

# MODUL PRAKTIKUM PEMROGRAMAN UNTUK PERANGKAT BERGERAK 1

Indra Azimi, S.T., M.T.

Reza Budiawan, S.T., M.T., OCA

Cahyana S.T., M.Kom.



D3 Rekayasa Perangkat Lunak Aplikasi  
Telkom University

# Daftar Isi

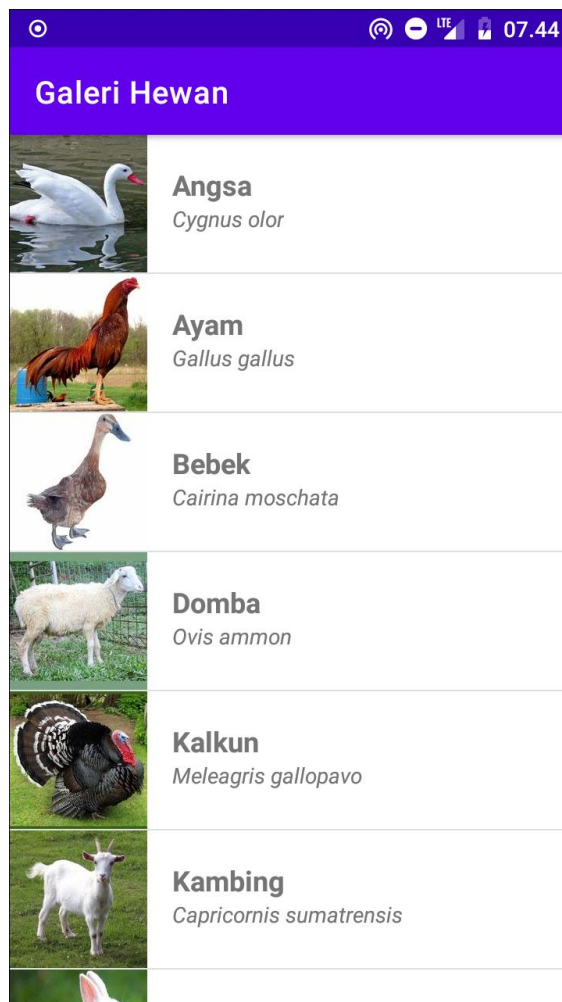
Modul 04: RecyclerView .....	2
1. Overview .....	2
2. Getting Started.....	3
3. Task .....	8
3.1. Menambahkan data dummy.....	8
3.2. Membuat layout untuk list .....	10
3.3. Menambahkan view binding.....	13
3.4. Menampilkan data dengan adapter.....	14
3.5. Menambahkan gambar pada list .....	17
4. Summary .....	21
5. Challenge.....	21

# Modul 04: RecyclerView

## 1. Overview

Pada modul kali ini, kita akan membuat sebuah aplikasi sederhana bernama “Galeri Hewan”. Aplikasi ini menampilkan gambar hewan dan namanya seperti yang pernah dilakukan pada modul 02 sebelumnya. Akan tetapi, kali ini gambar dan nama hewan ditampilkan dalam bentuk list/daftar. Untuk menghasilkan tampilan berupa list, kita akan menggunakan RecyclerView.

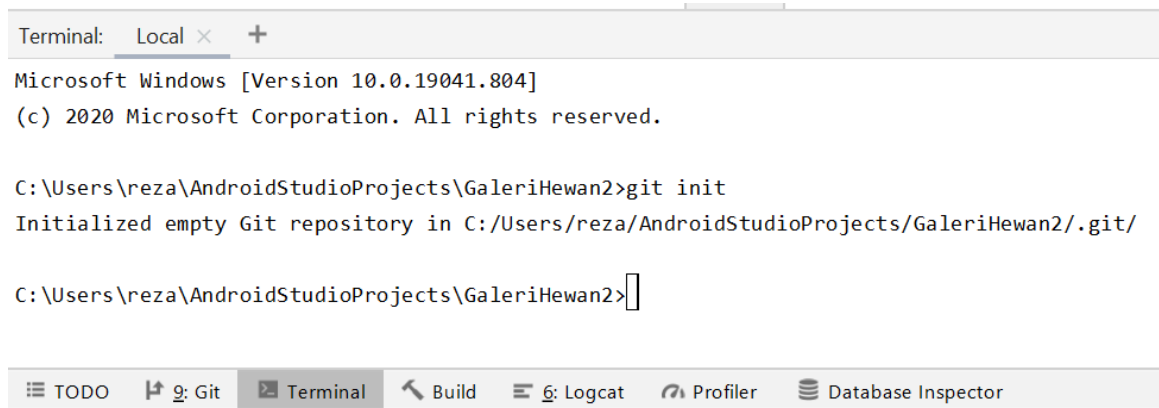
Selain itu, pada modul ini, kita juga akan berlatih menggunakan version control Git dan layanan Github dalam men-develop aplikasi tersebut. Aplikasi yang kita buat dapat dilihat versi lengkapnya di repository Github ini: <https://github.com/indraazimi/galeri-hewan> (lihat di branch “recyclerview”). Berikut adalah tampilan akhir dari aplikasi yang akan dibuat.



## 2. Getting Started

Kali ini kita akan menggunakan version control Git. Jadi unduh terlebih dahulu Git dari [sini](#), lalu install di komputer/laptop masing-masing. Kemudian sama seperti biasa, jalankan Android Studio, lalu buat project baru bernama “Galeri Hewan”. Tuliskan “org.d3ifxxx.galerihewan” sebagai nama package. Pilih minimum SDK 19, dengan Empty Activity sebagai template awal. Silahkan baca kembali modul pertama jika diperlukan.

Berikutnya, jadikan project Android Studio yang sudah dibuat sebelumnya agar menjadi project yang menggunakan versioning Git. Caranya, buka Terminal yang terletak di bagian bawah Android Studio. Di terminal tersebut, ketikkan perintah “git init” lalu Enter.



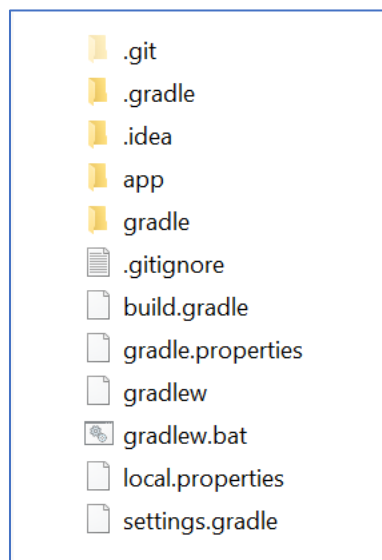
```
Terminal: Local x +
Microsoft Windows [Version 10.0.19041.804]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\reza\AndroidStudioProjects\GaleriHewan2>git init
Initialized empty Git repository in C:/Users/reza/AndroidStudioProjects/GaleriHewan2/.git/

C:\Users\reza\AndroidStudioProjects\GaleriHewan2>
```

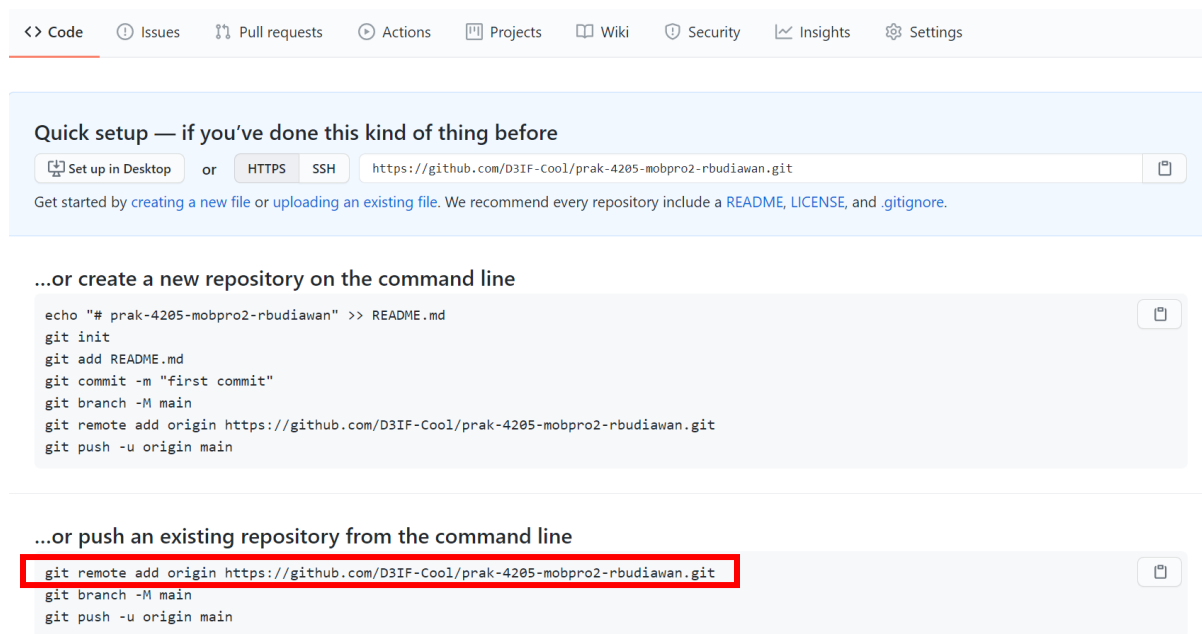
The screenshot shows the Android Studio interface with the Terminal tab selected. The terminal output confirms that a new Git repository has been initialized in the project's root directory.

Jika sudah terdapat keterangan “Initialized empty Git repository in ...” tandanya, project sudah mendukung version control. Selain itu, di folder project akan terdapat hidden folder “.git”.



Project Android Studio yang kita buat kali ini, telah dimungkinkan untuk dilakukan operasi berupa commit, push, dan lainnya. Operasi commit memiliki arti yang mirip dengan checkpoint pada sebuah game. Kita dapat kembali ke titik commit tersebut jika ada kode yang tidak sesuai harapan saat proses membangun aplikasi. Sedangkan operasi push merupakan operasi untuk mengirimkan kode kita agar tersimpan di repository online seperti Github.

Langkah selanjutnya adalah menyiapkan repository di Github untuk menyimpan project kita ini. Buat akun di [Github](#) terlebih dahulu jika masih belum punya. Lalu buka link yang disediakan oleh dosen untuk membuat repository praktikum. Setelah repository jadi, carilah perintah yang diberikan tanda merah pada gambar di bawah ini.



Quick setup — if you've done this kind of thing before

Set up in Desktop or HTTPS SSH `https://github.com/D3IF-Cool/prak-4205-mobpro2-rbudiawan.git`

Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

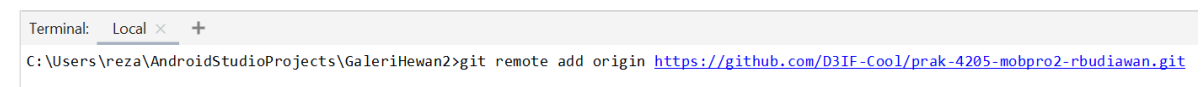
...or create a new repository on the command line

```
echo "# prak-4205-mobpro2-rbudiawan" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/D3IF-Cool/prak-4205-mobpro2-rbudiawan.git
git push -u origin main
```

...or push an existing repository from the command line

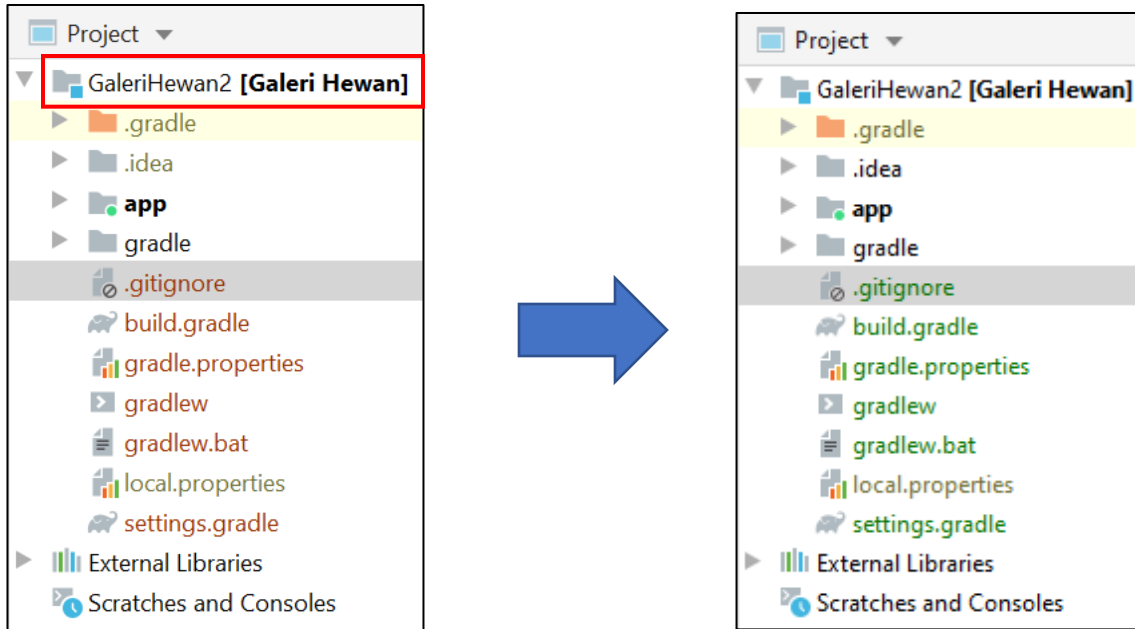
```
git remote add origin https://github.com/D3IF-Cool/prak-4205-mobpro2-rbudiawan.git
git branch -M main
git push -u origin main
```

Pada Terminal Android Studio, ketikkan baris tersebut lalu tekan Enter. Jika tidak keluar pesan apa-apa, berarti sudah benar. Pastikan perintahnya telah sesuai dengan repo masing-masing ya!



```
Terminal: Local x +
C:\Users\reza\AndroidStudioProjects\GaleriHewan2>git remote add origin https://github.com/D3IF-Cool/prak-4205-mobpro2-rbudiawan.git
```

Langkah selanjutnya adalah melakukan penambahan file ke dalam stage yang di-tracked untuk versioning. Ubah tampilan file dari Android ke Project. Setelah itu klik kanan pada nama project (yang ditandai merah), pilih menu Git > Add. Efek dari berhasilnya kode ini adalah, nama file akan berubah dari merah menjadi hijau.



Selanjutnya, pastikan info nama dan email dari Git memperlihatkan identitas masing-masing. Ketik dua perintah berikut di Terminal Android Studio, lalu perhatikan outputnya.

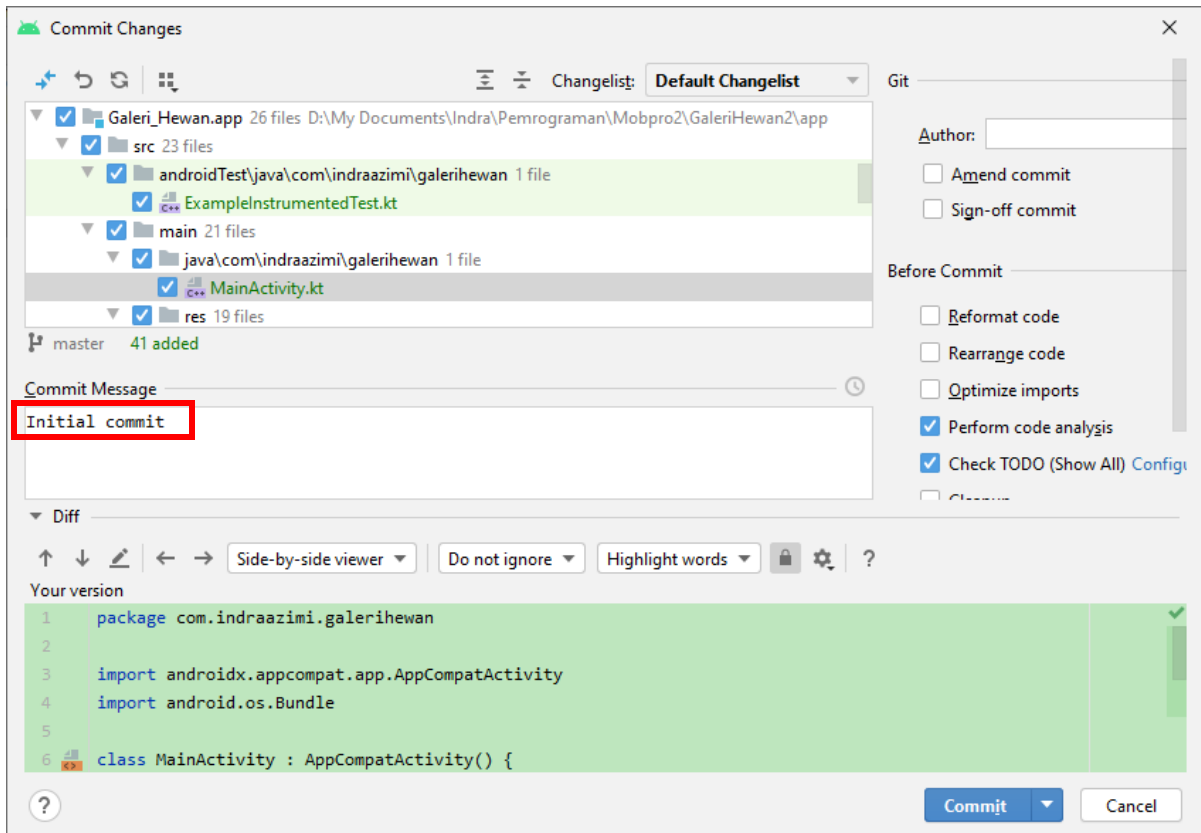
```
Terminal: Local x +  
  
C:\Users\reza\AndroidStudioProjects\GaleriHewanMod4>git config user.name  
reza  
  
C:\Users\reza\AndroidStudioProjects\GaleriHewanMod4>git config user.email  
rbudiawan@tass.telkomuniversity.ac.id
```

Jika setelah dicek ternyata email dan nama (*ingat, ini nama lengkap ya, **bukan** username Github*) masih belum benar, lakukan perubahan nama dan email dengan perintah berikut:

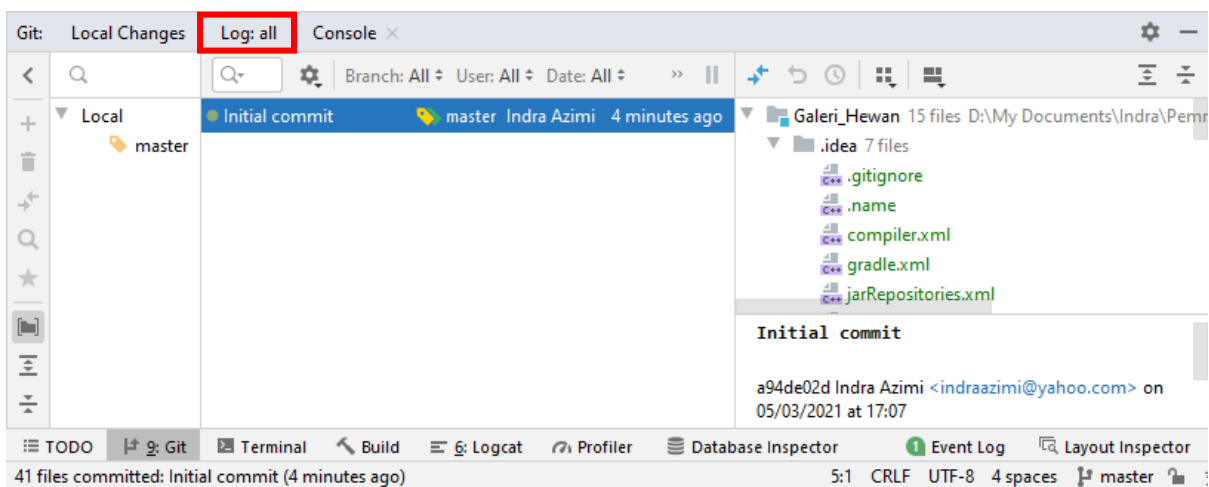
Mengubah nama → `git config --global user.name nama_lengkap`

Mengubah email → `git config --global user.mail email_yang_dipakai_di_Github`

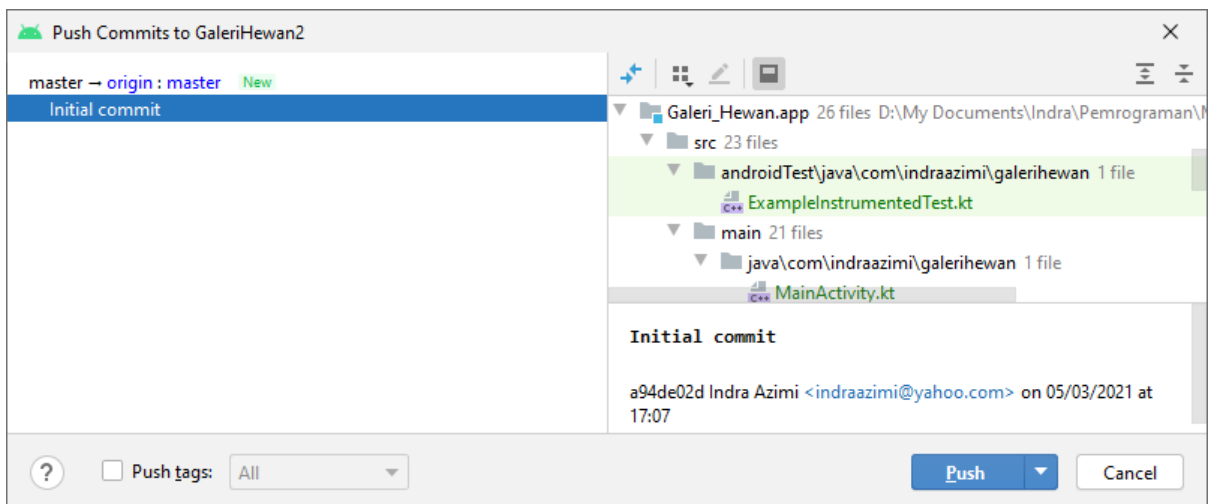
Jika informasi sudah sesuai, lakukan commit pertama pada project ini dengan memilih menu VCS > Commit... Isi commit message dengan "Initial commit" lalu klik Commit. Jika muncul peringatan, klik Commit lagi. Efek dari berhasilnya langkah ini adalah, nama file pada project explorer akan berubah warna dari hijau menjadi hitam.



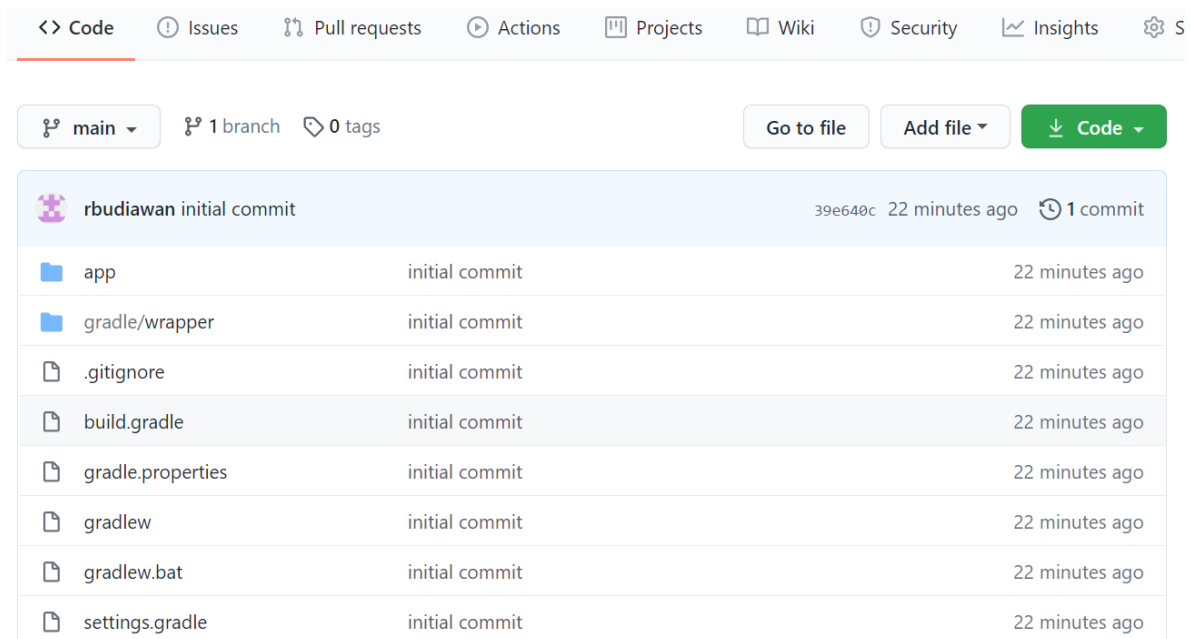
Selain itu, jika kita cek di bagian “Git” > “Log: all”, akan muncul pesan commit yang telah dibuat sebelumnya. Selain itu terlihat juga nama dan email yang melakukan commit, waktu commit, dan file-file yang ditambahkan di commit ini.



Berikutnya, kita akan melakukan operasi push ke repository Github. Caranya, klik menu VCS > Git > Push... Pada dialog yang muncul, pastikan initial commit telah tampil, lalu klik Push.



Pada dialog yang muncul selanjutnya, masukkan username Github pada bagian username, dan masukkan Personal Access Token (PAT) pada bagian password. Cara mendapatkan PAT dari Github dapat mengikuti langkah-langkah yang terdapat pada link [ini](#). Push dapat dikatakan berhasil jika setelah di-refresh, repository Github yang awalnya kosong menjadi berisi file-file sesuai project dari Android Studio kita masing-masing.





Berikutnya, kita akan menambahkan beberapa komponen pada project Android Studio ini. Silahkan ikuti task di modul ini langkah per langkah. Lakukan commit setiap 1 sub-task berhasil dilakukan.

**Dilarang keras untuk copy – paste kode dari modul/sumber lain!**

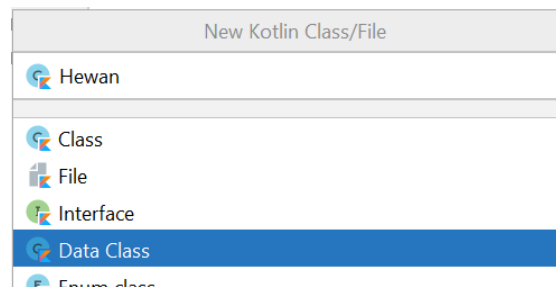
**Ngoding pelan-pelan akan membuat kamu lebih jago di masa depan.  
Lakukan commit setiap selesai 1 sub-task. Selamat ngoding!**

### 3. Task

#### 3.1. Menambahkan data dummy

Pada aplikasi ini, kita akan menampilkan daftar hewan ke dalam tampilan list. Untuk itu, diperlukan datanya terlebih dahulu. Data ini biasanya didapat dari database, hasil request ke server melalui web service, dan lainnya. Akan tetapi karena kita belum belajar mengambil data dari database ataupun dari internet, kali ini kita akan menggunakan data dummy yang kita bentuk dalam list (seperti ArrayList di bahasa pemrograman Java).

Untuk membuat data seperti yang dideskripsikan sebelumnya, langkah pertama yang kita lakukan adalah membentuk class Hewan. Klik kanan pada package yang ada dalam folder Java, pilih “New” → “Kotlin Class/File”. Masukkan nama “Hewan”, dan pilih “Data Class” sebagai file yang dibentuk. Android Studio akan bertanya, tambahkan file ini ke Git? Pilih Add. Pilih opsi ini untuk semua file yang kita tambahkan di task-task selanjutnya.



Ubah isi dari class Hewan.kt sebagai berikut:

```
data class Hewan(  
    val nama: String,  
    val namaLatin: String  
)
```

Berikutnya, gunakan class tersebut untuk membuat list of Hewan pada MainActivity.kt. Untuk memastikan bahwa dummy data sudah terbentuk, kita akan menampilkan jumlah hewan di layar Logcat saat aplikasi dijalankan.

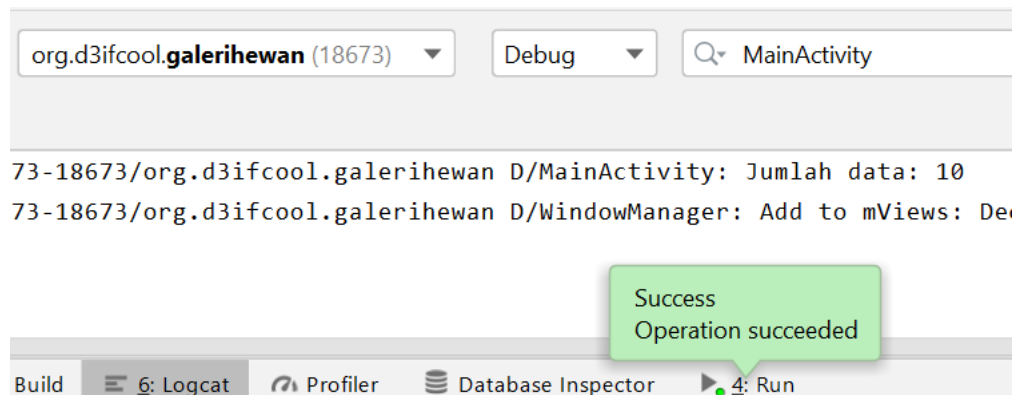
```
class MainActivity : AppCompatActivity() {

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        Log.d("MainActivity", "Jumlah data: " + getData().size)
    }

    private fun getData(): List<Hewan> {
        return listOf(
            Hewan("Angsa", "Cygnus olor"),
            Hewan("Ayam", "Gallus gallus"),
            Hewan("Bebek", "Cairina moschata"),
            Hewan("Domba", "Ovis ammon"),
            Hewan("Kalkun", "Meleagris gallopavo"),
            Hewan("Kambing", "Capricornis sumatrensis"),
            Hewan("Kelinci", "Oryctolagus cuniculus"),
            Hewan("Kerbau", "Bubalus bubalis"),
            Hewan("Kuda", "Equus caballus"),
            Hewan("Sapi", "Bos taurus"),
        )
    }
}
```

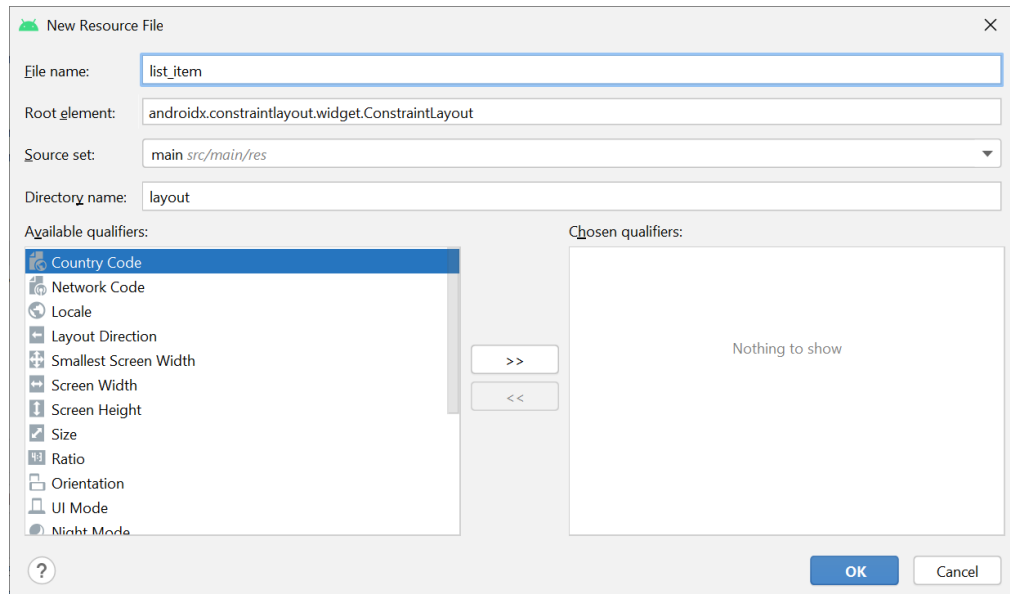
Import bagian Log agar kode tidak error. Jalankan aplikasi, perhatikan pada bagian Logcat, akan ada informasi jumlah hewan yang terdapat pada list.



Jika telah berhasil mengeluarkan hasil seperti contoh di atas, lakukan commit kembali dengan commit message: "Menambahkan data dummy".

### 3.2. Membuat layout untuk list

Pada task ini kita akan membuat tampilan/layout berupa list untuk menampilkan daftar hewan kepada pengguna. Layout yang akan dibuat terdiri dari 2: layout untuk menampilkan list/daftar, dan layout untuk menampilkan 1 item pada list/daftar tersebut. Pertama, kita akan membuat layout untuk menampilkan 1 item terlebih dahulu. Buat sebuah layout baru dengan cara klik kanan pada folder `/res/layout` pilih “New” → “Layout Resource File”. Berikan nama “list\_item”.



Berikutnya, atur tampilan dengan menggunakan informasi berikut: Terdapat 2 TextView pada sebuah ConstraintLayout dengan masing-masing konfigurasi terlihat pada tabel di bawah ini.

<b>ConstraintLayout</b> <ol style="list-style-type: none"><li>1. Ukuran lebar: mengikuti ukuran layar smartphone (<code>match_parent</code>)</li><li>2. Ukuran tinggi: mengikuti ukuran konten di dalamnya (<code>wrap_content</code>)</li></ol>	
<b>TextView menampilkan “Angsa”</b> <ol style="list-style-type: none"><li>1. id: <code>namaTextView</code></li><li>2. Lebar &amp; tinggi: sesuai ukuran konten (<code>wrap_content</code>)</li><li>3. Jarak tepi kiri: 16dp</li><li>4. Jarak tepi atas: 16dp</li><li>5. Constraint: sisi kiri &amp; atas terhadap parent (tepi layar)</li><li>6. Ukuran text (<code>textSize</code>): 18sp</li><li>7. Style tulisan (<code>textStyle</code>): bold</li></ol>	<b>TextView menampilkan “Cygnus olor”</b> <ol style="list-style-type: none"><li>1. id: <code>latinTextView</code></li><li>2. Lebar &amp; tinggi: sesuai ukuran konten (<code>wrap_content</code>)</li><li>3. Jarak tepi bawah: 16dp</li><li>4. Constraint: sisi bawah terhadap parent (tepi layar); sisi kiri &amp; atas terhadap “namaTextView”</li><li>5. Style tulisan (<code>textStyle</code>): italic</li></ol>

Kode tampilan sesuai konfigurasi di atas adalah sebagai berikut:

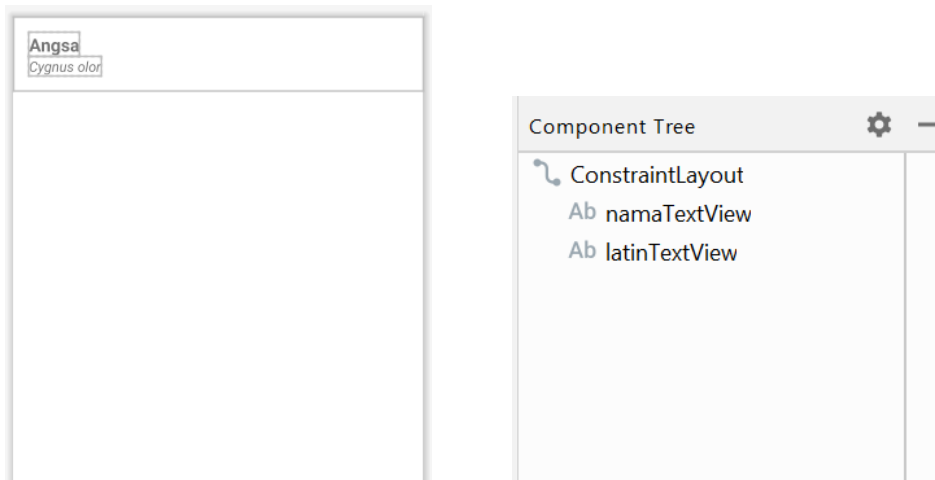
```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">

    <TextView
        android:id="@+id/namaTextView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="16dp"
        android:layout_marginTop="16dp"
        android:textSize="18sp"
        android:textStyle="bold"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        tools:text="Angsa" />

    <TextView
        android:id="@+id/latinTextView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="16dp"
        android:textStyle="italic"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintStart_toStartOf="@+id/namaTextView"
        app:layout_constraintTop_toBottomOf="@+id/namaTextView"
        tools:text="Cygnus olor" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

Hasil tampilan akhir di bagian design, diperlihatkan sebagai berikut:

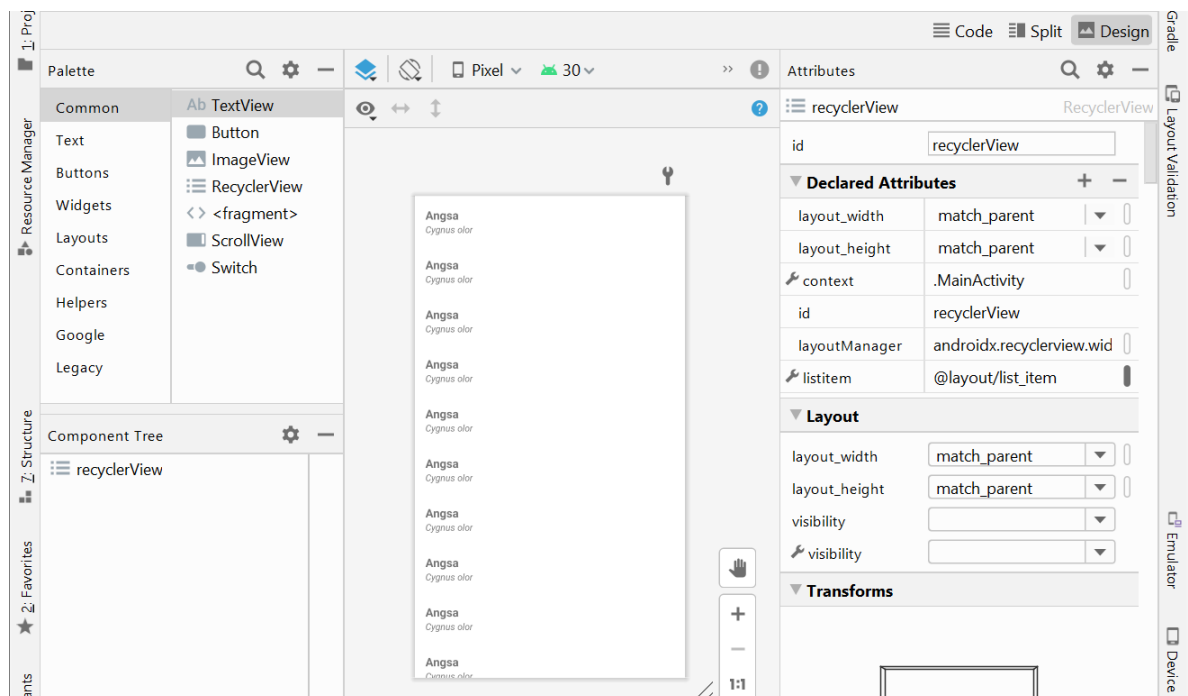


Setelah mengatur tampilan 1 item, langkah berikutnya mengatur tampilan list pada activity\_main.xml. Untuk mengatur tampilan, diperlukan komponen berupa RecyclerView. Hapus komponen default dari activity\_main.xml (ConstraintLayout & TextView), dan tambahkan komponen RecyclerView sehingga kode dari file tersebut sebagai berikut:

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.recyclerview.widget.RecyclerView
    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:id="@+id/recyclerView"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    app:layoutManager="androidx.recyclerview.widget.LinearLayoutManager"
    tools:context=".MainActivity"
    tools:listitem="@layout/list_item" />
```

Pada kode di atas, terlihat bahwa kita menggunakan sebuah komponen RecyclerView dengan id "recyclerView" yang memiliki tinggi serta lebar sesuai ukuran smartphone/emulator (match\_parent). Pada komponen ini, kita menggunakan layout "LinearLayoutManager" yang menampilkan data secara linear. Selain LinearLayoutManager, ada bentuk layout lain yang dapat diberikan pada RecyclerView seperti GridLayoutManager atau StaggeredGridLayoutManager.

Berikut tampilan pada mode design untuk activity\_main.xml.



Jika sudah selesai, jalankan aplikasi. Pastikan kode tidak error, dan aplikasi berjalan dengan baik di smartphone/emulator. Aplikasi Android yang dijalankan sekarang masih menampilkan halaman yang kosong, jadi tidak perlu merasa aneh. Setelah memastikan tidak ada masalah, lakukan commit dengan message “Membuat layout untuk list”.

### 3.3. Menambahkan view binding

Pada task ini, kita akan menambahkan view binding ke project Android Studio. Caranya kurang lebih sama dengan modul sebelumnya. Pertama, tambahkan kode berikut pada app/build.gradle:

```
defaultConfig {  
    ...  
}  
  
buildFeatures {  
    viewBinding = true  
}  
  
buildTypes {  
    ...  
}
```

Selanjutnya gunakan binding pada MainActivity.kt dengan mengubah kode menjadi sebagai berikut:

```
class MainActivity : AppCompatActivity() {  
  
    private lateinit var binding: ActivityMainBinding  
  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContentView(R.layout.activity_main)  
        binding = ActivityMainBinding.inflate(layoutInflater)  
        setContentView(binding.root)  
  
        Log.d("MainActivity", "Jumlah data: " + getData().size)  
    }  
    ...  
}
```

Untuk memastikan tidak ada kesalahan, lakukan running aplikasi. Pastikan aplikasi berjalan seperti biasa, dan tetap menampilkan tampilan yang sama seperti task sebelumnya. Jika sudah berjalan dengan baik, lakukan commit dengan message sesuai dengan judul dari task 3.3 ini.

### 3.4. Menampilkan data dengan adapter

Pada task ini, kita mulai menampilkan data di dalam list sehingga saat aplikasi dijalankan akan muncul list yang menampilkan nama hewan, baik dari penamaan bahasa Indonesia dan bahasa latin. Langkah pertama adalah menambahkan adapter. Adapter ini berguna untuk menghubungkan antara data dengan tampilan XML. Pada adapter, kita akan mengatur apa yang ditampilkan di TextView dengan id namaTextView dan juga latinTextView.

Cara membuat adapter, klik kanan pada package di bawah folder “java” pilih “New” → “Kotlin Class/File”. Berikan nama “MainAdapter” dengan tipe class.

```
class MainAdapter {  
}
```

Selanjutnya atur agar class ini menerima list of Hewan sebagai parameter di bagian konstruktornya. Variable “data” ini bersifat global untuk class MainAdapter. Selain itu, terdapat keyword private yang membuat variable “data” tersebut tidak bisa diakses oleh class lain selain class MainAdapter. List ini merupakan daftar yang akan ditampilkan kepada pengguna.

```
class MainAdapter(private val data: List<Hewan>) {  
}
```

Berikutnya, tambahkan sebuah inner class “ViewHolder” pada class MainAdapter. Class ini, sesuai namanya berfungsi sebagai view holder (penampung dari view), sehingga proses menampilkan data bisa berjalan lebih efisien. Berikan sifat “RecyclerView.ViewHolder” pada class ini dan buat juga sebuah fungsi bernama bind().

```
class MainAdapter(private val data: List<Hewan>) {  
  
    class ViewHolder(private val binding: ListItemBinding) :  
        RecyclerView.ViewHolder(binding.root) {  
  
        fun bind(hewan: Hewan) = with(binding) {  
            namaTextView.text = hewan.nama  
            latinTextView.text = hewan.namaLatin  
        }  
    }  
}
```

Pada class di atas, terlihat bahwa class ViewHolder memiliki 1 parameter berupa “binding” pada konstruktornya. Tipe dari binding ini adalah ListItemBinding, yang merupakan bentuk binding dari list\_item.xml. Dengan kata lain, pada class ini dapat diakses variable yang merupakan komponen dari layout list\_item.xml. Hal tersebut, dilakukan pada method bind yang terdapat pada class ini.

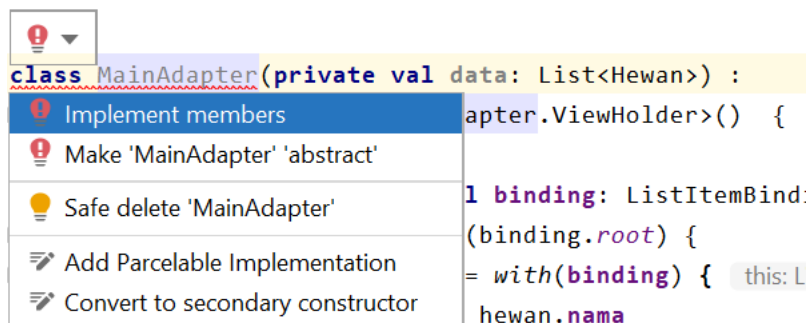
Sesuai namanya, “bind”, pada method ini kita akan melakukan “pengikatan” antara komponen XML dan data yang kita miliki (objek Hewan). Objek hewan ini merupakan input parameter pada method bind. Di dalam method bind ini, kita melakukan pengaturan bahwa namaTextView akan menampilkan nama dari objek Hewan, dan latinTextView menampilkan namaLatin dari objek Hewan.

Berikutnya, kita kembali lagi pada class MainAdapter. Berikan sifat “RecyclerView.Adapter<Holder>” pada class ini. Holder yang diberikan sesuai dengan class ViewHolder yang kita telah buat sebelumnya. Sehingga, kode dari MainAdapter menjadi sebagai berikut:

```
class MainAdapter(private val data: List<Hewan>) :  
    RecyclerView.Adapter<MainAdapter.ViewHolder>() {  
  
    class ViewHolder(private val binding: ListItemBinding) :  
        ...  
    }  
}
```

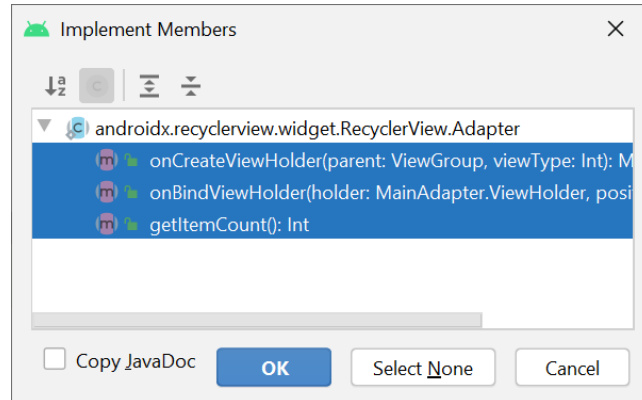
Class ini akan menjadi error karena class “RecyclerView.Adapter” merupakan abstract class yang memiliki abstract method. Sifat dari abstract method adalah wajib di-override pada subclass-nya. Sehingga, pada class MainAdapter ini terdapat tiga buah method yang harus di-override, yaitu: “onCreateViewHolder”, “onBindViewHolder”, dan “getItemCount”.

Untungnya, kita dapat meng-override secara otomatis sesuai suggest dari Android Studio. Caranya, arahkan kursor pada class MainAdapter yang sekarang sedang diberi garis merah (penanda kode error). Lalu, arahkan ke lampu merah dan pilih “Implement members”.





Pilih ketiga method yang harus di-override lalu klik OK.



Tugas selanjutnya adalah mengubah satu per satu isi dari ketiga method ini. Pertama, pada method `onCreateViewHolder`, kita akan melakukan pengambilan nilai binding dari "ListItemBinding" dengan menggunakan inflater. Jika ada kode yang error, tambahkan import untuk kode tersebut.

```
override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): ViewHolder {  
    val inflater = LayoutInflater.from(parent.context)  
    val binding = ListItemBinding.inflate(inflater, parent, false)  
    return ViewHolder(binding)  
}
```

Berikutnya, pada method `onBindViewHolder`, kita memanggil method `bind` dari class `ViewHolder` yang telah kita buat sebelumnya dan melewati data hewan sesuai posisinya.

```
override fun onBindViewHolder(holder: ViewHolder, position: Int) {  
    holder.bind(data[position])  
}
```

Selanjutnya, pada method `getItemCount`, tambahkan kode untuk memberikan jumlah dari data yang ditampilkan. Karena kita ingin semua data tampil, maka kita menggunakan nilai `size` dari list-nya.

```
override fun getItemCount(): Int {  
    return data.size  
}
```

Terakhir, gunakan class MainAdapter.kt pada MainActivity.kt seperti ini:

```
override fun onCreate(savedInstanceState: Bundle?) {
    ...
    Log.d("MainActivity", "Jumlah data: " + getData().size)
    with(binding.recyclerView) {
        addItemDecoration(DividerItemDecoration(context, RecyclerView.VERTICAL))
        adapter = MainAdapter(getData())
        setHasFixedSize(true)
    }
}
```

Pada kode di atas, terlihat bahwa kita menambahkan adapter pada RecyclerView dengan objek dari class MainAdapter yang telah kita buat sebelumnya. Pada pembuatan objek ini, kita juga memberikan data hewan dengan pemanggilan method getData(). Kode “addItemDecoration” digunakan untuk menambahkan garis pembatas horizontal antar data hewan yang ditampilkan.

Setelah menuliskan kode dengan baik dan memastikan tidak ada kode yang error, jalankan aplikasi di smartphone/emulator masing-masing. Pada tahap ini, akan ditampilkan daftar hewan berupa nama dalam bahasa Indonesia dan juga bahasa latinnya seperti pada gambar di samping.

Setelah memastikan aplikasi kita dapat berjalan dengan baik, lakukan commit dengan message sesuai judul dari task 3.4 ini.

Galeri Hewan	
Angsa	<i>Cygnus olor</i>
Ayam	<i>Gallus gallus</i>
Bebek	<i>Cairina moschata</i>
Domba	<i>Ovis ammon</i>

### 3.5. Menambahkan gambar pada list

Pada task ini, kita akan menambahkan gambar pada tampilan list yang sudah dihasilkan pada task sebelumnya. Langkah pertama yang harus dilakukan adalah tambahkan gambar (file terlampir di Zip) ke folder /res/drawable pada project Android Studio masing-masing. Setelah memasukkan gambar ke project, pastikan untuk menambahkan gambar tersebut ke Git. Jika sudah, nama file gambar tersebut akan berubah menjadi warna hijau di explorer.

Berikutnya, ubah tampilan 1 item sehingga memiliki ImageView dan guideline horizontal 50% untuk menampilkan teks di tengah-tengah. Ubah kode list\_item.xml sebagai berikut:

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    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="wrap_content">

    <ImageView
        android:id="@+id/imageView"
        android:layout_width="88dp"
        android:layout_height="88dp"
        android:contentDescription="@null"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        tools:src="@drawable/angsa" />

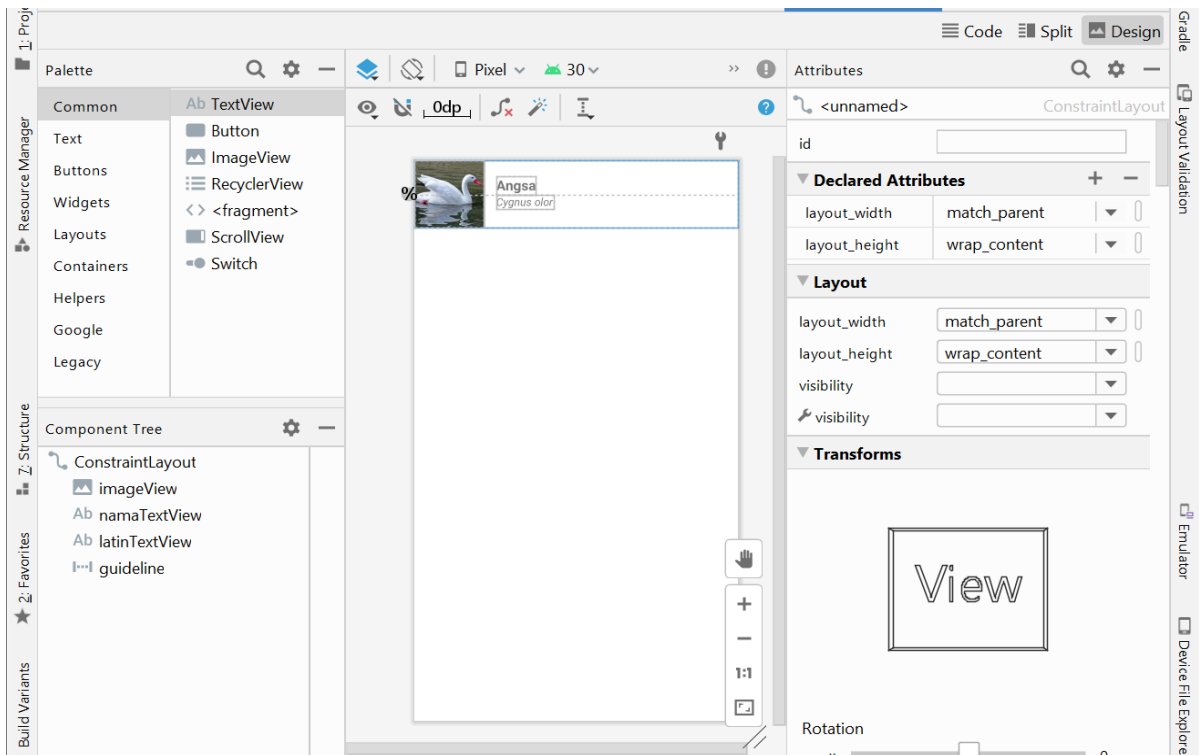
    <TextView
        android:id="@+id/namaTextView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="16dp"
        android:layout_marginTop="16dp"
        android:textSize="18sp"
        android:textStyle="bold"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintBottom_toTopOf="@+id/guideline"
        app:layout_constraintStart_toEndOf="@id/imageView"
        tools:text="Angsa" />

    <TextView
        android:id="@+id/latinTextView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="16dp"
        android:textStyle="italic"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintStart_toStartOf="@+id/namaTextView"
        app:layout_constraintTop_toBottomOf="@+id/namaTextView"
        app:layout_constraintTop_toBottomOf="@+id/guideline"
        tools:text="Cygnus olor" />

    <androidx.constraintlayout.widget.Guideline
        android:id="@+id/guideline"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        app:layout_constraintGuide_percent="0.5" />

</androidx.constraintlayout.widget.ConstraintLayout>

```



Karena adanya tambahan informasi berupa gambar, ubah class Hewan.kt menjadi sebagai berikut:

```
data class Hewan(  
    val nama: String,  
    val namaLatin: String  
    val namaLatin: String,  
    val imageResId: Int  
)
```

Perubahan class ini berpengaruh pada method `getData()` pada `MainActivity.kt`. Hal ini dikarenakan, parameter konstruktor pada class `Hewan` bertambah 1, yaitu `imageResId` dengan tipe `Int`. Ubah kodenya sebagai berikut:

```
private fun getData(): List<Hewan> {  
    return listOf(  
        Hewan("Angsa", "Cygnus olor"),  
        Hewan("Ayam", "Gallus gallus"),  
        Hewan("Bebek", "Cairina moschata"),  
        Hewan("Domba", "Ovis ammon"),  
        Hewan("Kalkun", "Meleagris gallopavo"),  
        Hewan("Kambing", "Capricornis sumatrensis"),  
        Hewan("Kelinci", "Oryctolagus cuniculus"),  
        Hewan("Kerbau", "Bubalus bubalis"),  
        Hewan("Kuda", "Equus caballus"),  
        Hewan("Sapi", "Bos taurus"),  
    )  
}
```

```

        Hewan("Angsa", "Cygnus olor", R.drawable.angsa),
        Hewan("Ayam", "Gallus gallus", R.drawable.ayam),
        Hewan("Bebek", "Cairina moschata", R.drawable.bebek),
        Hewan("Domba", "Ovis ammon", R.drawable.domba),
        Hewan("Kalkun", "Meleagris gallopavo", R.drawable.kalkun),
        Hewan("Kambing", "Capricornis sumatrensis", R.drawable.kambing),
        Hewan("Kelinci", "Oryctolagus cuniculus", R.drawable.kelinci),
        Hewan("Kerbau", "Bubalus bubalis", R.drawable.kerbau),
        Hewan("Kuda", "Equus caballus", R.drawable.kuda),
        Hewan("Sapi", "Bos taurus", R.drawable.sapi),
    )
}

```

Sampai di sini, tidak ada error yang terjadi. Aplikasi dapat dijalankan. Akan tetapi, gambar tetap tidak dimunculkan dan terlihat adanya ruang kosong yang muncul saat aplikasi berjalan. Hal ini dikarenakan kita belum melakukan perubahan adapter yang bertanggung jawab untuk menghubungkan antara data dengan tampilan yang dimiliki. Kode yang menghubungkan antara data dan tampilan terdapat pada MainAdapter.kt, tepatnya pada method bind yang merupakan method pada class ViewHolder.

```

fun bind(hewan: Hewan) = with(binding) {
    namaTextView.text = hewan.nama
    latinTextView.text = hewan.namaLatin
    imageView.setImageResource(hewan.imageResId)
}

```

Jika sudah mengubah kode method bind, jalankan aplikasi kembali. Pada aplikasi yang berjalan di smartphone/emulator akan muncul daftar hewan yang memunculkan gambar, nama bahasa Indonesia, dan juga nama dalam bahasa latinnya.

Setelah memastikan aplikasi berjalan dengan baik, lakukan commit dengan message sesuai judul task 3.5. Cek daftar commit di tab Git di bagian bawah Android Studio. Pastikan commit yang dilakukan sudah sesuai jumlah task ditambah initial commit. Berikutnya, lakukan push kode project ke repository Github menggunakan menu VCS > Git > Push... Refresh repository di browser dan pastikan kodenya sudah sesuai dengan kode versi lokal (komputer/laptop masing-masing).

## 4. Summary

Pada modul kali ini kita telah membuat sebuah aplikasi Android yang menampilkan daftar hewan dalam bentuk list menggunakan komponen RecyclerView. Untuk menggunakan komponen ini, kita perlu menambahkan sebuah layout tambahan untuk menampilkan 1 item dari list dan juga adapter yang memiliki inner class ViewHolder.

Selain mengkodekan pembuatan aplikasi, pada modul ini, kita juga menggunakan version control berupa Git dan juga layanan Github. Terdapat operasi dasar berupa init, add, commit, define remote, dan push untuk menaruh kode versioning ke repository Github sehingga bisa diakses oleh siapa saja.

## 5. Challenge

Pada kode yang telah dibangun sebelumnya, kita membangun tampilan daftar hewan menggunakan 3 informasi: gambar, nama Indonesia, dan nama latin. Coba tambahkan lagi informasi dari hewan tersebut, misalnya jenis hewan (mamalia/unggas/dll). Selain itu, cobalah juga untuk mengubah tampilan list dengan menggunakan layout manager selain LinearLayoutManager.