BAB 3

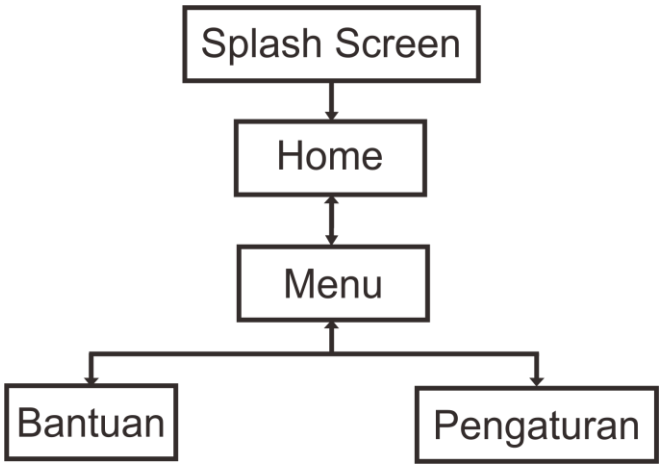
PEMBUATAN DAN IMPLEMENTASI APLIKASI

3.1 Perancangan Aplikasi

Rancangan tampilan merupakan tahap pertama yang dilakukan dalam pembuatan aplikasi Pengingat Stasiun Tujuan. Tahap ini merupakan tahap yang sangat penting dalam pembuatan aplikasi karena berkaitan dengan visualisasi aplikasi. Dalam merancang tampilan sebuah aplikasi dibutuhkan kesederhanaan dari arsitektur aplikasi dan kemudahan dalam penggunaannya. Tentunya rancangan pada aplikasi ini akan meliputi beberapa tampilan yang mendukung keutuhan dari aplikasi ini.

3.2 Struktur Navigasi

Dalam pembuatan aplikasi Pengingat Stasiun Tujuan membutuhkan Struktur Navigasi untuk mengetahui alur atau jalannya aplikasi yang dibuat. Gambar 3.1 adalah Struktur Navigasi Hirarki. Berikut ini adalah Struktur Navigasi pada aplikasi Pengingat Stasiun Tujuan.



Gambar 3.1 Struktur Navigasi Aplikasi

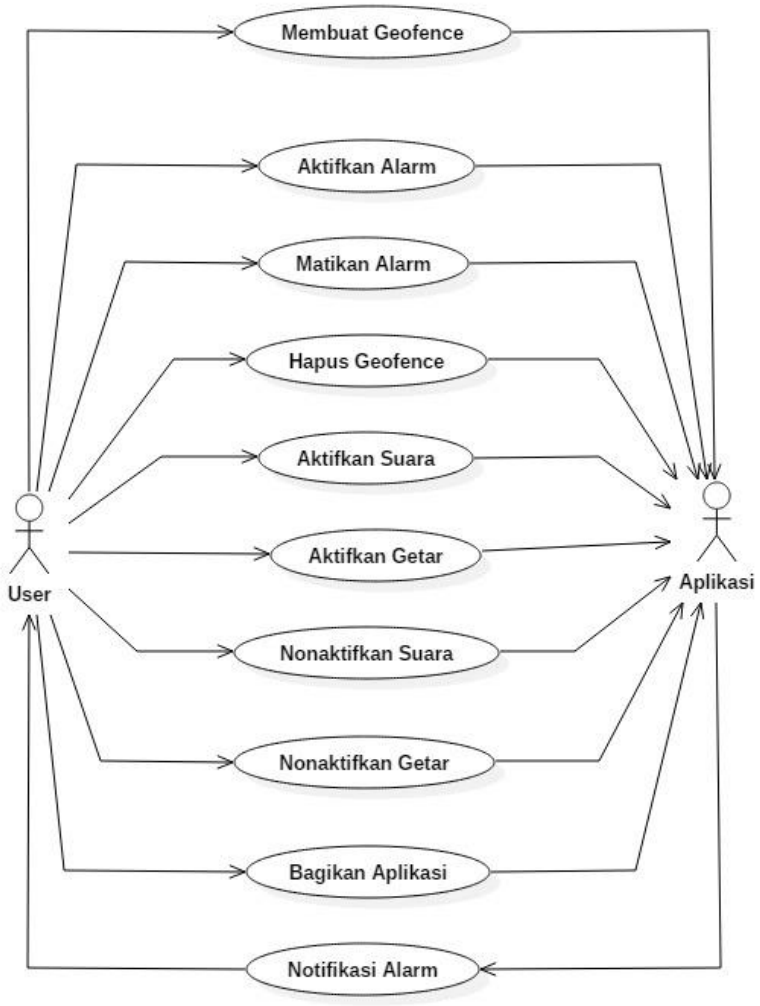
Gambar 3.1 merupakan penjelasan alur aplikasi Pengingat Stasiun Tujuan setiap tampilan. Saat aplikasi dimulai, akan menampilkan *SplashScreen* selama kurang lebih 5 detik, kemudian akan muncul tampilan *home* yang berisi *Spinner* atau *dropdown*, *button on/off* dan *menu* . *Dropdown* atau *Spinner* berisi daftar 24 stasiun yang dilewati oleh kereta *Commuterline* Jakartakota menuju Bogor dan sebaliknya, daftar stasiun tersebut berfungsi agar menyalanya alarm sesuai dengan tujuan yang telah ditentukan. *Button on/off* berfungsi untuk menyalakan alarm yang telah ditentukan di menu *dropdown* dan mematikan nada alarm apabila alarm berbunyi ketika sampai ditujuan. Di dalam *menu* terdapat enam pilihan *menu* yaitu, *home,* bantuan, tentang, pengaturan , bagikan dan keluar. *Menu home* berfungsi untuk pindah ke tampilan utama, tampilan utama tersebut adalah tampilan setelah *splashscreen. Menu* bantuan berfungsi untuk pindah kehalaman baru atau tampilan baru yang berisi penjelasan cara menggunakan aplikasi Pengingat Stasiun Tujuan. *Menu* tentang berfungsi agar memunculkan *messagebox* untuk melihat infromasi aplikasi. *Menu* pengaturan berfungsi untuk pindah kehalaman baru atau tampilan baru yang berisi pengaturan, di dalam halaman pengaturan terdapat dua *switch, switch* yang pertama digunakan untuk mengaktifkan atau menonaktifkan suara sedangkan *switch* yang kedua digunakan untuk mengaktifkan atau menonaktifkan getar ketika alarm sedang berjalan. *Menu* bagikan adalah menu yang digunakan untuk membagikan informasi aplikasi Pengingat Stasiun Tujuan kepada seseorang atau ke *grup* dengan menggunakan aplikasi lain tentang adanya aplikasi Pengingat Stasiun Tujuan. Dan yang terakhir adalah *menu* keluar, menu tersebut digunakan untuk keluar dari aplikasi.

3.3 UML

UML(*Unified Modeling Language)* pada aplikasi Pengingat Stasiun Tujuan terdiri dari *use case diagram,* dan *activity diagram.* Masing masing diagram menggambarkan secara umum sistem dari aplikasi ini.

3.3.1 *Use Case* Diagram

Use case diagram menggambarkan aktivitas yang dilakukan oleh suatu sistem dari sudut pandang pengamatan luar dan yang menjadi titik beratnya adalah sesuatu yang dapat dilakukan *user*, bukan cara melakukannya. Pada gambar 3.2 merupakan *use case* yang digunakan di dalam sistem



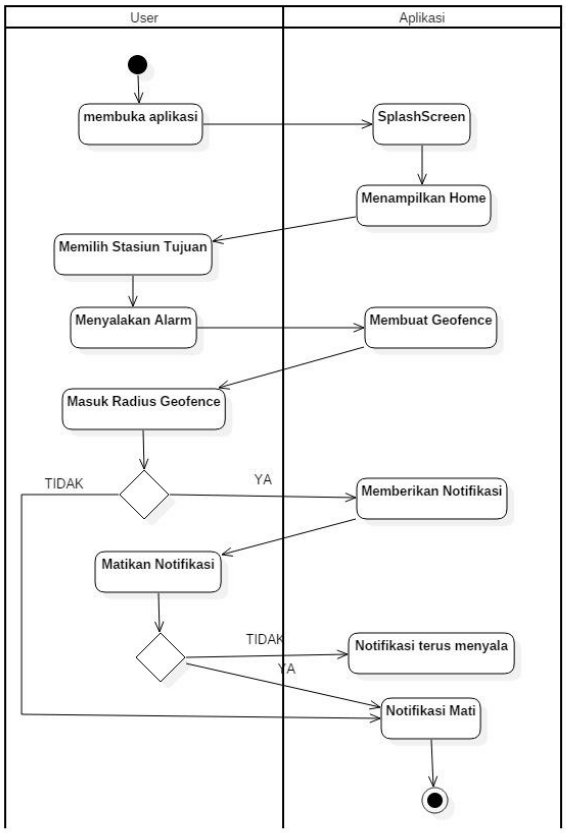
Gambar 3.2 *Use Case* Diagram

Pada gambar 3.2 merupakan *use case* diagram yang menggambarkan kegiatan yang dapat dilakukan oleh user pada aplikasi ini. Pada aktor User dapat membuat Geofence dan dapat menghapus Geofence, meng aktifkan alarm dan

mematikan alarm, mengaktifkan atau menonaktifkan suara dan getar notifkasi, membagikan aplikasi informasi Pengingat Stasiun Tujuan kepada pengguna android lain, dan aplikasi memberikan notifikasi kepada user berupa suara dan getar apabila telah sampai di stasiun tujuan.

3.3.2 Activity Diagram

Activity Diagram adalah representasi grafis dari seluruh tahapan alur kerja. Pada *activity* diagram terdapat langkah-langkah dari komponen dalam suatu sistem. Diagram ini menggambarkan alur berawal, percabangan yang mungkin terjadi dan alur berakhir.



Gambar 3.3 *Activity Diagram*

InitialState menggambarkan awal dari proses dalam aplikasi dan *finalstate* merupakan penggamabaran akhir dari sebuah proses. *User* masuk pada tampilan awal aplikasi aplikasi akan menampilkan *Splashscreen*, berikutnya aplikasi akan

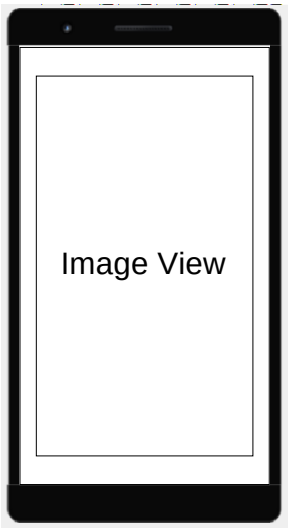
menampilkan *home,* di dalam tampilan *home* user dapat memilih stasiun tujuan dan mengaktifkan alarm, setelah stasiun dan alarm telah dibuat selanjutnya aplikasi akan membuat *geofence* dari stasiun yang dituju. Selanjutnya apabila user masuk ke dalam radius *geofence,* maka aplikasi akan memberikan notifikasi berupa getar dan bunyi pada *smartphone* kapada *user*, apabila *user* belum atau tidak masuk ke dalam radius *geofence,* maka aplikasi tidak memberikan notifikasi atau notifikasi mati, apabila user masuk ke dalam radius *geofence* tetapi tidak menonaktifkan notifikasi maka notifikasi tetap bergetar dan berbunyi sampai notifikasi di nonaktifkan.

3.4 Rancangan Tampilan

Rancangan tampilan merupakan tahap lanjutan dalam pembuatan aplikasi Pengingat Stasiun Tujuan. Tahap ini merupakan tahapan yang penting dalam pembuatan aplikasi karena berkaitan dengan tampilan aplikasi yang akan digunakan. Dalam merancang tampilan sebuah aplikasi dibutuhkan kesederhanaan dari arsitektur aplikasi dan kemudahan dalam penggunaannya. Rancangan pada aplikasi ini merupakan gambaran dari aplikasi yang akan dibuat dan meliputi beberapa tampilan yang mendukung keutuhan dari aplikasi ini.

3.4.1 Rancangan Tampilan Splash Screen

Tampilan *splash screen* merupakan tampilan awal sebelum masuk ke tampilan *home*. Rancangan tampilan *splash screen* dapat dilihat pada gambar 3.4.

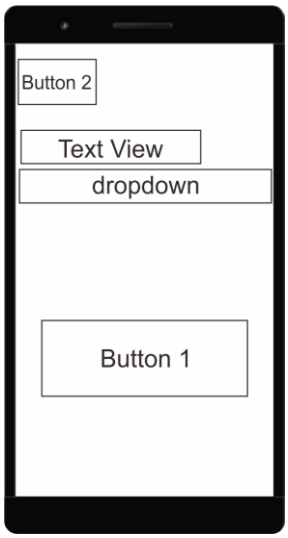


Gambar 3.4 Rancangan Tampilan Splash Screen

Pada gambar 3.4 terdapat satu *imageview* dengan format *png* untuk menampilkan logo dari apliksi, dengan *background* selama 5 detik sebelum masuk ke dalam tampilan *home*.

3.4.2 Rancangan Tampilan *Home*

Tampilan pilihan *home* merupakan tampilan kedua yang dapat dilihat pada gambar 3.5.

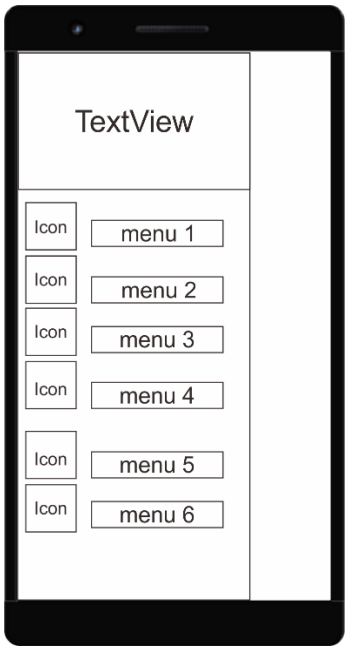


Gambar 3.5 Rancangan Tampilan *Home*

Pada gambar 3.5 terdapat satu *textview* yang digunakan sebagai petunjuk untuk memilih stasiun yang dituju. Terdapat satu *dropdown* atau *menu* pilihan kebawah yang berisi daftar nama stasiun yang dilewati kereta tujuan Bogor dari Jakartakota atau sebaliknya. Terdapat dua buah *button*, *button* satu berfungsi sebagai tombol mengaktifkan alarm dan membuat *geofence* setelah tujuan dipilih di *menu dropdown* atau menonaktifkan bunyi alarm ketika sampai pada stasiun yang dituju atau ketika sudah masuk ke dalam radius *geofence* dan menghapus *geofence* yang telah dibuat, *button* dua berfungsi sebagai pembuka *menu* aplikasi Pengingat Stasiun Tujuan.

3.4.3 Rancangan Tampilan *Menu* aplikasi

Tampilan pilihan *menu* merupakan tampilan yang akan terlihat atau keluar pada saat *button* dua ditekan. Tampilan *menu* ini berisi *menu* apa saja yang terdapat di aplikasi Pengingat Stasiun Tujuan. Rancangan tampilan *menu* ini dapat dilihat pada gambar 3.6.



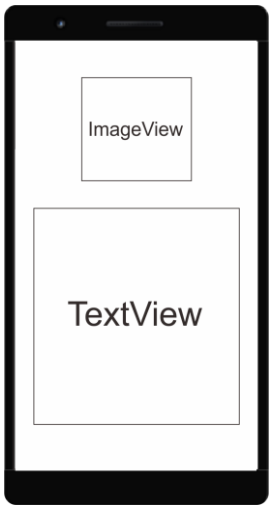
Gambar 3.6 Rancangan Tampilan *Menu*

Pada gambar 3.6 terdapat *textview* yang digunakan untuk menuliskan judul aplikasi, lalu terdapat enam icon dan enam *menu*. *Menu* satu berfungsi sebagai

tombol kembali ke tampilan *home* atau membuka kembali tampilan *home. Menu* dua berfungsi untuk menampilkan tampilan bantuan. *Menu* tiga berfungsi untuk menampilkan *messagebox* pada layar yang berisi tentang *developer* aplikasi. *Menu* empat berfungsi untuk menampilkan tampilan pengaturan. *Menu* lima berfungsi untuk membagikan informasi aplikasi kepada orang lain melalui aplikasi *message* yang ada pada *smartphone*. Lalu yang terakhir adalah *menu* enam, *menu* tersebut digunakan untuk keluar aplikasi Pengingat Stasiun Tujuan.

3.4.4 Rancangan Tampilan Bantuan

Tampilan bantuan berguna untuk menampilkan *step by step* atau cara menggunakan aplikasi dan cara melaporkan jika terjadi bug kepada penulis. Rancangan tampilan bantuan ini dapat dilihat pada gambar 3.7.



Gambar 3.7 Rancangan Tampilan bantuan

Pada gambar 3.7 terdapat satu *imageview* dan satu *textview*, *imageview* digunakan untuk menaruh gambar logo atau icon dari aplikasi Pengingat Stasiun Tujuan. Untuk *textview* berfungsi untuk menampilkan instruksi dari aplikasi dan bantuan laporan jika adanya bug kepada *developer*.

3.4.5 Rancangan Tampilan Tentang

Pada tampilan tentang ini hanya menggunakan *messagebox*, *messagebox* tersebut sudah tersedia pada library android studio, sehingga dapat langsung

menggunakan tanpa adanya desain tampilan terlebih dahulu. Pada gambar 3.8 ini terdapat tampilan layar ketika *messagebox* muncul.

Gambar 3.8 Tampilan ketika *messagebox* muncul

Pada tampilan diatas terdapat *messagebox* yang berfungsi untuk memunuculkan pesan yang berisi tentang nama aplikasi, versi aplikasi, dan profil *developer*.

3.5 Langkah Pembuatan Aplikasi

Setelah membuat rancangan, struktur navigasi dan algoritma aplikasi, tahap berikutnya adalah pembuatan aplikasi. Berikut ini langkah-langkah pembuatan aplikasi Pengingat Stasiun Tujuan :

3.5.1 Instalasi JDK

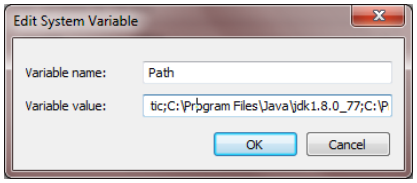
Pada langkah pertama penulis melakukan instalasi *software* JDK (*Java Development Kit).* JDK yang digunakan yaitu versi jdk1.8.0_77. Berikut adalah langkah-langkah instalasi *software* JDK:

a. Penulis mengunduh *software* JDK di situs resmi oracle [8]
b. Pada langkah-langkah instalasi JDK, penulis hanya menekan tombol *next*, seperti gambar 3.9.
c. Jika proses instalasi telah selesai dan berhasil, maka akan terdapat folder JDK dan JRE pada C:\Program Files\Java.

(a) (b)

Gambar 3.9 (a) Instalasi Awal JDK, (d) Proses Instalasi JDK Selesai

d. Selanjutnya penulis melakukan pengaturan *PATH* Java, dengan cara memilih *Control Panel* kemudian penulis memilih *System*, penulis melanjutkan dengan memilih *Advanced* dan mengklik *Environment Variables*, kemudian penulis mengisi *variabel name : PATH* dan *variable value* : C:\Program Files\Java\ jdk1.8.0_77\bin; (menunjuk pada folder \bin JDK yang di*install*). Gambar 3.9 adalah proses yang dilakukan penulis dalam pengaturan *PATH* Java.



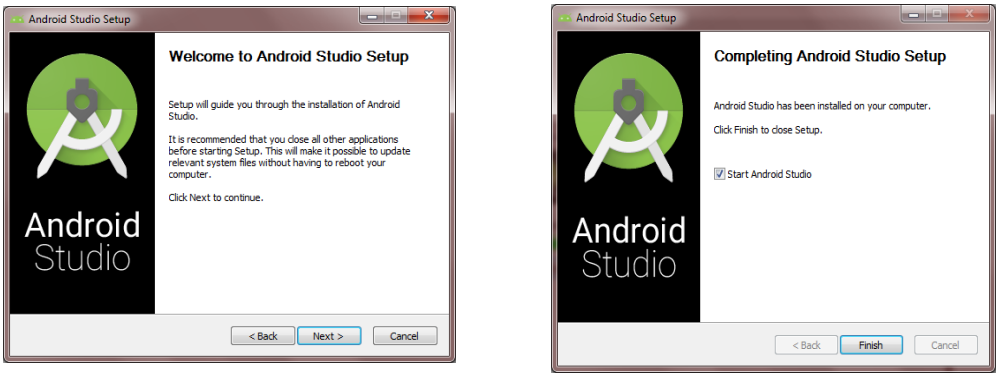
Gambar 3.10 Tampilan *System Variable*

3.5.2 Instalasi Android Studio

Setelah instalasi JDK selesai dilakukan pada langkah pertama, selanjutnya penulis menginstalasi Android Studio. Berikut adalah langkah-langkah instalasi Android Studio:

a. Penulis mengunduh *software* Android Studio di situs resmi Android Studio[2]
b. Setelah selesai mengunduh, penulis mencari *file* Android Studio instalasi *executable* (bernama **android-studio-bundle-<version>.exe**) di jendela *Windows Explorer* dan penulis mengklik dua kali untuk memulai proses

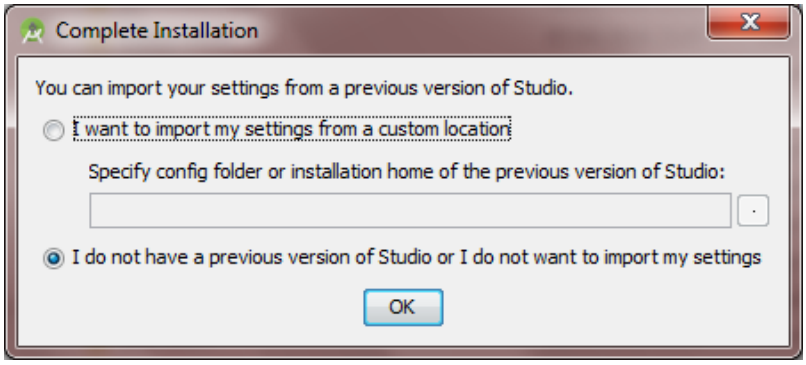
instalasi, setelah itu langkah-langkah instalasi Android Studio berikutnya penulis hanya menekan tombol *next* seperti pada gambar 3.6.



(a) (b)

Gambar 3.11 (a) Proses Instalisasi Awal Android Studio,
(b) Instalasi Android Studio selesai

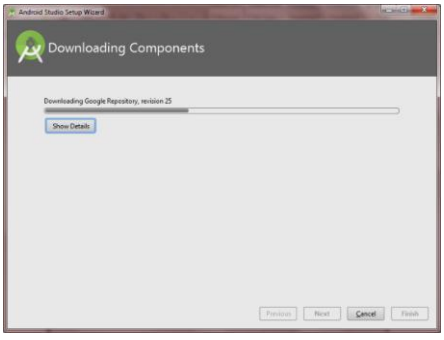
c. Setelah langkah instalasi selesai seperti pada gambar 3.11 (b), kemudian akan muncul jendela yang menyediakan opsi untuk mengimpor pengaturan dari versi Android Studio sebelumnya seperti pada gambar 3.12. Jika memiliki pengaturan dari versi sebelumnya dan ingin mengimpor mereka ke dalam instalasi terbaru, maka opsi yang sesuai dengan lokasi dipilih. Karena penulis baru pertama kali meng*install* Android Studio maka pernyataan *"I do not have a previous version of Android Studio or I do not want to import my settings"* dipilih dan penulis menekan tombol *OK* untuk melanjutkan. Setelah penulis memilih pernyataan tersebut, kemudian jendela untuk memulai Android Studio akan tampil seperti pada gambar 3.13.



Gambar 3.12 Jendela Kelengkapan Instalasi.

Gambar 3.13 Jendela Memulai Android Studio

d. Setelah Android Studio berhasil dimuat, selanjutnya akan tampil jendela *Android Studio Setup Wizard* seperti pada gambar 3.14. Pada jendela ini penulis mengunduh dan menginstalasi komponen Android SDK Tools. Sebelumnya penulis memastikan komputer terhubung dengan internet



Gambar 3.14 Proses Unduh dan Instalasi Android SDK tools

3.5.3 Instalasi ADB Drivers Android Studio

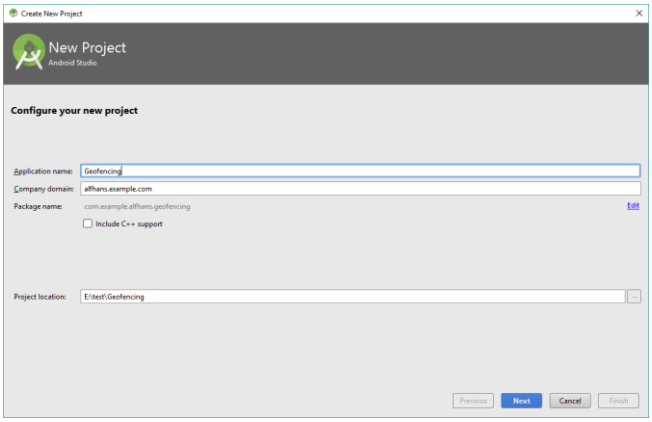
Setelah melakukan instalasi Android Studio pada langkah ke dua, selanjutnya penulis melakukan instalasi ADB Drivers pada komputer/PC, agar nantinya aplikasi dapat di uji coba dengan menggunakan *smartphone*. Berikut adalah langkah-langkah dalam melakukan instalasi ADB Drivers:

a. Penulis menghubungkan *Smartphone* Android dengan komputer menggunakan kabel USB.
b. Pada komputer yang terdapat di *desktop* atau *Windows Explorer*, penulis menekan tombol klik pada bagian kanan *mouse* kemudian penulis memilih *Manage.*
c. Selanjutnya penulis memilih *Devices* yang terdapat di panel kiri.
d. Penulis mencari dan memperluas perangkat lain di panel kanan.
e. *Update Driver Software* dipilih setelah penulis menekan tombol klik dibagian kanan *mouse* pada nama perangkat. Langkah ini akan meluncurkan *Hardware Update Wizard.*
f. Penulis memilih *Browse my computer for driver software* kemudian menekan tombol *next*.
g. Kemudian penulis mencari folder driver USB. (The Google USB Driver terletak di <sdk>\extras\google\usb_driver\.) dengan menekan *Browse*.
h. Penulis menekan tombol *Next* untuk menginstalasi *driver*.

3.5.4 Pembuatan Project Baru Android Studio

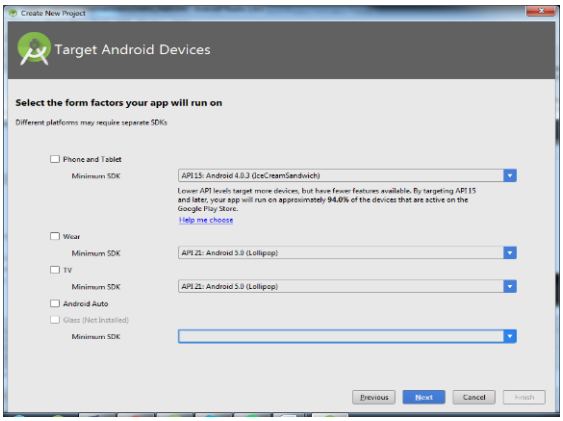
Setelah selesai menginstall android studio maka tahap selanjutnya adalah pembuatan aplikasi dengan memilih *Start an new Android Studio Project.*

a. Langkah pertama adalah penulis mengisi nama projek "*Geofencing*", *Company Domain* (alfhan.example.com), kemudian penulis menentukan direktori kerja atau *workspace* yang digunakan untuk menyimpan projek Android. Lokasi direktori kerja yang ditetapkan penulis yaitu "D:\Geofencing". Lalu penulis menekan tombol *Next* seperti pada gambar 3.15.



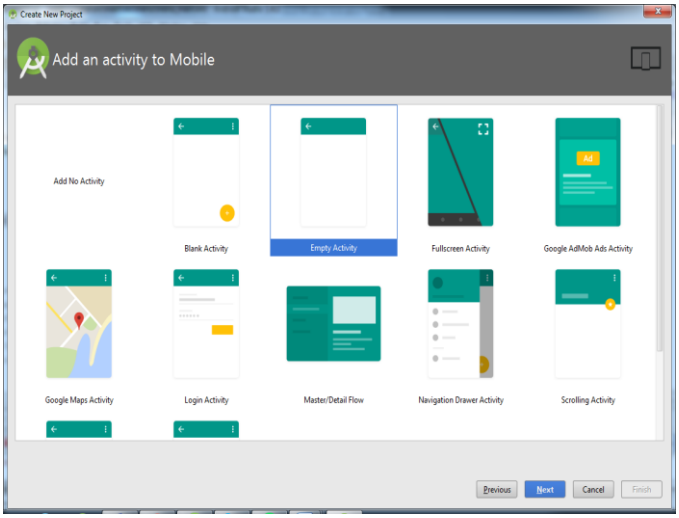
Gambar 3.15 Konfigurasi Projek Baru

b. Setelah membuat projek pada gambar 3.15, penulis melanjutkan dengan memillih target *run* aplikasi yang ada pada gambar 3.16, dalam hal ini penulis memilih *Phone and Tablet.* Dibagian minimum SDK, penulis memilih API 15: Android 4.0.3 (IceCreamSandwich). Kemudian penulis menekan tombol *Next*.



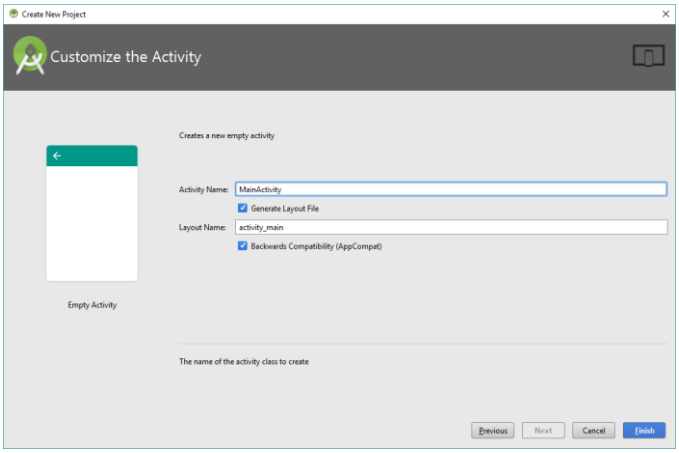
Gambar 3.16 Menentukan Target Run Aplikasi

c. Dari gambar 3.16 kemudian akan muncul tampilan untuk pilihan *Layout* (*Activity*) aplikasi. Disini pengguna Android Studio bebas memilih *activity* yang disediakan tanpa harus membuat *layout* nya lagi dari awal. Untuk menyesuaikan dengan desain yang telah dibuat, maka penulis memilih *Empty Activity* seperti pada gambar 3.17. Setelah itu penulis menekan tombol *Next*.



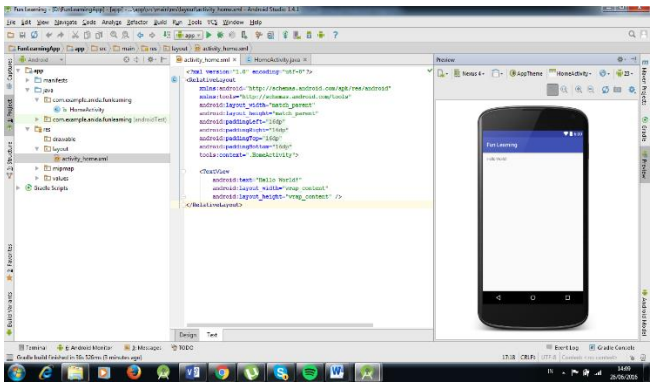
Gambar 3.17 Memilih Tampilan *Layout Activity*

d. Pada tahap ini, penulis menentukan nama dari *activity* yang telah dipilih pada langkah lima.c. Disini penulis memberi nama *MainActivity* kemudian penulis menekan tombol *Finish* seperti pada gambar 3.18.



Gambar 3.18 Mengatur Nama *Activity*

e. Setelah penulis menekan tombol *finish*, maka akan tampil lembar projek Android Studio seperti pada gambar 3.19

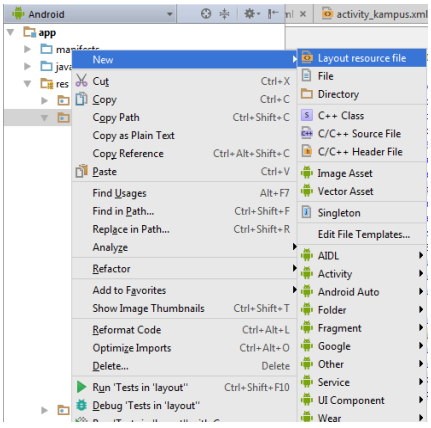


Gambar 3.19 Tampilan Lembar Kerja

3.5.5 Pembuatan *Splash Screen*

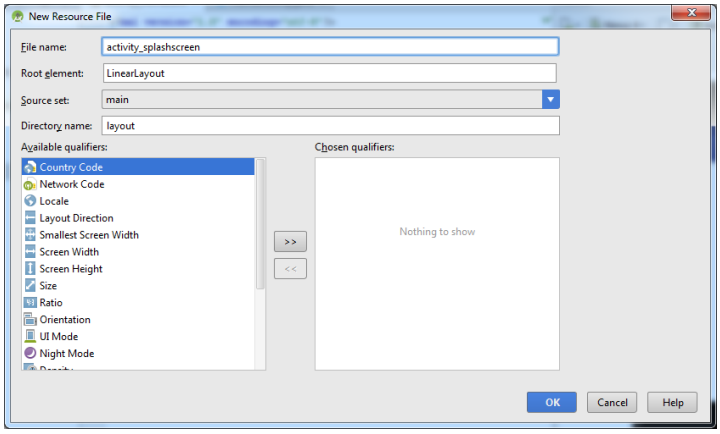
Setelah projek baru berhasil dibuat seperti pada gambar 3.19, selanjutnya penulis membuat *activity* XML untuk mendesain tampilan pogram aplikasi dan *activity* Java untuk memasukkan kode aplikasi. Berikut adalah langkah-langkah dalam pembuatan *activity* XML dan *activity* Java pada Android Studio:

a. Untuk membuat *activity* XML baru, penulis menekan tombol klik bagian kanan *mouse* pada folder *layout* yang terdapat di folder *res* di Android Studio. Kemudian penulis memilih *layout resource file* seperti pada gambar 3.20.



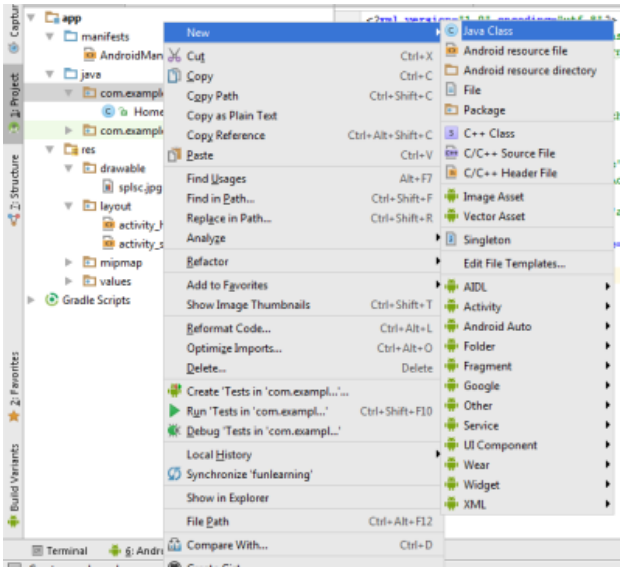
Gambar 3.20 Membuat *Activity* XML Baru

b. Setelah penulis memilih *layout resource file* seperti pada gambar 3.20, akan muncul jendela untuk memberi nama *activity* XML seperti pada gambar 3.21. Setelah penulis memberi nama *activity*, kemudian penulis menekan tombol *OK*.



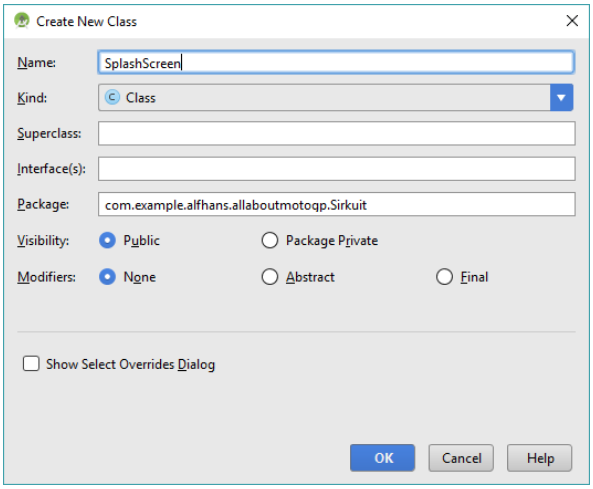
Gambar 3.21 Memberi Nama *Activity* XML Baru

c. Setelah *activity* XML berhasil dibuat, penulis melanjutkan dengan membuat *activity* Java baru. Untuk membuat *activity* Java baru, penulis mengklik kanan *mouse* pada folder Java di Android Studio, kemudian penulis memilih *New* lalu memilih *Java Class* seperti pada gambar 3.22.



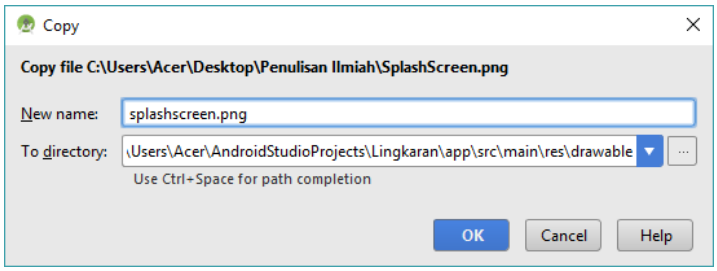
Gambar 3.22 Membuat *Java Class* Baru

d. Setelah penulis memilih *Java Class* seperti gambar 3.22, akan muncul jendela untuk memberi nama *activity* Java seperti pada gambar 3.23. Penulis memberi nama *activity* kemudian menekan tombol *OK*.



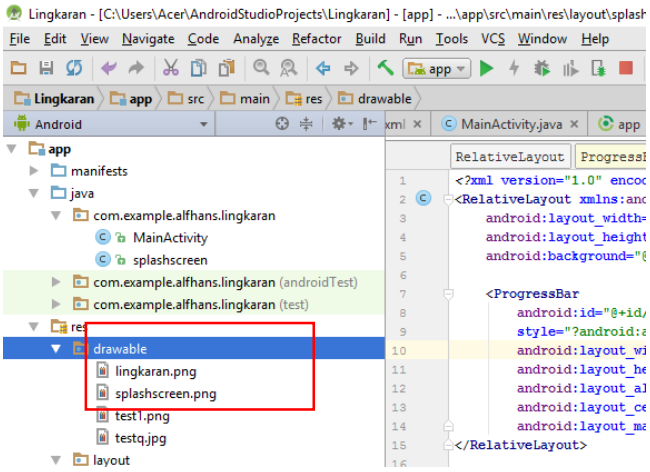
Gambar 3.23 Memberi Nama *class* Java Baru

e. Setelah langkah selanjutnya adalah memasukan gambar terlebih dahulu ke dalam folder drawable yang terdapat disebelah kiri, penulis dapat meng-copy kan saja langsung dari peyimpanan ke dalam langsung drawable seperti gambar 3.24 ini.



Gambar 3.24 Memasukan Gambar ke Android Studio

Setelah itu klik ‘OK’ dan gambar akan masuk ke dalam android studio dan dapat digunakan sebagai *resource* pada folder gambar seperti gambar 3.25



Gambar 3.25 Folder Gambar di Android Studio

Setelah membuat class dan memasukan gambar yang penulis ingin jadikan untuk *splashscreen* pada android studio. Berikut langkah-langkah selanjutnya dalam pembuatan *splashscreen* aplikasi Pengingat Stasiun Tujuan pada Android Studio :

f. Selanjutnya dalam pembuatan *splashscreen* aplikasi Pengingat Stasiun Tujuan yaitu penulis mendesain tampilan *splashscreen* pada *activity* xml yang bernama activity_splashscreen.xml yang sebelumnya sudah dibuat pada langkah ke enam. Untuk mendesain tampilan *splashscreen*, penulis menggunakan *tag LinearLayout* agar *widget* terurut secara vertikal. Di dalam *tag LinearLayout*, penulis juga menggunakan *android:orientation* vertikal karena perangkat yang nantinya digunakan untuk aplikasi Pengingat Stasiun Tujuan berorientasi vertikal. Untuk mengatur tinggi dan lebar layar, penulis menggunakan *match_parent* supaya ukuran mengikuti ukuran layar. Penulis menggunakan *android:gravity center* supaya tampilan berada ditengah layar. Penulis menggunakan background #FFFFFF yang berarti *background* berwarna putih. Berikut adalah potongan kode aplikasi pada activity_splashscreen.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center"
    android:background="#FFFFFF">
```

g. Kemudian penulis menambahkan *ImageView* di dalam tag *LinearLayout*. *ImageView* berfungsi untuk menampilkan gambar. Pada *ImageView* penulis menggunakan *match_parent* untuk mengatur tinggi dan lebar gambar, kemudian penulis menggunakan *layout_gravity center* supaya gambar berada ditengah. Untuk menampilkan gambar yang akan dijadikan *splashscreen*, penulis memanggil gambar yang sudah tersimpan di folder *drawable* dengan menggunakan *android:src* yang berarti *source* atau sumber folder penyimpanan gambar. Penyimpanan gambar pada folder *drawable* sudah dijelaskan pada langkah ke lima. Gambar yang digunakan untuk *splashscreen* yaitu bernama splashscreen.png yang ada pada gambar 3.20. Selain itu penulis membuat gambar mengikuti skala layar dengan menambahkan tipe skala *fitXY*. Berikut potongan kode aplikasi pada activity_splashscreen.xml:

```
<ImageView
    android:src="@drawable/xz"

    android:layout_width="fill_parent"

    android:layout_height="fill_parent"

    android:scaleType="fitXY"/>
```

h. Kemudian penulis menambahkan *widget* progress bar pada tampilan, progress ini berfungsi seperti tampilan loading pada umumnya, yaitu seperti lingkaran yang beruputar agar terlihat seperti sedang melakukan progress atau melakukan aktifitas tertentu yangmemerlukan waktu untuk membuka atau memulai aktiftas tersebut. *Style* pada kode pada dasarnya seperti tema untuk proggress bar tersebut, *width* dan *height* atau lebar dan tinggi penulis membuat ukuran

wrap_content yang artinya adalah sesuai ukuran dengan progress bar secara default. *Center_horizontal* adalah posisi yang bersifat boolean apakah posisi progress bar nanti akan diletakan ditengah layar atau tidak, *margin_botom* adalah jarak progress bar kebawah layar, berikut potongan kode xml activity_splashscreen.xml:

```
<ProgressBar
    android:id="@+id/progressBar1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentBottom="true"
    android:layout_centerHorizontal="true"
    android:layout_marginBottom="58dp" />
```

i. Setelah penulis mendesain tampilan *splashscreen* pada activity_splashscreen.xml, kemudian penulis mengetikkan kode aplikasi Pengingat Stasiun Tujuan pada *activity* Java yang bernama Splashscreen.java yang sudah dibuat dilangkah ke enam.Pada Splashscreen.java, penulis menggunakan ***new*** *Handler().postDelayed(****new*** *Runnable()* yang berfungsi supaya *splashscreen delay* selama waktu yang ditentukan, disini penulis menentukan waktu 5000ms yang berarti 5 detik pada *splashscreen*, kemudian setelah waktunya selesai, *splashscreen* akan langsung masuk ke MenuActivity.java. Berikut potongan kode aplikasi Pengingat Stasiun Tujuan pada Splashscreen.java:

```
        new Handler().postDelayed(new Runnable() {
            @Override
            public void run() {
                Intent i = new
Intent(SplashscreenActivity.this, MenuActivity.class);
                startActivity(i);
                finish();
            }
        },5000);
```

3.5.6 Pembuatan *Main Activity*

Pada pembuataan *main activity* sama seperti *splashscreen* namun sudah tidak memerlukan untuk pembuatan file baru activity_main.xml nya dan *java class* karena sudah secara *default* tersedia ketika membuat project baru, dalam pembuatan tampilan *main activity* terdapat 5 widget yang digunakan berikut *widget* pada tampilan tersebut :

a. Agar *geofence* dapat dibuat, ditambahkan permintaan izin dari pengguna untuk menggunakan akses GPS, internet dan getar. Berikut merupakan potongan kode untuk meminta perizinan.

```
<uses-permission
android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission
android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission
android:name="android.permission.ACCESS_COARSE_LOCATION" />
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.VIBRATE"/>
```

android:name="android.permission.ACCESS_FINE_LOCATION digunakan untuk memberikan perizinan tentang melakukan deteksi lokasi *smartphone* secara terperinci.

android:name="android.permission.ACCESS_NETWORK_STATE" digunakan untuk mengecheck apakah *smartphone* terdapat koneksi internet.

android.permission.ACCESS_COARSE_LOCATION digunakan untuk melakukan akses lokasi Anda melalui WIFI yang sedan digunakan.

android.permission.INTERNET digunakan untuk mengakses internet ketika melakukan download API Google Maps.

<uses-permission android:name="android.permission.VIBRATE"/> digunnakan untuk memberikan izin untuk vibrasi atau getar perangkat.

b. langkah pertama adalah mengatur tampilan awal atau dasar dari *widget,* penulis membuat *width* dan *height* dengan *match_parent, match_parent* berfungsi agar tampilan memenuhi ukuran layar ponsel, pada tampilan ini penulis memberikan *id* sebagai identitas tampilan nanti jika tampilan ini dipanggil kembali oleh sebuah fungsi.

id tampilan *main_activity* adalah "drawer" dapat dilihat dari "android:id="@+id/drawer". *alignParentTop* berfungsi agar tampilan tepat berada dibawah *header* aplikasi. Dan android.support.v4.widget.DrawerLayout ini library yang digunakan pada *menu* aplikasi apabila menggunakan *drawer* atau tampilan yang bergeser pada tampilan *main_activity.*

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.v4.widget.DrawerLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:id="@+id/drawer">

<RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical"
        android:weightSum="1"
        android:gravity="center"
        android:layout_alignParentTop="true"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true">
```

c. selanjutnya adalah memasukan *widget* yang dibutuhkan, penulis menambahkan *textview* sebagai petunjuk agar user memilih stasiun tujuan nya. Penulis menggunak *width* dan *height* dengan *wrap_content, wrap_content* digunakan sebagai ukuran yang sesuasi atau default yang mengikuti ukuran *content* nya. *Textcolor* digunakan sebagai pemberi warna pada text, format warna pada Android Studio berbentuk *hexa* dan warna #B22222 adalah warna merah marun. android:text berfungsi untuk mengisi *textview* dengan sebuah kata-kata atau dengan kalimat.

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textColor="#B22222"
    android:text="Pilih Stasiun Tujuan Anda :"
    android:textSize="20dp"
    android:id="@+id/textView"
    android:layout_alignParentTop="true"
    android:layout_alignLeft="@+id/spinner"
    android:layout_alignStart="@+id/spinner"
    android:layout_marginTop="10dp" />
```

d. Selanjutnya menambahkan *widget spinner, spinner* ini berisikan 24 nama stasiun dengan rute Jakartakota menuju Bogor atau sebaliknya yang dapat dijadikan sebagai pengingat tujuan. *string*.xml ini penulis berikan nama *id* "stasiun". 24 daftar nama stasiun tersebut dimasukan ke dalam *string*.xml yang berada pada folder *String*. berikut kode xml yang ditambakan pada tampilan home dengan menambahkan *widget spinner* pada *main_activity*.xml seperti berikut :

1. *Widget Spinner*

```
//pembuatan widget spinner
<Spinner
    android:layout_width="300dp"
    android:layout_height="wrap_content"
    android:id="@+id/spinner"
    android:spinnerMode="dropdown"
    android:entries="@array/stasiun"
    android:layout_below="@+id/textView"
    android:layout_centerHorizontal="true"
```

2. String .xml

```
// daftar nama stasiun didalam String.xml
<string-array name="stasiun">
        <item> ---- Select -----</item>
        <item>Bogor</item>
        <item>Cilebut</item>
        <item>Bojong Gede</item>
        <item>Citayam</item>
        <item>Depok</item>
        <item>Depok Baru</item>
        <item>Pondok Cina</item>
        <item>Universitas Indonesia</item>
        <item>Universitas Pancasila</item>
        <item>Lenteng Agung</item>
       <item>Tanjung Barat</item>
        <item>Pasar Minggu</item>
        <item>Pasar Minggu Baru</item>
       <item>Duren Kalibata</item>
         <item>Cawang</item>
         <item>Tebet</item>
         <item>Manggarai</item>
         <item>Cikini</item>
         <item>Gondangdia</item>
         <item>Juanda</item>
         <item>Sawah Besar</item>
         <item>Mangga Besar</item>
         <item>Jayakarta</item>
         <item>Jakarta Kota</item>
         </string-array>
         </resources>
```

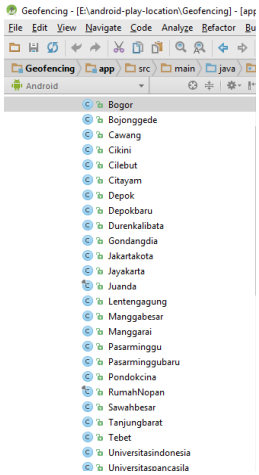
Untuk menampilkan *spinner* tersebut diperlukan *adapter*. Berikut adalah kode *adapter* dari *spinner*.

```
Spinner spinner = (Spinner) findViewById(R.id.spinner);

ArrayAdapter adapter =
ArrayAdapter.createFromResource(this,R.array.stasiun,
android.R.layout.simple_spinner_item);

adapter.setDropDownViewResource(android.R.layout.simple_sp
inner_dropdown_item);
spinner.setAdapter(adapter);
```

Setelah ini penulis membuat *class* yang berisikan nama stasiun rute Jakartakota menuju bogor dan juga sebaliknya. Pembuatan *class* sama seperti sebelumnya berikut daftar nama *class* yang dibuat :



Gambar 3. 26 Daftar Nama *Class Stasiun* yang Dibuat

Selanjutnya kode dimasukan ke dalam *main_activitiy.java* untuk menentukan coordinat *geofence* dan radius yang akan digunakan pada setiap daftar nama stasiun pada *spinner*. Berikut adalah pemanggilan *class geofence* dari setiap daftar nama *spinner* :

1. Pemanggilan *class geofence* pada *spinner*

```
// pemanggilan class geofence pada spinner
spinner.setOnItemSelectedListener(new
AdapterView.OnItemSelectedListener() {
    @Override
    public void onItemSelected(AdapterView<?> parent, View
view, int i, long l) {
if (i == 0) {
```

```
            AlertDialog alertDialog = new
AlertDialog.Builder(MainActivity.this).create();
            alertDialog.setTitle("Pengingat Stasiun
Tujuan");
            alertDialog.setMessage("Pilihlah Stasiun
Tujuan Anda");
            alertDialog.show();
        }
        if (i == 1) {
            Bogor();
        }
        if (i == 2) {
            Cilebut();
        }
        if (i == 3) {
```

Berikut kode *class* pembuatan *geofence* stasiun Bogor yang mewakili seluruh *class* daftar nama stasiun karena logika yang digunakan sama:

2. pemanggilan class pembuatan geofence stasiun Bogor

```
public class Bogor {
    private Bogor() {
    }
  public static final long GEOFENCE_EXPIRATION_IN_HOURS =
3;
    public static final long
GEOFENCE_EXPIRATION_IN_MILLISECONDS =
            GEOFENCE_EXPIRATION_IN_HOURS * 60 * 60 * 1000;
    public static final float GEOFENCE_RADIUS_IN_METERS =
500;
    public static final HashMap<String, LatLng>
BAY_AREA_LANDMARKS = new HashMap<String, LatLng>();
    static {
        BAY_AREA_LANDMARKS.put("Stasiun Bogor", new
LatLng(-6.595576, 106.790457));
    }
}
```

Isi kode dari *class* pembuatan *geofence* menyerupai dengan semua daftar *class* dari nama stasiun, yang berbeda hanya posisi *latitude* dan *longitude* di setiap stasiunya.

e. Selanjutnya adalah memasukan dua *widget button* pada *main_activity*.xml, *button* satu digunakan untuk mengaktifkan alarm dan membuat radius atau daerah *geofence* secara virtual. Sedangkan *button* dua digunakan untuk menghapus *geofence* dan menghentikan atau menonaktifkan notifikasi aplikasi ketika alarm berbunyi. Berikut adalah potongan kode xml dan java pada dua *button* tersebut :

1. *Widget button* satu

```
<Button
    android:paddingLeft="50dp"
    android:id="@+id/addgeofences"
    android:layout_width="186dp"
    android:layout_height="186dp"
    android:background="@drawable/btn"
    android:layout_weight="0.28"
    android:onClick="addgeo"
    android:layout_centerHorizontal="true" />
```

2. *Widget button* dua

```
<Button
    android:id="@+id/offgeofences"
    android:layout_width="186dp"
    android:layout_height="186dp"
    android:layout_weight="0.28"
    android:background="@drawable/button_click"
    android:onClick="remgeo"
    android:paddingLeft="50dp"
    android:layout_alignParentBottom="true"
    android:layout_alignLeft="@+id/addgeofences"
    android:layout_alignStart="@+id/addgeofences"
    android:layout_marginBottom="65dp" />
```

Pada potongan xml diatas, *button* satu dan dua sengaja dibuat bertumpuk agar terihat seperti tombol *on/off* pada umumnya, dan terdapat onClick yang berguna apabila salah satu tombol di klik maka tombol otomatis tidak dapat diklik untuk kedua kalinya, ketika *button* satu di klik maka *button satu* langsung digantikan dengan *button* dua.

Selanjutnya adalah memberikan fungsi pada dua *button* yang telah dibuat, berikut potongan kode nya.

3. *Button* satu

```
public void addgeo(View view) {
    if (!mGoogleApiClient.isConnected()) {
        Toast.makeText(this,
getString(R.string.not_connected),
Toast.LENGTH_SHORT).show();
        return;
    }
```

```
        LocationServices.GeofencingApi.addGeofences(
                mGoogleApiClient,
                getGeofencingRequest(),

                getGeofencePendingIntent()
        ).setResultCallback(this);
    } catch (SecurityException securityException) {.
        logSecurityException(securityException);
    }
```

Button satu ini berfungsi sebagai pembuat *geofence,* saat tombol ini diklik maka akan mengecek apakah *spinner* sudah dipiliha atau belum. Jika sudah maka *button* ini akan meminta layanan dari *GoogleApiClient* untuk pembuatan *geofence.*

4. *Button* dua

```
public void remgeo(View view) {
    if (!mGoogleApiClient.isConnected()) {
        Toast.makeText(this,
getString(R.string.not_connected),
Toast.LENGTH_SHORT).show();
        return;
    }
    try {
        LocationServices.GeofencingApi.removeGeofences(
                mGoogleApiClient,

                getGeofencePendingIntent()
        ).setResultCallback(this);
        GeofenceTransitionsIntentService.r.stop();
        GeofenceTransitionsIntentService.v.cancel();
```

Pada *button* dua ini berfungsi sebaliknya dari *button* satu, *button* ini digunakan untuk menghapus *geofence* yang sudah dibuat serta menghentikan suara dan getar dari notifikasi yang diberikan *smartphone* ketika telah mendekati stasiun tujuan.

3.5.7 Pembuatan *Menu Drawer*

Menu Drawer adalah *menu* atau navigasi yang tersembunyi pada sebelah kiri atau sebelah kanan layar android. *Menu* ini dapat ditampilkan apabila pengguna menggeser dari tepi kiri layar atau menekan tombol *toggle* yang tedapat di sudut atas kiri layar. Berikut adalah cara pembuatan *menu* drawer :

a. Buatlah kode xml di dalam tampilan *main_activity*.xml. *menu drawer* ini sudah tersedia dia di library android studio, penulis hanya perlu meng *import* dan mendeklarasikan library yang digunakan. Berikut kode xml pembuatan navigasi *drawer* :

```
 <android.support.design.widget.NavigationView
            android:id="@+id/navigation"
            android:layout_height="match_parent"
            android:layout_width="match_parent"
            app:menu="@menu/menu_drawer"
            android:layout_gravity = "bottom|start"
            app:headerLayout="@layout/drawer_header"
            >

        </android.support.design.widget.NavigationView>

</android.support.v4.widget.DrawerLayout>
```

Dari kode diatas penulis dapat menggunakan library dari </android.support.v4.widget.DrawerLayout> untuk menggunakan dan memanggil library *drawer*. *Drawer* ini di beri id *menu_drawer* agar dapat dipanggil oleh fungsi di dalam *main_activity.java* untuk menampilkan *drawer* dan menyembunyikan *drawer* dan juga untuk membuat action klik dari setiap daftar *menu* yang terdapat di *drawer* tersebut.

b. Setelah membuat navigasi *drawer*, langkah selanjutnya adalah membuat file xml untuk membuat header dari menu *drawer* yang dapat di isi tulisan dan gambar untuk judul dari navigasi. Buatlah terlebih dahulu file xml yang bernama *drawer_header*.xml. Untuk tampilan ini hanya perlu membuat file xml nya saja tanpa perlu membuat file.Java dari xml ini. Berikut kode xml untuk header navigasi *drawer. :*

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="140dp"
    android:orientation="vertical"
    tools:background="#c0392b">

    <TextView
        android:textStyle="bold"
        android:padding="20dp"
        android:id="@+id/textView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="12dp"
        android:text="Pengingat Stasiun Tujuan Menu"
        android:textSize="30dp"
        android:layout_alignParentBottom="true"
        android:layout_alignParentLeft="true"
```

Pada kode diatas *header* navigasi Pengingat Stasiun Tujuan hanya berisi *widget textview* yang hanya berisi tulisan judul *menu.*

c. Setelah tampilan sudah dibuat, langkah selanjutnya adalah mengisi daftar *menu* apa saja yang akan disediakan pada *menu drawer*. Lalu penulis membuat file xml terlebih dahulu untuk membuat tampilan dibawah *header* yang telah penulis buat pada langkah sebelumnya. Buatlah xml bernama *menu_drawer*.xml pada layout. kode dari menu_drawer.xml

```
<?xml version="1.0" encoding="utf-8"?>
<menu
xmlns:android="http://schemas.android.com/apk/res/android"
    >

    <group android:checkableBehavior="single">
    <item android:id="@+id/home"
        android:icon="@drawable/ic_home_black_24dp"
        android:title="Home"/>

    <item android:id="@+id/bantuan"
        android:icon="@drawable/ic_error_black_24dp"
        android:title="Bantuan"/>

    <item android:id="@+id/tentang"
    android:icon="@drawable/ic_person_outline_black_24dp"
        android:title="Tentang"/>
```

```
    <item android:id="@+id/pengaturan"
        android:icon="@drawable/ic_menu_manage"
        android:title="Pengaturan"/>
    </group>

    <item android:title = "Lain nya">
        <menu>
            <item
        android:id="@+id/bagikan"
        android:icon="@drawable/ic_menu_share"
        android:title="Bagikan"/>

    <item android:id="@+id/keluar"

android:icon="@drawable/ic_power_settings_new_black_24dp"
        android:title="Keluar"/>
            </menu>
    </item>

</menu>
```

dari kode diatas terdapat enam daftar *menu* yang disediakan di aplikasi, *menu* tersebut diantaranya home, bantuan,tentang, pengaturan, bagikan dan keluar. setiap *menu* diberikan *icon* dan *id. Icon* digunakan untuk memudahkan untuk pemilihan *menu* dan membuat tampilan menjadi lebih menarik, icon yang digunakan sudah tersedia pada android studio sehingga tidak perlu untuk mencari gambar *icon* dan memasukannya ke dalam android stuio. *Id* ini digunakan untuk pemanggilan fungsi di *MainActivity.java.*

d. Selanjutnya penulis adalah memberikan fungsi kepada *menu drawer* agar dapat digunakan. Fungsi ini adalah pedeklarasian *drawer* dan pemanggilan *id drawer* pada *main_activity.java* agar *drawer* dapat muncul dan bergeser ketika *buttin toggle* diklik atau menggeser layar dari tepi kiri. Berikut adalah kode java *drawer* pada *main_activity*.java :

```
drawerLayout = (DrawerLayout) findViewById(R.id.drawer);
toggle = new ActionBarDrawerToggle(MainActivity.this,
drawerLayout, R.string.open, R.string.close);

drawerLayout.addDrawerListener(toggle);
toggle.syncState();

getSupportActionBar().setDisplayHomeAsUpEnabled(true);
getSupportActionBar().setHomeButtonEnabled(true);
```

e. Langkah terakhir pembuatan navigasi *drawer* adalah memberikan fungsi atau action ketika salah satu *menu* dalam navigasi dipilih akan melakukan sesuatu atau menamipilkan sesuatu, berikut adalah kode fungsi untuk memberikan action pada navigasi *drawer :*

```
naigation = (NavigationView)
findViewById(R.id.navigation);
navigation.setNavigationItemSelectedListener(new
NavigationView.OnNavigationItemSelectedListener() {
    @Override
    public boolean onNavigationItemSelected(MenuItem
menuItem) {
        int id = menuItem.getItemId();
        switch (id){
            case R.id.home:
              i  = new
Intent(getApplicationContext(), MainActivity.class);
                startActivity(i);
                finish();
                break;
```

Kode diatas berfungsi untuk menjalakan pilihan *menu* home pada *menu* aplikasi. Pada menu home instruksi yang dijalankan pertama kali adalah memanggil *id* home dan membuka halaman utama kembali.

```
case R.id.bantuan:
i =  new Intent(getApplication(), Bantuan.class);
            startActivity(i);
            break;
```

Diatas adalah kode atau fungsi untuk pemilihan *menu* bantuan pada aplikasi . Pada *menu* bantuan ini berfungsi untuk pindah tampilan yang berawal dari tampilan home ke tampilan bantuan.

```
case R.id.tentang:
AlertDialog.Builder info = new
AlertDialog.Builder(context);
info.setMessage("Aplikasi Pengingat Stasiun Tujuan versi
1.0 beta")
.setCancelable(false).setPositiveButton("OKE",
                        new AlertDialog.OnClickListener()
{
                            @Override
                            public void
onClick(DialogInterface arg0, int i) {
                                arg0.cancel();
                            }
                        });
```

```
                AlertDialog Dialog = info.create();
                Dialog.setTitle("Tentang Aplikasi");
                Dialog.show();
                TextView text = (TextView)
Dialog.findViewById(android.R.id.message);
                text.setTextSize(20);
                break;
```

Diatas adalah kode atau fungsi untuk pemilihan *menu* tentang pada aplikasi. Pada *menu* tentang ini aplikasi menampilkan *messagebox* yang sudah terdapat pada library android studio. *Messagebox* tersebut berisi tentang nama aplikasi dan versi aplikasi Pengingat Stasiun Tujuan.

```
    case R.id.pengaturan:
    i =  new Intent(getApplication(), Setting.class);
    startActivity(i);
    break;
```

Diatas adalah kode atau fungsi untuk pemilihan *menu* pengaturan pada aplikasi . Pada *menu* pengaturan ini berfungsi untuk pindah tampilan yang berawal dari tampilan home ke tampilan pengaturan.

```
Intent i = new Intent(Intent.ACTION_SEND);
String Judul = "Pengingat Stasiun Tujuan";
i.setType("text/plain");
i.putExtra(Intent.EXTRA_SUBJECT, Judul);
String sAux = "Jadikan aplikasi ini sebagai pengingat
stasiun tujuan anda dalam perjalan menggunakan
CommuterLine\n\n";
sAux = sAux + "https://ibb.co/k87Opa \n\n";
i.putExtra(Intent.EXTRA_TEXT, sAux);
startActivity(Intent.createChooser(i, "Sharing Apps"));
return true;
```

Kode diatas berfungsi untuk membagikan dan mengirimkan informasi tentang adanya aplikasi Pengingat Stasiun Tujuan yang berfungsi sebagai pengingat stasiun tujuan bagi pengguna jasa *commuterline*.

```
case R.id.keluar:
AlertDialog.Builder keluar = new
AlertDialog.Builder(context);
                keluar.setTitle("Keluar Aplikasi");
                keluar.setMessage(" Anda Yakin Akan Keluar
? ");
                keluar.setCancelable(false);
                keluar.setPositiveButton("Ya", new
AlertDialog.OnClickListener() {
 @Override
 public void onClick(DialogInterface arg0, int arg1) {
   MainActivity.this.finish();
                    }
                });
                keluar.setNegativeButton("Tidak", new
DialogInterface.OnClickListener() {
@Override
public void onClick(DialogInterface arg0, int arg1) {
 arg0.cancel();
 }
  });
 AlertDialog confirm = keluar.create();
confirm.show();
 break;
}
return false;
}
});
```

Kode diatas adalah pilihan *menu* keluar pada aplikasi. Saat memilih *menu* keluar pada aplikasi, aplikasi akan mengeluarkan output berupa *messagebox* yang terdapat dua pilihan yaitu ya atau tidak. Jika memilih ya maka aplikasi akan langsung keluar dan tertutup dari *smartphone*, jika tidak *smartphone* tetap menampilkan aplikasi.

3.5.8 Pembuatan Tampilan Bantuan

Tampilan bantuan ini berfungsi untuk memberikan informasi kepada pengguna aplikasi Pengingat Stasiun Tujuan. Pada tampilan ini menggunakan *imageview* dan *textview* yang berisi langkah-langkah cara menggunakan aplikasi. Seperti pembuatan tampilan sebelumnya, pada tampilan diperlukan pembuatan file xml pada aplikasi Pengingat Stasiun Tujuan. Pada tampilan bantuan ini menggunakan *widget imageview, textview* dan *view.* Berikut kode xml tampilan bantuan sebagai berikut :

```
<ScrollView
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"

tools:context="com.google.android.gms.location.sample.geofe
ncing.Bantuan"
    >
    <RelativeLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content">
        <TextView
            android:id="@+id/judulbantuan"
            android:textStyle="bold"
            android:text="Bantuan"
            android:textColor="#000000"
            android:textSize="25dp"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginTop="16dp"
            android:layout_alignParentTop="true"
            android:layout_alignParentLeft="true"
            android:layout_alignParentStart="true"
            android:layout_marginLeft="19dp"
            android:layout_marginStart="19dp" />
        <ImageView
            android:id="@+id/bantuan"
            android:layout_marginTop="16dp"
            android:layout_width="150dp"
            android:layout_height="150dp"
            android:background="@mipmap/ic_launcher"
            android:layout_below="@+id/judulbantuan"
            android:layout_centerHorizontal="true" />
```

Kode diatas adalah potongan xml dari tampilan bantuan. Pada tampilan bantuan menggunakan *scrollview* yang berfungsi agar tampilan layar dapat di geser kebawah untuk menampilkan seluruh informasi yang berupa *textview* dapat tersampaikan seluruhnya

```
<TextView
    android:layout_marginLeft="20dp"
    android:layout_marginRight="20dp"
    android:paddingTop="10dp"
    android:layout_below="@+id/bantuan"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textColor="#000000"
    android:text="Aplikasi Pengingat Stasiun Tujuan adalah
aplikasi pengingat stasiun tujuan bagi pengguna jasa kereta
listrik CommuterLine. aplikasi ini dapat dengan cara
sebagai berikut :"
    android:id="@+id/textView5" />

......... // textview langkah-langkah menggunakan aplikasi
<View
    android:layout_width="match_parent"
    android:layout_height="1dp"
    android:layout_marginTop="10dp"
    android:id="@+id/view1"
    android:background="@android:color/darker_gray"
    android:layout_below="@id/textView12"
    >
</View>

<TextView
    android:layout_marginLeft="20dp"
    android:layout_marginRight="20dp"
    android:textSize="10dp"
    android:paddingTop="10dp"
    android:layout_below="@id/view1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textColor="#808080"
    android:text="jika terjadi kesalahan atau bug silahkan
laporkan pada developer dan emailkan ke alamat email
alfhanrf@gmail.com"
    android:id="@+id/textView13" />
```

Pada kode diatas diberikan *widget* berupa *view* untuk memberikan garis batas antara langkah-langkah menggunakan aplikasi dan informasi *developer*.

3.5.9 Pembuatan Tampilan Pengaturan

Tampilan ini dibuat untuk mengatur suara dan getar pada aplikasi yang menggunakan *widget textview , switch,* dan *view*. Berikut kode xml dan java yang digunakan pada pembuatan tampilan ini :

1. potongan kode xml pada tampilan pengaturan

```
<Switch
    android:id="@+id/SwitchGetar"
    android:layout_width="80dp"
    android:layout_height="30dp"
    android:layout_alignLeft="@+id/SwitchSuara"
    android:layout_alignStart="@+id/SwitchSuara"
    android:layout_alignTop="@+id/Getar"
    android:theme="@style/SCBSwitch" />

<TextView
    android:id="@+id/Suara"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Suara"
    android:textColor="#000000"
    android:textSize="20dp"
    android:layout_above="@+id/ketalarm"
    android:layout_alignLeft="@+id/Getar"
    android:layout_alignStart="@+id/Getar" />

.............. // terdapat dua switch dan textview
```

Fungsi *switch* digunakan untuk mengaktifkan dan menonaktifkan getar atau suara. *Textview* digunakan sebagai penjelasan penggunaan *switch.*

2. Potongan kode java *switch* di *main_activity.java*

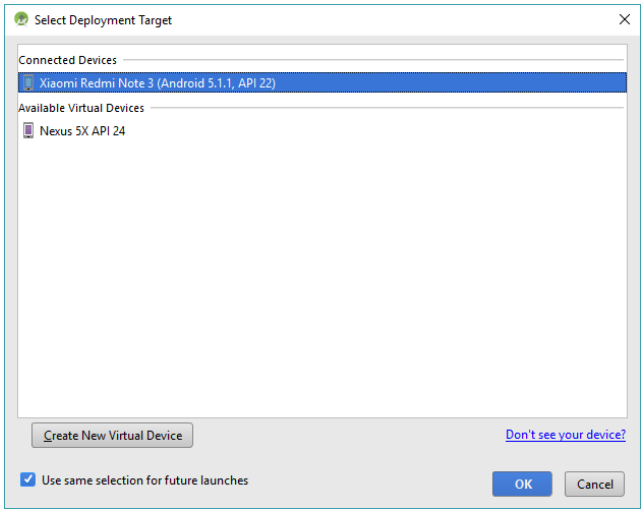
```
SS.setOnCheckedChangeListener(new
CompoundButton.OnCheckedChangeListener() {
    @Override
    public void onCheckedChanged(CompoundButton buttonView,
boolean isChecked) {
        if (SS.isChecked()) {
            SharedPreferences.Editor editor =
getSharedPreferences("com.google.android.gms.location.sampl
e.geofencing", MODE_PRIVATE).edit();
            editor.putBoolean("suara", true);
            editor.commit();
```

```
SS.setOnCheckedChangeListener(new
CompoundButton.OnCheckedChangeListener() {
    @Override
    public void onCheckedChanged(CompoundButton buttonView,
boolean isChecked) {
        if (SS.isChecked()) {
            SharedPreferences.Editor editor =
getSharedPreferences("com.google.android.gms.location.sampl
e.geofencing", MODE_PRIVATE).edit();
            editor.putBoolean("suara", true);
            editor.commit();
```

kode diatas adalah kode pengaturan suara pada aplikasi, jika *switch mode on* maka aplikasi akan memberikan output suara apabila sudah mendekati stasiun tujuan, jika *switch mode off* maka aplikasi hanya memberikan notifikasi pada bar *smartphone*. Untuk kode java pada *switch* di pengaturan getar hampir menyerupai kode pengaturan suara dan fungsi dari *switch* nya sama dengan fungsi *switch* suara.

3.6 Implementasi Aplikasi

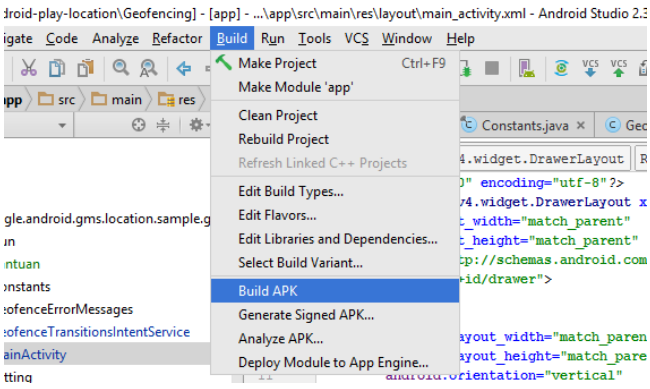
Aplikasi yang sudah dibuat kemudian diimplementasikan pada *smartphone* berbasis Android. Penulis menggunakan ADB Drivers yang sudah di*install* pada langkah ke tiga di subbab 3.8 pada *smartphone* yang dijadikan *device launcher* dengan menghubungkan *smartphone* ke laptop melalui kabel USB. Implementasi aplikasi pada *device launcher* bertujuan untuk memastikan bahwa aplikasi dapat berjalan dengan baik. Untuk melakukan implementasi pada *device launcher*, penulis mengklik tombol *run* yang ada pada *menu bar* di Android Studio. Kemudian akan tampil jendela *device chooser* seperti pada gambar 3.26. Setelah *device* terdeteksi, penulis menceklis *Use same device for future launches* supaya jendela tidak muncul lagi saat di *Run* ulang di *device* yang sama. Kemudian penulis menekan tombol OK. Aplikasi Pengingat Stasiun Tujuan akan muncul secara otomatis pada *smartphone* yang dijadikan *device launcher*.



Gambar 3.27 Jendela *Device Chooser*

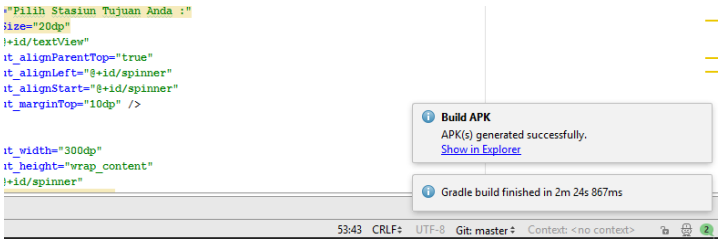
Setelah penulis melakukan implementasi pada *smartphone* yang menjadi *device launcher* dan aplikasi Pengingat Stasiun Tujuan dapat berjalan dengan baik, kemudian penulis membuat *file* .apk karena sistem operasi Android mendukung pemasangan aplikasi dengan format *file Android package (.apk).* Berikut adalah langkah langkah dalam membuat *file* .apk:

1. Langkah pertama, penulis memilih menu *build* dilanjutkan dengan memilih built apk seperti gambar 3.28.



Gambar 3.28 *Menu Build*

2. Setelah penulis memilih *Build APK* android studio langsung mengeksekusi project android yang penulis buat, setelah selesai maka akan ada dialog ketika *APK* telah selesai di *build* dan untuk membuka folder dimana letak *file APK* tersebut seperti gambar 3.29.



Gambar 3.29 *dialog* Notifikasi *APK*

3. Jika *file* .apk sudah terbentuk, maka selanjutnya penulis menyalin *file* .apk tersebut ke *smartphone* Android dengan cara mengirimkannya melalui *bluetooth* atau menyalinnya langsung ke *SDCard* Android. Setelah tersalin, penulis memilih *file StationReminder.apk* yang kemudian di*install*, sehingga Android akan memulai proses peng*installan*.

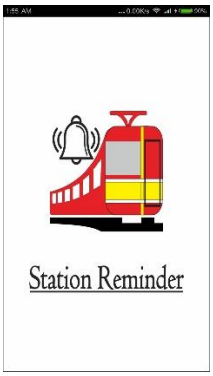
3.7 Uji Coba Aplikasi

Setelah proses implementasi aplikasi Pengingat Stasiun Tujuan pada *device launcher* dan pembuatan *file* .apk selesai, selanjutnya dilakukan uji coba aplikasi Pengingat Stasiun Tujuan. Uji coba akan dilakukan pada lima *device* berbeda dengan menggunakan *file* .apk dari aplikasi Pengingat Stasiun Tujuan yang telah dibuat pada implementasi aplikasi di subbab 3.9. Uji coba aplikasi yang dilakukan pada *smartphone* yang menjadi *device launcher* menghasilkan tampilan *output* aplikasi Pengingat Stasiun Tujuan seperti berikut:

1. Bentuk Icon aplikasi Pengingat Stasiun Tujuan pada *smartphone* setelah terinstall, dapat dilihat pada gambar 3.30

Gambar 3.30 Icon Aplikasi Pengingat Stasiun Tujuan

2. Aplikasi Pengingat Stasiun Tujuan akan muncul pada *device launcher* secara otomatis, pertama diawali dengan *Splashscreen* aplikasi Pengingat Stasiun Tujuan seperti pada gambar 3.31

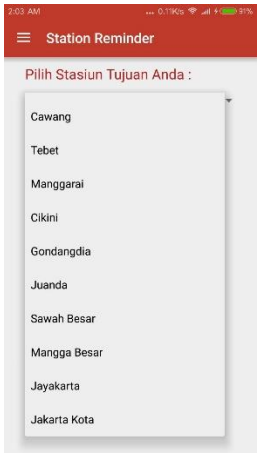


Gambar 3.31 Tampilan Splashscreen

3. Setelah *splashscreen*, aplikasi Pengingat Stasiun Tujuan langsung masuk ke menu utama aplikasi Pengingat Stasiun Tujuan, terdpat *textview*, menu *dropdown* dan *button On/Off* seperti pada gambar 3.32.

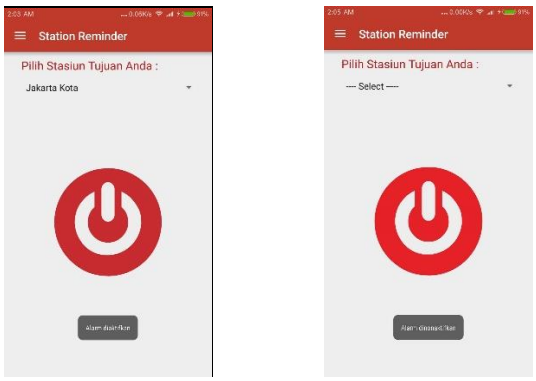
Gambar 3.32 Tampilan *Menu Utama*

4. Ketika *menu dropdown* ditekan maka akan mucul 24 nama stasiun dengan rute Jakartakota menuju Bogor dan sebaliknya, seperti pada gambar 3.33.



Gambar 3.33 Tampilan Isi *Menu Dropdown*

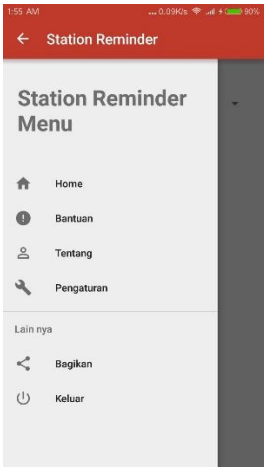
5. Setelah memilih stasiun tujuan dan menekan tombol *On/Off* maka jika alarm baru akan dibuat akan muncul pesan seperti gambar 3.34(a), jika alarm sudah diset dan ingin dimatikan atau mengganti tujuan akan muncul pesan seperti gambar 3.34(b).



(a) (b)

Gambar 3.34 (a) Tampillan Alarm On, (b) Tampilan Alarm *Off*

6. Lalu tampilan daftar *menu drawer* apabila penulis menggeser layar dari tepi kiri ke kanan atau dengan menekan tombol toggle pada kiri atas tampilan, seperti pada gambar 3.35.

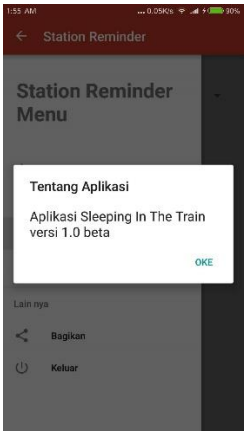


Gambar 3.35 Tampilan Daftar *Menu Drawer*

7. Untuk tampilan *menu home* pada *menu drawer* hanya membuka kembali tampilan utama setelah *splashscreen*.
8. Selanjutnya adalah tampilan bantuan apabila dipilih maka akan berganti *activity* atau tampilan pada aplikasi, Seperti pada gambar 3.36.

Gambar 3.36 Tampilan *Menu* Bantuan

9. Kemudian adalah tampilan *menu* tentang pada aplikasi, jika ditekan akan mengeluarkan *messagebox* atau pesan, seperti pada gambar 3.37.

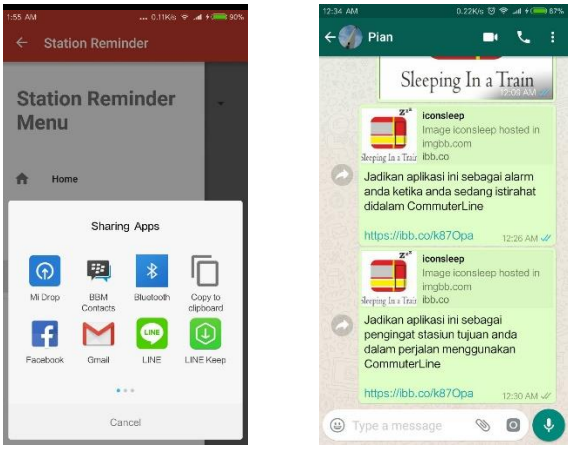


Gambar 3.37 Tampilan Dialog *Messagebox Menu* Tentang

10. Selanjutnya tampilan pengaturan pada aplikasi, jika menu ini dipilih maka akan berganti *activity* atau tampilan. seperti pada gambar 3.38.

Gambar 3.38 Tampilan *Menu* Pengaturan

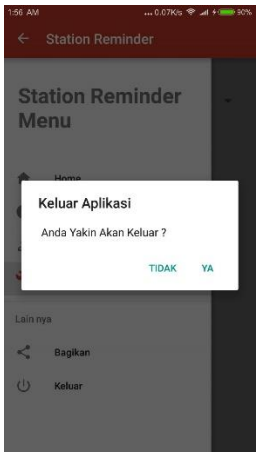
11. Ketika menu bagikan dipilih pada *menu drawer*, maka akan tampil jendela membuka aplikasi pesan untuk membagikan informasi yang dibagikan, seperti pada gambar 3.39 adalah aplikasi yang dapat membagikan informasi dan juga informasi yang dibagikan.



(a) (b)

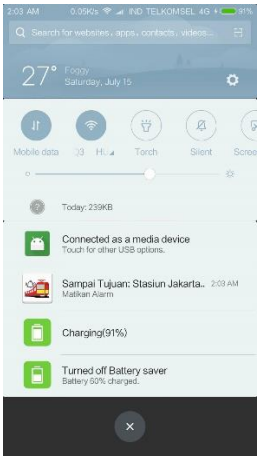
Gambar 3.39 (a) Tampilan *Menu* Bagikan, (b) Informasi yang dibagikan tentang aplikasi.

12. Sedangkan jika menu *Exit* dipilih maka akan muncul *Alert Dialog* seperti pada gambar 3.40.



Gambar 3.40 Tampilan Menu Exit

13. Jika sudah mendekati stasiun tujuan maka aplikasi akan memberikan notifikasi kapada user. Seperti pada gambar 3.41.



Gambar 3.40 Notifikasi Aplikasi Pengingat Stasiun Tujuan

3.7.1 Hasil Perbandingan Aplikasi pada Perangkat Berbeda

Uji coba aplikasi telah dilakukan pada lima *device* yang berbeda spesifikasi resolusi, ukuran layar *smartphone*, sistem operasi yang digunakan di masing-masing *smartphone*, memori internal, dan RAM yang ada pada *smartphone* tersebut. Tabel 3.1 adalah tabel hasil perbandingan uji coba yang dilakukan pada lima *device* berbeda.

Tabel 3.1 Hasil Perbandingan Aplikasi pada Perangkat Berbeda

Merk	Spesifikasi	Hasil
Xiomi Note 3 Pro	Resolusi: Full HD 1080 x 1920 pixels Layar: 5.5 inches OS: Android 5.1.1 (Lollipop) Memori Internal 32GB, RAM 3GB	Tampilan sesuai dengan ukuran layar ponsel. Aplikasi berjalan lancar.
Xiomi Note 3 Pro	Resolusi: Full HD 1080 x 1920 pixels Layar: 5.5 inches OS: Android 5.1.1 (Lollipop)	Tampilan sesuai dengan ukuran layar ponsel. Aplikasi berjalan lancar

	Memori internal 16 GB, RAM 2 GB	
Xiaomi 4x	Resolusi: 720 x 1.280 piksel Layar: 5 inches OS: Android 6.0.1 (Jelly Bean) Memori Internal 32GB, RAM 3GB	Tampilan sesuai dengan ukuran layar ponsel. Aplikasi berjalan lancar.
Samsung Galaxy J2 Prime	Resolusi : 540 x 960 pixels Layar : 4.7 inches OS: Android 5.1.1 (Lollipop) OS: Android 5.1.1 (Lollipop) Memori Internal 8, RAM 1GB	Tampilan sesuai dengan ukuran layar ponsel. Aplikasi berjalan lancar.
Samsung Galaxy Note 4	Resolusi : 1440 x 2560 pixels Layar : 5.7 inches OS: Android 6.0.1 (Jelly Bean) Memori Internal 32GB, RAM 3GB	Tampilan sesuai dengan ukuran layar ponsel. Aplikasi berjalan lancar.

Tabel 3.1 Hasil Perbandingan Aplikasi pada Perangkat Berbeda

Dari tabel 3.1 dapat disimpulkan bahwa aplikasi berjalan dengan baik di lima *device* yang berbeda spesifikasi resolusi, ukuran layar smatphone, sistem operasi, memori internal, dan RAM.

3.8 Evaluasi

Pada tahap ini merupakan tahapan evaluasi yang merupakan tahapan yang dilakukan setelah implementasi dan uji coba. Tahapan evaluasi ini diharapakan dapat membantu penulis dalam memperoleh pendapat mengenai aplikasi yang telah penulis buat. Untuk melakukan evaluasi penulis menggunakan metode kuisioner.

Pembuatan kuisioner dibantu dengan fasilitas *google docs,* yaitu *form online* untuk pengisian kuisioner. Pada kuisioner ini terdapat lima buah pertanyaan yang wajib untuk dijawab. Di dalam kuisioner ini terdapat empat buah pilihan jawaban yaitu sangat setuju, setuju, tidak setuju, sangat tidak setuju. Berikut ini merupakan daftar pertanyaan kuisioner:

1. Tampilan aplikasi Pengingat Stasiun Tujuan *simple* dan menarik.
2. Aplikasi Pengingat Stasiun Tujuan berjalan dengan lancar pada perangkat *smartphone* anda.
3. Aplikasi Pengingat Stasiun Tujuan mudah digunakan.
4. Aplikasi Pengingat Stasiun Tujuan bermanfaat sebagai pengingat stasiun tujuan anda pada saat anda di *commuterline.*
5. Aplikasi Pengingat Stasiun Tujuan tidak menggangu kinerja dari *smartphone,* seperti daya baterai, suhu dan memori.

3.8.1 Hasil Kuisioner

Kuisioner telah berhasil diisi oleh 24 responden, dengan setiap pertanyaan memiliki hasil yang berbeda – beda. Perhitungan kuisioner menggunakan rumus perhitungan skala Likert. Berikut adalah Rumus perhitungan skala likert:

a. Nilai Jawaban(NJ)

   - Sangat Setuju(SS)= 4
   - Setuju (S)= 3
   - Tidak Setuju(TS)= 2
   - Sangat Tidak Setuju(STS)= 1

b. Rumus Mencari Nilai Total Jawaban

TJ = ((T(SS) x NJ(SS)) + (T(S) x NJ(S)) + (T(TS) x NJ(TS)) + (T(STS) x NJ(STS))

Ket. [TJ = Nilai Total jawaban, T = Total Responden, NJ= Nilai Jawaban]

c. Interpretasi Jawaban Perhitungan

Y = NJ Tertinggi x Jumlah Responden

X = NJ Terendah x Jumlah Responden

d. Rumus Interval :

I = 100 / Jumlah Pilihan Jawaban

Maka I = 100 / 4 = 25

Berikut kriteria interpretasi jawaban berdasarkan interval

- Angka 0% - 24,99% = Sangat Tidak Setuju
- Angka 25% - 49,99% = Tidak Setuju
- Angka 50% - 74,99% = Setuju
- Angka 75% - 100% = Sangat Setuju

e. Penyelesaian Akhir % = TJ/Y x 100

Berikut ini penulis akan melampirkan rumus perhitungan skala likert beserta hasil hitungan dari masing-masing pertanyaan yang terdapat pada kuisioner. Tabel 3.2 merupakan hasil perhitungan kuisioner dari masing-masing pertanyaan.

1. Tampilan Aplikasi Pengingat Stasiun Tujuan *simple* dan menarik.

Jawaban	Jumlah Responden	Nilai Persen
Sangat Setuju	8	33.33%
Setuju	9	37.50%
Tidak Setuju	6	25.00%
Sangat Tidak Setuju	1	4.16%

Tabel 3.2 Hasil Kuisoner Pertanyaan 1

Rumus Perhitungan Skala Likert :

- NJ : SS = 4, S = 3, TS = 2, STS = 1
- Mencari Nilai Total Jawaban :
  TJ = ((8x4) + (9x3) + (6 x 2) + (1x1))
  TJ = 24 + 27 + 12 + 1 = 64
- Interpretasi Jawaban Perhitungan :
  Y = 4 x 24 = 98
  N = 1 x 24 = 24
- Penyelesaian Akhir :
  Penyelesaian Akhir % = 64/98 x 100
  Penyelesaian Akhir % = 65.30 % (kategori Setuju)

2. Aplikasi Pengingat Stasiun Tujuan berjalan dengan lancar pada perangkat *smartphone* anda.

Jawaban	Jumlah Responden	Nilai Persen

Sangat Setuju	8	33.33%
Setuju	16	66,66%
Tidak Setuju	0	0%
Sangat Tidak Setuju	0	%

Tabel 3.3 Hasil Kuisoner Pertanyaan 2

Rumus Perhitungan Skala Likert :

- NJ : SS = 4, S = 3, TS = 2, STS = 1
- Mencari Nilai Total Jawaban :

  TJ = ((8x4) + (16x3) + (0 x 2) + (0x1))

  TJ = 32 + 48 + 0 + 0 = 80
- Interpretasi Jawaban Perhitungan :

  Y = 4 x 24 = 98

  N = 1 x 24 = 24
- Penyelesaian Akhir :

  Penyelesaian Akhir % = 80/98 x 100

  Penyelesaian Akhir % = 81, 63 % (kategori Sangat Setuju)

3. Aplikasi Pengingat Stasiun Tujuan mudah digunakan.

Jawaban	**Jumlah Responden**	**Nilai Persen**
Sangat Setuju	11	45,83%
Setuju	13	54,16%
Tidak Setuju	0	0%
Sangat Tidak Setuju	0	%

Tabel 3.4 Hasil Kuisoner Pertanyaan 3

Rumus Perhitungan Skala Likert :

- NJ : SS = 4, S = 3, TS = 2, STS = 1

- Mencari Nilai Total Jawaban :

  TJ = ((11x4) + (13x3) + (0 x 2) + (0x1))

  TJ = 44+ 39 + 0 + 0 = 83

- Interpretasi Jawaban Perhitungan :

  Y = 4 x 24 = 98

  N = 1 x 24 = 24

- Penyelesaian Akhir :

  Penyelesaian Akhir % = 83/98 x 100

  Penyelesaian Akhir % = 84,69 % (kategori Sangat Setuju)

4. Aplikasi Pengingat Stasiun Tujuan bermanfaat sebagai pengingat stasiun tujuan anda pada saat anda di *commuterline.*

Jawaban	**Jumlah Responden**	**Nilai Persen**
Sangat Setuju	9	37,50%
Setuju	15	62.50%
Tidak Setuju	0	0%
Sangat Tidak Setuju	0	%

Tabel 3.5 Hasil Kuisoner Pertanyaan 4

Rumus Perhitungan Skala Likert :

- NJ : SS = 4, S = 3, TS = 2, STS = 1
- Mencari Nilai Total Jawaban :

  TJ = ((9x4) + (15x3) + (0 x 2) + (0x1))

  TJ = 36+ 45 + 0 + 0 = 81

- Interpretasi Jawaban Perhitungan :

  Y = 4 x 24 = 98

  N = 1 x 24 = 24

- Penyelesaian Akhir :

  Penyelesaian Akhir % = 81/98 x 100

  Penyelesaian Akhir % = 82,65 % (kategori Sangat Setuju)

5. Aplikasi Pengingat Stasiun Tujuan tidak menggangu kinerja dari *smartphone*, seperti daya baterai, suhu dan memori.

Jawaban	Jumlah Responden	Nilai Persen
Sangat Setuju	3	12,50%
Setuju	16	62.50%
Tidak Setuju	5	25,00%
Sangat Tidak Setuju	0	0%

Tabel 3.6 Hasil Kuisoner Pertanyaan 5

Rumus Perhitungan Skala Likert :

- NJ : SS = 4, S = 3, TS = 2, STS = 1
- Mencari Nilai Total Jawaban :

  TJ = ((3x4) + (15x3) + (5 x 2) + (0x1))

  TJ = 12+ 45 + 10 + 0 = 77
- Interpretasi Jawaban Perhitungan :

  Y = 4 x 24 = 98

  N = 1 x 24 = 24
- Penyelesaian Akhir :

  Penyelesaian Akhir % = 77/98 x 100

  Penyelesaian Akhir % = 78,57 % (kategori Sangat Setuju)

Nilai rata – rata (NR) dari semua jawaban :

(NR) = Total Nilai Perhitungan Skala Likert / Jumlah Pertanyaan

(NR) = (65.30+ 81,63+ 84,69+ 82,65 + 78,57) / 5

(NR) = 392,84 / 5

(NR) = 78,568 %

Dari hasil perhitungan Skala Likert pada Tabel 3.2, 3.3, 3.4, 3.5 dan 3.6 dapat disimpulkan bahwa :

- 65.30% responden Setuju bahwa aplikasi Pengingat Stasiun Tujuan memiliki tampilan yang *simple* dan menarik.
- 81,63% responden Sangat Setuju bahwa aplikasi Pengingat Stasiun Tujuan dapat berjalan baik pada *smartphone.*
- 84,69% responden Sangat Setuju bahwa aplikasi Pengingat Stasiun Tujuan mudah digunakan atau *user friendly.*
- 82,65 % responden Sangat Setuju bahwa aplikasi Pengingat Stasiun Tujuan bermanfaat sebagai pengingat stasiun tujuan bagi pengguna jasa *commuterline.*
- 78,57% responden Sangat Setuju bahwa aplikasi Pengingat Stasiun Tujuan tidak menggangu kinerja dari *smartphone*, seperti daya baterai, suhu dan memori.

Kemudian setelah dilakukan perhitungan nilai rata-rata dari seluruh hasil kuisioner, dapat disimpulkan bahwa 78,56% dari 24 responden memberikan tanggapan positif, aplikasi Pengingat Stasiun Tujuan dapat menjadi pengingat stasiun bagi pengguna jasa *commuterline*, memiliki tampilan yang menarik dan *simple* sehingga mudah digunakan, dan tidak menggunakan banyak *resource* pada *smartphone*.