



Institut Teknologi Telkom Purwokerto 2020/2021

PEMROGRAMAN PERANGKAT BERGERAK

Modul Praktikum Topik 7

- Scrolling Activity
- Google Maps Activity

7.1 Persiapan

1. Android Studio
2. Device Android/Emulator Android
3. Kabel Data
4. Praktikum kali ini masih menggunakan module praktikum 6

7.2 Scrolling Activity

1. Buat activity baru.

Activity	ScrollingActivity
Name	DetailActivity

1. Edit **DetailActivity**, tambahkan kode berikut untuk menangkap data yang dikirim dari activity sebelumnya.

```
companion object {
    const val EXTRA_MYDATA = "extra_mydata"
}

inline fun <reified T : Parcelable> Activity.getParcelableExtra(key: String) = lazy {
    intent.getParcelableExtra<T>(key)
}
```

2. Edit **CardViewMyDataAdapter**, ubah isi dari **holder.itemView.setOnClickListener** menjadi intent dengan berpindah ke halaman **DetailActivity** dengan membawa data detail dari item yang dipilih. Hasil perubahannya adalah seperti berikut.

```
3. override fun onBindViewHolder(holder: CardViewViewHolder, position: Int) {
    ...
    ...
    holder.btnShare.setOnClickListener {
        Toast.makeText(holder.itemView.context, "Share " + listMyDatas[holder.adapterPosition].name,
        Toast.LENGTH_SHORT).show()
    }
    holder.itemView.setOnClickListener {
        val moveWithObjectIntent = Intent(context, DetailActivity::class.java)
        moveWithObjectIntent.putExtra(DetailActivity.EXTRA_MYDATA, myData)
        context.startActivity(moveWithObjectIntent)
    }
}
```

4. Pada bagian context akan berwarna merah, hal tersebut dikarenakan belum ada variabel context yang di deklarasikan. Variabel context berisi sebuah informasi dari activity yang sedang aktif, context disini diperlukan karena pada kondisi saat ini intent di panggil melalui sebuah adapter, sehingga adapter perlu diberikan informasi tentang activity yang aktif. Context akan di ambil dari MainActivity. Edit **MainActivity**, udah kode pada bagian fungsi showRecyclerView.

```
private fun showRecyclerCardView() {
    rv_mydata.layoutManager = LinearLayoutManager(this)
    val cardViewMyDataAdapter = CardViewMyDataAdapter (list,this@MainActivity)
    rv_mydata.adapter = cardViewMyDataAdapter
}
```

5. Edit **CardViewMyDataAdapter**, tambahkan kode seperti dibawah ini.

```
class CardViewMyDataAdapter(private val listMyDatas: ArrayList<MyData>, val context: Context) :
    RecyclerView.Adapter<CardViewMyDataAdapter.CardViewHolder>() {
    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): CardViewHolder {
        val view: View = LayoutInflater.from(parent.context).inflate(R.layout.item_cardview_practice3,
            parent, false)
        return CardViewHolder(view)
    }
}
```

6. Edit **activity_detail**, tambahkan ImageView untuk menampilkan gambar item yang dipilih.

```
<androidx.coordinatorlayout.widget.CoordinatorLayout
...
tools:context=". DetailActivity">
<com.google.android.material.appbar.AppBarLayout
...
    android:theme="@style/AppTheme.AppBarOverlay">
<com.google.android.material.appbar.CollapsingToolbarLayout
...
    app:toolbarId="@+id/toolbar">
    <ImageView
        android:id="@+id/iv_detail_photo"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:fitsSystemWindows="true"
        android:scaleType="centerCrop"
        android:src="@drawable/background"
        app:layout_collapseMode="parallax" />
    <androidx.appcompat.widget.Toolbar
        ...
        app:popupTheme="@style/AppTheme.PopupOverlay" />
    </com.google.android.material.appbar.CollapsingToolbarLayout>
</com.google.android.material.appbar.AppBarLayout>
<include layout="@layout/content_scrolling" />
<com.google.android.material.floatingactionbutton.FloatingActionButton
```

7. Edit **content_scrolling**, tambahkan atribut id pada TextView agar component dapat dikenali melalui id.

```
<TextView
    android:id="@+id/tv_detail_description"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_margin="@dimen/text_margin"
    android:text="@string/description" />
```

8. Edit **DetailActivity**, tambahkan kode berikut didalam fungsi onCreate untuk menampilkan data pada title, tv_detail_description dan iv_detail_photo.

```
val myData by getParcelableExtra<MyData>(DetailActivity.EXTRA_MYDATA)
supportActionBar?.title = myData.name.toString()
supportActionBar?.setDisplayHomeAsUpEnabled(true)
tv_detail_description.text = myData.description.toString()

Glide.with(this)
    .load(myData.photo.toString())
    .apply(RequestOptions().override(700, 700))
    .into(iv_detail_photo)
```

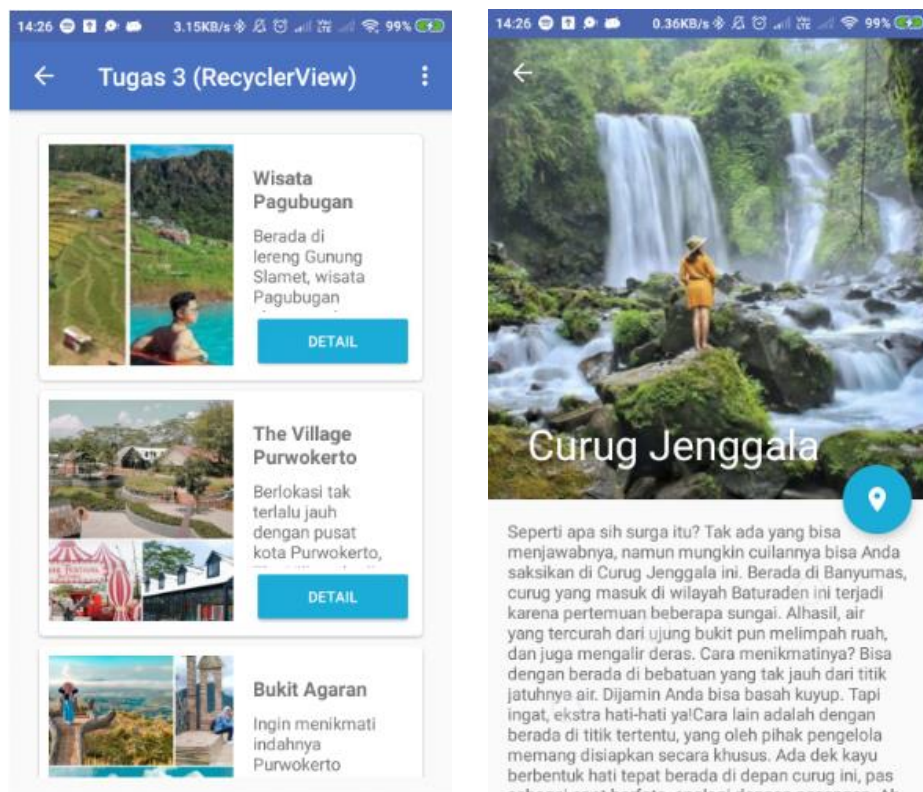
9. Edit **DetailActivity**, tambahkan fungsi baru untuk kembali ke activity sebelumnya saat tombol panah back di tekan.

```
override fun onSupportNavigateUp(): Boolean {
    onBackPressed()
    return true
}
```

10. Edit **activity_detail**, ubah nilai **layout_height** menjadi 400dp pada komponen **app_bar**.

```
<com.google.android.material.appbar.AppBarLayout
    android:id="@+id/app_bar"
    android:layout_width="match_parent"
    android:layout_height="400dp"
    android:fitsSystemWindows="true"
    android:theme="@style/AppTheme.AppBarOverlay">
```

11. Jalankan aplikasi, hasilnya seperti dibawah ini.



7.3 Google Maps Activity

1. Edit **MyData**, tambahkan 2 variabel yaitu latitude dan longitude untuk menampung data berupa koordinat lokasi. Dan buat implementasi parcelable untuk mengirim data ke activity lain dalam bentuk object. Sehingga isi dari **MyData** adalah seperti dibawah ini.

```
@Parcelize
data class MyData(
    val name: String,
    val description: String,
    val photo: String,
    val lat: Double,
    val lang: Double
) : Parcelable
```

2. Edit **String.xml**, tambahkan data latitude dan longitude.

```
<string-array name="data_lat">
    <item>-7.333797</item>
    <item>-7.374473</item>
</string-array>
<string-array name="data_lang">
    <item>109.206959</item>
    <item>109.240532</item>
</string-array>
```

3. Edit **MainActivity**, ubah isi dari fungsi **getListMyDatas** dan sesuaikan dengan penambahan latitude dan longitude.

```
fun getListMyDatas(): ArrayList<MyData> {
    val dataName = resources.getStringArray(R.array.data_name)
    val dataDescription = resources.getStringArray(R.array.data_description)
    val dataPhoto = resources.getStringArray(R.array.data_photo)
    val dataLat = resources.getStringArray(R.array.data_lat)
    val dataLang = resources.getStringArray(R.array.data_lang)
    val listMyData = ArrayList<MyData>()
    for (position in dataName.indices) {
        val myData = MyData(
            dataName[position],
            dataDescription[position],
            dataPhoto[position],
            dataLat[position].toDouble(),
            dataLang[position].toDouble()
        )
        listMyData.add(myData)
    }
    return listMyData
}
```

4. Buat activity.

Activity	Google Maps Activity
Name	MapsActivity

5. Edit **MapsActivity**, tambahkan kode berikut untuk mengambil member variabel tanpa melalui objek.

```
companion object {
    const val EXTRA_MYDATA = "extra_mydata"
}

val myData by getParcelableExtra<MyData>(DetailActivity.EXTRA_MYDATA)
inline fun <reified T : Parcelable> Activity.getParcelableExtra(key: String) = lazy {
    intent.getParcelableExtra<T>(key)
}
```

6. Edit **MapsActivity**, tambahkan kode berikut pada fungsi **onCreate** untuk menangkap data yang dikirim dari activity sebelumnya.

```
supportActionBar?.title = myData?.name.toString()
supportActionBar?.setDisplayHomeAsUpEnabled(true)
```

7. Edit **MapsActivity**, tambahkan fungsi baru untuk kembali ke activity sebelumnya saat tombol panah back di tekan.

```
override fun onSupportNavigateUp(): Boolean {
    onBackPressed()
    return true
}
```

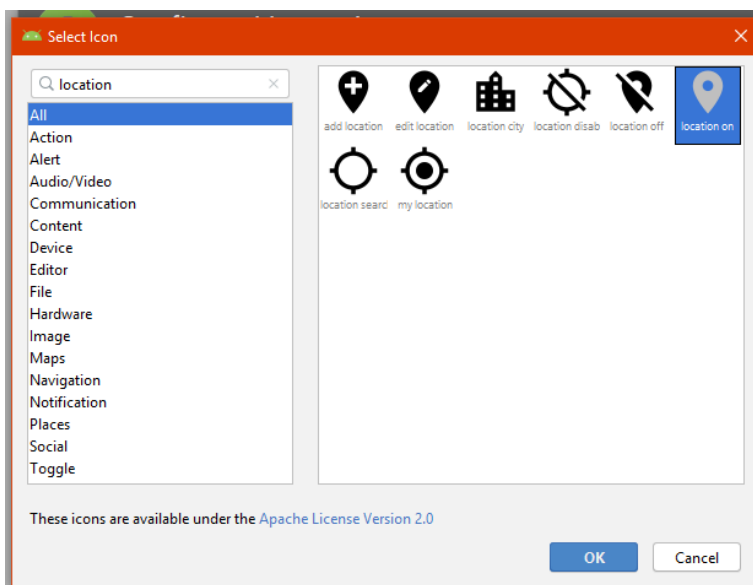
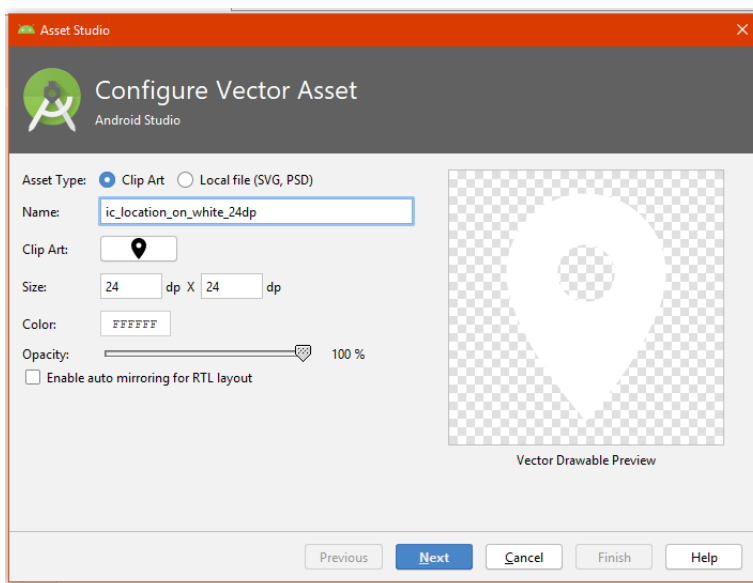
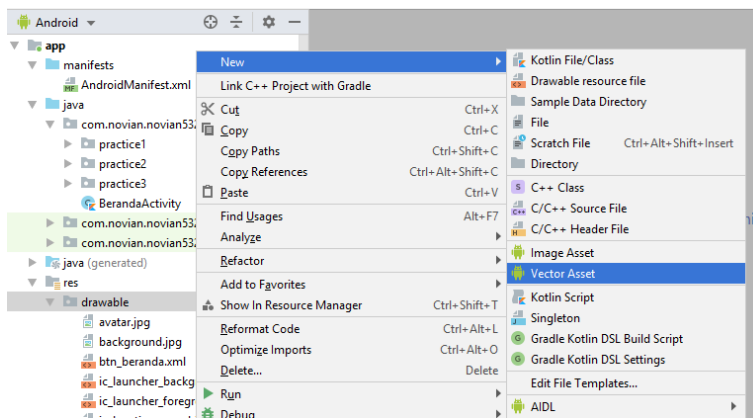
8. Edit **MapsActivity**, ubah fungsi **onMapReady** seperti dibawah ini.

```
override fun onMapReady(googleMap: GoogleMap) {
    mMap = googleMap
    var lat: Double = myData!!.lat
    var lng: Double = myData!!.lng
    val location = LatLng(lat, lng)
    mMap.addMarker(MarkerOptions().position(location).title(myData!!.name.toString()))
    mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(location, 16.0f))
}
```

9. Edit **google_maps_api.xml**, tambahkan **key google maps** pada komponen **google_maps_api**.

```
<string name="google_maps_key" templateMergeStrategy="preserve"
    translatable="false">AlzaSyC7Xyv2yogvVBSrblPZArZZ6muk1kyA7L0</string>
```

10. Tambahkan icon lokasi.



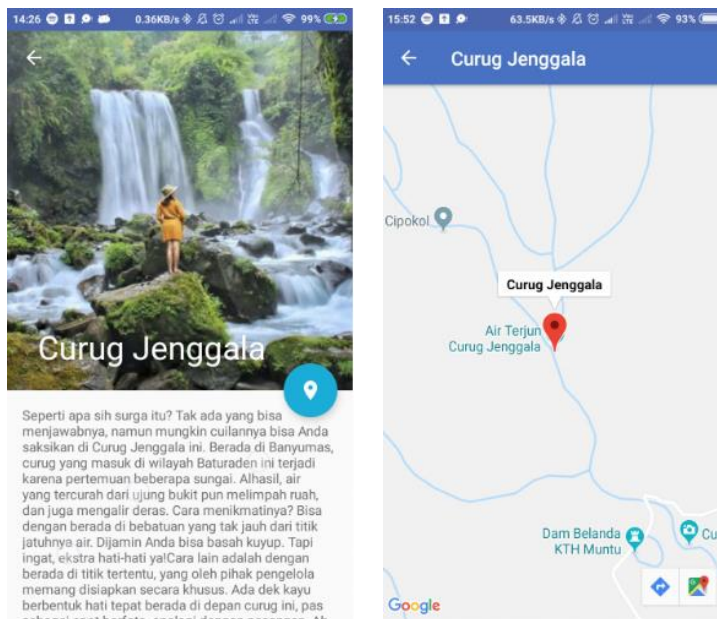
11. Edit **activity_detail.xml**, ubah isi atribut **app:srcCompat** dengan icon yang telah di buat sebelumnya.

```
<androidx.coordinatorlayout.widget.CoordinatorLayout
...
tools:context=". DetailActivity">
<com.google.android.material.appbar.AppBarLayout
...
android:theme="@style/AppTheme.AppBarOverlay">
<com.google.android.material.appbar.CollapsingToolbarLayout
...
app:toolbarId="@+id/toolbar">
<ImageView
...
app:layout_collapseMode="parallax" />
<androidx.appcompat.widget.Toolbar
...
app:popupTheme="@style/AppTheme.PopupOverlay" />
</com.google.android.material.appbar.CollapsingToolbarLayout>
</com.google.android.material.appbar.AppBarLayout>
<include layout="@layout/content_scrolling" />
<com.google.android.material.floatingactionbutton.FloatingActionButton
android:id="@+id/fab"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_margin="@dimen/fab_margin"
app:layout_anchor="@id/app_bar"
app:layout_anchorGravity="bottom|end"
app:srcCompat="@drawable/ic_location_on_white_24dp" />
</androidx.coordinatorlayout.widget.CoordinatorLayout>
```

12. Edit **DetailActivity**, tambahkan kode pada **fab.setOnClickListener**.

```
val moveWithObjectIntent = Intent(this, MapsActivity::class.java)
moveWithObjectIntent.putExtra(MapsActivity.EXTRA_MYDATA, myData)
startActivity(moveWithObjectIntent)
```


13. Jalankan aplikasi, hasilnya seperti dibawah ini.



7.4 Tugas

1. Selesaikan modul praktikum
2. Step tambahan : Buatlah agar list dan grid saat di tekan juga dapat membuka halaman detail item seperti pada cardview.
3. Pastikan tidak ada terjadi error pada aplikasi, demo aplikasi dapat dilihat pada -> <https://youtu.be/EKA15McjaTA>
4. Build aplikasi menjadi .apk.
5. Upload file .apk ke lms, deadline dapat dilihat pada lms.
6. Kriteria Penilaian :
 - a. Melanjutkan project pada pertemuan kemarin dengan melakukan push ke github branch praktikum7 (30)
 - b. Semua fitur pada module praktikum dapat dijalankan dengan baik (50)
 - c. Step tambahan di kerjakan dengan benar (10)
 - d. Improvisasi (10)
 - e. Keterlambatan pengurangan nilai 20