

## Dasar Android

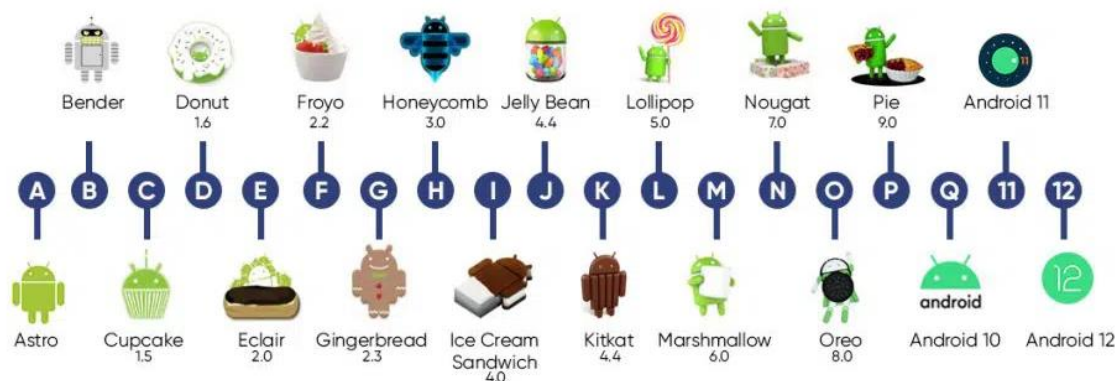
Hai gengs, kalau kemarin kita belajar Kotlin, sekarang kita bakal belajar terkait Dasar dari Pemrograman Android! Sebelum kita kenalan sama Pemrograman Android, kita bakal kenalan dulu nih sama Android itu sendiri.

### Apa itu Android?

Jadi, android itu adalah sistem operasi berbasis linux yang dibuat oleh Google dan dirancang untuk digunakan oleh smartphone atau tablet berbasis touch screen. Berbagai macam produsen telah menggunakan Android sebagai sistem operasi untuk peranti (device) yang mereka produksi. Android juga mempunyai store dengan lebih dari 2.5 miliar pengguna aktif per bulannya per Mei 2019.

### Versi Android

Sejauh ini, Android sudah merilis berbagai macam versi, mulai dari versi lama seperti Jelly Bean, hingga versi terbaru seperti Tiramisu. Berikut ini adalah daftar versi android yang sudah rilis sejauh ini:



Gambar 1. Kumpulan Versi Android

### Pengenalan Android Studio

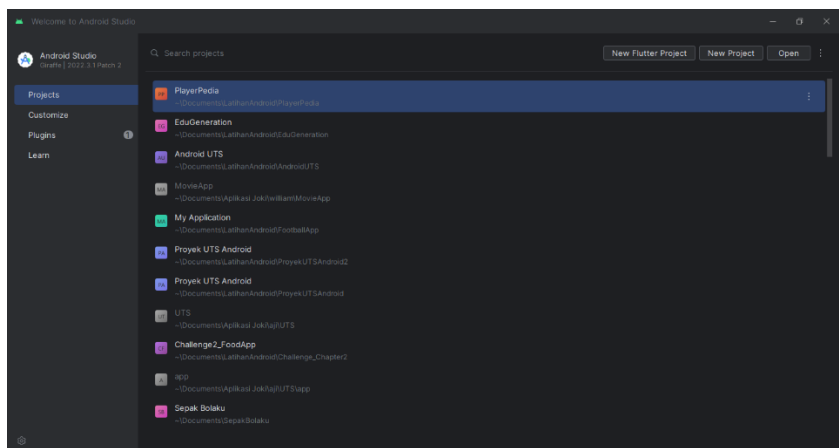
Android Studio adalah Integrated Development Environment (IDE) resmi dari Google untuk pengembangan aplikasi Android. Android studio dibuat berdasarkan IntelliJ IDEA (buatan JetBrains). Ketika menjalankan Android Studio, ada spesifikasi minimal dari sebuah device untuk menjalankannya, spesifikasi minimal ini dibuat agar pengguna lebih nyaman ketika menggunakan Android Studio. Bagi yang laptopnya di bawah standar dari spesifikasi minimal, tetap bisa menjalankan Android Studio namun harus bersabar karena proses menjalankan Android Studio cukup memakan waktu. Berikut ini adalah detail spesifikasi minimal untuk menjalankan Android Studio:

Requirement	Minimum	Recommended
OS	64-bit Microsoft Windows 8	Latest 64-bit version of Windows
RAM	8 GB RAM	16 GB RAM or more
CPU	x86_64 CPU architecture; 2nd generation Intel Core or newer, or AMD CPU with support for a Windows Hypervisor Framework.	Latest Intel Core processor
Disk space	8 GB (IDE and Android SDK and Emulator)	Solid state drive with 16 GB or more
Screen resolution	1280 x 800	1920 x 1080

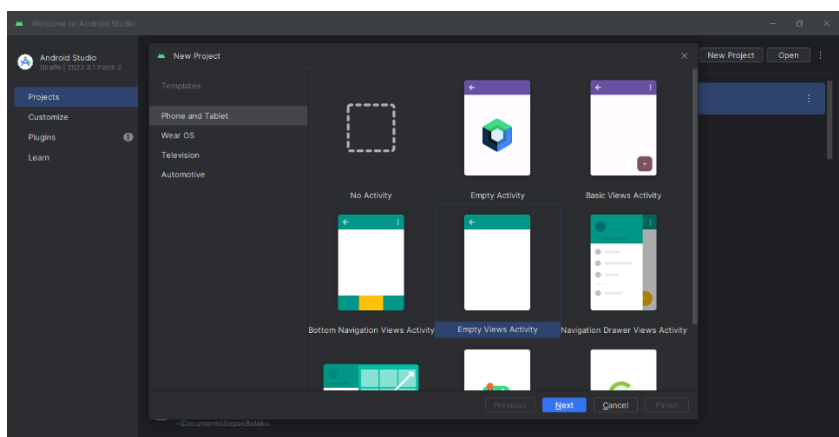
Gambar 2. Spesifikasi Minimal Android Studio

## Membuat Project Android Studio

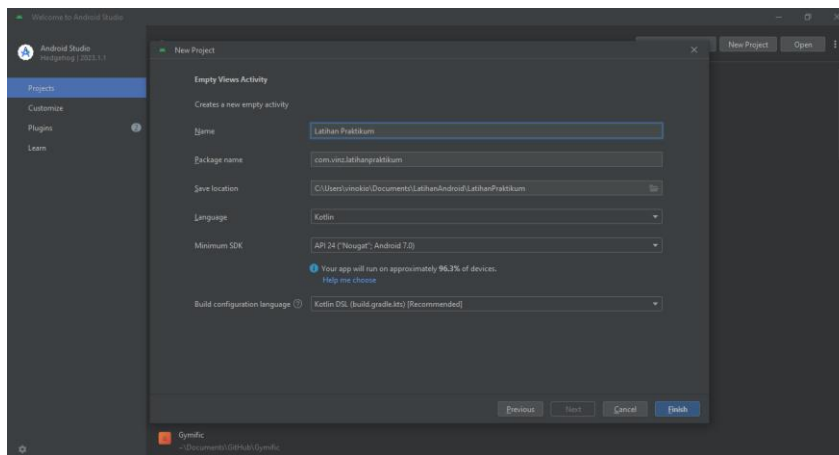
Sebagai langkah pertama, kita akan belajar terkait cara membuat project di Android Studio. Langkah-langkahnya adalah sebagai berikut:



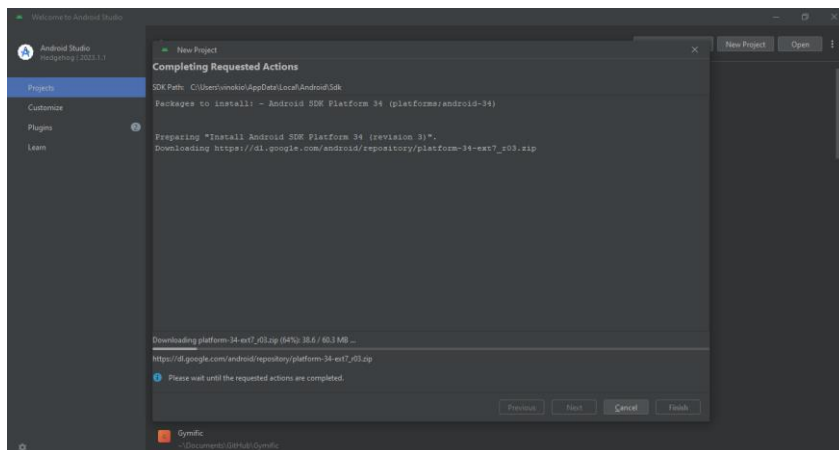
Klik tombol **New Project** di pojok kanan atas



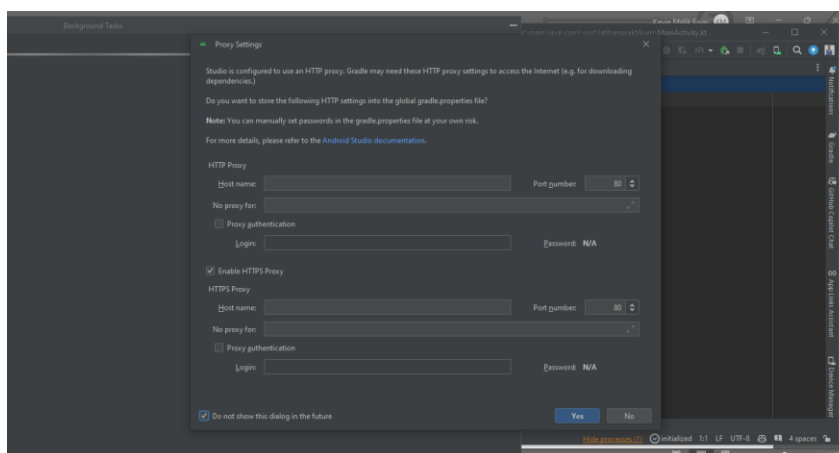
Sebagai awalan, kita pilih yang **Empty Views Activity**



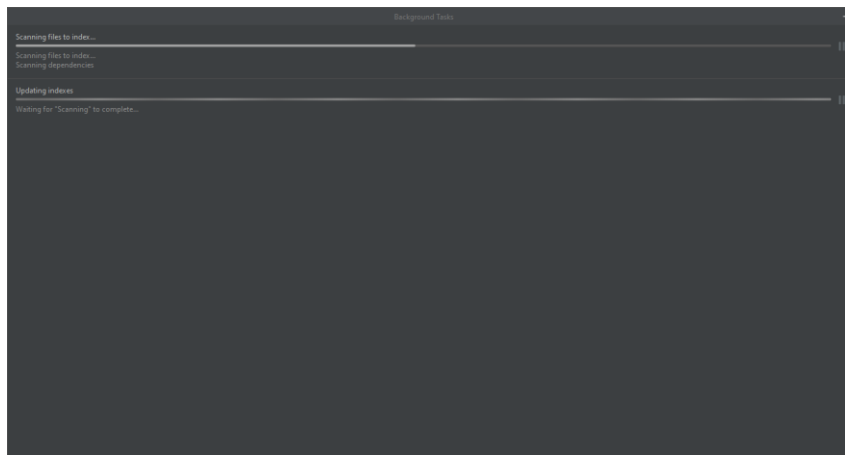
Lalu isi detail dari aplikasi yang mau kamu buat, kalau sudah klik finish



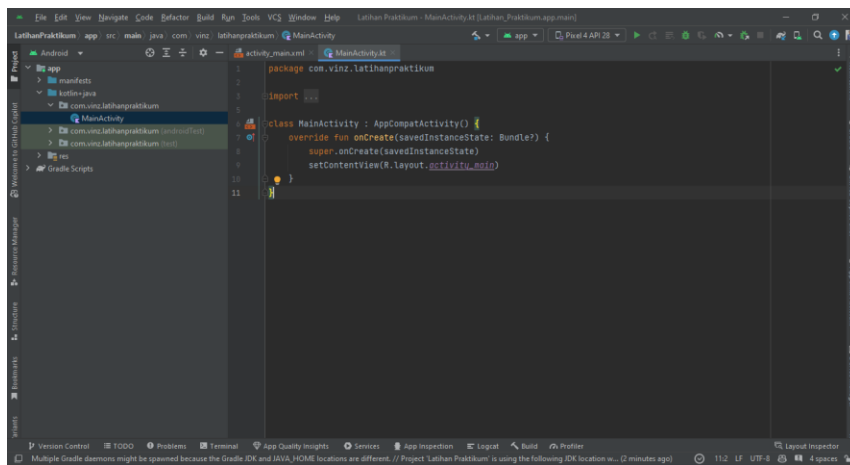
Kalau ada download kaya gini, tungguin aja, nanti kalau udah jangan lupa klik finish.



Kalau ada dialog kaya gini, klik aja yang **Do not show this dialog in the Future**, abis tu klik **Yes**

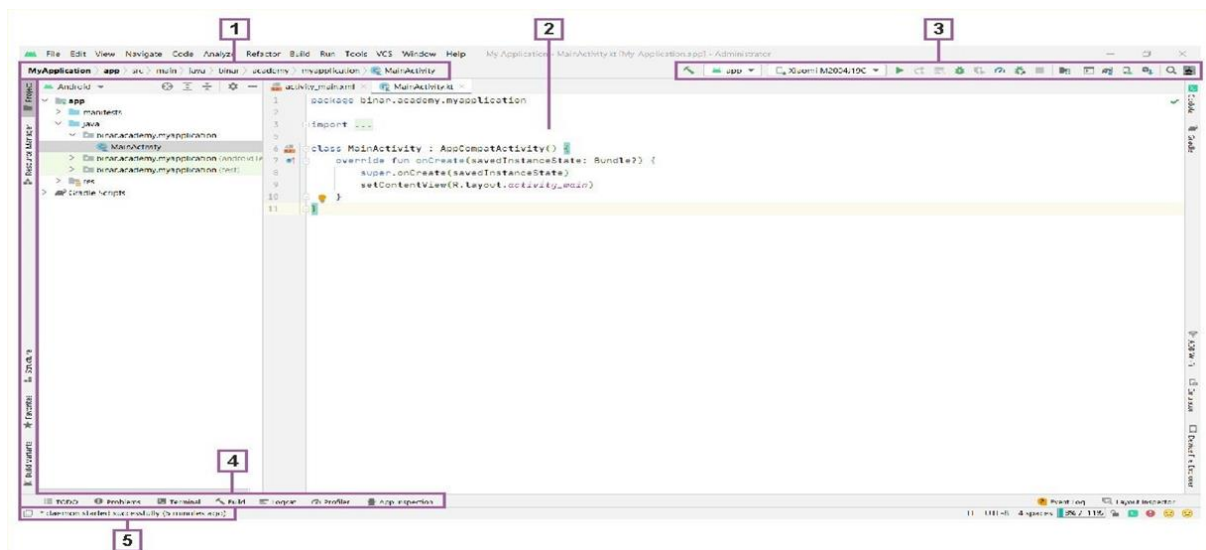


Tunggu loadingnya sampai selesai (agak sedikit lama, harap bersabar)



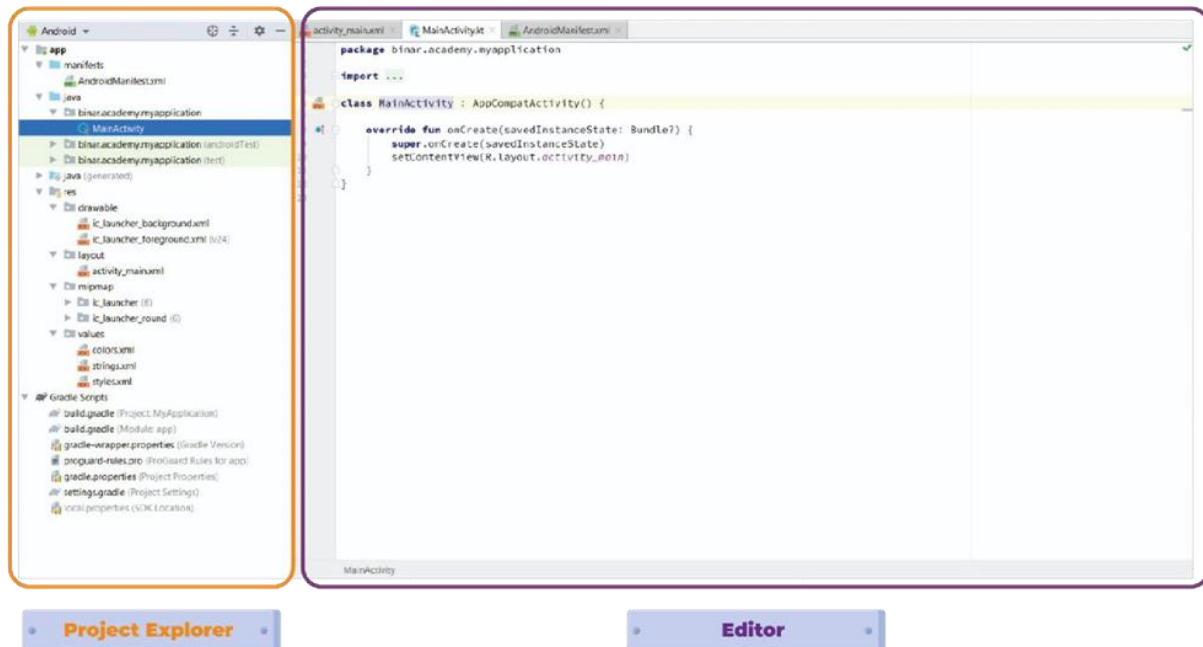
Kalau loadingnya sudah selesai maka akan muncul tampilan seperti berikut

## Interface Android Studio



MyApplication > app > src > main > java > binar > academy > myapplication > MainActivity

Navigation bar ini menunjukkan jalur file yang sedang kita kerjakan. Melalui navigation bar kita juga bisa melihat struktur dari project kita.



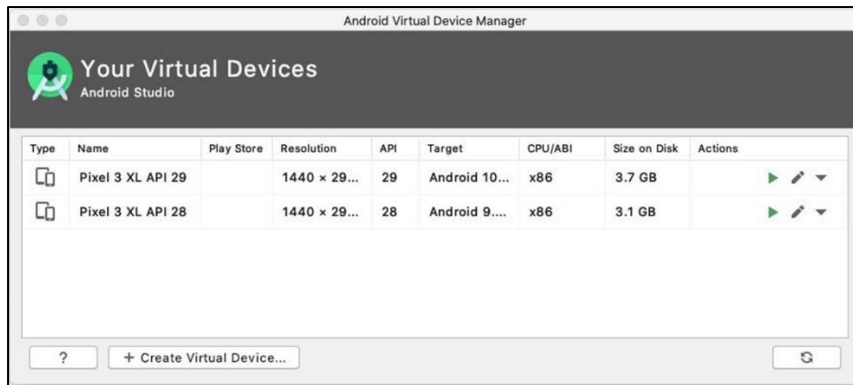
Bagian editor ini adalah layar tempat kita membuka semua file, menuliskan kode, merancang layout, dll. Tergantung pada jenis file yang sedang dibuka. Sedangkan project explorer (sebelah kiri dari editor) adalah keseluruhan file yang terdapat di project milik kita.



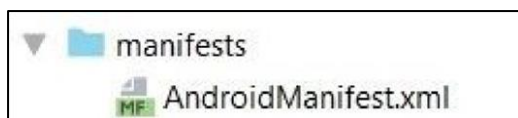
Toolbar menyediakan fitur untuk menjalankan aplikasi, menjalankan emulator, buka SDK manager, dan tools lainnya.



Tool Window Bar merupakan menu tools yang mengelilingi IDE Android Studio. Ini adalah jendela yang terdiri dari beberapa opsi yang bisa kita expand pas diklik.



Berikutnya adalah Android Virtual Device Manager, kalau ini adalah antarmuka yang bisa kamu luncurkan dari Android Studio, yang membantu untuk membuat dan mengelola perangkat virtual Android.



File manifest menjelaskan informasi penting tentang aplikasi kita ke Android build tools, Android operating system, dan Google Play.



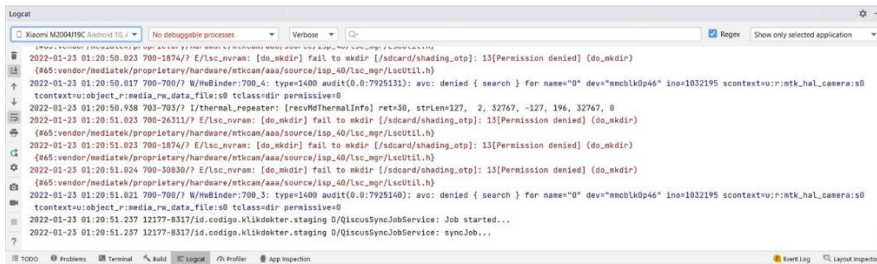
Pada folder Java, terdapat berbagai file source code yang nantinya kita tulis dalam Bahasa Java/Kotlin, termasuk juga unit test maupun android test.



Folder ini terdiri dari semua media aplikasi yang kita butuhkan, kayak layout, gambar, font, warna, string, animation, dan dimensi.

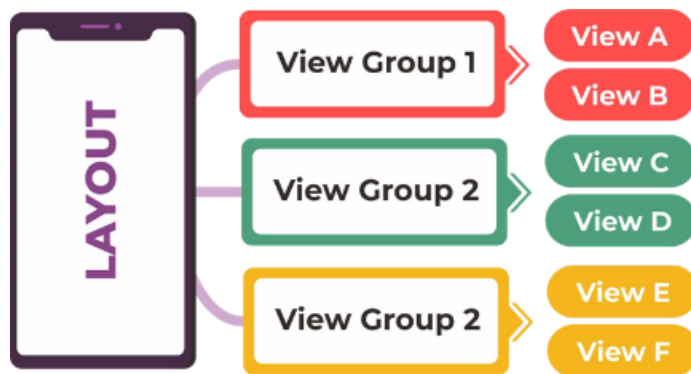


Android Studio menggunakan Gradle sebagai dasar dari build system. Doi punya lebih banyak kemampuan khusus yang disediakan sama plugin Android buat Gradle.

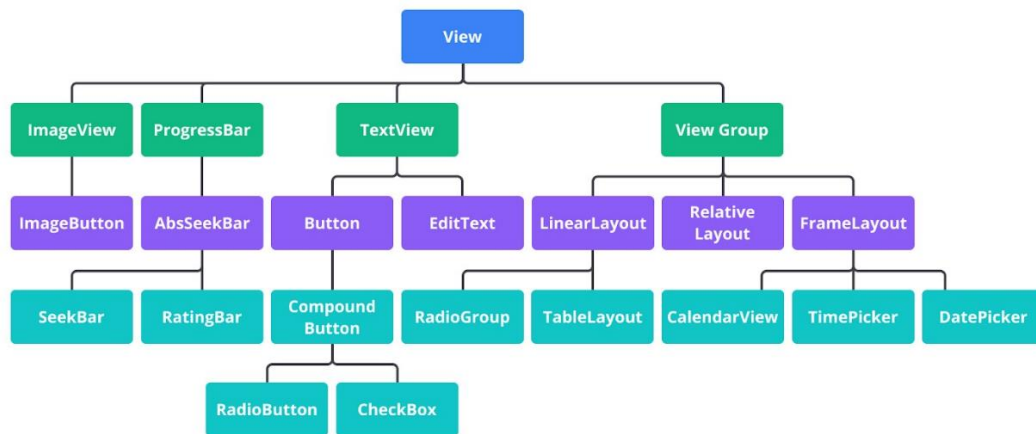


Dengan adanya fitur Logging, kamu bakal tahu letak baris code yang bikin aplikasi kamu error.

## Pengenalan Layout di Android Studio

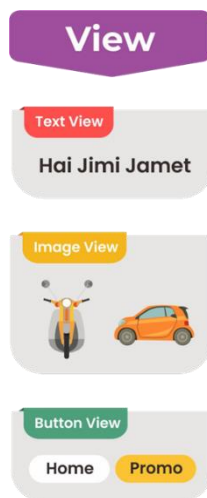


View mewakili komponen yang dapat kita lihat seperti teks, tombol, dan gambar. Sedangkan ViewGroup adalah yang mengatur semua komponen itu tertata rapi. Gampangnya, ViewGroup adalah parent (orang tua) dan views di dalamnya adalah child-nya (anaknya).



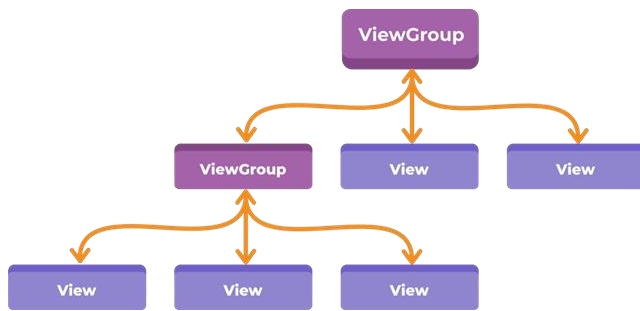
Gambar 3. Struktur View

Kalau kita lihat gambar di atas, beberapa view yang terdapat di dalam android studio sebenarnya berasal dari sumber yang sama. Seperti misalnya Radio Button dan CheckBox yang ternyata berasal dari Compound Button. Lalu ada ViewGroup seperti RadioGroup dan TableLayout yang berasal dari LinearLayout.

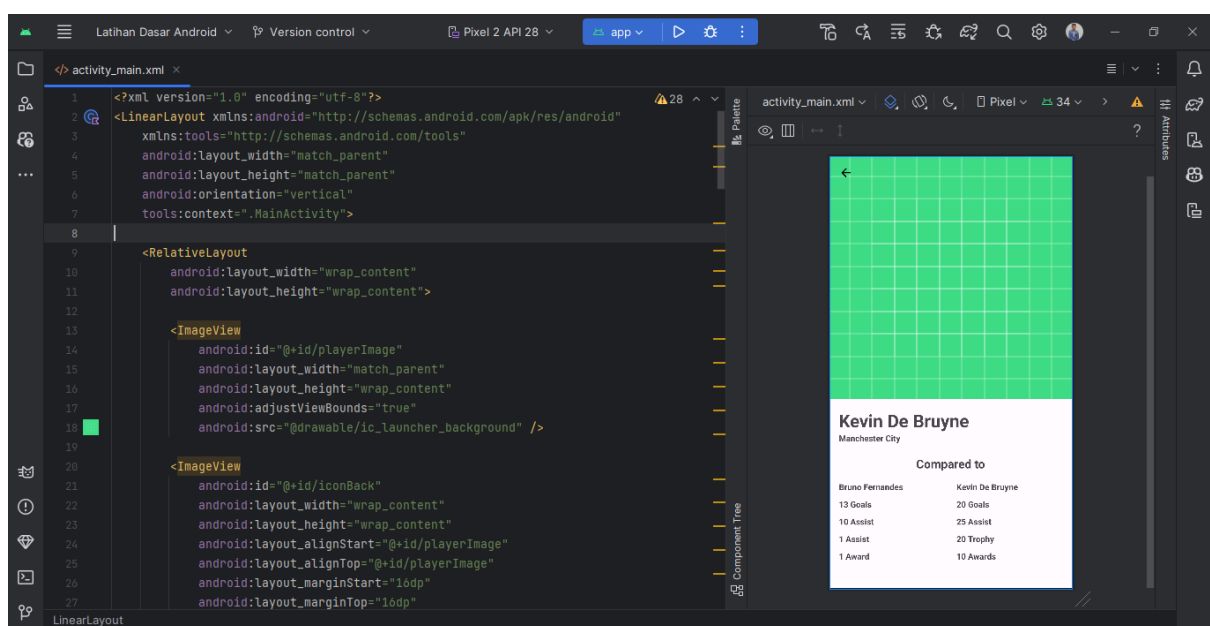


Android View adalah class dasar untuk semua komponen UI di android. Misalnya, class ImageView digunakan untuk menampilkan gambar di aplikasi android yang merupakan subclass dari View.

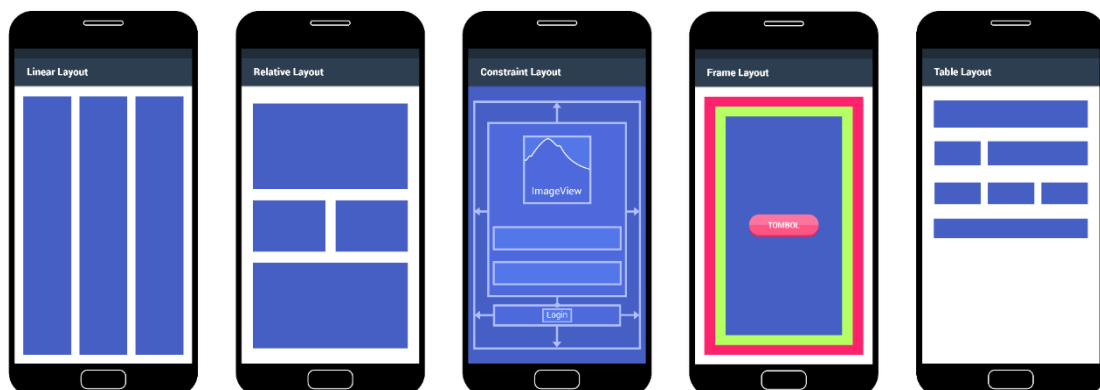




Dalam sebuah ViewGroup nantinya bisa menampung dua buah komponen View, di mana satu komponen ViewGroup terdiri dari 3 buah komponen View. Terus, di dalam ViewGroup tersebut bisa ada View ataupun ViewGroup lagi. Hal ini disebut dengan Nested ViewGroup.



Contoh nested view group di samping, di mana kita menggunakan RelativeLayout untuk membuat button Back di dalam LinearLayout.



Gambar 4. Berbagai macam tipe ViewGroup

Gambar di atas adalah contoh dari berbagai macam ViewGroup. Mulai dari yang paling kiri yaitu Linear Layout dan yang paling kanan yaitu Table Layout.

## Linear Layout



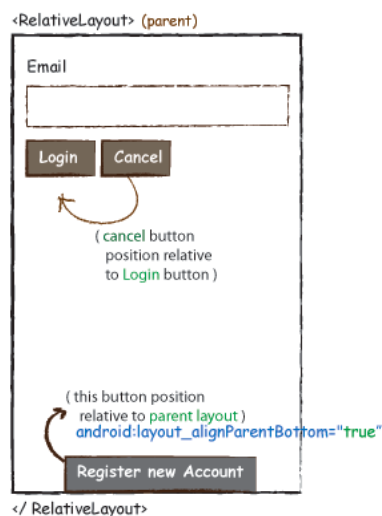
`android:orientation="vertical"`



`android:orientation="horizontal"`

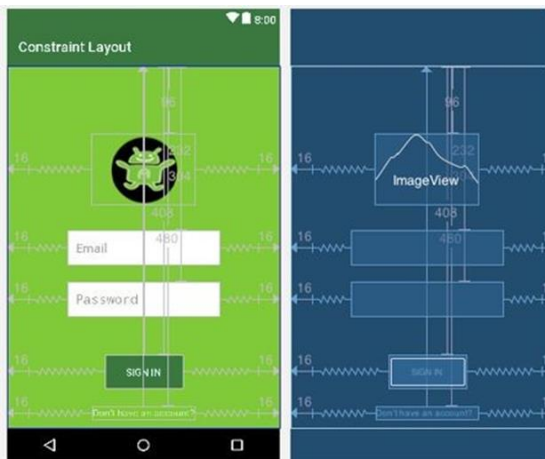
LinearLayout adalah layout yang akan menempatkan komponen-komponen di dalamnya secara berkelompok. Kita bisa memilih apakah layout bakal dimuat secara horizontal maupun vertikal.

## Relative Layout



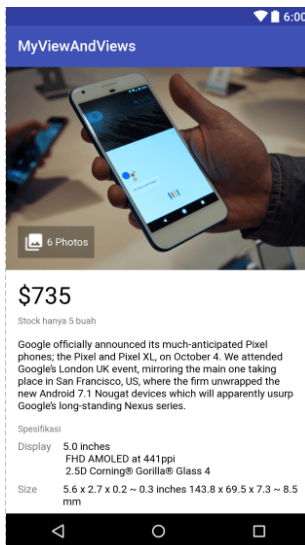
RelativeLayout digunakan buat menentukan posisi setiap komponen secara relatif terhadap komponen yang lain. Posisi setiap View bisa ditentukan sebagai relatif terhadap komponen lain (kayak di sebelah kirinya atau di bawahnya tampilan lain).

## ConstraintLayout



ConstraintLayout memungkinkan kita membuat layout yang kompleks dengan hierarki flat (tanpa Nested ViewGroup). Mirip dengan dengan RelativeLayout, di mana semua View ditata sesuai dengan keterkaitannya dengan View, ViewGroup, ataupun pada parent-nya. **Tapi funfact-nya, ConstraintLayout lebih fleksibel daripada RelativeLayout.**

## ScrollView

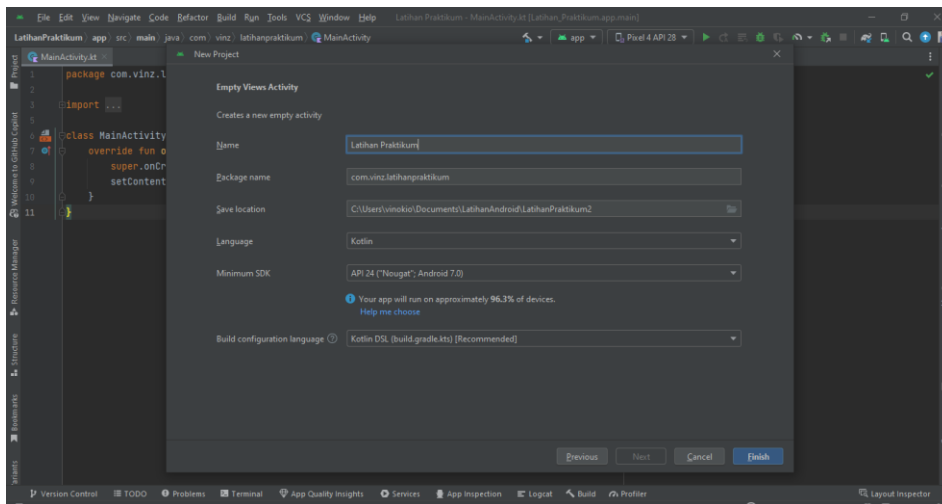


```
<?xml version="1.0" encoding="utf-8"?>
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/activity_main"
    android:layout_height="match_parent"
    android:layout_width="match_parent"
    android:background="@android:color/white"
    tools:context="com.dicoding.myviewandviews.MainActivity">
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical">
        <FrameLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content">
            <ImageView
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:adjustViewBounds="true"
                android:src="@drawable/pixel_google"
                android:scaleType="fitXY"/>
            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:padding="8dp"
                android:text="6 Photos"
                android:gravity="center_vertical"
                android:drawableLeft="@drawable/ic_collections_white_18dp"
                android:drawablePadding="4dp"
                android:textAppearance="@style/TextAppearance.AppCompat.Small"
                android:background="@android:color/white"
                android:textColor="@android:color/white"
                android:layout_marginLeft="@dimen/activity_horizontal_margin"
                android:layout_marginBottom="@dimen/activity_vertical_margin"
                android:layout_gravity="bottom" />
        </FrameLayout>
        <FrameLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginTop="8dp">
            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:padding="8dp"
                android:text="$735"
                android:layout_marginLeft="@dimen/activity_horizontal_margin"
                android:layout_marginRight="@dimen/activity_horizontal_margin"
                android:layout_marginTop="@dimen/activity_vertical_margin"
                android:layout_marginBottom="8dp"
                android:textColor="@android:color/black" />
            <TextView
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:padding="8dp"
                android:text="Google officially announced its much-anticipated Pixel phones; the Pixel and Pixel XL, on October 4. We attended Google's London UK event, mirroring the main one taking place in San Francisco, US, where the firm unwrapped the new Android 7.1 Nougat devices which will apparently usurp Google's long-standing Nexus series." />
            <Table
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:padding="8dp">
                <tr>
                    <th>Spesifikasi</th>
                    <th>Detail</th>
                </tr>
                <tr>
                    <td>Display</td>
                    <td>5.0 inches FHD AMOLED at 441ppi 2.5D Corning® Gorilla® Glass 4</td>
                </tr>
                <tr>
                    <td>Size</td>
                    <td>5.6 x 2.7 x 0.2 ~ 0.3 inches 143.8 x 69.5 x 7.3 ~ 8.5 mm</td>
                </tr>
            </Table>
        </FrameLayout>
    </LinearLayout>
</ScrollView>
```

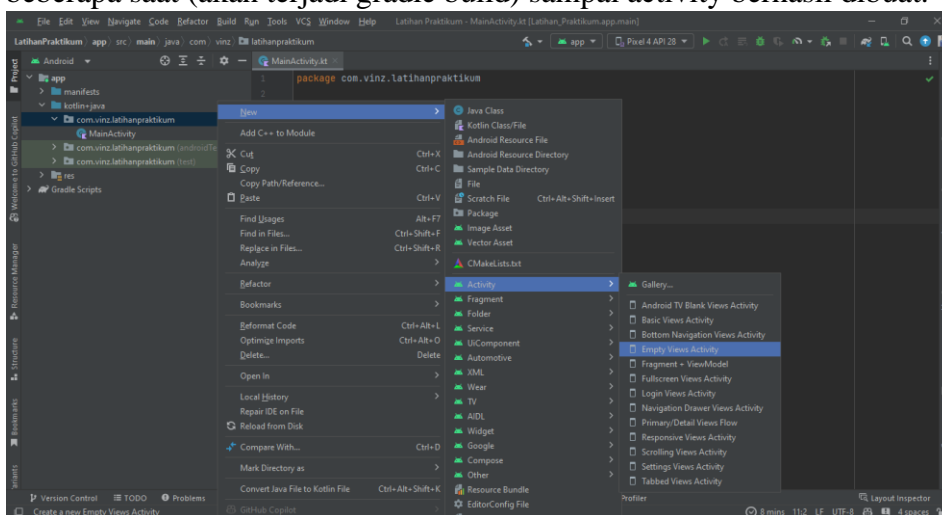
Ketika aplikasi memuat informasi yang banyak dan melebihi ukuran layar, Anda membutuhkan tampilan yang memungkinkan pengguna untuk membaca informasi dengan lengkap. Kita membutuhkan komponen ScrollView untuk mengatasi masalah ini.

## Let's Practice!

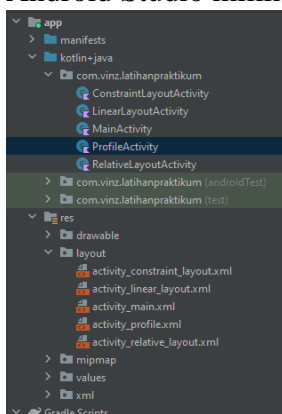
1. Silakan buat project baru (ikuti langkah untuk membuat project baru di atas), beri nama project tersebut dengan nama terserah kalian, kalau sudah klik finish dan tunggu beberapa saat sampai projectnya selesai terbuat.



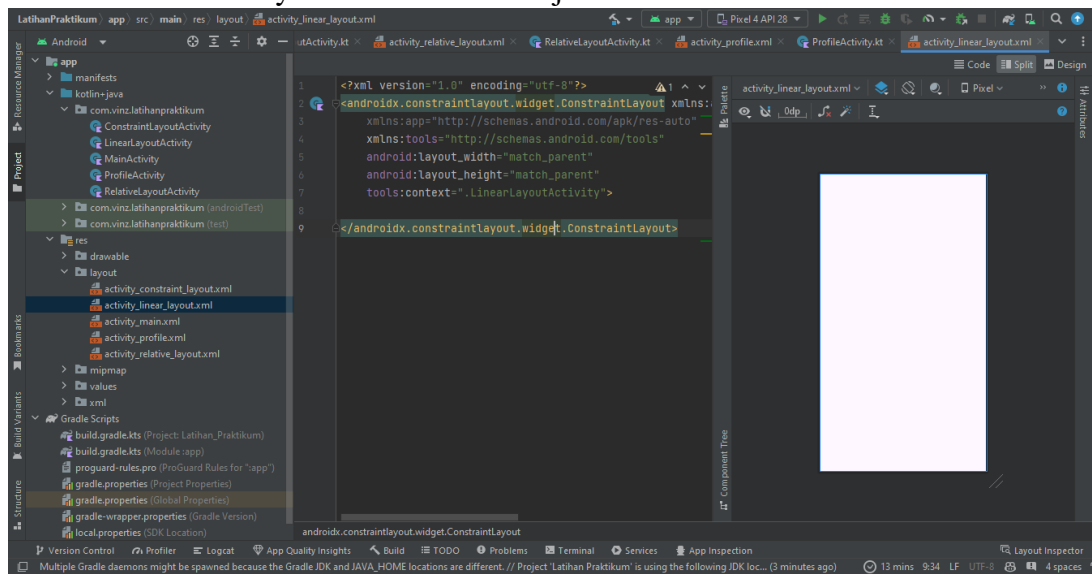
2. Kalau sudah, silakan klik project dan klik kanan di package milik kita, lalu pilih new – activity – empty views activity, kita akan buat 4 activity baru, silakan beri nama activity tersebut dengan nama **LinearLayoutActivity**, **ConstraintLayoutActivity**, **RelativeLayoutActivity**, dan **ProfileActivity**. kalau sudah klik finish dan tunggu beberapa saat (akan terjadi gradle build) sampai activity berhasil dibuat.



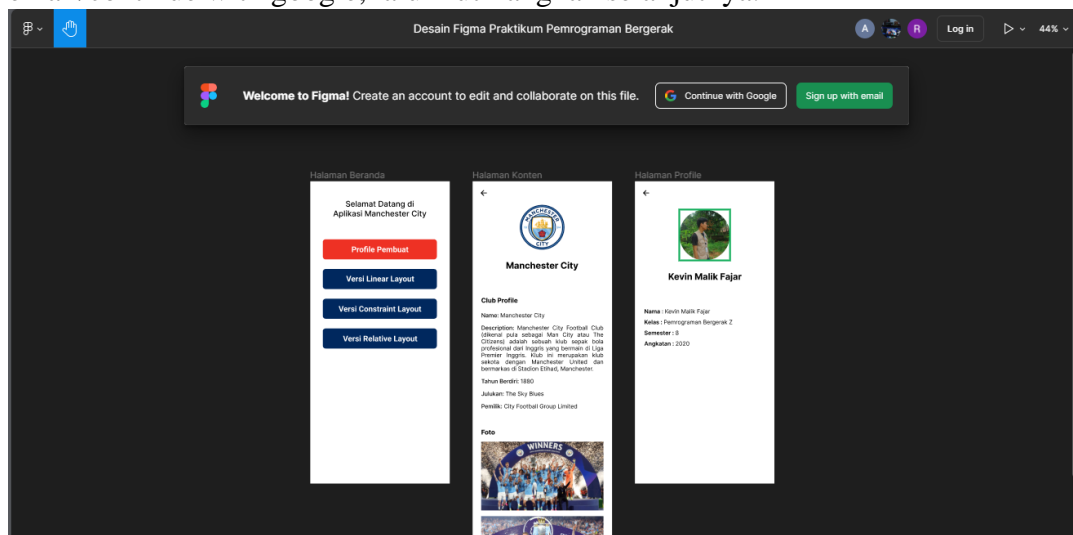
3. Kalau sudah berhasil terbuat, nantinya akan ada file LinearLayoutActivity, ConstraintLayoutActivity, RelativeLayoutActivity, dan ProfileActivity. di project Android Studio milik kamu. Setelah itu coba buka file activity\_linear\_layout.xml



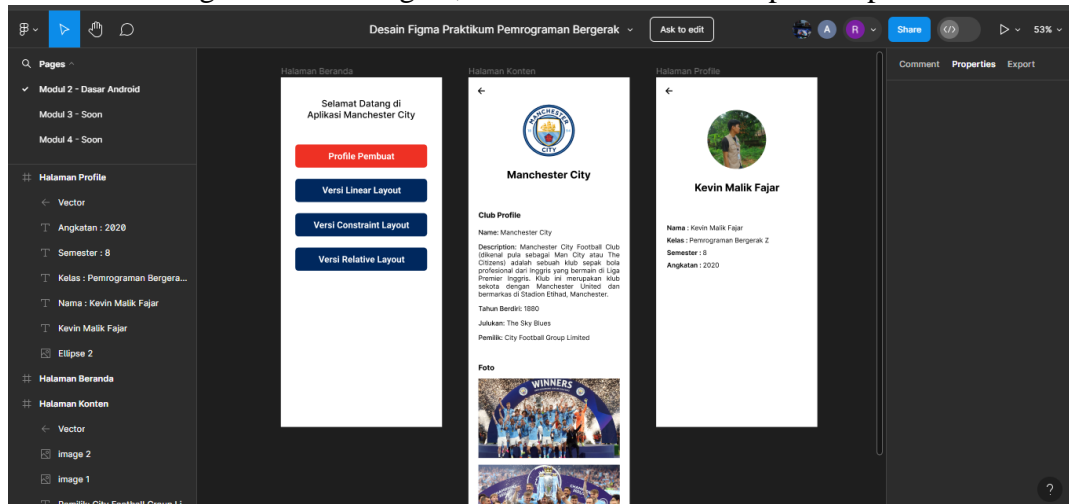
4. Kalau sudah terbuka nanti akan muncul tampilan seperti ini, file XML berfungsi untuk melakukan modifikasi terhadap tampilan dari sebuah activity, gampangnya sih dia adalah html dan css dari aplikasi android milik kamu, jadi bener-bener fokus dari file XML adalah hanya di user interface saja.



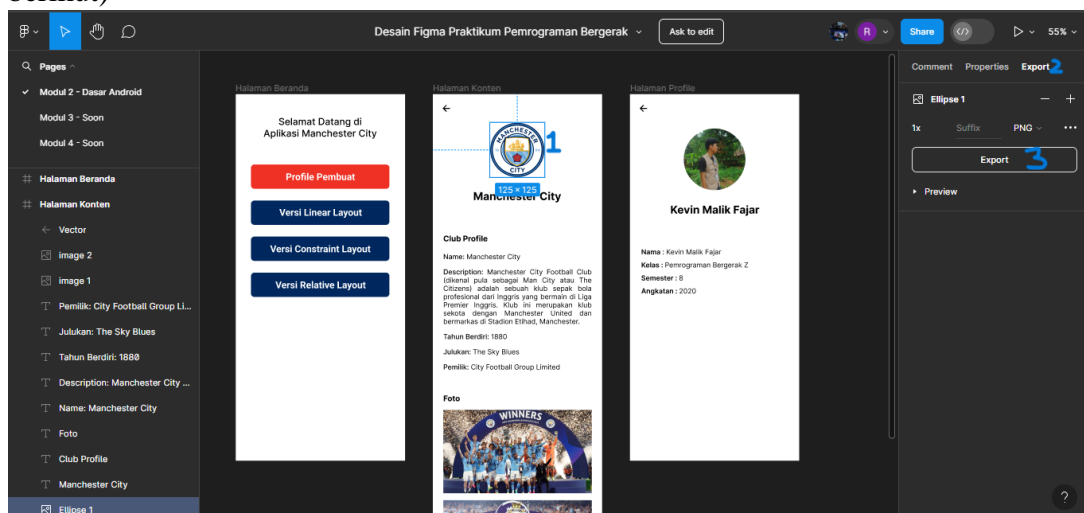
5. Kalau sudah terbuka, kita akan mencoba menerapkan desain yang terdapat di dalam figma [berikut ini](#) (silakan buka linknya)
6. Kalau sudah dibuka, maka akan muncul tampilan seperti ini (jika kalian belum login ke figma), silakan login ke figma terlebih dahulu dengan cara klik sign up with email/continue with google, lalu ikuti langkah selanjutnya.



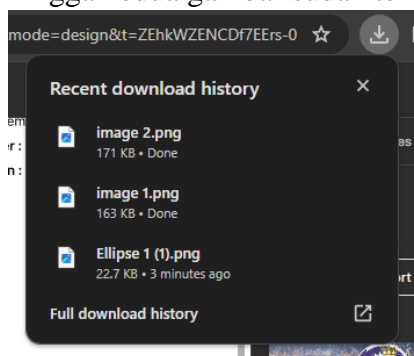
7. Kalau sudah login ke dalam Figma, maka akan muncul tampilan seperti ini



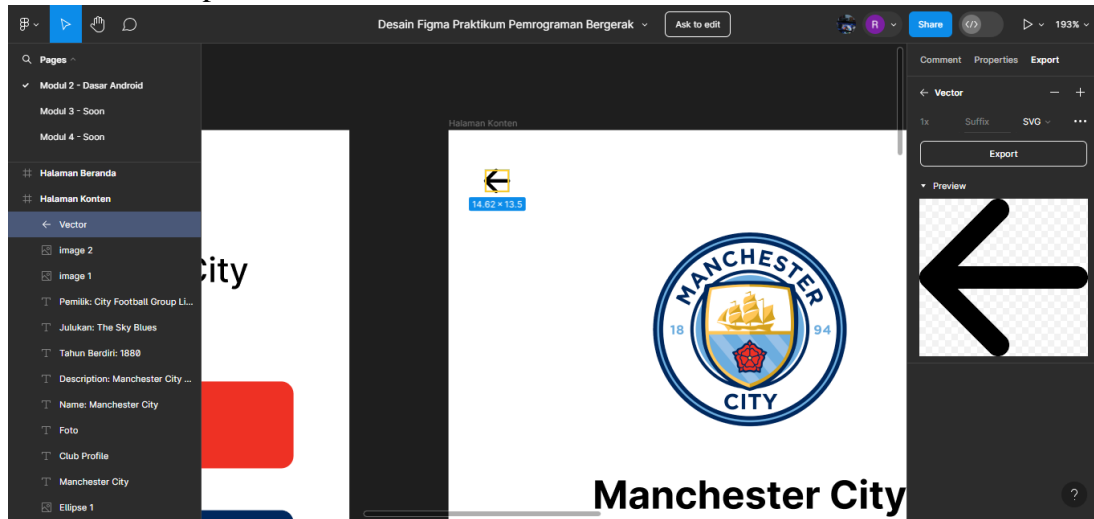
8. Kita akan mulai terapkan desain aplikasi tersebut ke dalam aplikasi yang kita miliki, langkah pertama silakan klik pada gambar Manchester city di atas, lalu kita akan export gambar tersebut dengan cara klik export (di samping properties), lalu klik lagi tombol export. (Kalau bingung silakan liat urutan penomoran klik pada gambar berikut)



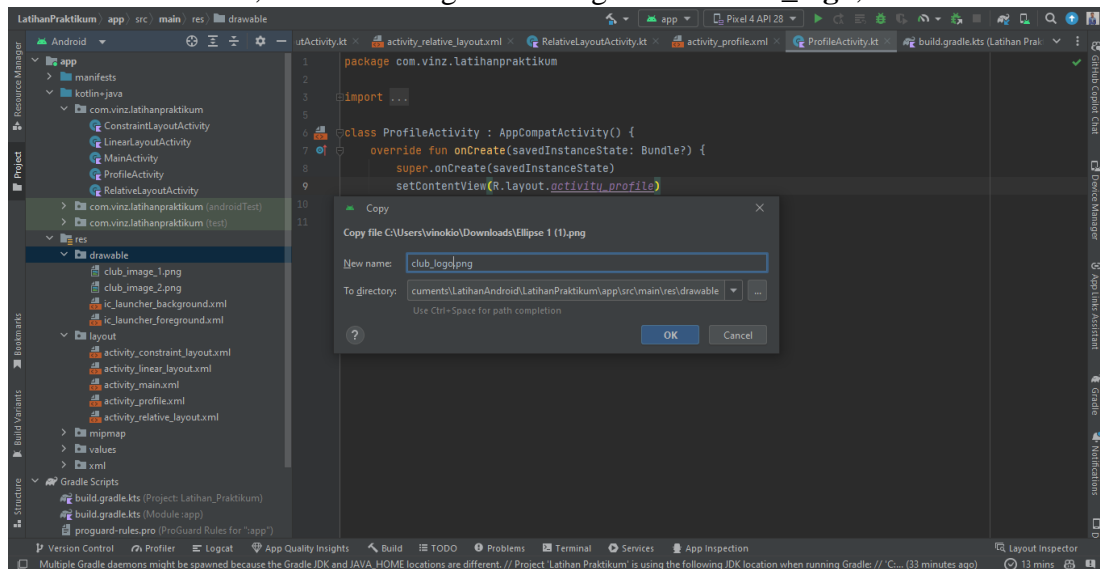
9. Kalau sudah nanti gambar akan otomatis ter-download. Ulangi langkah tersebut untuk 2 gambar yang terdapat di bawah tulisan **Foto** (letaknya masih di Halaman Konten) hingga kedua gambar sudah ter-download secara sempurna.



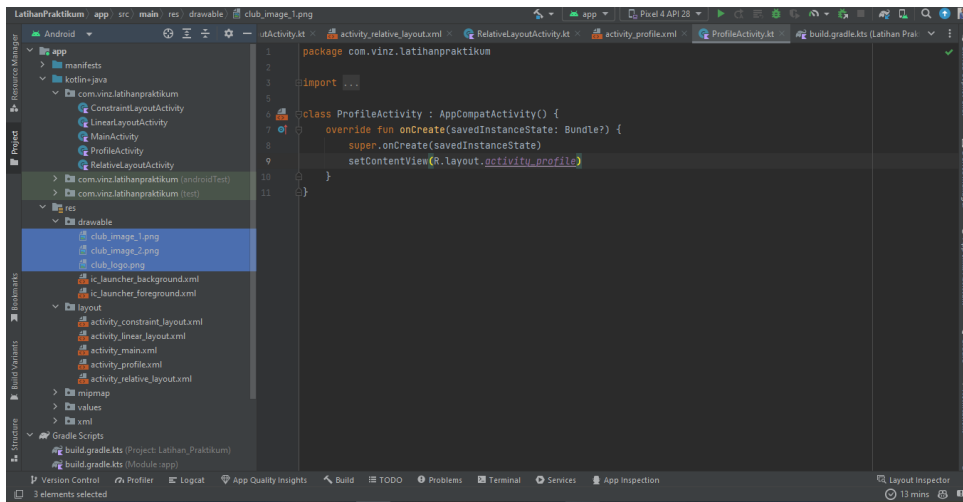
10. Ulangi langkah tersebut untuk men-download **Tombol Back** (letaknya di pojok kiri atas di Halaman Konten), namun untuk tombol back, silakan pilih opsi **SVG**, setelah itu baru klik export



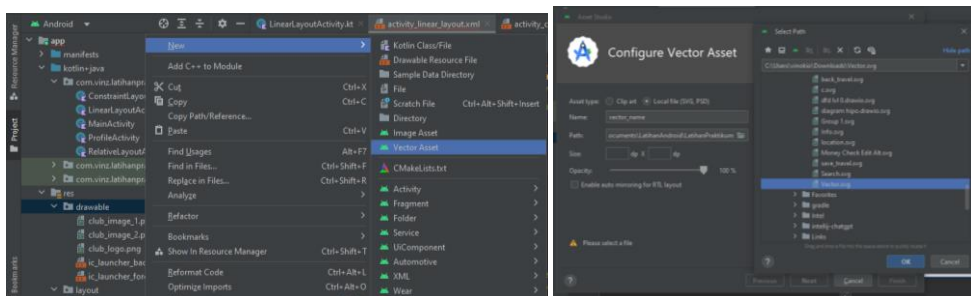
11. Kalau sudah silakan buka **File Manager** dari laptop milik teman-teman, buka folder **Download**, lalu copy gambar tersebut (CTRL+C) mulai dari gambar dengan judul ellipse, dan paste-kan (CTRL+V) gambar tersebut ke folder **Drawable** yang terdapat di **Android Studio**, lalu isi nama gambar dengan nama **club\_logo**, lalu klik **Ok**



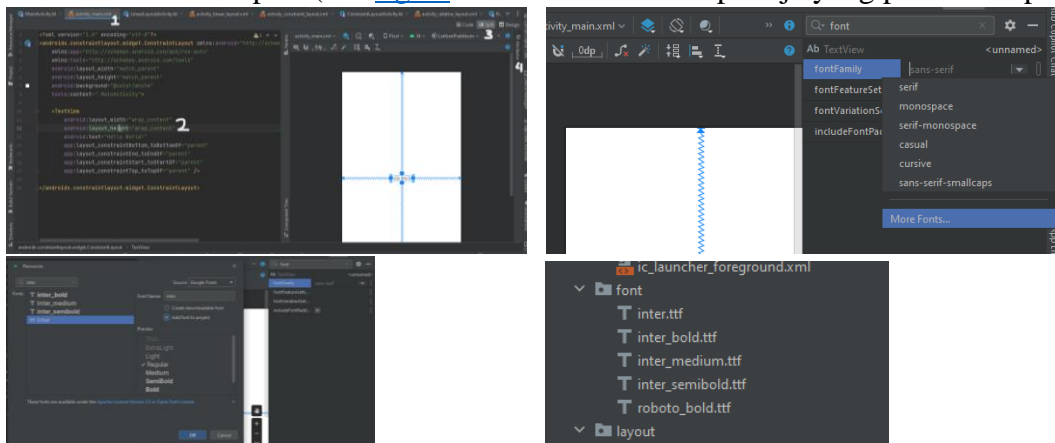
12. Ulangi langkah tersebut untuk gambar yang lain, yaitu image 1 dan image 2. Ubah nama **image 1** menjadi **club\_photo\_1** dan **image 2** menjadi **club\_photo\_2**, kalau sudah sukses maka gambar akan sukses dipaste ke **Drawable**.



13. Lalu yang terakhir, silakan masukkan Tombol Back ke dalam drawable dengan cara klik kanan pada Folder **Drawable**, lalu klik **New**, Klik **Vector Asset** dan di asset type pilih **Local File** dan klik icon **folder** di kolom **Path**, lalu arahkan ke folder **Download** dan pilih **Vector.svg**, kalau sudah klik **ok** dan ubah namanya menjadi **ic\_back**, lalu klik **next** dan **finish**.

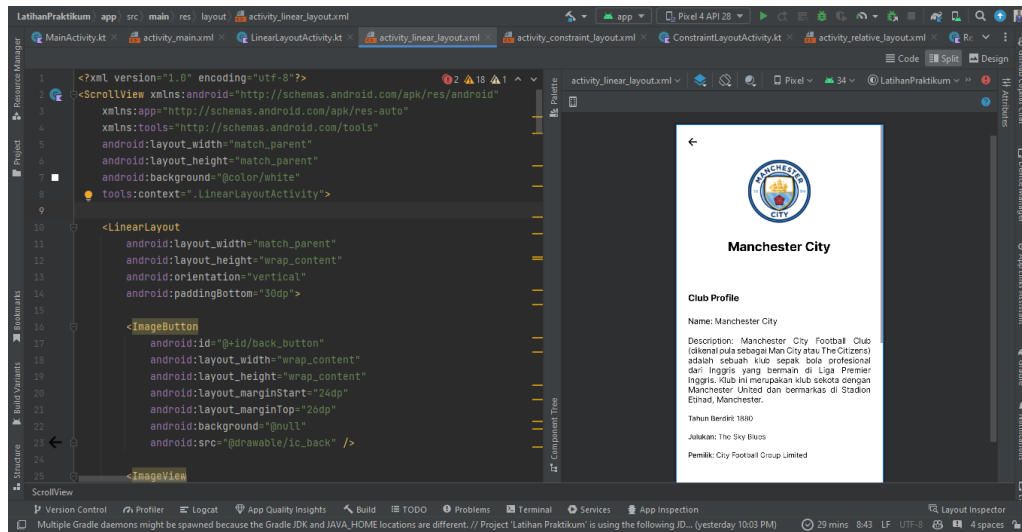


14. Selanjutnya, kita akan coba untuk import font yang akan kita butuhkan di dalam aplikasi ini. Silakan temen-temen buka **activity\_main.xml**, lalu klik **TextView** dan pastikan sudah mengaktifkan mode **split**, lalu kalau sudah silakan klik **attributes** dan di kolom pencarian silakan ketik **font** dan klik dropdown di kolom **fontFamily**, lalu scroll ke bawah dan klik **more font** dan cari font yang ingin kamu import, pilih yang **add font to project**, setelah itu klik **OK** dan ulangi langkah tersebut sampai semua font berhasil ter-import (cek [figma](#) untuk melihat font apa saja yang perlu di import)

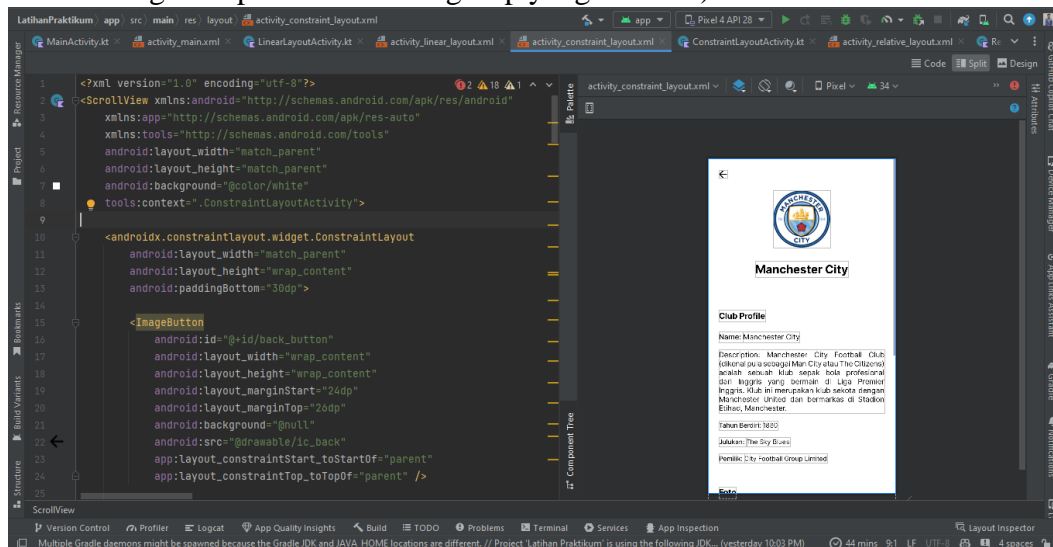




15. Lalu silakan paste-kan kode di [link berikut](#) ke dalam **activity\_linear\_layout.xml**, pastikan nama file yang kamu import (seperti gambar, font) sudah sesuai seperti yang ditentukan di atas, jika sudah sesuai harusnya akan muncul tampilan seperti berikut ini.



16. Selanjutnya, silakan buka **activity\_constraint\_layout.xml** di dalam folder **layout** dan paste-kan kode di [link berikut](#) ke dalam file **activity\_constraint\_layout.xml**, jika sudah sesuai harusnya akan muncul tampilan seperti berikut ini (tampilan yang sama namun dengan implementasi viewgroup yang berbeda)



17. Lalu buka file **activity\_relative\_layout.xml** di dalam folder **layout** dan paste-kan kode di [link berikut](#) ke dalam file **activity\_relative\_layout.xml**, jika sudah sesuai harusnya akan muncul tampilan seperti berikut ini (tampilan yang sama namun dengan implementasi viewgroup yang berbeda)



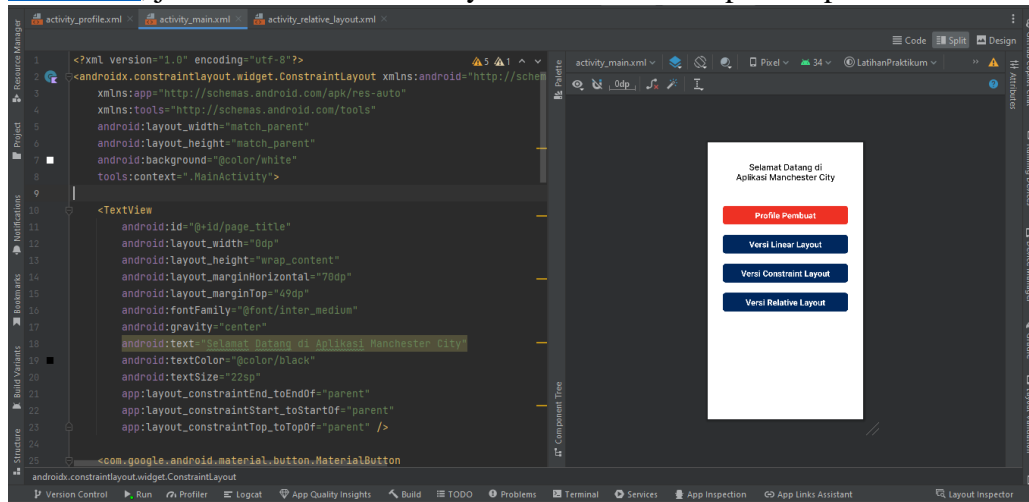
bg\_blue\_btn.xml

```
<?xml version="1.0" encoding="utf-8"?>
<shape android:shape="rectangle" xmlns:android="http://schemas.android.com/apk/res-auto"
    <solid android:color="#00285E"/>
    <corners android:radius="8dp"/>
</shape>
```

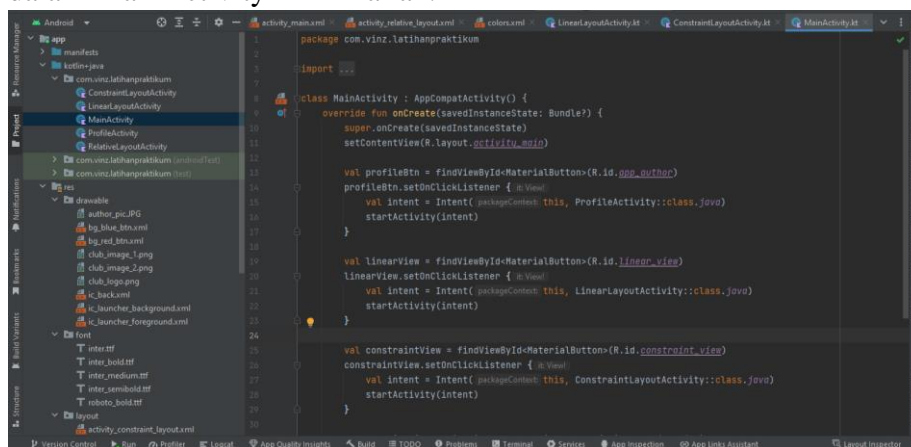
bg\_red\_btn.xml

```
<?xml version="1.0" encoding="utf-8"?>
<shape android:shape="rectangle" xmlns:android="http://schemas.android.com/apk/res-auto"
    <solid android:color="#EE3124"/>
    <corners android:radius="8dp"/>
</shape>
```

20. Silakan buka file **activity\_main.xml** di folder **layout** dan masukan kode dari link [berikut ini](#), jika sudah sesuai harusnya akan muncul tampilan seperti berikut ini



21. Lalu silakan buka file **MainActivity**, dan paste-kan kode dari link [berikut ini](#) ke dalam MainActivity milik kalian.



22. Terakhir, silakan paste-kan kode berikut di **ConstraintLayoutActivity**, **LinearLayoutActivity**, **ProfileActivity**, dan **RelativeLayoutActivity**.

```
val backButton = findViewById<ImageButton>(R.id.back_button)
backButton.setOnClickListener {
    onBackPressedDispatcher.onBackPressed()
}
```

23. Lalu coba jalankan program tersebut dengan cara klik tombol **Run** (kotak biru), namun sebelum menjalankan program, pastikan **Sudah Ada Device** (garis merah) yang tersedia (entah hp/emulator) untuk menjalankan program, jika semua mengikuti prosedur dan tidak ada masalah seharusnya akan tampil aplikasi seperti berikut ini dengan tingkat kemiripan hampir 100% (ada kemungkinan terjadi error ketika melakukan Run karena kesalahan dari Android Studio, namun bukan error di Program)

```

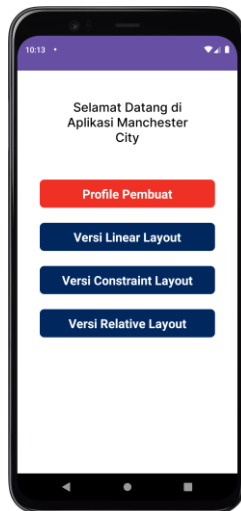
package com.vinz.latihanpraktikum

import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.view.View
import android.widget.Button

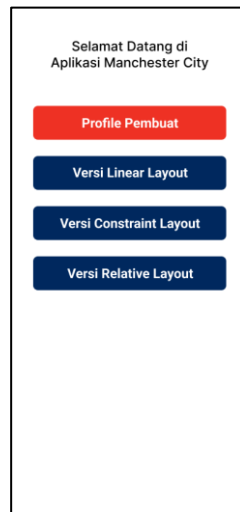
class RelativeLayoutActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_relative_layout)
    }
}

```

**Aplikasi**



**Desain**



**Aplikasi**



**Desain**



**Aplikasi**



**Desain**



Oiya, seluruh source code dari aplikasi latihan di atas bisa temen-temen akses di link [berikut ini](#)

## **Tugas Pendahuluan**

1. Apa itu View dan ViewGroup di dalam Layout Android?
2. Apa itu activity di dalam sebuah project Android?
3. Apa perbedaan antara file Kotlin dan XML di dalam sebuah activity?

## **Tugas Praktikum**

Silahkan buka link [berikut ini](#), amati desainnya lalu pilih 1 style ViewGroup yang ingin kalian gunakan (misalkan ConstraintLayout), setelah itu implementasikan semua desain screen tersebut menggunakan ViewGroup yang kalian pilih dan berikan alasannya kenapa kalian menggunakan ViewGroup tersebut (sertakan alasan menggunakan ViewGroup di laporan praktikum) dan berikan navigasi untuk perpindahan antar screen!

Alur Aplikasi (pastikan alur aplikasinya seperti ini)

1. Ketika user pertama kali membuka aplikasi, maka akan diarahkan ke Halaman Masuk
2. Jika pengguna menekan tombol Lanjut di Halaman Masuk, maka akan diarahkan ke Halaman Beranda
3. Halaman Beranda bisa discroll karena kontennya yang cukup banyak
4. Jika logo Universitas Trunojoyo di pojok kanan atas diklik (dari Halaman Beranda), maka akan diarahkan ke Halaman Profile

Style: Seragamkan style dengan desain yang terdapat di Figma