

# **LAPORAN PRAKTIKUM**

## **PEMROGRAMAN BERBASIS MOBILE**

### **PERTEMUAN KE-4**



**Disusun Oleh :**

**NAMA** : Raden Isnawan Argi Aryasatya  
**NIM** : 195410257  
**JURUSAN** : Informatika  
**JENJANG** : S1  
**KELAS** : 5

**Laboratorium Terpadu**  
**Sekolah Tinggi Manajemen Informatika Komputer**  
**AKAKOM**  
**YOGYAKARTA**

**2021**

## **PERTEMUAN KE-4** **(LINEAR DAN CONSTRAINT LAYOUT)**

### **TUJUAN**

1. Mahasiswa mampu memahami dan mengetahui tentang Linear Layout dan Constraint Layout
2. Mahasiswa mampu menggunakan komponen view dasar (TextView, EditText, Button) dalam aplikasi

### **DASAR TEORI**

**Layout** menentukan struktur untuk antarmuka pengguna di aplikasi Anda, seperti di dalam activity. Semua elemen pada layout dibuat menggunakan hierarki objek View dan ViewGroup. View biasanya menggambar sesuatu yang pengguna bisa melihat dan berinteraksi dengannya. Sedangkan ViewGroup adalah container yang tidak terlihat yang menentukan struktur layout untuk View dan objek ViewGroup lainnya. Di modul ini kita membahas tentang dua komponen viewgroup yaitu:

**LinearLayout** adalah sekelompok tampilan yang menyejajarkan semua anak dalam satu arah. Anda bisa menetapkan arah layout dengan atribut `android:orientation`. Dengan kata lain, Linear Layout adalah jenis layout dimana user menempatkan 1 objek per baris atau kolom secara sejajar. Jadi di dalam setiap baris atau kolom hanya ada 1 objek yang bisa ditempatkan. Linear Layout memiliki dua jenis yaitu Linear Layout Vertical (Objek per baris/kesamping) dan Linear Layout Horizontal (Objek per kolom/kebawah)

**ConstraintLayout** memungkinkan Anda membuat tata letak yang kompleks dan besar dengan hierarki tampilan datar (tidak ada grup tampilan tersarang). ConstraintLayout mirip dengan RelativeLayout yang semua tampilannya diletakkan sesuai dengan hubungan antara tampilan yang setara dan tata letak induk, tetapi lebih fleksibel dari RelativeLayout dan lebih mudah digunakan dengan Layout Editor Android Studio. Untuk menggunakan ConstraintLayout di project Anda, ikuti langkah-langkah berikut:

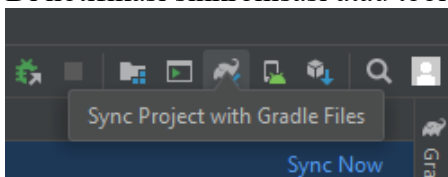
1. Pastikan Anda memiliki repositori `maven.google.com` yang dinyatakan di file `build.gradle` tingkat modul:

```
allprojects {
    repositories {
        google()
    }
}
```

2. Tambahkan library sebagai dependensi di file `build.gradle` (Module app), seperti yang ditunjukkan di contoh berikut. Perlu diperhatikan bahwa versi terbaru mungkin berbeda dengan yang ditunjukkan di contoh:

```
dependencies {
    implementation fileTree(dir: 'libs', include: ['*.jar'])
    implementation "org.jetbrains.kotlin:kotlin-stdlib-jdk7:$kotlin_version"
    implementation 'androidx.appcompat:appcompat:1.0.2'
    implementation 'androidx.core:core-ktx:1.0.2'
    implementation 'androidx.constraintlayout:constraintlayout:1.1.3'
```

3. Di notifikasi sinkronisasi atau toolbar, klik Sync Project with Gradle Files.



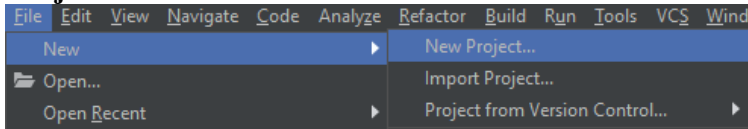
Sekarang Anda sudah siap untuk membuat tata letak menggunakan ConstraintLayout.

---

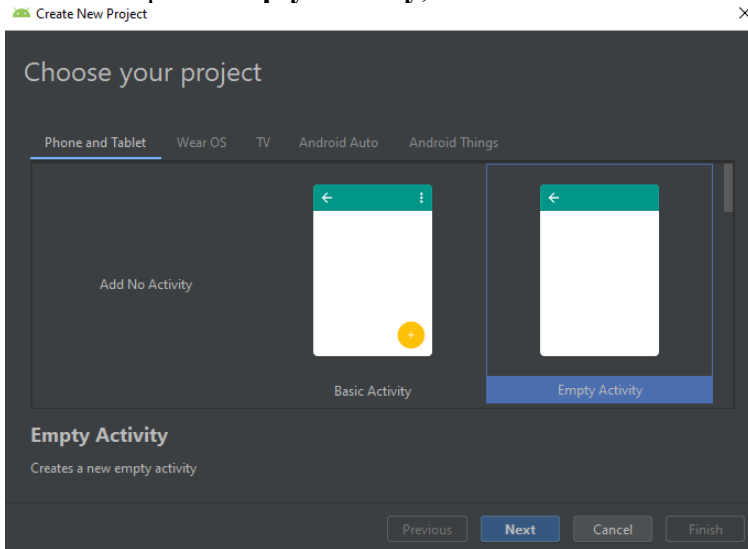
## PRAKTIK

### Praktik 1. Proyek LinearLayout

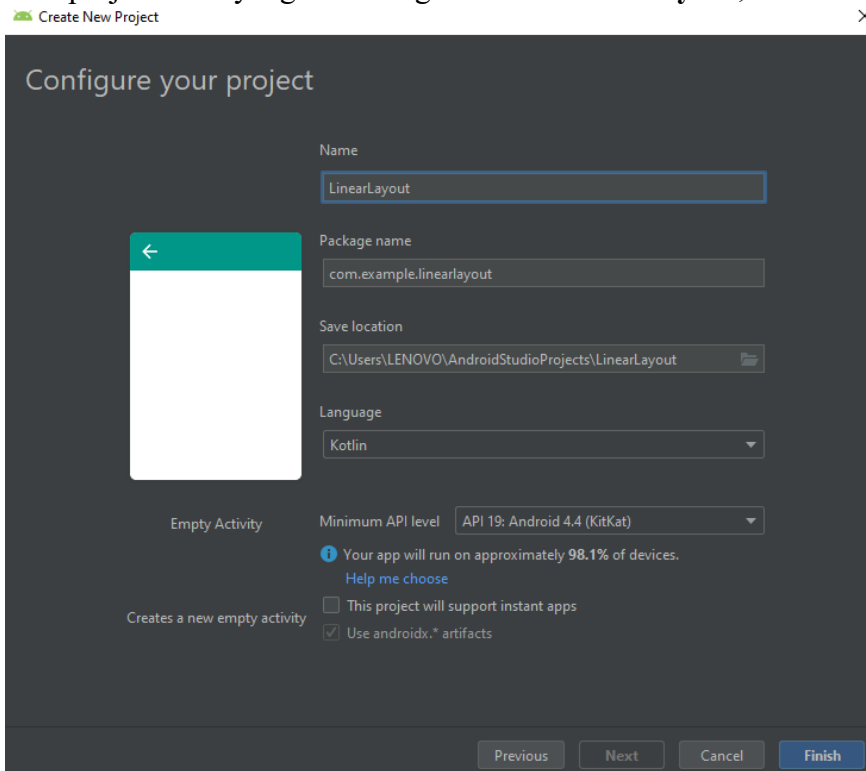
1. Buatlah project dengan nama LinearLayout dengan cara klik menu **File -> New -> New Project ...**



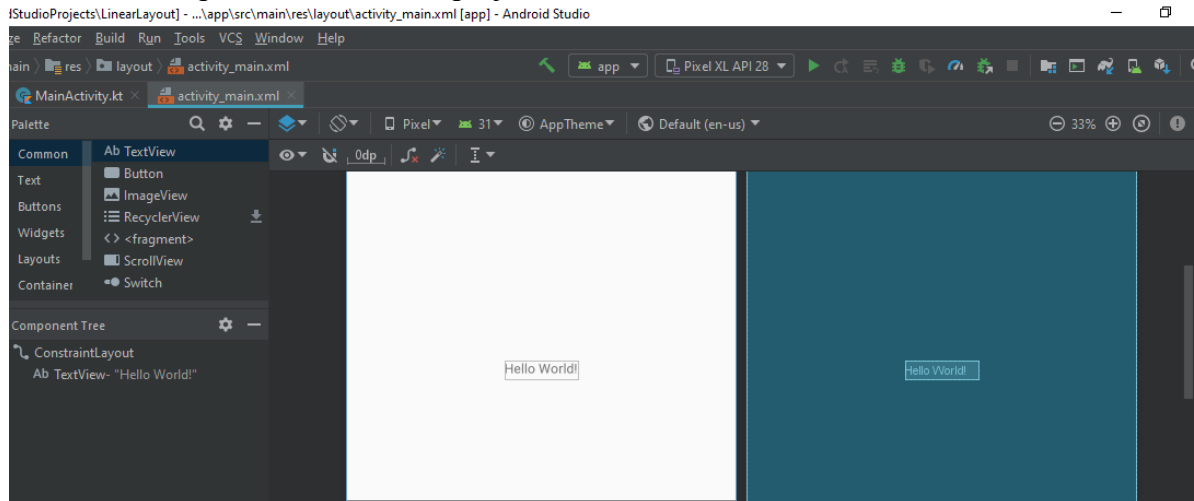
2. Kemudian pilih **Empty Activity**, lalu klik button **Next**



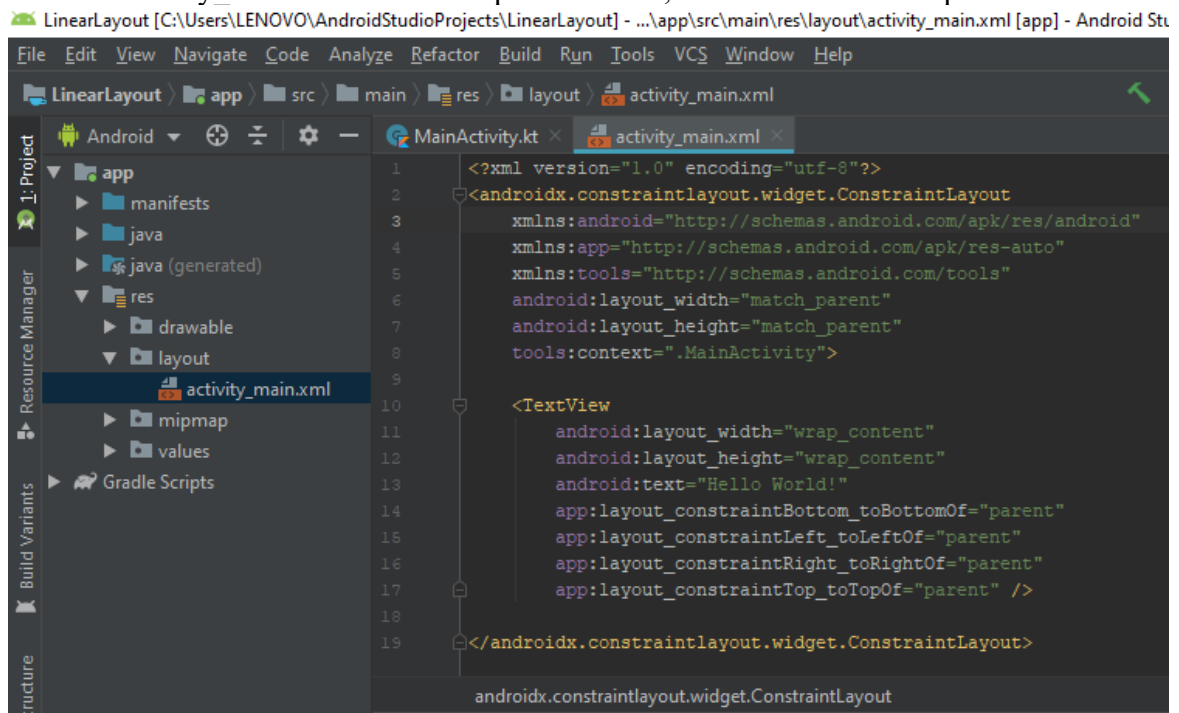
3. Beri project anda yang baru dengan nama **LinearLayout**, kemudian klik button **Finish**.



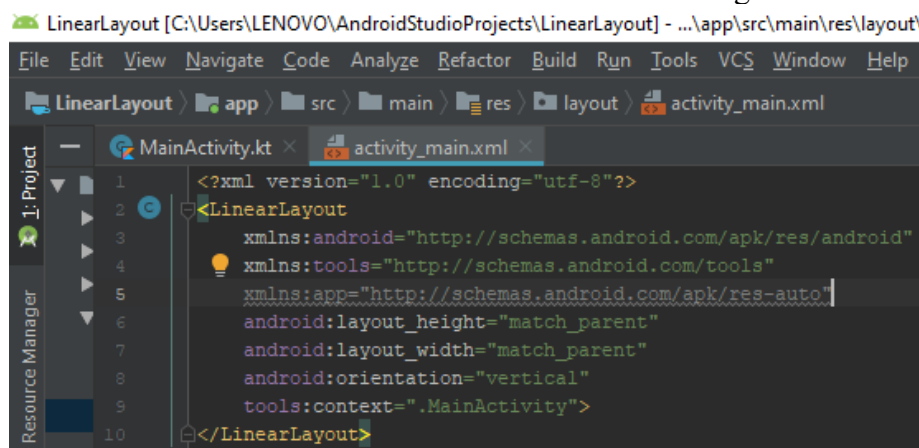
4. Langkah selanjutnya, buka file **activity\_main.xml** yang terdapat pada **app -> res -> layout**. Secara default pada saat membuat project baru maka akan muncul teks **Hello World**.



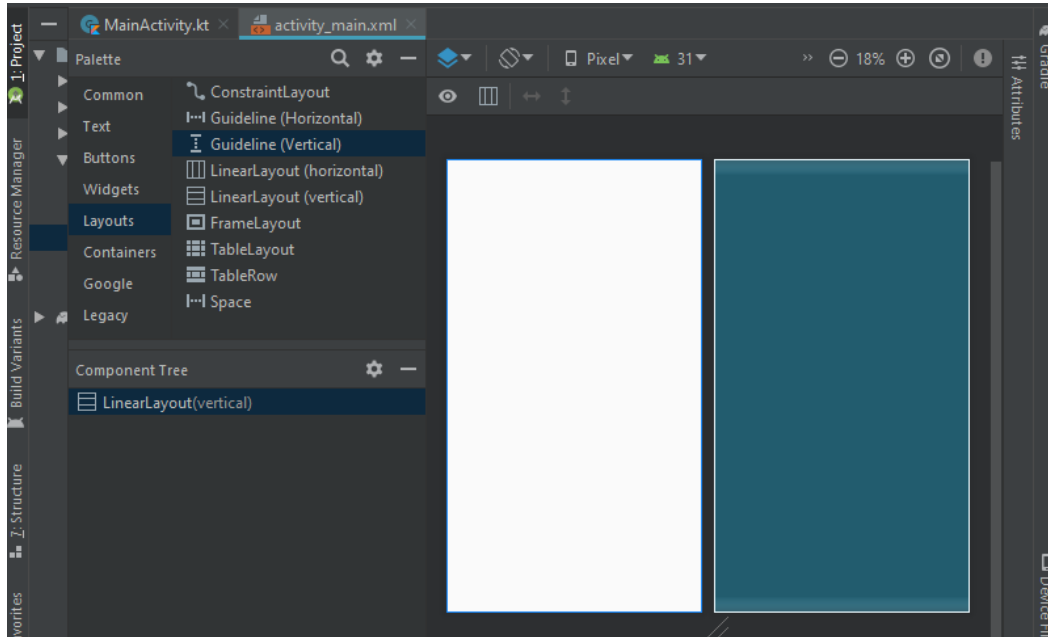
5. Buka file **activity\_main.xml** kemudian pilih tab **Text**, akan terlihat kode seperti berikut.



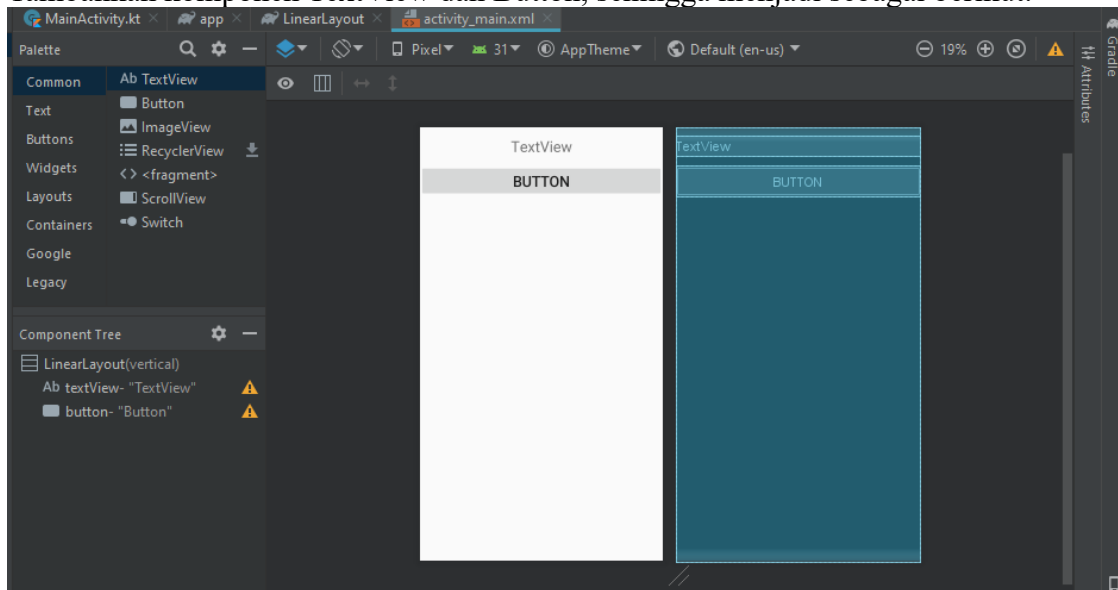
6. Standar layout pertama adalah Constraint Layout. Ubahlah menjadi Linear Layout dengan kode berikut. Tambahkan atribut **android:orientation** dengan nilai **vertical**



7. Maka hasilnya akan terlihat sebuah linear layout berhasil ditempatkan di bagian area atau canvas.



8. Tambahkan komponen TextView dan Button, sehingga menjadi sebagai berikut.

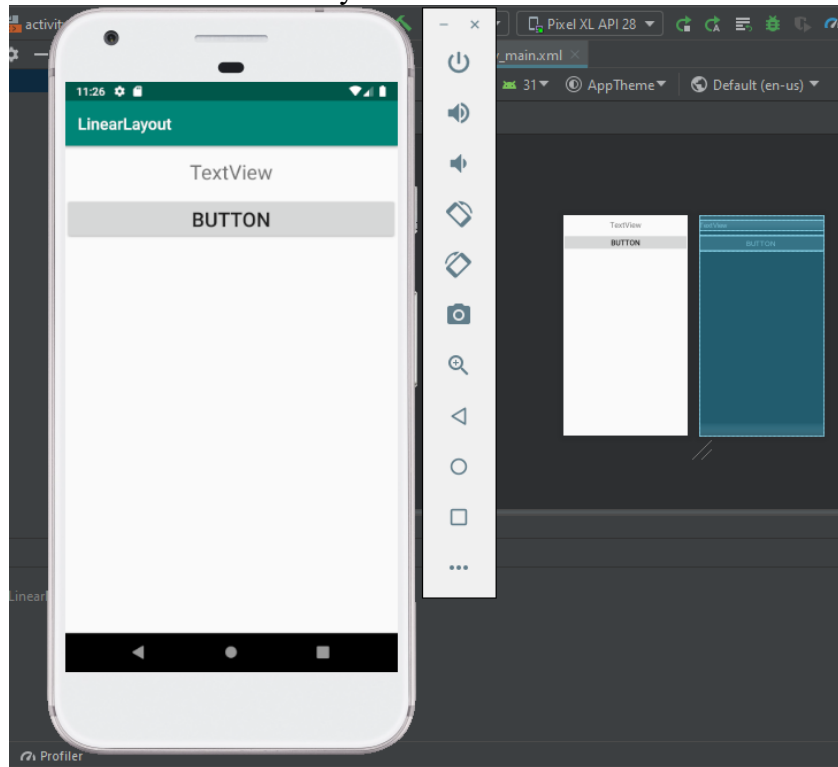


Tampilan kode TextView dan Button:

```
<TextView
    android:id="@+id/textView"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:gravity="center"
    android:text="TextView"
    android:layout_marginTop="15dp"
    android:textSize="25sp"
/>

<Button
    android:id="@+id/button"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:gravity="center"
    android:text="Button"
    android:layout_marginTop="15dp"
    android:textSize="25sp" />
```

9. Jalankan dan amati hasilnya.



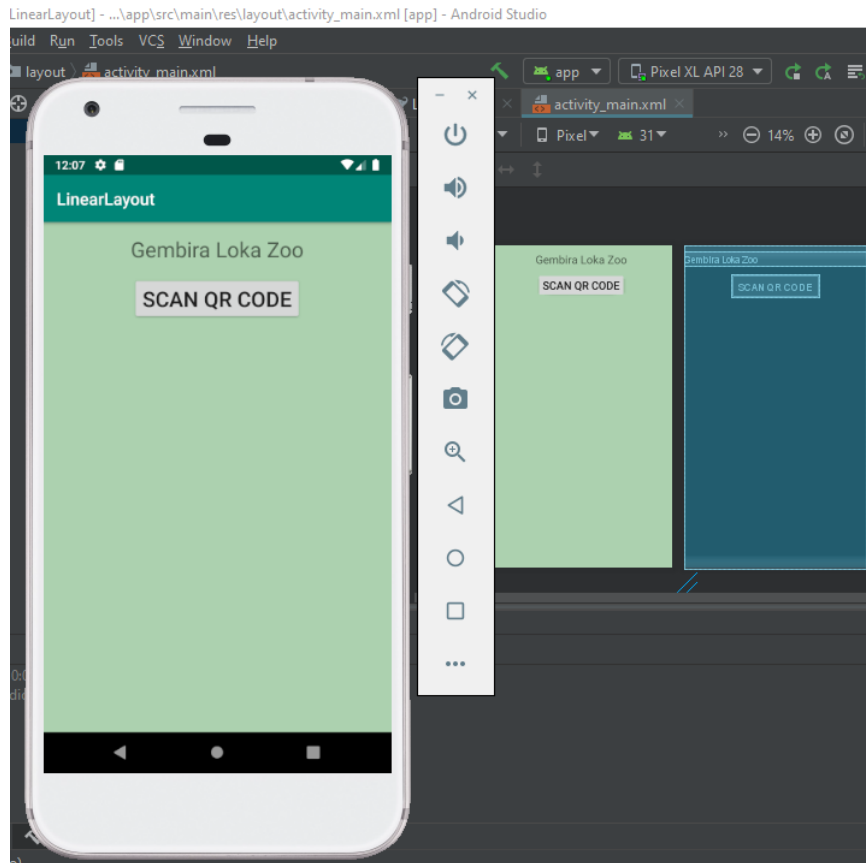
10. Ubah atribut-atribut yang ada dan jalankan lagi, amati perubahannya

```
MainActivity.kt x app x LinearLayout x activity_main.xml x
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     xmlns:tools="http://schemas.android.com/tools"
5     xmlns:app="http://schemas.android.com/apk/res-auto"
6     android:layout_height="match_parent"
7     android:layout_width="match_parent"
8     android:orientation="vertical"
9     android:gravity="center_horizontal"
10    tools:context=".MainActivity"
11    android:background="#ACD1AF">
12    <TextView
13        android:id="@+id/textView"
14        android:layout_width="match_parent"
15        android:layout_height="wrap_content"
16        android:gravity="center"
17        android:text="Gembira Loka Zoo"
18        android:layout_marginTop="15dp"
19        android:textSize="25sp"
20    />
21    <Button
22        android:id="@+id/button"
23        android:layout_width="wrap_content"
24        android:layout_height="wrap_content"
25        android:gravity="center"
26        android:text="SCAN QR CODE"
27        android:layout_marginTop="15dp"
28        android:textSize="25sp" />
29 </LinearLayout>
```

Penjelasan:

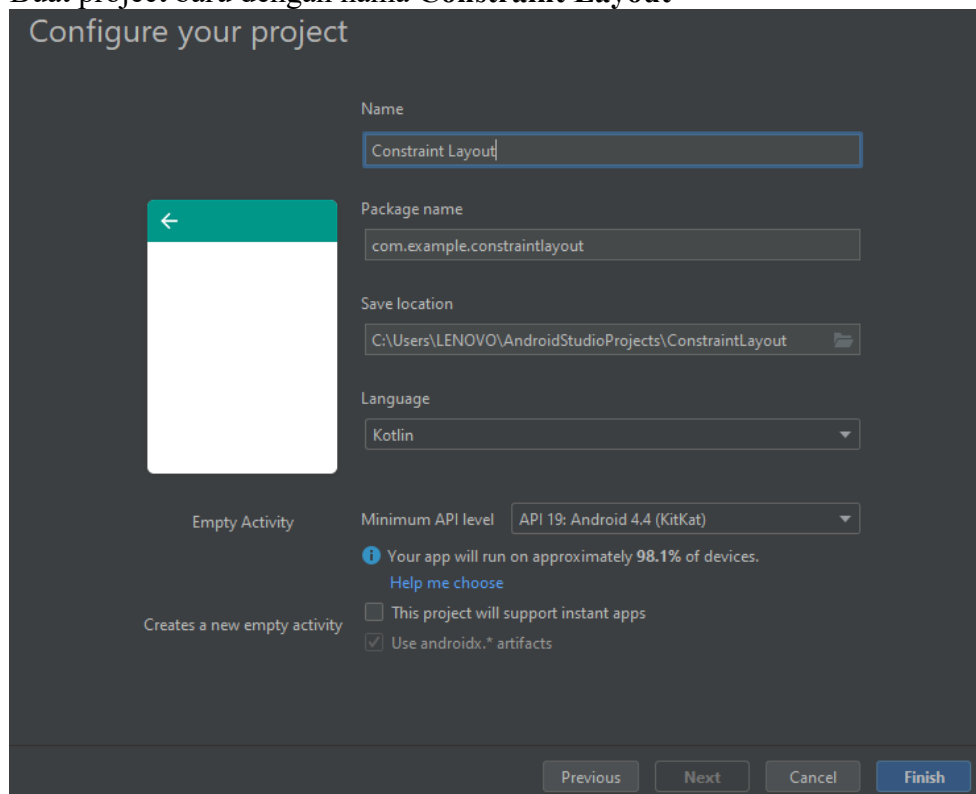
Atribut yang saya ganti/tambahkan adalah ***android:background="#ACD1AF"*** untuk mengubah warna background app menjadi hijau soft. Lalu saya ubah text di TextView dengan kode ***android:text="Gembira Loka Zoo"***. Lalu tulis ***android:layout\_width="wrap\_content"*** supaya panjang dan lebar Button mengikuti konten di dalamnya. Terakhir saya ubah text di dalam Button dengan ***android:text="SCAN QR CODE"***.

Tampilan di emulator:

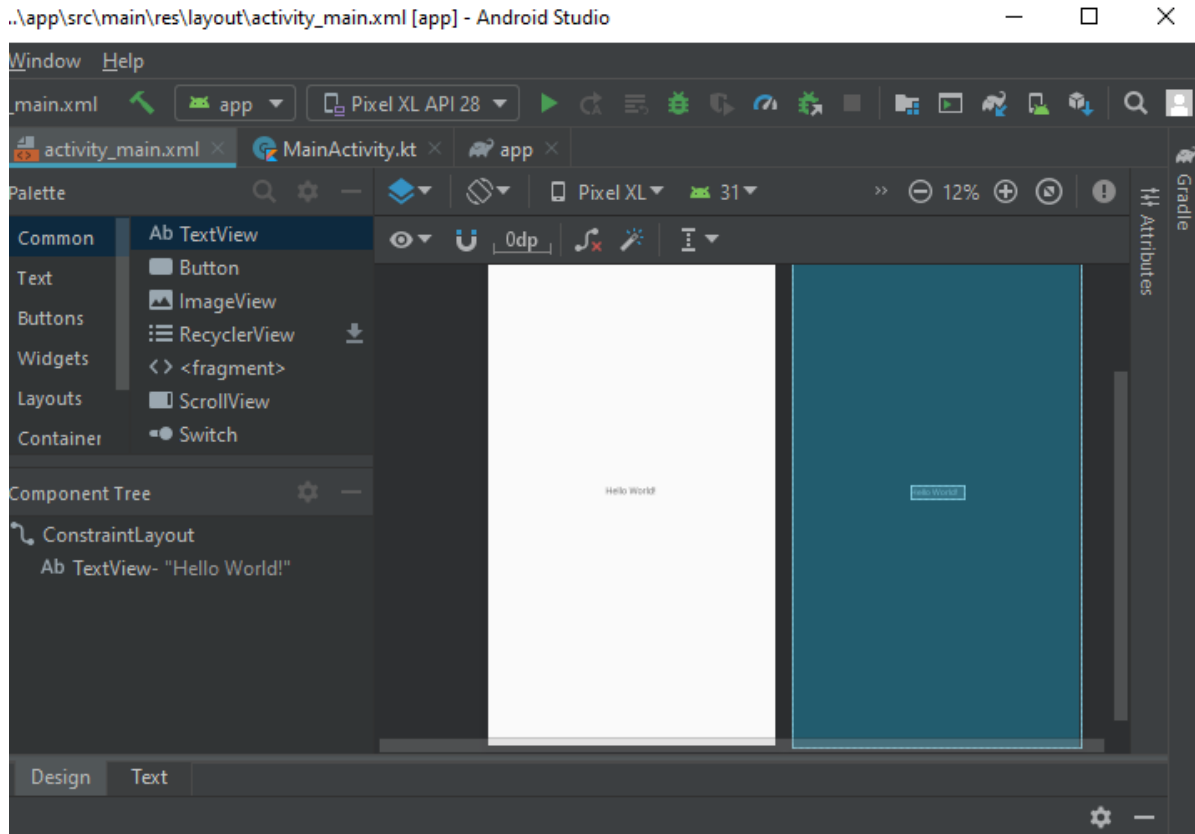


## Praktik 2. Proyek ConstraintLayout

1. Buat project baru dengan nama **Constraint Layout**



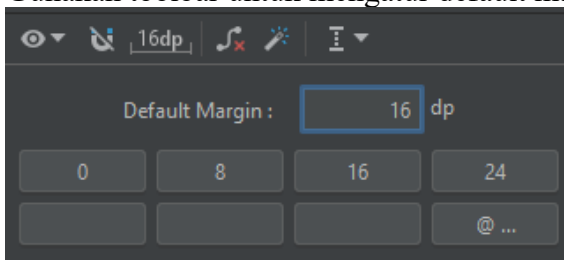
2. Buka file **activity\_main.xml** dan klik tab **Desain**



3. Anda akan menambahkan constraint (batasan) secara manual, maka koneksi otomatis kita matikan. Di toolbar, temukan Turn Off/On Autoconnect toggle button, yang ditunjukkan di bawah ini. (Jika Anda tidak dapat melihat toolbar, klik di dalam area editor desain dari Layout Editor.) Pastikan autoconnect tidak aktif.

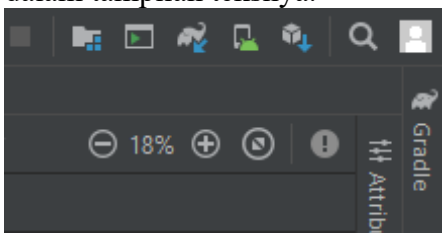


4. Gunakan toolbar untuk mengatur default margins ke 16dp.



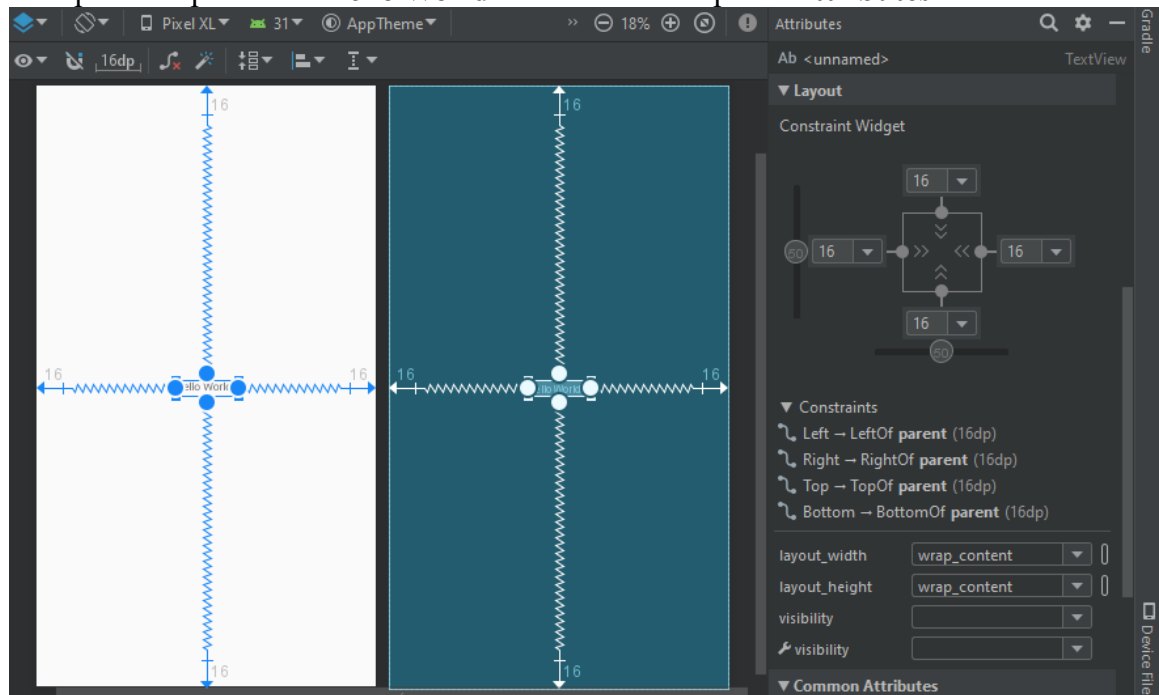
Ketika Anda mengatur margin default ke 16dp, constraint baru dibuat dengan margin ini, jadi Anda tidak perlu menambahkan margin setiap kali Anda menambahkan constraint

5. Perbesar menggunakan ikon + di sebelah kanan toolbar, hingga teks Hello World terlihat di dalam tampilan teksnya.

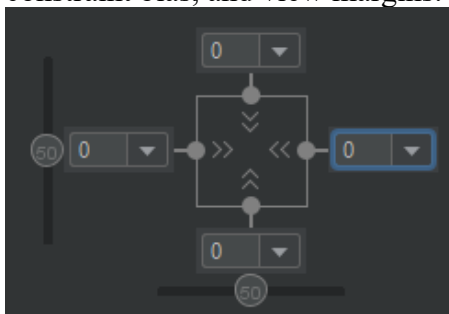




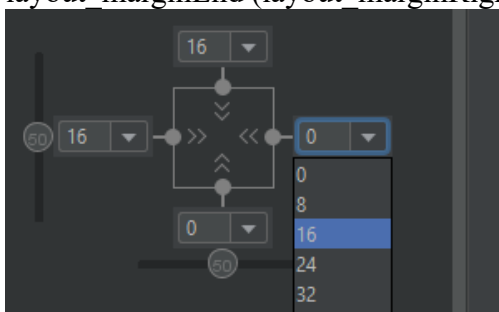
6. Klik pada tampilan teks **Hello World** untuk membuka panel **Attributes**



7. View Inspector, yang ditunjukkan pada gambar di bawah, adalah bagian dari panel Atribut. View Inspector mencakup kontrol untuk atribut layout seperti constraint, constraint types, constraint bias, and view margins.

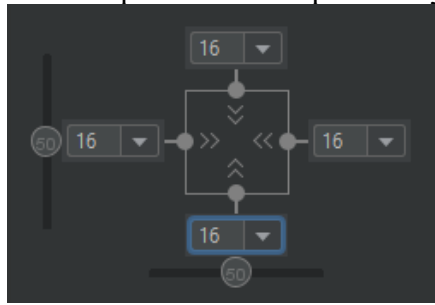


8. Constraint bias menempatkan elemen tampilan di sepanjang sumbu horizontal dan vertikal. Secara default, tampilan dipusatkan di antara dua constraint dengan bias 50%. Untuk menyesuaikan bias, Anda dapat menarik slider bias di view inspector. Menarik slider bias mengubah posisi tampilan sepanjang sumbu.
9. Tambahkan margin untuk TextView Hello World. Perhatikan bahwa dalam view inspector, margin kiri, kanan, atas, dan bawah untuk tampilan teks adalah 0. Margin default tidak ditambahkan secara otomatis, karena tampilan ini dibuat sebelum Anda mengubah margin default. Untuk margin kiri, kanan, dan atas, pilih 16dp dari menu drop-down di inspektur tampilan. Misalnya, dalam tangkapan layar berikut ini Anda menambahkan layout marginEnd (layout marginRight).

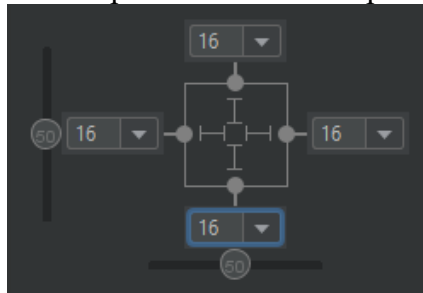


10. Sesuaikan batasan dan margin untuk TextView. Di view inspector, panah >>> di dalam kotak mewakili tipe constrain:

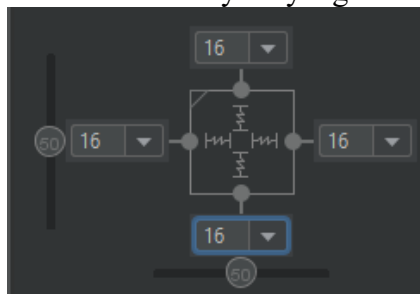
- a. >>> wrap content: Tampilan hanya selebar kontennya.



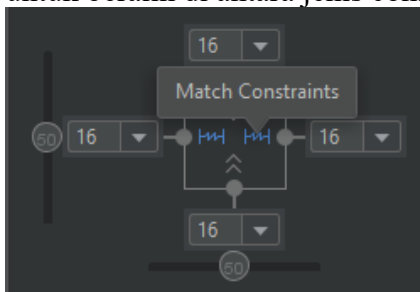
- b. | fixed: Anda dapat menentukan dimensi sebagai margin tampilan di kotak teks di sebelah panah constrain tetap.



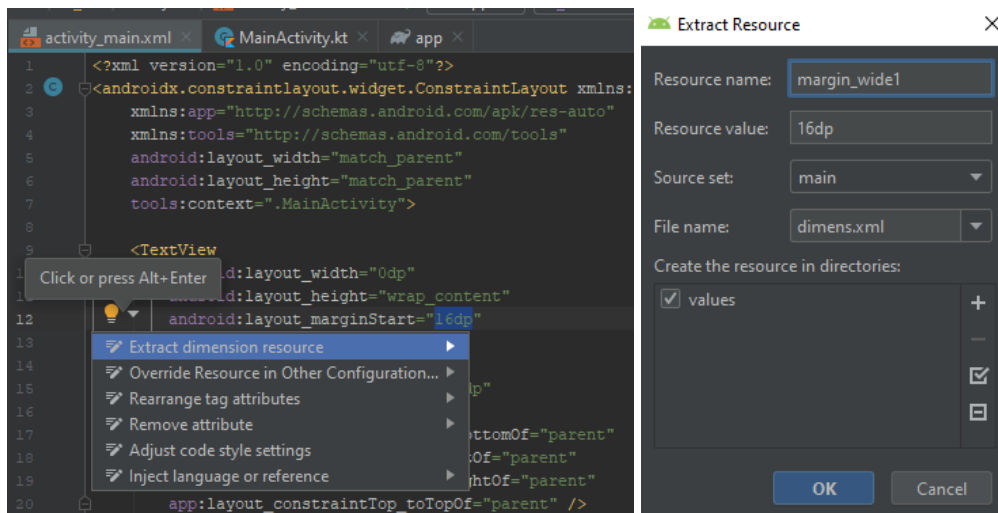
- c. match constraint: Tampilan melebar sebanyak mungkin untuk memenuhi constraint di setiap sisi, setelah memperhitungkan margin tampilan sendiri. Constrain ini sangat fleksibel, karena memungkinkan layout untuk beradaptasi dengan berbagai ukuran dan orientasi layar. Dengan membiarkan tampilan sesuai dengan constraint, Anda membutuhkan layout yang lebih sedikit untuk aplikasi yang Anda buat.



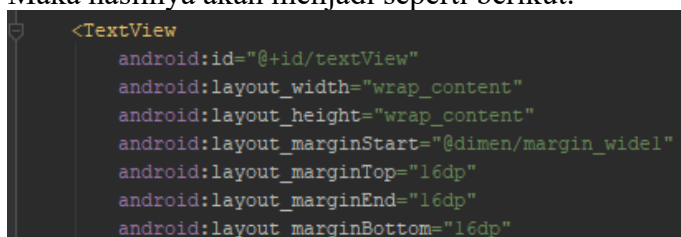
11. Di view inspector, ubah constraint kiri dan kanan ke Match Constraints. (Klik simbol panah untuk beralih di antara jenis constraint.)



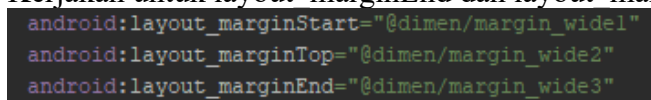
12. Pindah ke tab Text, Ekstrak resource dimensi untuk layout\_marginStart, dan atur nama Resource ke margin\_wide1. (blok pada bagian isian dari layout\_marginStart, pilih extract dimensions resource).  
(screenshot di halaman selanjutnya)



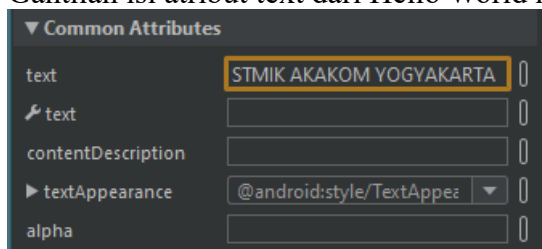
Maka hasilnya akan menjadi seperti berikut.



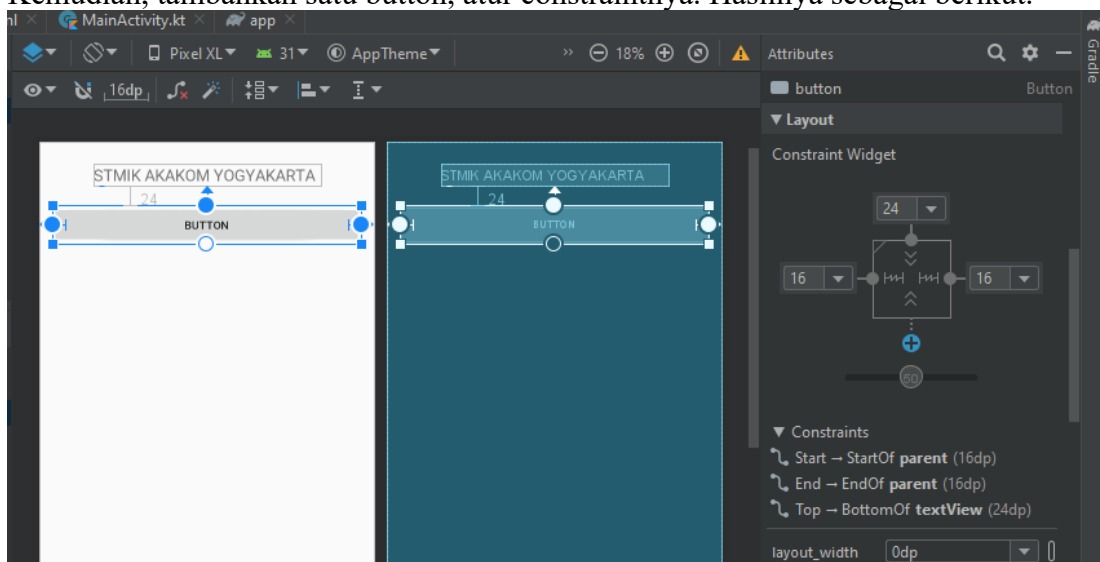
13. Kerjakan untuk layout marginEnd dan layout marginTop. Hasilnya adalah sebagai berikut



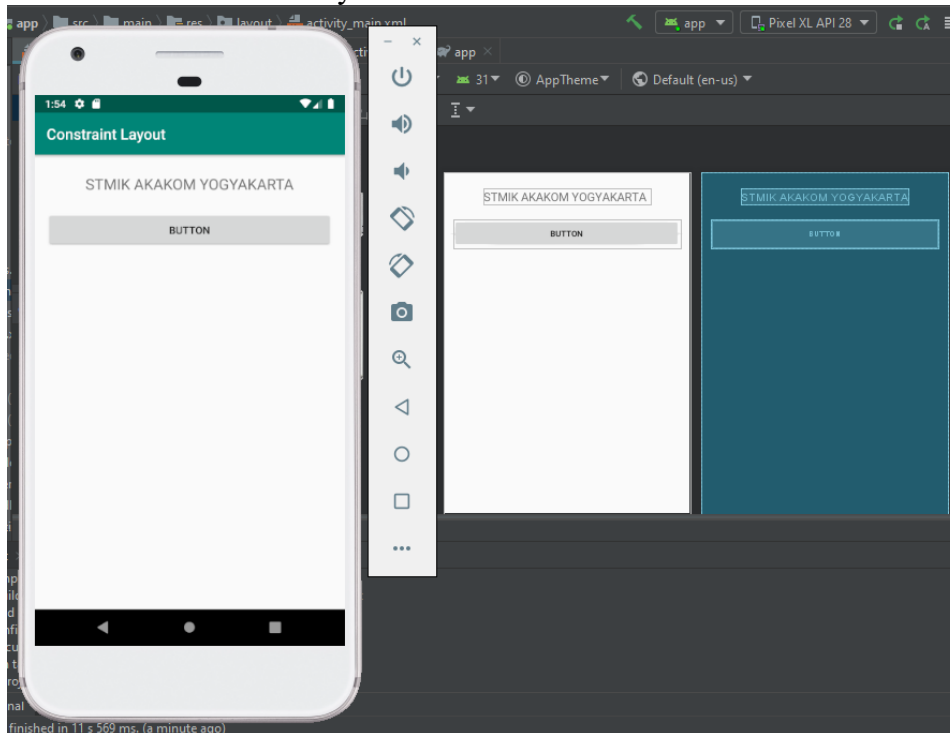
14. Gantilah isi atribut text dari Hello World menjadi STMIK AKAKOM YOGYAKARTA.



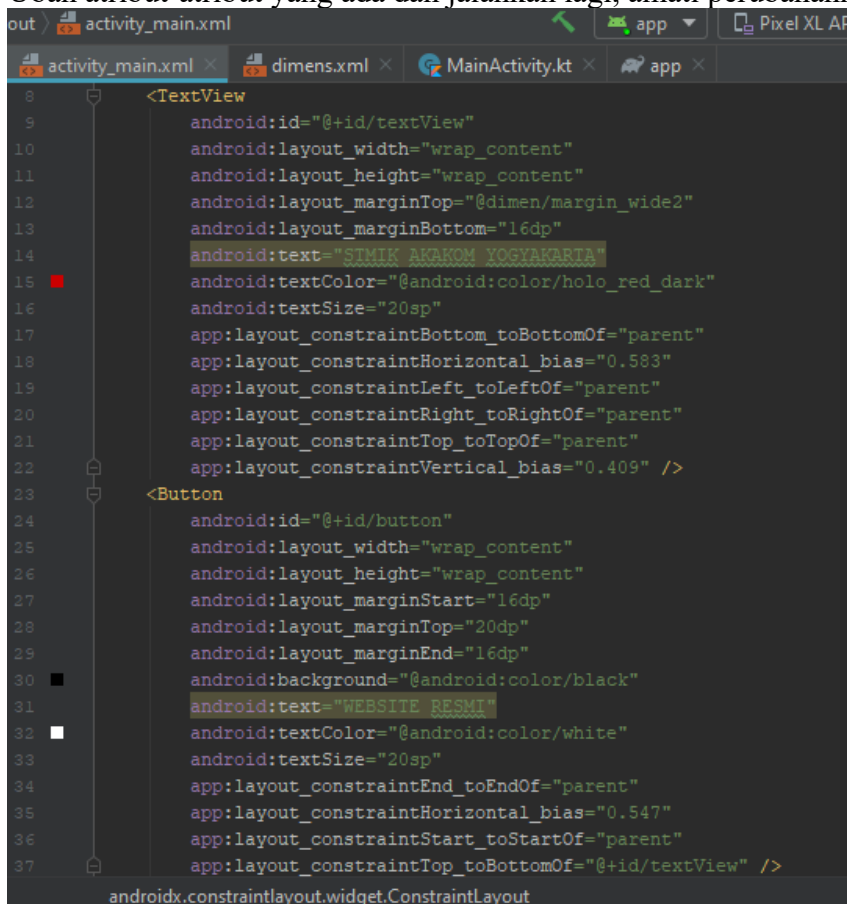
15. Kemudian, tambahkan satu button, atur constraintnya. Hasilnya sebagai berikut.



## 16. Jalankan dan amati hasilnya



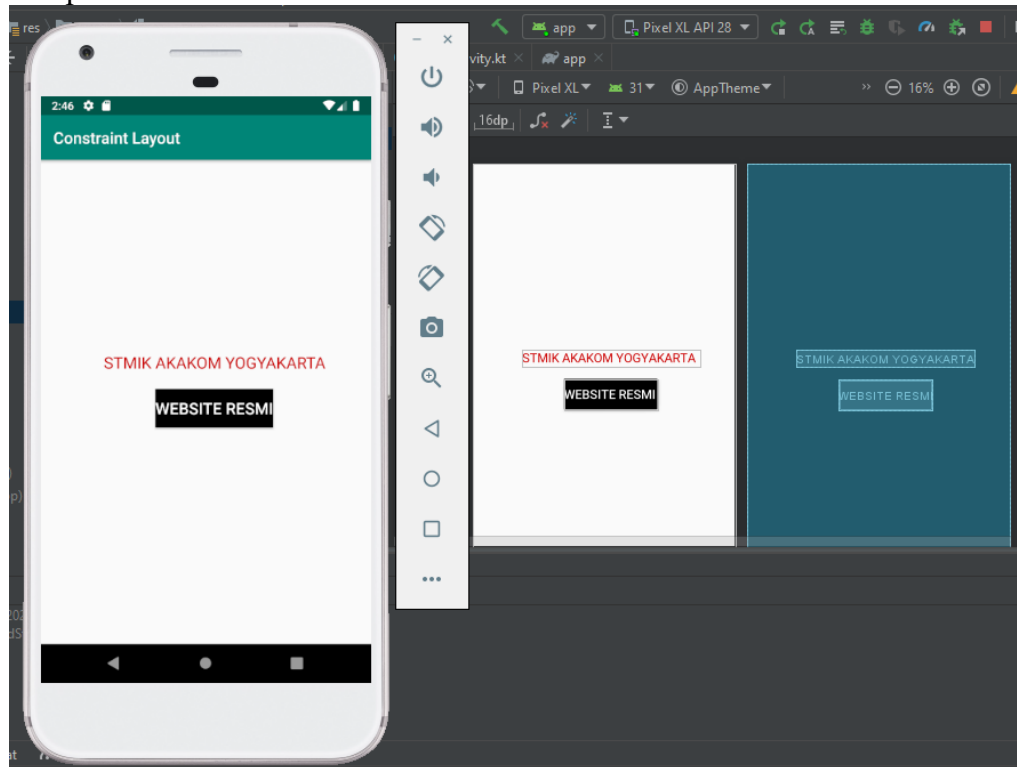
## 17. Ubah atribut-atribut yang ada dan jalankan lagi, amati perubahannya.



### Penjelasan:

Saya mengubah textview menjadi merah, kemudian mengubah text button menjadi “WEBSITE RESMI” berwarna putih dengan background hitam. Setelah itu saya pindah letak komponen ke tengah layout.

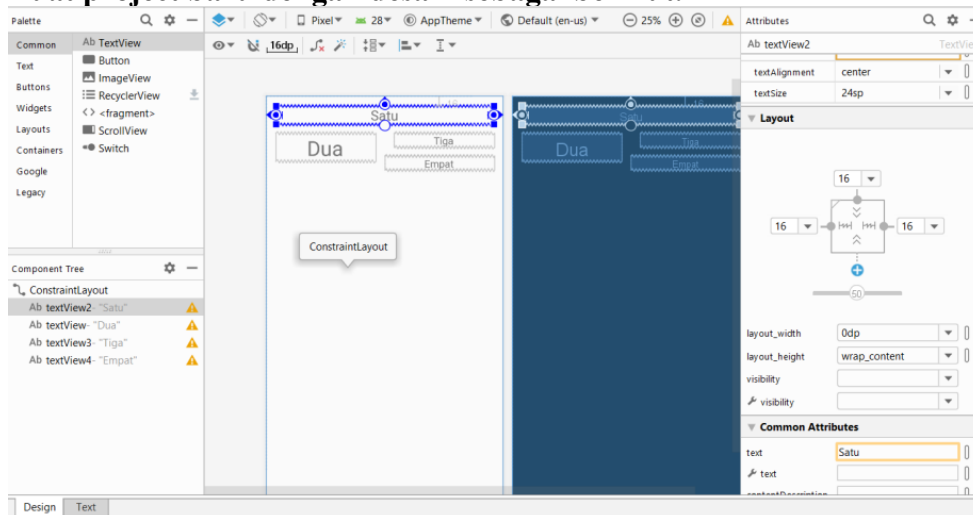
Tampilan di emulator:



## LATIHAN

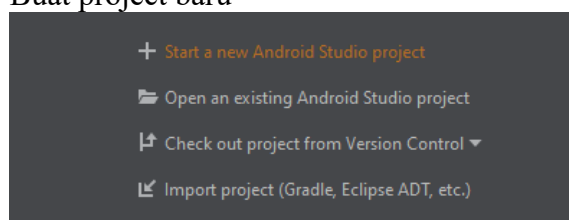
### Latihan 1:

**Buat project baru dengan desain sebagai berikut.**

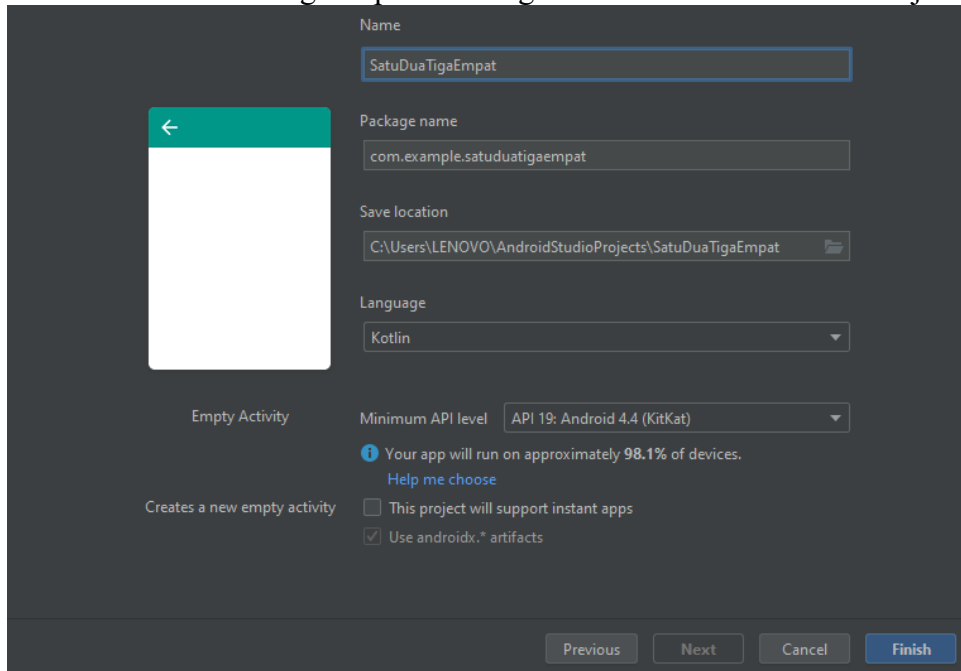


Langkah-langkah:

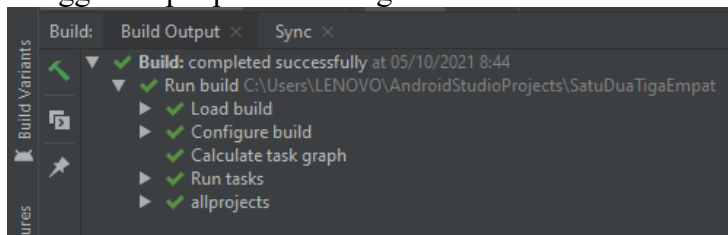
1. Buat project baru



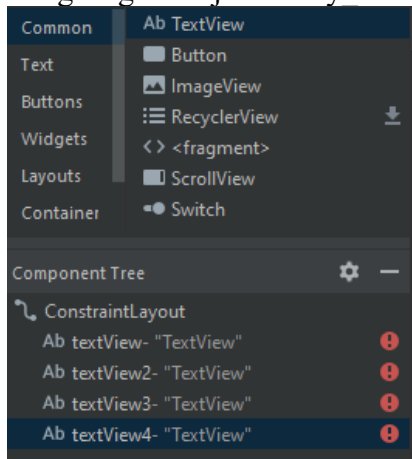
2. Beri nama “SatuDuaTigaEmpat”. Konfigurasi lain dibiarkan default saja. Klik Finish



3. Tunggu sampai proses build gradle selesai



4. Langsung menuju activity main.xml. Lalu tambahkan 4 komponen TextView seperti berikut



5. Kemudian kita setting supaya text align center, lalu ganti nama setiap textview menjadi Satu, Dua, Tiga, Empat. Selanjutnya kita tentukan besar text.

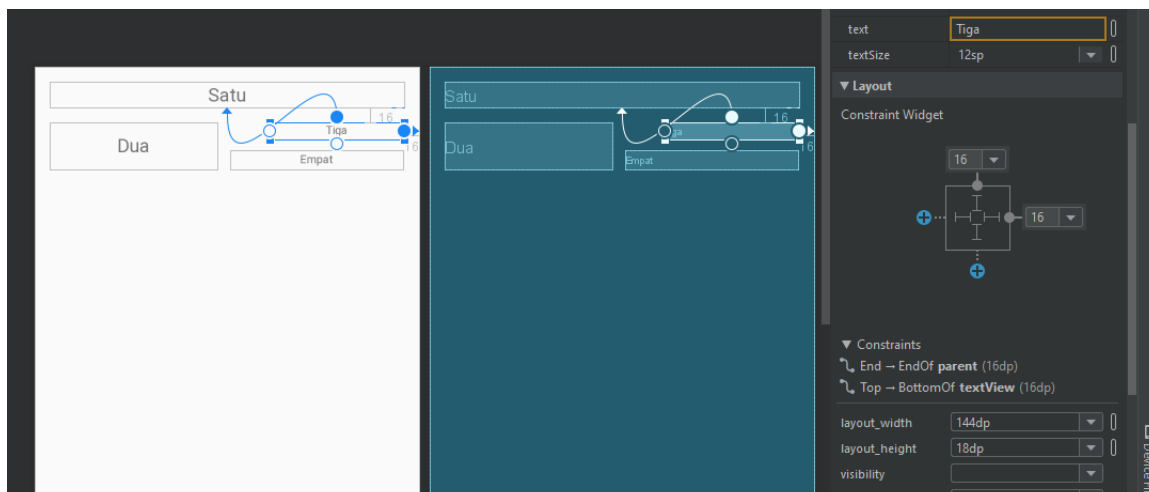
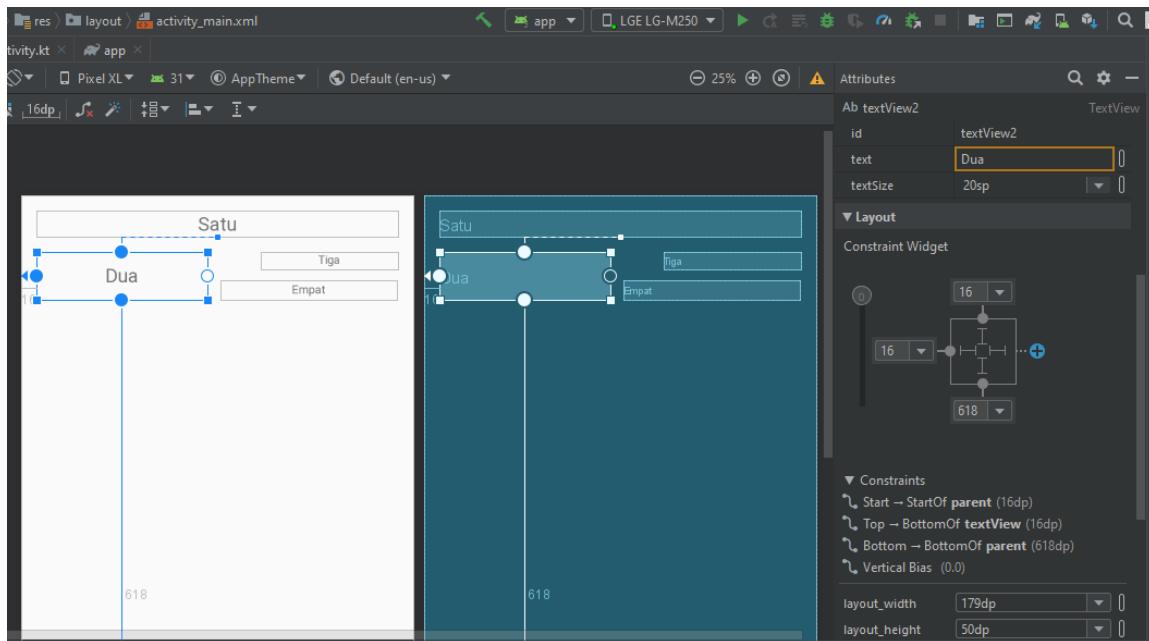
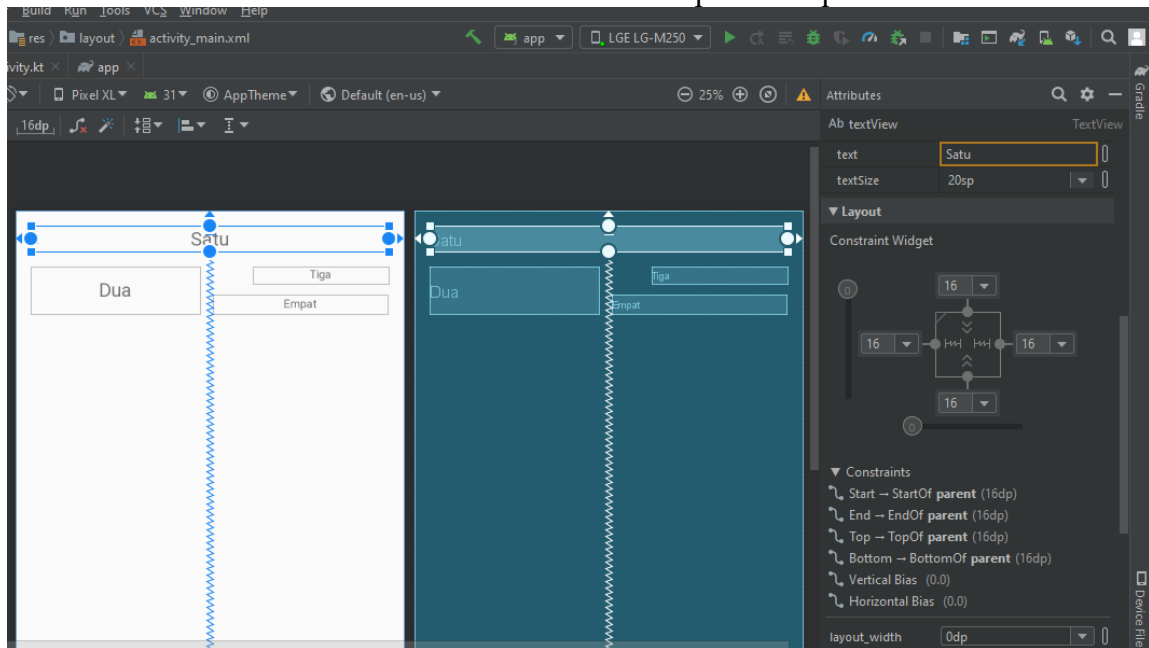
```
android:gravity="center"
android:text="Satu"
android:textSize="20sp"
```

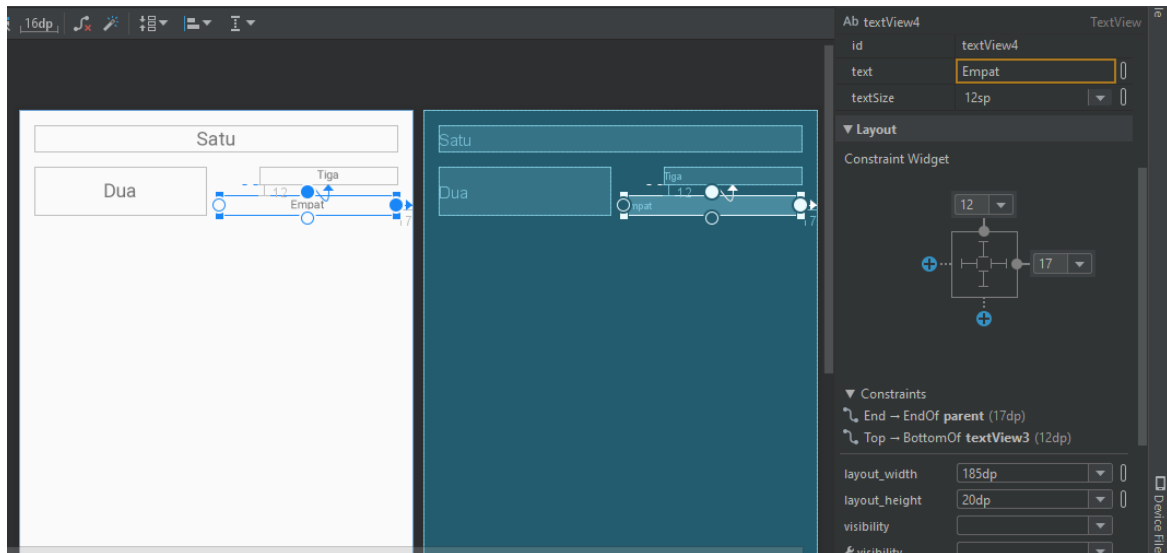
```
android:gravity="center"
android:text="Dua"
android:textSize="20sp"
```

```
android:gravity="center"
android:text="Tiga"
android:textSize="12sp"
```

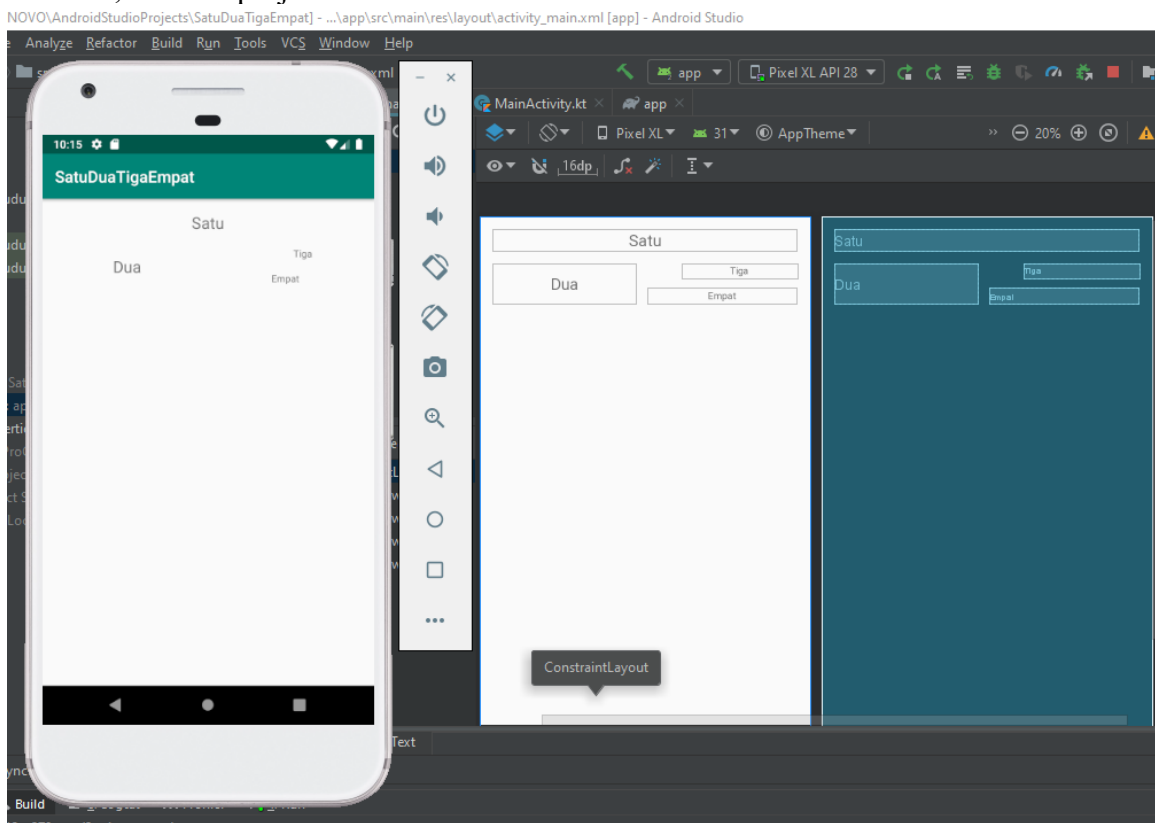
```
android:gravity="center"
android:text="Empat"
android:textSize="12sp"
```

6. Kemudian kita beri constraint vertikal dan horizontal pada setiap atribut textView





7. Terakhir, kita run project di emulator virtual Pixel XL API 28

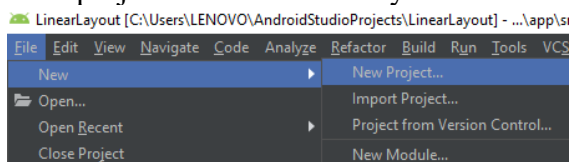


## Latihan 2:

Buat project baru dengan menggunakan Linear Layout dengan minimal 3 komponen (TextView/Button) yang ditambahkan dan eksplorasilah atribut-atribut yang ada.

Langkah-langkah:

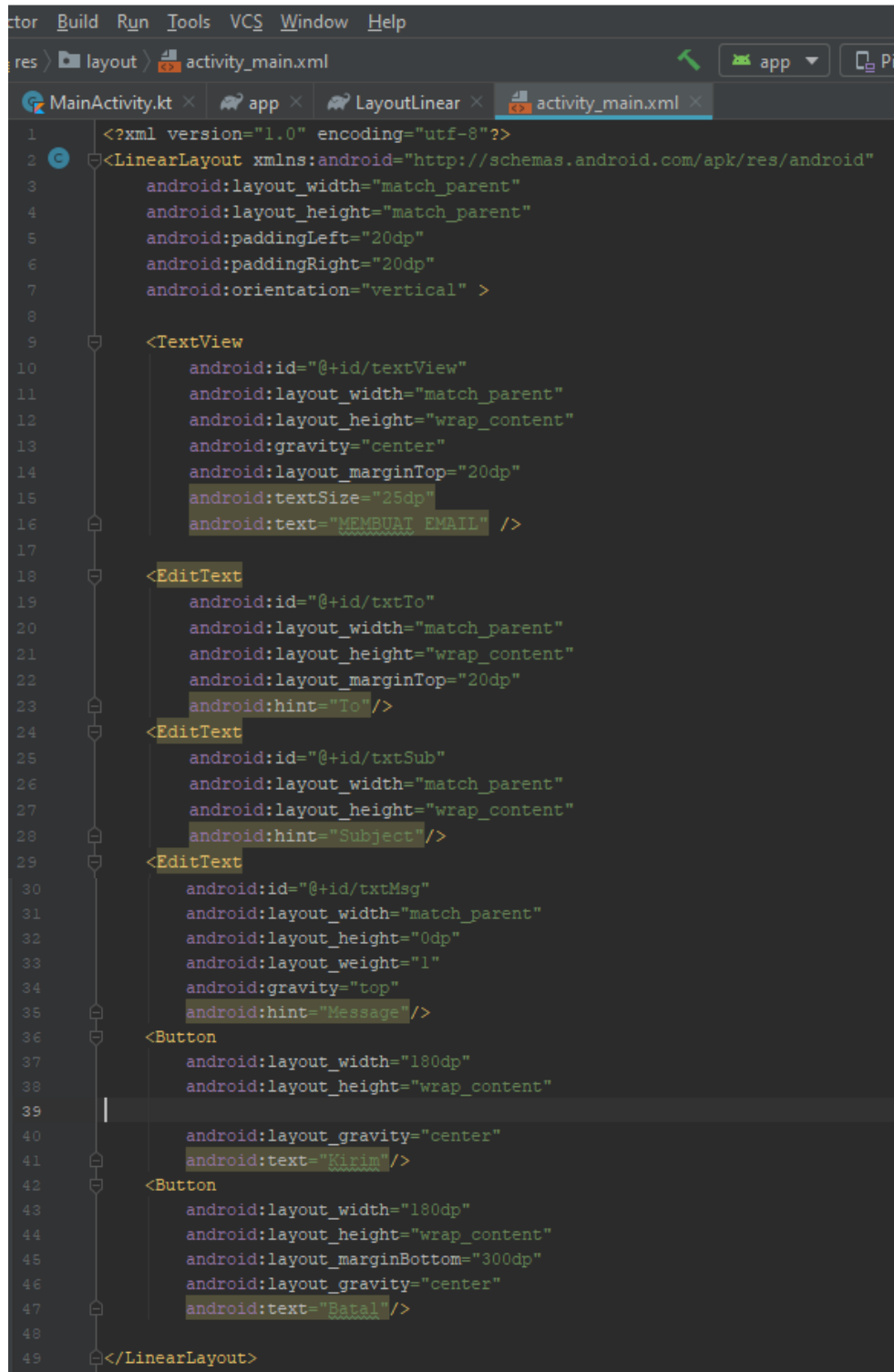
1. Buat project baru bernama Layout Linear





2. Jika project sudah selesai build gradle, langsung saja ke TextView lalu ke Text. Kita edit kode nya menjadi seperti ini

objects\LayoutLinear] - ...app\src\main\res\layout\activity\_main.xml [app] - Android Studio



```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     android:layout_width="match_parent"
4     android:layout_height="match_parent"
5     android:paddingLeft="20dp"
6     android:paddingRight="20dp"
7     android:orientation="vertical" >
8
9     <TextView
10         android:id="@+id/textView"
11         android:layout_width="match_parent"
12         android:layout_height="wrap_content"
13         android:gravity="center"
14         android:layout_marginTop="20dp"
15         android:textSize="25dp"
16         android:text="MEMBUAT EMAIL" />
17
18     <EditText
19         android:id="@+id/txtTo"
20         android:layout_width="match_parent"
21         android:layout_height="wrap_content"
22         android:layout_marginTop="20dp"
23         android:hint="To"/>
24
25     <EditText
26         android:id="@+id/txtSub"
27         android:layout_width="match_parent"
28         android:layout_height="wrap_content"
29         android:hint="Subject"/>
30
31     <EditText
32         android:id="@+id/txtMsg"
33         android:layout_width="match_parent"
34         android:layout_height="0dp"
35         android:layout_weight="1"
36         android:gravity="top"
37         android:hint="Message"/>
38
39     <Button
40         android:layout_width="180dp"
41         android:layout_height="wrap_content"
42         android:layout_gravity="center"
43         android:text="Kirim"/>
44
45     <Button
46         android:layout_width="180dp"
47         android:layout_height="wrap_content"
48         android:layout_marginBottom="300dp"
49         android:layout_gravity="center"
50         android:text="Batal"/>
51 </LinearLayout>
```

Penjelasan:

Pertama ada textview yang kita gunakan untuk memberi judul “MEMBUAT EMAIL”.

Kedua ada EditText “To” yang sebagai tempat memasukkan alamat email yang dituju.

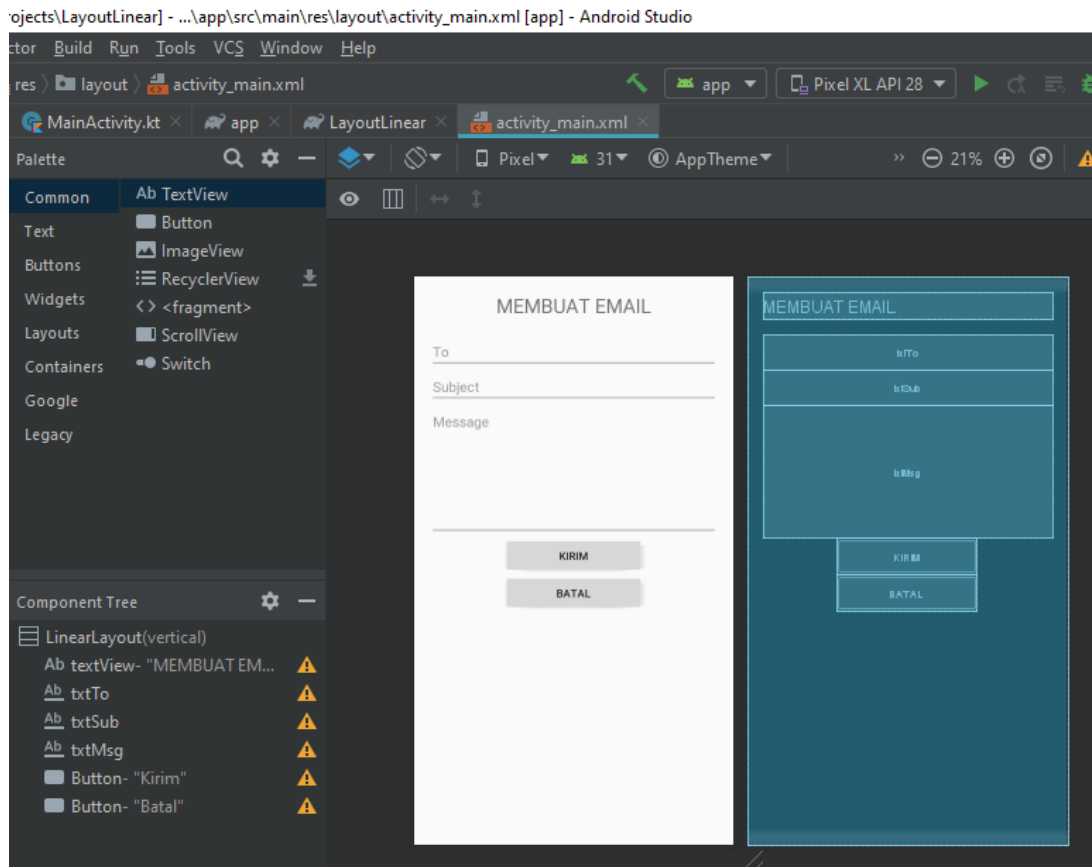
Ketiga ada EditText “Subject” sebagai subjek email

Keempat ada EditText “Message” sebagai pesan yang akan dikirim

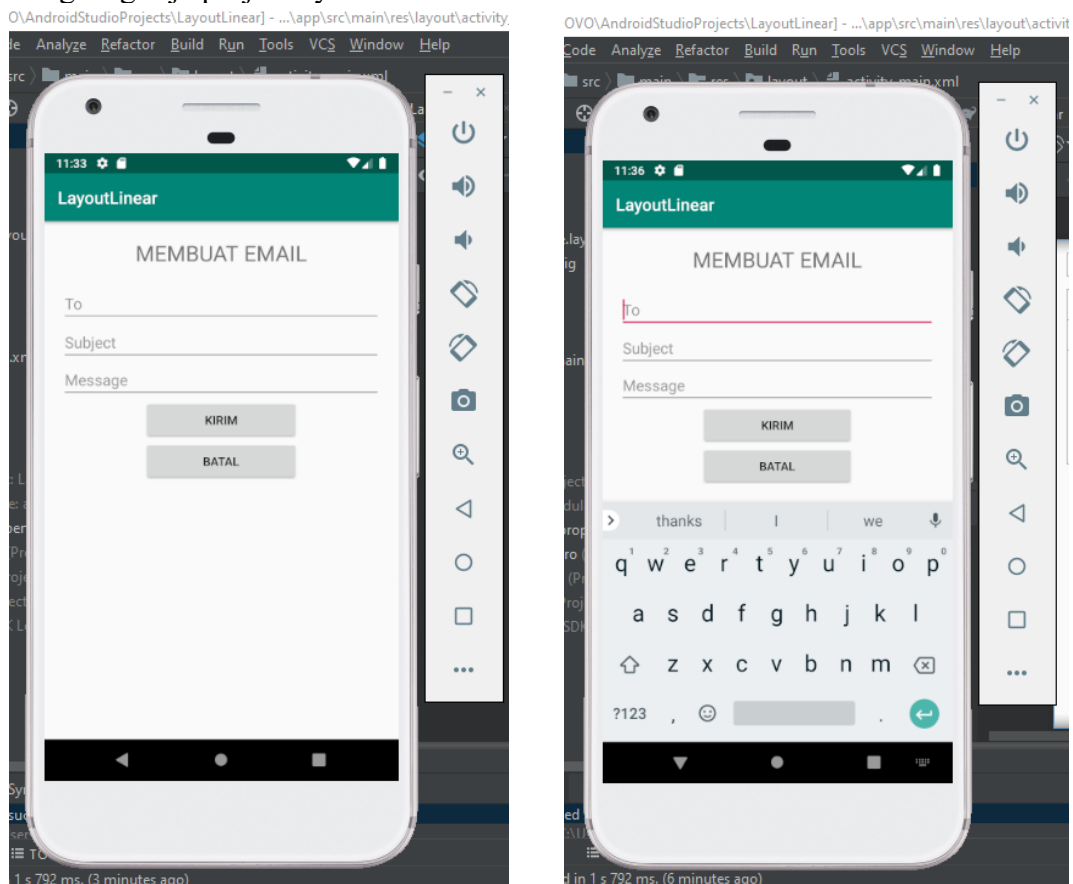
Kelima ada button untuk tombol “Kirim” untuk mengirim pesan

Terakhir ada button lagi untuk tombol “Batal” untuk membatalkan pesan

### 3. Tampilan di tab Design



### 4. Langsung saja project nya kita run di emulator virtual Pixel XL API 28

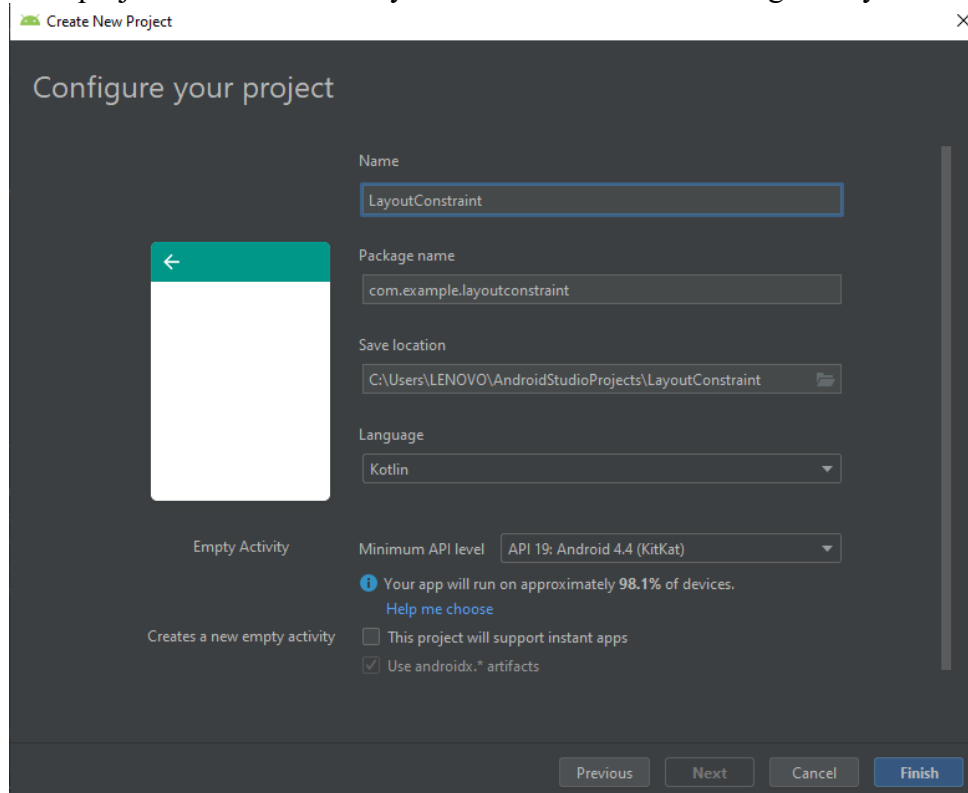


### Latihan 3:

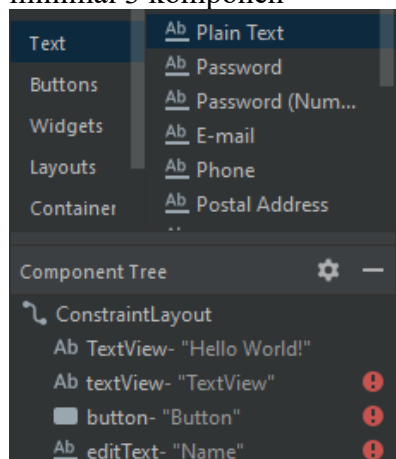
Buat project baru dengan menggunakan Constrain Layout dengan minimal 3 komponen (TextView/Button) yang ditambahkan dan eksplorasilah atribut-atribut yang ada.

Langkah-langkah:

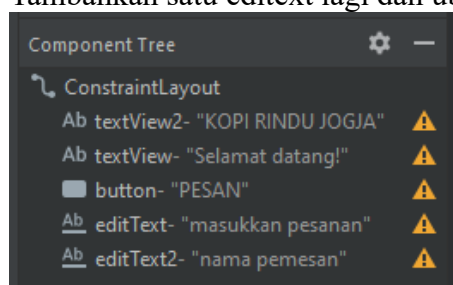
1. Buat project baru bernama LayoutConstraint. Berikut konfigurasinya



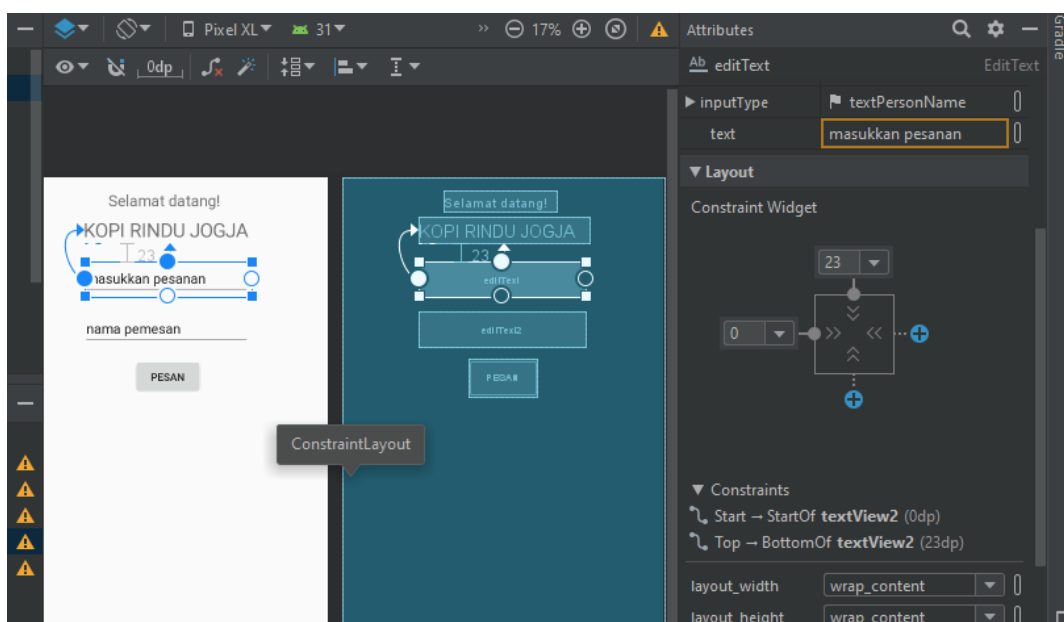
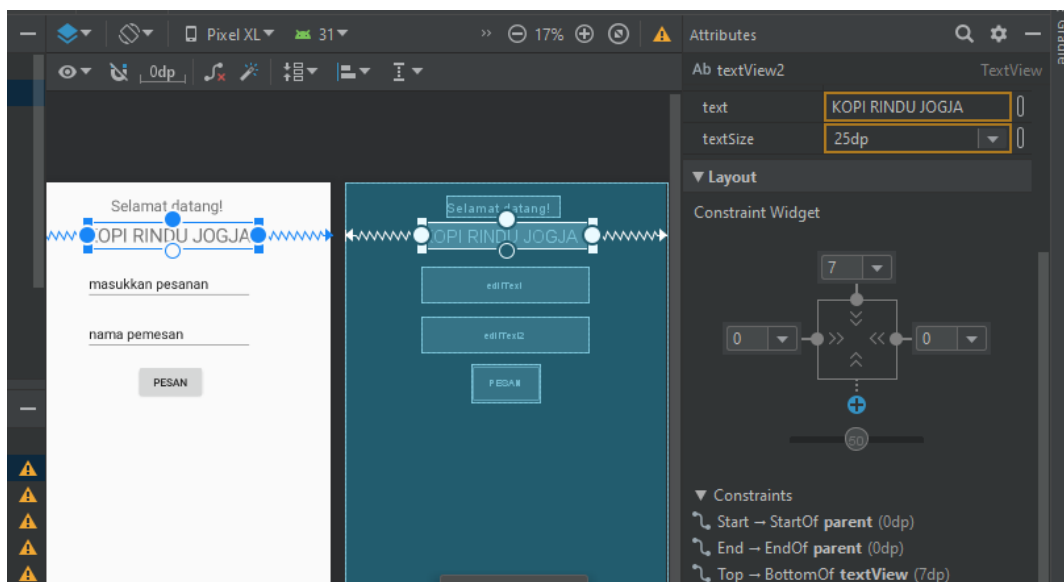
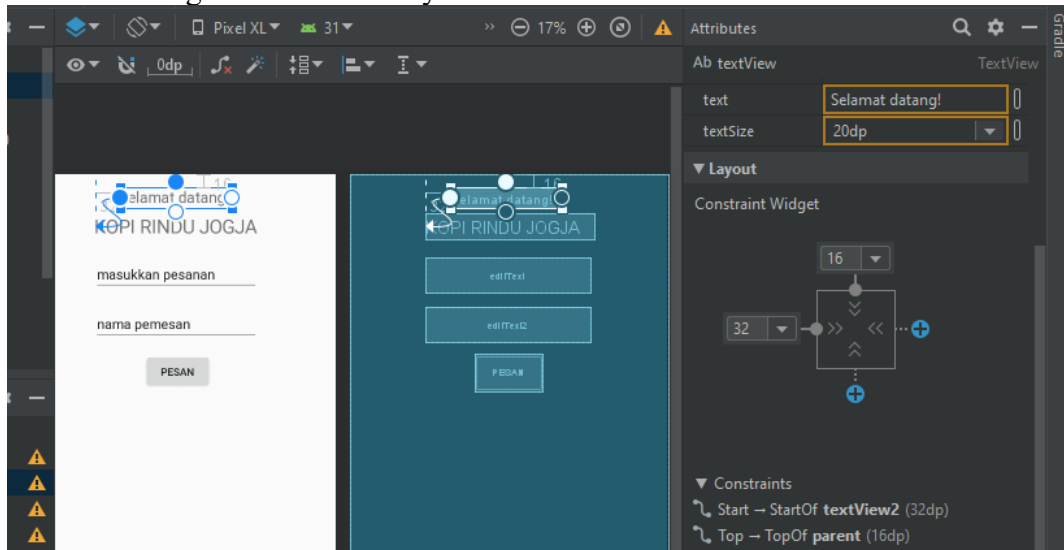
2. Tunggu hingga build gradle selesai, lalu langsung masuk ke activity\_main.xml. tambahkan minimal 3 komponen

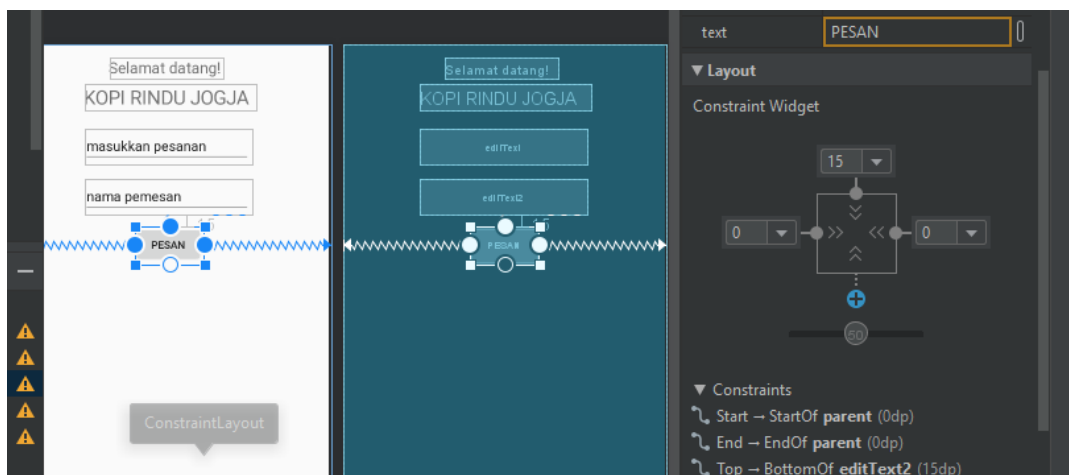
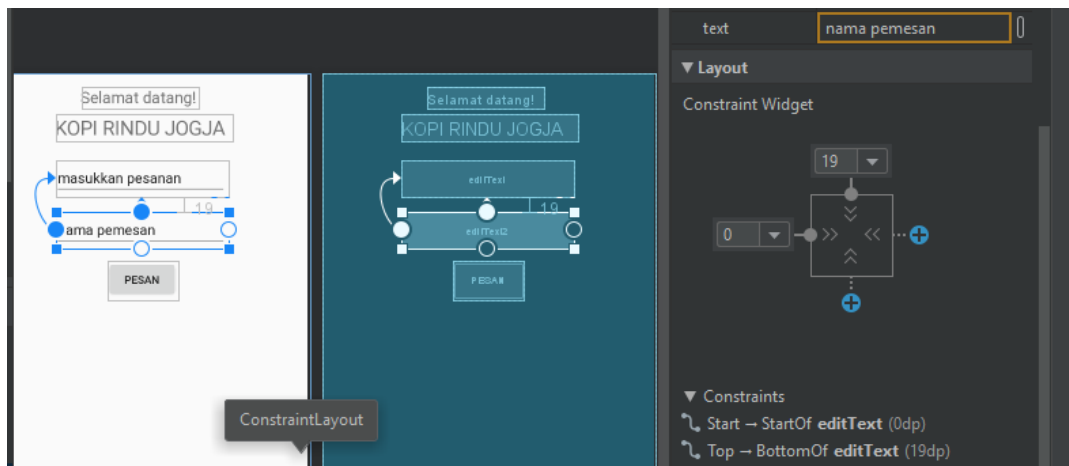


3. Tambahkan satu editext lagi dan ubah nama semua komponen

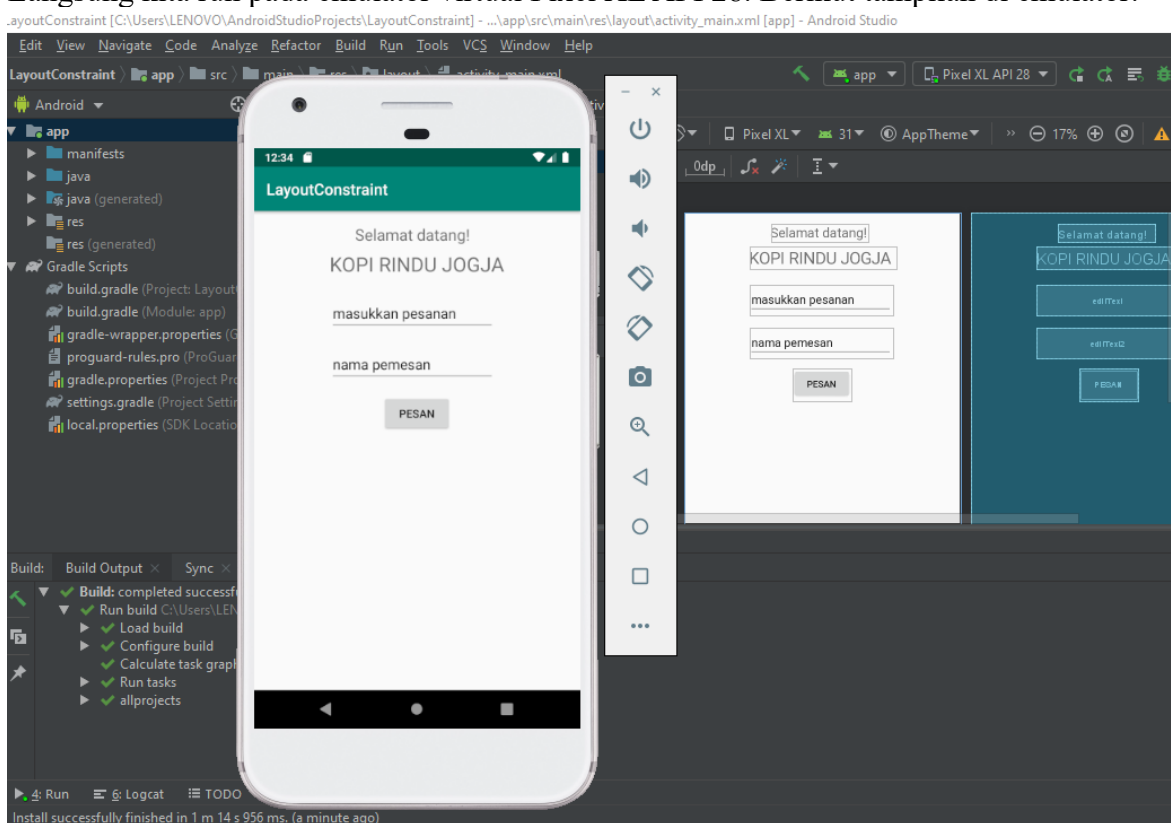


4. Kemudian kita beri constraint vertikal dan horizontal pada setiap atribut/komponen. Seperti ini hasil konfigurasi constraint nya:





5. Langsung kita run pada emulator virtual Pixel XL API 28. Berikut tampilan di emulator:

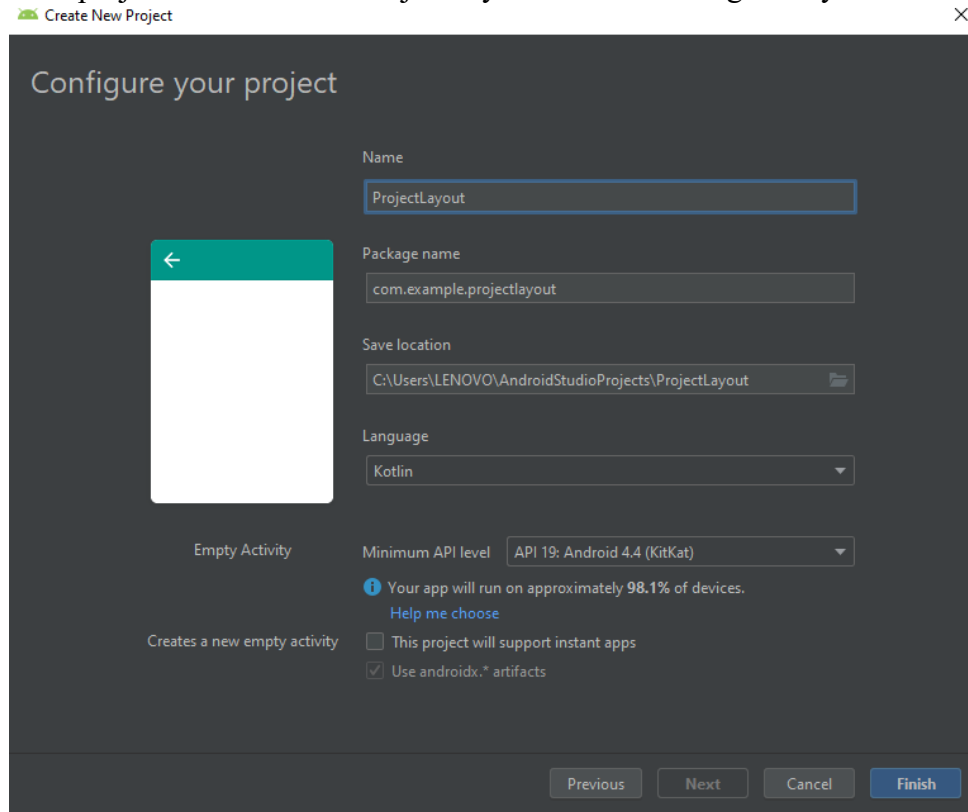


## TUGAS

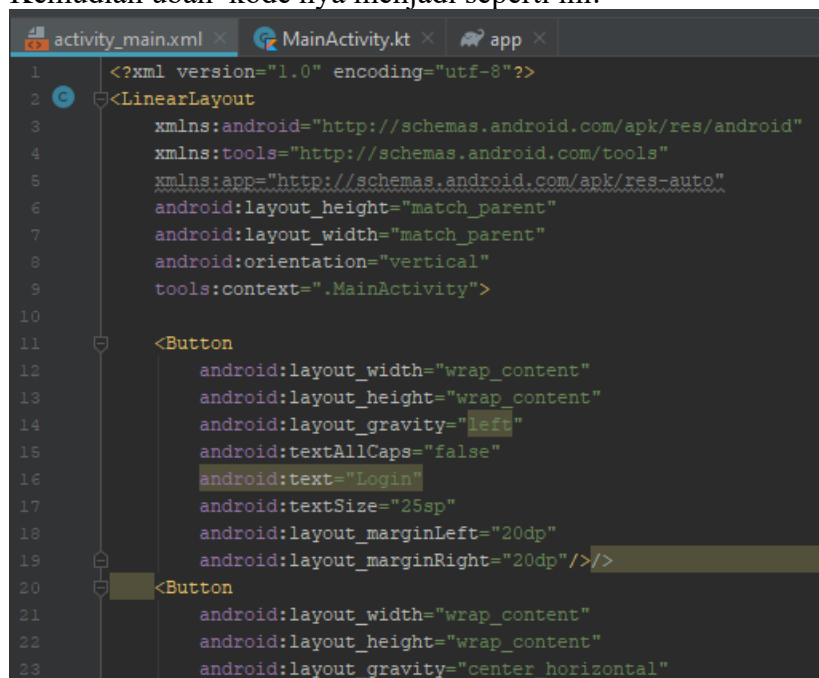
**Buat Project pada perangkat komputer anda untuk mengimplementasikan layout dengan berbagai bentuk tampilan.**

Langkah-langkah:

1. Buat project baru bernama ProjectLayout. Berikut konfigurasinya:



2. Jika proses gradle sudah selesai, langsung saja ke activity\_main.xml, lalu ke tab Text. Kemudian ubah kode nya menjadi seperti ini:



```

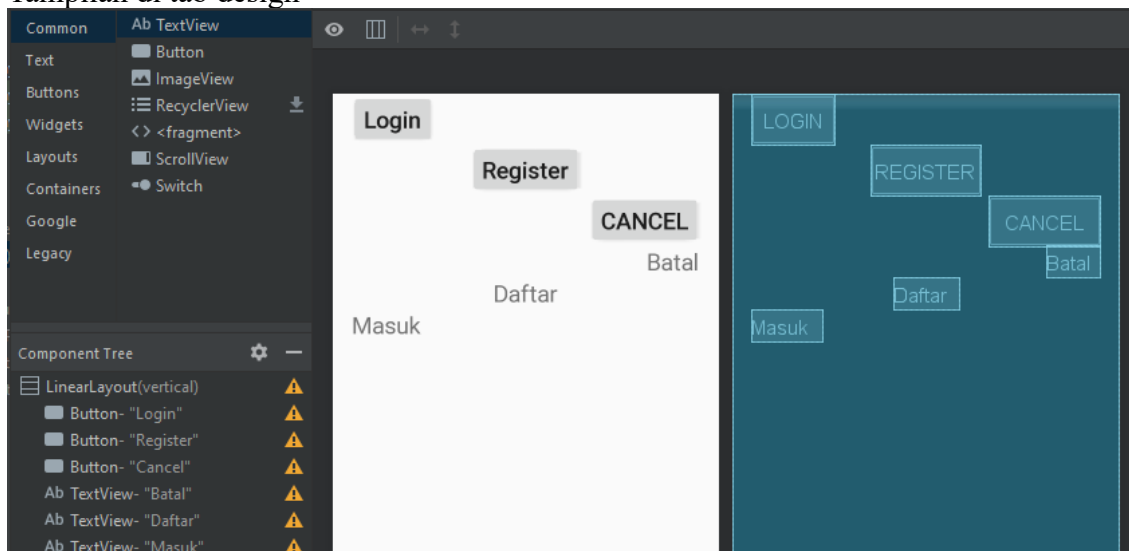
24         android:textAllCaps="false"
25         android:text="Register"
26         android:textSize="25sp"/>
27     <Button
28         android:layout_width="wrap_content"
29         android:layout_height="wrap_content"
30         android:layout_gravity="right"
31         android:text="Cancel"
32         android:textSize="25sp"
33         android:layout_marginLeft="20dp"
34         android:layout_marginRight="20dp"/>/>
35     <TextView
36         android:layout_width="wrap_content"
37         android:layout_height="wrap_content"
38         android:layout_gravity="right"
39         android:text="Batal"
40         android:textSize="25sp"
41         android:layout_marginLeft="20dp"
42         android:layout_marginRight="20dp"/>/>
43     <TextView
44         android:layout_width="wrap_content"
45         android:layout_height="wrap_content"
46         android:layout_gravity="center_horizontal"
47         android:text="Daftar"
48         android:textSize="25sp"/>
49     <TextView
50         android:layout_width="wrap_content"
51         android:layout_height="wrap_content"
52         android:layout_gravity="left"
53         android:text="Masuk"
54         android:textSize="25sp"
55         android:layout_marginLeft="20dp"
56         android:layout_marginRight="20dp"/>
57 </LinearLayout>

```

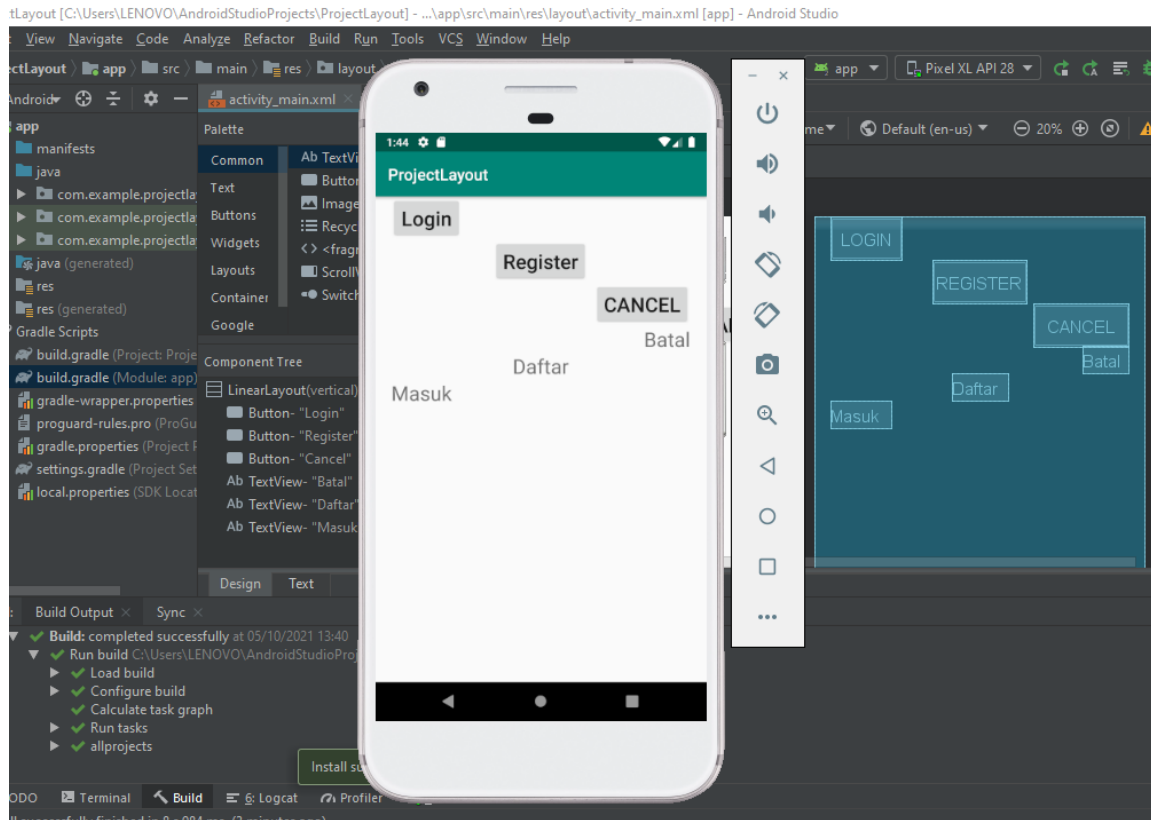
Penjelasan:

Pertama, kita buat 3 buah Button yang berjalan vertikal urut dari kiri ke kanan seperti anak tangga ke bawah. Text dalam seluruh button tersebut menggunakan bahasa inggris. Kedua, kita buat TextView yang juga membentuk tangga kebawah, tetapi dimulai dari kanan ke kiri. Isi TextView adalah terjemahan bahasa indonesia dari text bahasa inggris yang ada di button.

### 3. Tampilan di tab design



4. Sekarang kita run menggunakan virtual device emulator. Berikut tampilannya:

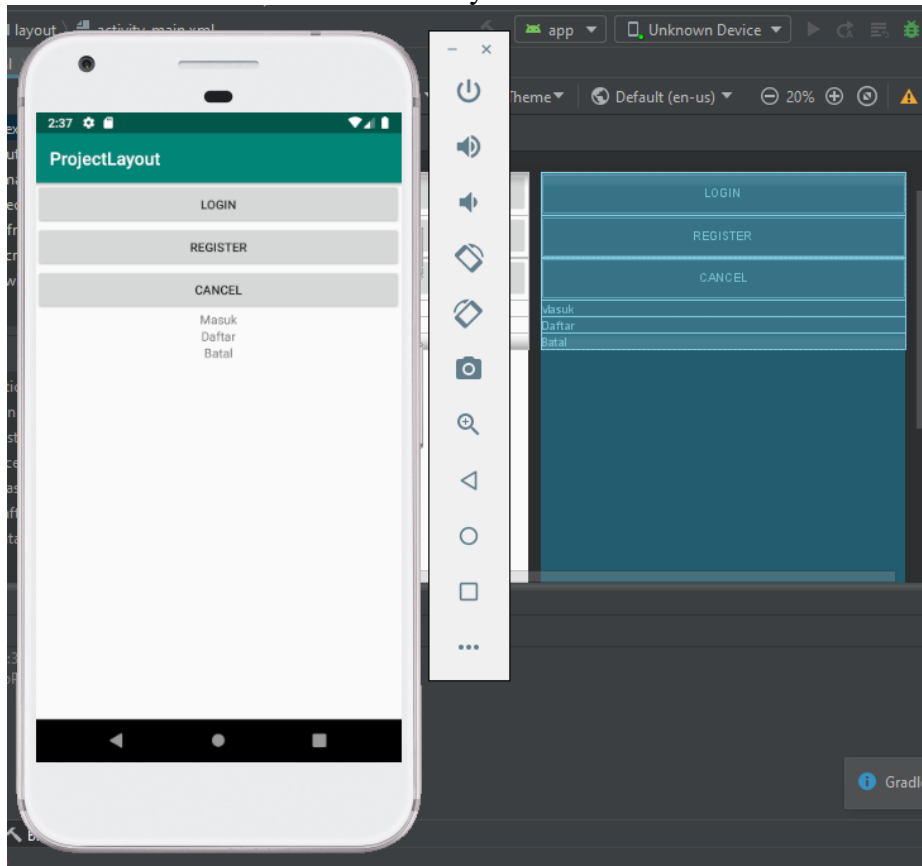


5. Sekarang kita coba kembalikan button dan textview tersebut ke posisi vertikal normal. Ganti kode pada button dan textview menjadi seperti berikut:

```
activity_main.xml x MainActivity.kt x app x
10 <Button
11     android:layout_width="match_parent"
12     android:layout_height="wrap_content"
13     android:text="Login"/>
14
15 <Button
16     android:layout_width="match_parent"
17     android:layout_height="wrap_content"
18     android:text="Register"/>
19
20 <Button
21     android:layout_width="match_parent"
22     android:layout_height="wrap_content"
23     android:text="Cancel"/>
24
25 <TextView
26     android:layout_width="match_parent"
27     android:layout_height="wrap_content"
28     android:gravity="center"
29     android:text="Masuk"/>
30
31 <TextView
32     android:layout_width="match_parent"
33     android:layout_height="wrap_content"
34     android:gravity="center"
35     android:text="Daftar"/>
36
37 <TextView
38     android:layout_width="match_parent"
39     android:layout_height="wrap_content"
40     android:gravity="center"
41     android:text="Batal"/>
42 </LinearLayout>
```



6. Jalankan di emulator. Berikut hasilnya



---

## KESIMPULAN

Pada laporan ini saya belajar banyak tentang constraint layout dan linear layout. Walaupun di awal saya sempat kebingungan, saya tetap dapat memenuhi tujuan modul ini yaitu mampu memahami dan mengetahui tentang Linear Layout dan Constraint Layout dan mampu menggunakan komponen view dasar (TextView, EditText, Button) dalam aplikasi. Selain dua jenis layout tersebut, kita juga belajar tentang komponen TextView dan Button. TextView adalah komponen yang berguna untuk menampilkan teks ke layar, sedangkan Button adalah komponen yang membuat pengguna dapat berinteraksi dengan cara ditekan untuk melakukan sesuatu.

Terima Kasih