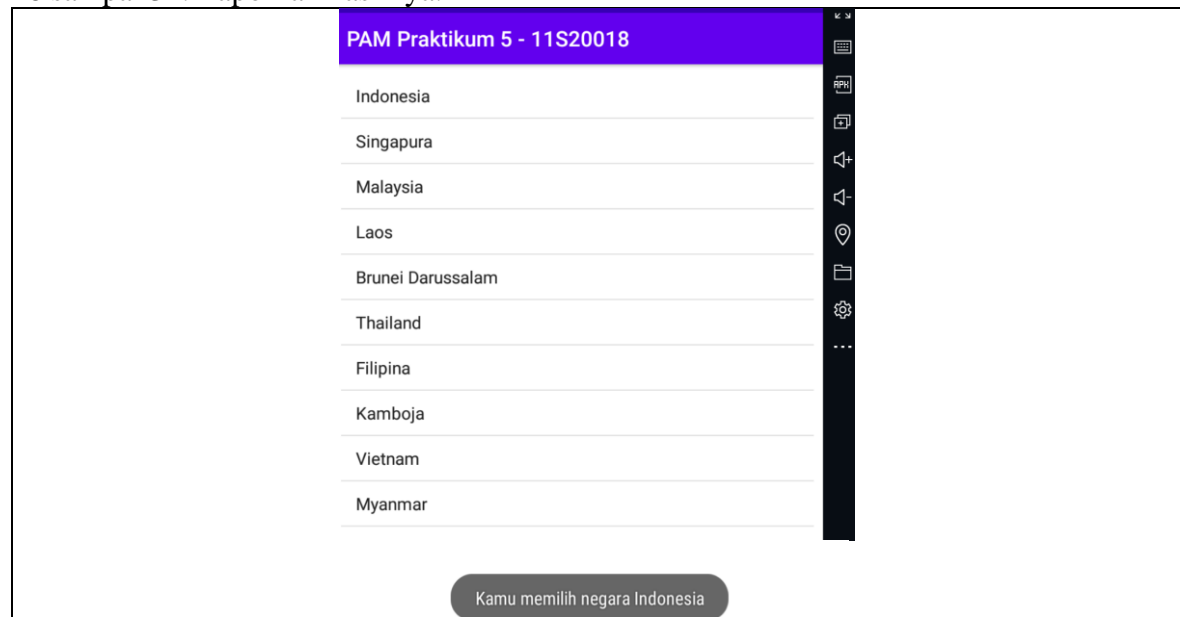


NIM	:	11S20018
Nama	:	Ruth Aulya Silalahi
Kelas	:	13IF1
Video Source Code	:	https://youtu.be/NwWQHDji-L8

A. List & View

List View

Jelaskan tujuan dari kode program pada line 19 sampai line 24 dan kode program pada line 26 sampai 34. Laporkan hasilnya.



Pada tahapan line 19 sampai line 24 merupakan kode program dalam bahasa pemrograman Kotlin untuk membuat sebuah ArrayAdapter di aplikasi Android. ArrayAdapter tersebut digunakan untuk menampilkan daftar negara-negara yang tergabung dalam ASEAN dalam bentuk list pada antarmuka pengguna aplikasi. Sedangkan pada line 26 sampai line 34 merupakan kode program dalam bahasa pemrograman Kotlin untuk menambahkan event listener pada ListView yang menampilkan daftar negara-negara ASEAN. Event listener tersebut akan merespon ketika pengguna melakukan klik pada salah satu negara yang ditampilkan, dan menampilkan pesan Toast dengan nama negara yang dipilih.

Recycler View

Cari tahu apa fungsi dari `OnCreateViewHolder`, `OnBindViewHolder`, dan `getItemCount`. Laporkan hasilnya.

`onCreateViewHolder` berfungsi untuk membuat objek `ViewHolder`, yang digunakan untuk menampilkan tampilan item pada `RecyclerView`. Fungsi ini akan dijalankan ketika `RecyclerView` memerlukan sebuah objek `ViewHolder` baru, dan akan membuatnya

Recycler View

dengan menggunakan layout yang telah ditentukan. Fungsi ini akan mengembalikan objek ViewHolder yang sudah dibuat.

onBindViewHolder berfungsi untuk mengikat data ke dalam tampilan item pada ViewHolder. Fungsi ini akan dijalankan setiap kali RecyclerView menampilkan sebuah tampilan item pada layar, dan akan mengambil data dari sumber data dan menampilkannya pada tampilan item yang terdapat pada ViewHolder yang sedang ditampilkan.

getItemCount berfungsi untuk mengembalikan jumlah total item pada daftar data yang akan ditampilkan pada RecyclerView. Fungsi ini akan dijalankan ketika RecyclerView memerlukan informasi tentang jumlah total item yang akan ditampilkan. Fungsi ini harus mengembalikan sebuah nilai integer yang merepresentasikan jumlah total item yang akan ditampilkan pada RecyclerView.



Grid View

Cari tahu kegunaan dari BaseAdapter() beserta fungsi-fungsinya. Laporkan hasilnya.

BaseAdapter adalah kelas abstrak dalam Android SDK yang digunakan sebagai dasar untuk membuat adapter yang menghubungkan data dengan antarmuka pengguna. Fungsi utama dari BaseAdapter adalah untuk menghubungkan data dengan antarmuka pengguna pada ListView atau GridView. Untuk mengimplementasikan BaseAdapter, kita perlu menyesuaikan kelas adapter kustom yang diwarisi dari BaseAdapter.

Beberapa fungsi dari BaseAdapter antara lain:

1. **getView()**: Method ini digunakan untuk mengembalikan tampilan yang akan ditampilkan pada posisi tertentu dalam ListView atau GridView. Method ini harus diimplementasikan oleh adapter kustom.

Grid View

2. `getCount()`: Method ini digunakan untuk mengembalikan jumlah item dalam data yang akan ditampilkan dalam ListView atau GridView.
3. `getItem()`: Method ini digunakan untuk mengembalikan item data pada posisi tertentu dalam adapter.
4. `getItemId()`: Method ini digunakan untuk mengembalikan ID item pada posisi tertentu dalam adapter.
5. `getViewTypeCount()` dan `getItemViewType()`: Method ini digunakan untuk mendukung multiple view types dalam adapter, misalnya untuk menampilkan tampilan yang berbeda untuk item data yang berbeda.
6. `notifyDataSetChanged()`: Method ini digunakan untuk memberi tahu adapter bahwa data yang ditampilkan telah berubah dan perlu diperbarui pada tampilan.

PAM Praktikum 5 - 11S20018



Elephant



Giraffe



Gorilla



Hippo



Lion



Panda



Rabbit



Rhino

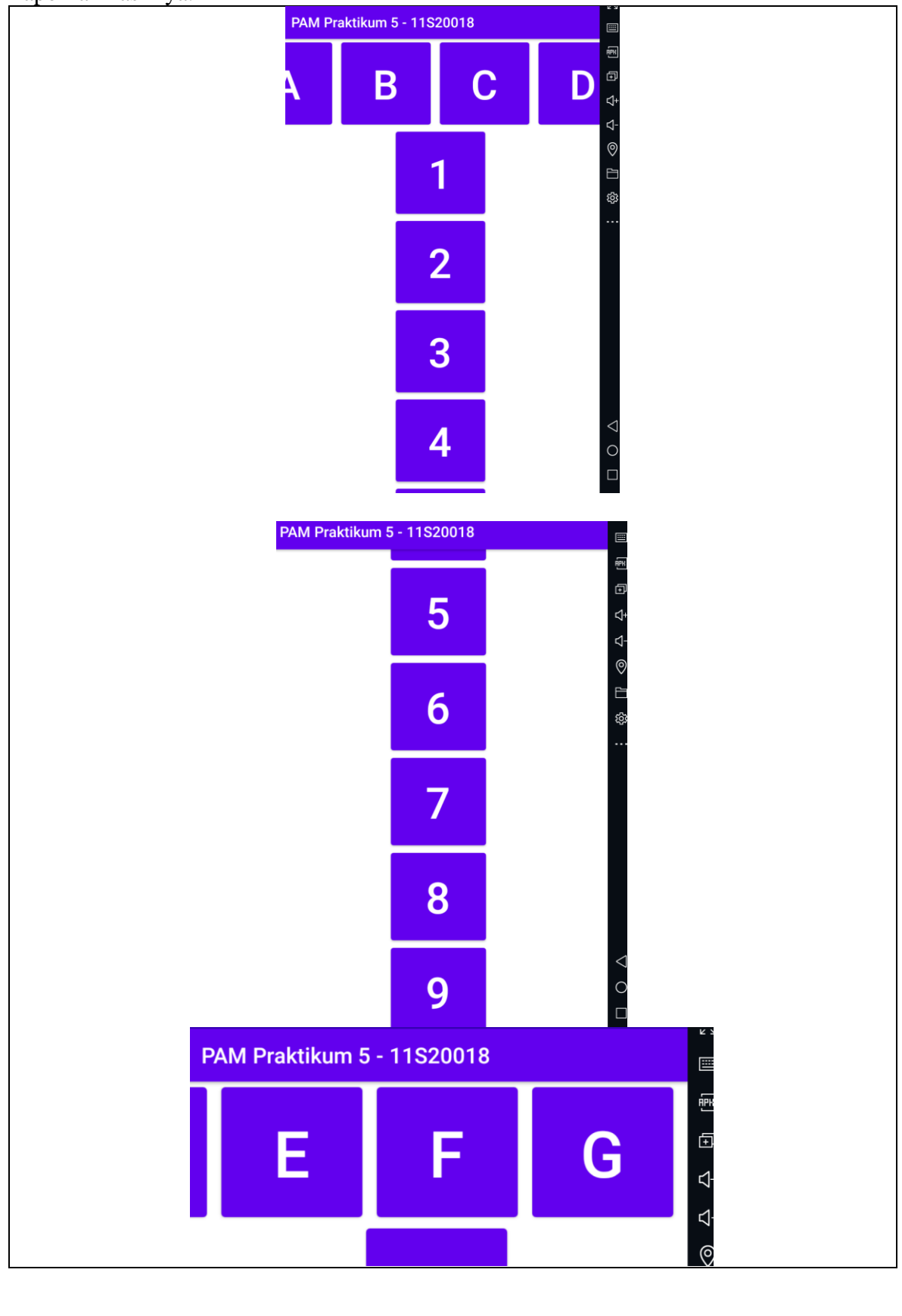


Zebra

Kamu memilih hewan Giraffe.

Scroll View

Laporkan hasilnya.



Web View

Cari tahu kegunaan dari fungsi `onBackPressed()`. Laporkan hasilnya.

`onBackPressed()` adalah sebuah fungsi dalam Android SDK yang digunakan untuk menangani tombol "back" pada perangkat Android. Ketika tombol "back" ditekan, sistem Android akan memanggil fungsi `onBackPressed()` yang dapat di-override oleh aktivitas (activity) untuk menentukan tindakan yang diambil ketika tombol "back" ditekan.

PAM Praktikum 5 - 11S20018

Halaman web tidak tersedia

Halaman web di <https://delcom.org/> tidak dapat dimuat karena:

net::ERR_CACHE_MISS

Caari tahu fungsi dari:

```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
dan
android:usesCleartextTraffic="true"
```

`<uses-permission android:name="android.permission.INTERNET"/>`: Digunakan untuk memberikan aplikasi akses ke internet. Dengan izin ini, aplikasi dapat melakukan koneksi ke internet seperti mengambil data dari server atau melakukan permintaan HTTP.

`<uses-permission android:name="android.permission.ACCESS_WIFI_STATE"/>`: Digunakan untuk memeriksa status koneksi WiFi pada perangkat. Dengan izin ini, aplikasi dapat memeriksa apakah perangkat terhubung ke jaringan WiFi atau tidak.

`android:usesCleartextTraffic="true"`: Digunakan untuk mengizinkan aplikasi untuk melakukan koneksi melalui HTTP tanpa enkripsi (clear text). Opsi ini biasanya diaktifkan jika aplikasi memerlukan koneksi melalui HTTP tanpa SSL/TLS (misalnya pada pengembangan aplikasi yang memerlukan debug atau testing).

PAM Praktikum 5 - 11S20018

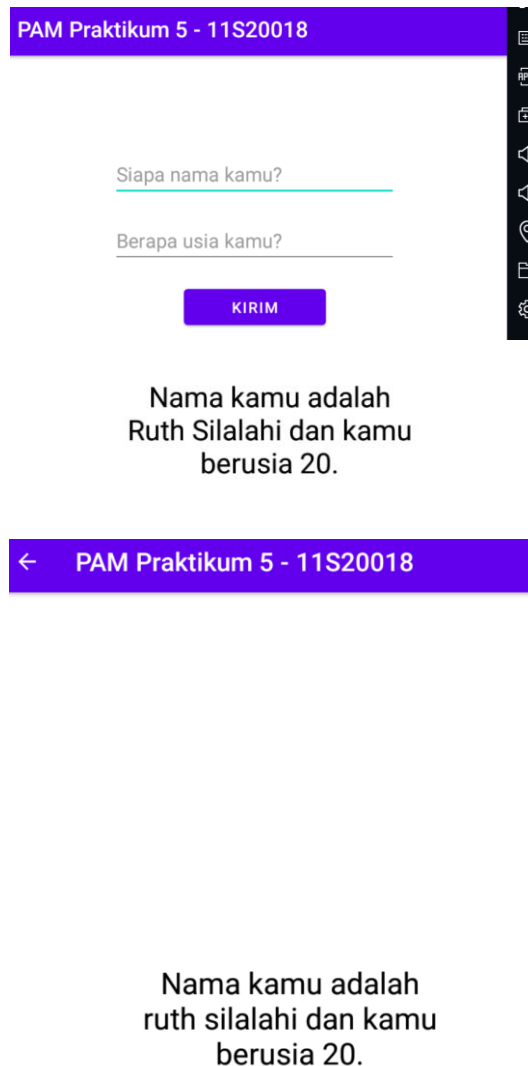


B. Intent and Lifecycles

Intent

Cari tahu apa kegunaan kode program pada line 28 sampai 30, line 32 dan line 34. Laporkan hasilnya.

1. `var intent= Intent(this@MainActivity, SecondActivity::class.java)`: Membuat objek Intent yang akan digunakan untuk membuka aktivitas baru (SecondActivity). `this@MainActivity` menunjukkan aktivitas saat ini, sedangkan `SecondActivity::class.java` menunjukkan kelas aktivitas tujuan yang akan dibuka.
2. `intent.putExtra("name", name)`: Mengirimkan data dengan nama "name" dan nilai dari variabel "name" ke aktivitas tujuan (SecondActivity).
3. `intent.putExtra("age", age)`: Mengirimkan data dengan nama "age" dan nilai dari variabel "age" ke aktivitas tujuan (SecondActivity).
4. `startActivity(intent)`: Memulai aktivitas baru (SecondActivity) dengan menggunakan objek Intent yang telah dibuat sebelumnya.



Application Lifecycle

Laporkan hasilnya.

```
023-02-25 22:52:35.462 11792-11792/com.example.pampraktikum5_11s20018 D/Message: onPause
023-02-25 22:52:35.487 11792-11792/com.example.pampraktikum5_11s20018 D/Message: onStop
023-02-25 22:52:39.712 11792-11792/com.example.pampraktikum5_11s20018 D/Message: onRestart
023-02-25 22:52:39.713 11792-11792/com.example.pampraktikum5_11s20018 D/Message: onStart
023-02-25 22:52:39.714 11792-11792/com.example.pampraktikum5_11s20018 D/Message: onResume
023-02-25 22:52:51.447 11792-11792/com.example.pampraktikum5_11s20018 D/Message: onPause
023-02-25 22:52:51.736 11792-11792/com.example.pampraktikum5_11s20018 D/Message: onStop
023-02-25 22:52:51.738 11792-11792/com.example.pampraktikum5_11s20018 D/Message: onDestroy
```

Run TODO Problems Terminal Logcat App Inspection Build Profiler

Activity Lifecycle

Laporkan hasilnya.

Try the new Logcat with improved formatting and filtering options.

```
2023-02-25 23:10:44.967 14724-14724/com.example.pampraktikum5_11s20018 D/Message: Main Activity onCreate
2023-02-25 23:10:44.970 14724-14724/com.example.pampraktikum5_11s20018 D/Message: Main Activity onStart
2023-02-25 23:10:44.971 14724-14724/com.example.pampraktikum5_11s20018 D/Message: Main Activity onResume
2023-02-25 23:11:39.453 14724-14724/com.example.pampraktikum5_11s20018 D/Message: Main Activity onPause
2023-02-25 23:11:39.488 14724-14724/com.example.pampraktikum5_11s20018 D/Message: Second Activity onCreate
2023-02-25 23:11:39.489 14724-14724/com.example.pampraktikum5_11s20018 D/Message: Second Activity onStart
2023-02-25 23:11:39.489 14724-14724/com.example.pampraktikum5_11s20018 D/Message: Second Activity onResume
2023-02-25 23:11:39.764 14724-14724/com.example.pampraktikum5_11s20018 D/Message: Main Activity onStop
2023-02-25 23:11:46.819 14724-14724/com.example.pampraktikum5_11s20018 D/Message: Second Activity onPause
2023-02-25 23:11:46.855 14724-14724/com.example.pampraktikum5_11s20018 D/Message: Main Activity onCreate
2023-02-25 23:11:46.856 14724-14724/com.example.pampraktikum5_11s20018 D/Message: Main Activity onStart
2023-02-25 23:11:46.858 14724-14724/com.example.pampraktikum5_11s20018 D/Message: Main Activity onResume
2023-02-25 23:11:47.123 14724-14724/com.example.pampraktikum5_11s20018 D/Message: Second Activity onStop
```

Fragment Lifecycle

Cari tahun masing-masing kegunaan dari fungsi fragment di atas. Laporkan hasilnya.

- `onAttach()`: dipanggil ketika Fragment terhubung ke Activity yang menggunakannya.
- `onCreate()`: dipanggil saat Fragment dibuat, digunakan untuk melakukan inisialisasi awal.
- `onCreateView()`: dipanggil ketika tampilan Fragment dibuat dan dapat disesuaikan.
- `onViewCreated()`: dipanggil ketika tampilan Fragment sudah terlihat dan dapat diakses.
- `onStart()`: dipanggil ketika Fragment mulai terlihat oleh pengguna.
- `onResume()`: dipanggil ketika Fragment aktif dan interaktif dengan pengguna.
- `onPause()`: dipanggil ketika pengguna meninggalkan Fragment atau ketika Fragment tidak lagi aktif.
- `onStop()`: dipanggil ketika Fragment tidak lagi terlihat oleh pengguna.
- `onDestroyView()`: dipanggil ketika tampilan Fragment dihancurkan atau tidak lagi dapat diakses.

Fragment Lifecycle

- `onDestroy()`: dipanggil ketika Fragment dihancurkan atau tidak lagi digunakan.
- `onDetach()`: dipanggil ketika Fragment terlepas dari Activity yang menggunakannya.

```
com.example.pampraktikum5_11s20018 I/fragment: onAttach
com.example.pampraktikum5_11s20018 I/fragment: onCreate
com.example.pampraktikum5_11s20018 I/fragment: onCreateView
com.example.pampraktikum5_11s20018 I/fragment: onViewCreated
com.example.pampraktikum5_11s20018 I/fragment: onStart
com.example.pampraktikum5_11s20018 I/fragment: onResume
```

PAM Praktikum 5 - 11S20018

My Fragement

```
com.example.pampraktikum5_11s20018 I/fragment: onPause
com.example.pampraktikum5_11s20018 I/fragment: onStop
com.example.pampraktikum5_11s20018 I/fragment: onDestroyView
com.example.pampraktikum5_11s20018 I/fragment: onDestroy
com.example.pampraktikum5_11s20018 I/fragment: onDetach
```

Service

Cari tahu kegunaan dari kode

```
<uses-permission android:name="android.permission.WAKE_LOCK"/>
```

```
<service android:name=".ClassicServiceExample"/>
<service android:name=".JobIntentServiceExample"
    android:permission="android.permission.BIND_JOB_SERVICE"/>
```


Service

`<uses-permission android:name="android.permission.WAKE_LOCK" />` adalah izin yang diberikan kepada aplikasi untuk mengunci perangkat agar tidak memasuki mode sleep saat aplikasi sedang berjalan. Hal ini berguna untuk memastikan aplikasi tetap berjalan meskipun perangkat dalam keadaan tidak aktif.

`<service android:name=".ClassicServiceExample"/>` adalah konfigurasi untuk mengaktifkan layanan klasik pada aplikasi. Layanan klasik merupakan komponen yang memungkinkan aplikasi untuk menjalankan proses yang berjalan di latar belakang tanpa memerlukan interaksi langsung dengan pengguna.

`<serviceandroid:name=".JobIntentServiceExample" android:permission="android.permission.BIND_JOB_SERVICE" />` adalah konfigurasi untuk mengaktifkan layanan JobIntentService pada aplikasi. Layanan JobIntentService memungkinkan aplikasi untuk menjalankan proses yang membutuhkan waktu yang lama atau berat di latar belakang dengan menggunakan job scheduler Android.

PAM Praktikum 5 - 11S20018

START CLASSIC SERVICE

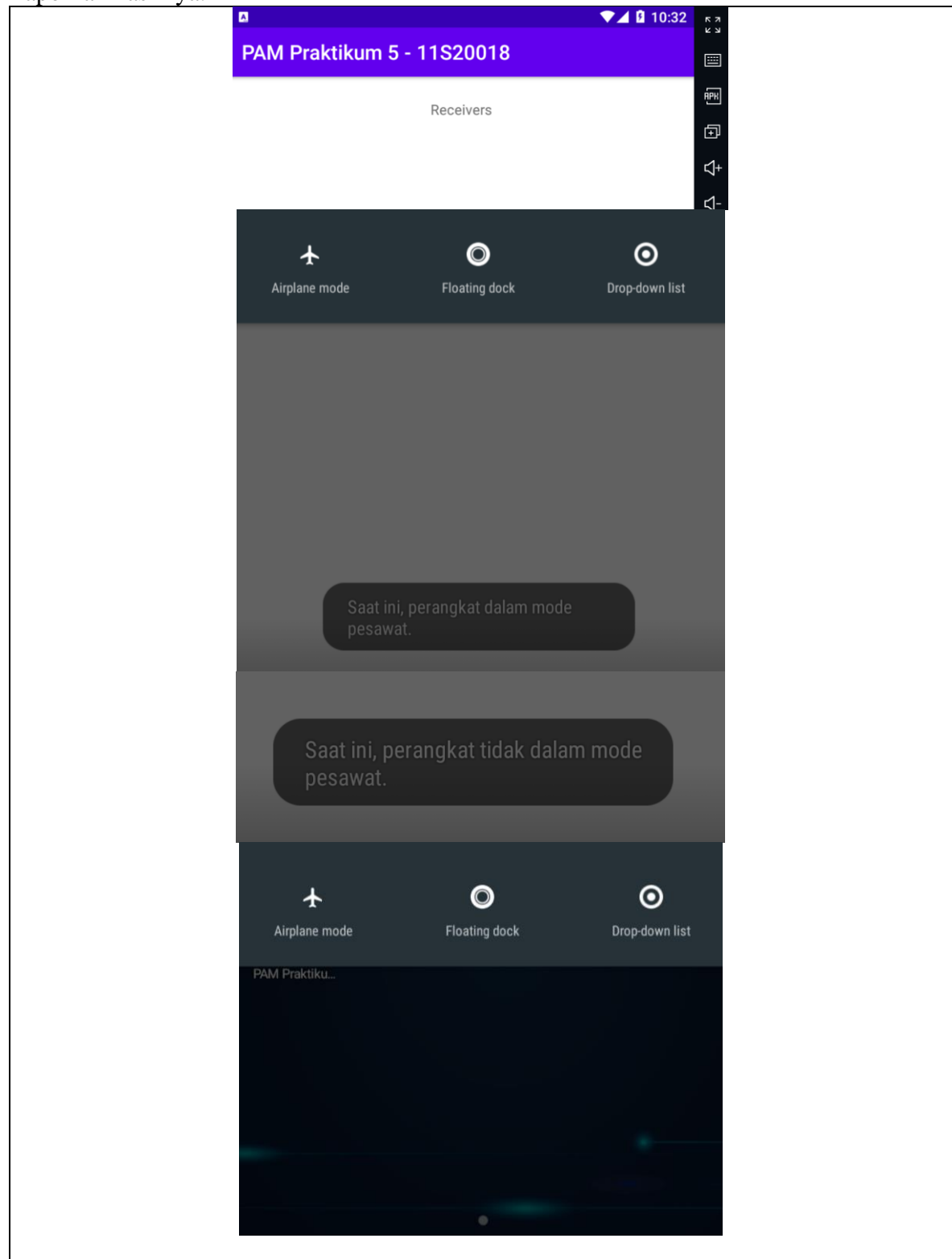
START JOB INTENT SERVICE

STOP SERVICE

```
2023-02-26 05:28:53.881 10006-10006/com.example.pampraktikum5_11s20018 D/Service: Classic Service is created.
2023-02-26 05:28:53.881 10006-10006/com.example.pampraktikum5_11s20018 D/Service: Classic Service is started.
2023-02-26 05:28:53.881 10006-10006/com.example.pampraktikum5_11s20018 D/Service Thread: main
2023-02-26 05:29:03.958 10006-10006/com.example.pampraktikum5_11s20018 D/Service: Classic Service is stopped.
2023-02-26 05:29:07.646 10006-10316/com.example.pampraktikum5_11s20018 D/Service: Job IntentService is started.
2023-02-26 05:29:07.646 10006-10316/com.example.pampraktikum5_11s20018 D/Service Thread: AsyncTask #1
2023-02-26 05:29:07.653 10006-10006/com.example.pampraktikum5_11s20018 D/Service: Job IntentService is stopped.
```

Receiver

Laporkan hasilnya.



Android View Binding

Cari tahu kegunaan kode berikut

```
buildFeatures{  
    viewBinding = true  
}
```

Kode `buildFeatures{ viewBinding = true }` adalah sebuah konfigurasi untuk Gradle build system pada proyek Android yang memungkinkan penggunaan fitur View Binding.

View Binding adalah fitur pada Android Studio yang memungkinkan akses yang lebih mudah ke komponen UI seperti `TextView`, `EditText`, dan `Button` dalam kode Kotlin atau Java tanpa perlu melakukan `findViewById()` secara manual. Dengan View Binding, kita dapat mengurangi jumlah kode yang dibutuhkan untuk mengakses komponen UI dan juga mengurangi kesalahan penulisan.

Dalam kode Gradle di atas, kita mengaktifkan fitur View Binding dengan menetapkan nilai `true` pada properti `viewBinding`. Dengan begitu, Android Studio akan secara otomatis menghasilkan kelas View Binding untuk setiap layout file XML dalam proyek. Kita dapat menggunakan kelas tersebut untuk mengakses komponen UI dalam kode Kotlin atau Java.

Laporkan hasilnya.

PAM Praktikum 5 - 11S20018

SUBMIT

PAM Praktikum 5 - 11S20018

SUBMIT

Hay, Ruth Silalahi