apk. *File* apk dapat di-*install* dan dijalankan di perangkat Android masing-masing. *Testing* dilakukan selama dua minggu yaitu pada minggu ke-6 dan ke-7. Jika ditemukan adanya *bug* atau bagian-bagian aplikasi yang perlu perbaikan, laporan tersebut akan langsung disampaikan agar aplikasi dapat segera direvisi.

### 3.3.3 Kendala dan Solusi yang Ditemukan

Kendala yang dihadapi selama proses perancangan dan pembangunan aplikasi Peduli ATM BCA berbasis Android adalah kurangnya pengalaman dalam mengembangkan aplikasi Android. Meski sebelumnya sudah pernah mengembangkan di perangkat Android, masih banyak aspek-aspek pemrograman Android yang belum pernah digunakan seperti *PullToRefreshList*, *Card View*, dan *Custom Adapter*, serta API seperti OneSignal, Google Sign In, dan Google Maps.

Solusi yang dilakukan untuk mengatasi kendala tersebut adalah dengan mempelajari pemrograman Android secara mandiri sebelum memulai magang. Selain itu, bertanya pada rekan kerja dan supervisi jika ada hal yang membingungkan atau memerlukan saran.

