

Modul Praktikum Pertemuan Ketiga

Membuat aplikasi dengan dua activity dan passing parameter

1. Buka projek yang sudah dibuat pada activity sebelumnya.
2. Buka MainActivity.java → tambahkan sintak untuk mendeklarasikan email dan password pada “button.onclick”

```
//membuat fungsi onclick pada button btnLogin
btnLogin.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        //menyimpan input user di editText password kedalam variabel nama
        nama = edemail.getText().toString();

        //menyimpan input user di editText password kedalam variabel password
        password = edpassword.getText().toString();

        //menengeset email yang benar
        String email = "admin@mail.com";

        //mengeset password yang benar
        String pass = "123";
```

→ pada

sintak yang mengatur kondisi jika input dari user benar → buatlah objek bundle

```
//mengecek apakah isi dari email dan password sudah sama dengan email dan
// password yang sudah diset
if (nama.equals(email) && password.equals(pass)) {
    //membuat variabel toast dan menampilkan pesan "Login sukses"
    Toast t = Toast.makeText(getApplicationContext(),
        text: "Login Sukses",
        Toast.LENGTH_LONG);

    //menampilkan toast
    t.show();

    //Membuat objek bundle
    Bundle b = new Bundle();
```

→ masukkan value dari email dan password kedalam bundle dan beri kunci untuk masing-masing variabel tersebut

```
//memsukkan value dari variabel nama dengan kunci "a"
// dan dimasukkan kedalam bundle
b.putString("a", nama.trim());

//memsukkan value dari variabel password dengan kunci "b"
// dan dimasukkan kedalam bundle
b.putString("b", password.trim());
```

→ buatlah objek intent

kemudian isi parameter dari objek intent tersebut dengan context “MainActivity.this” atau “getApplicationContext()” dan class yang dituju, yaitu class “ActivityHasil.class”

```
//membuat objek intent berpindah activity dari mainactivity ke ActivityHasil
Intent i = new Intent(getApplicationContext(), ActivityHasil.class);
```

→ masukkan bundle kedalam intent

```
//memasukkan bundle kedalam intent untuk dikirimkan ke ActivityHasil
i.putExtras(b);
```

→ buat method yang berfungsi untuk berpindah activity ke Activity Hasil

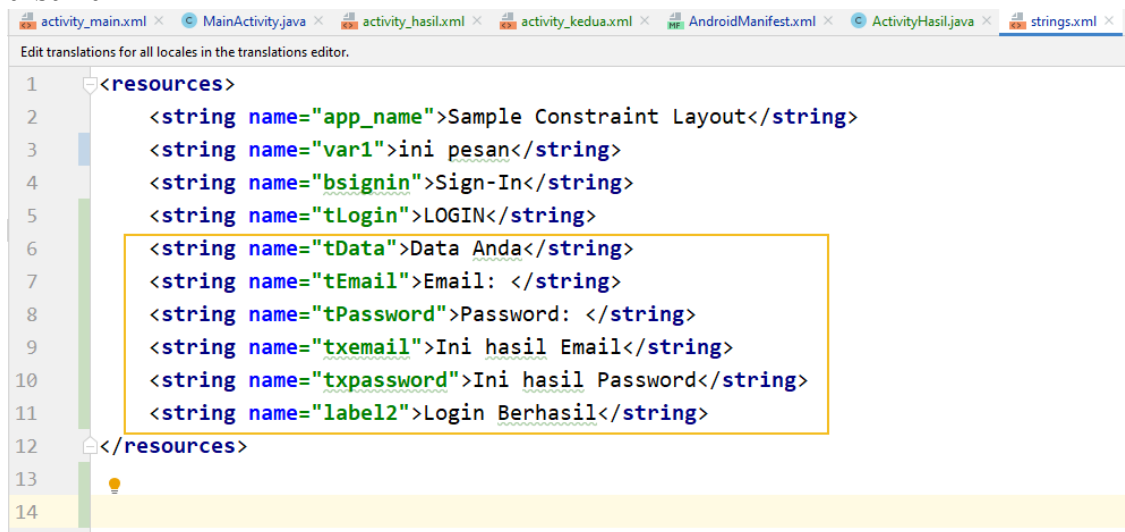
```
//berpindah ke ActivityHasil
startActivity(i);
```

. Sehingga tampilan dari method button.onclick akan menjadi seperti pada gambar dibawah.

File - E:\Materi Kuliah\Instruktur-Asisten\Pengembangan Aplikasi Mobile\Program\2021\Constraint Layout\app\src\main\java\com\example\sampleconstraintlayout\MainA

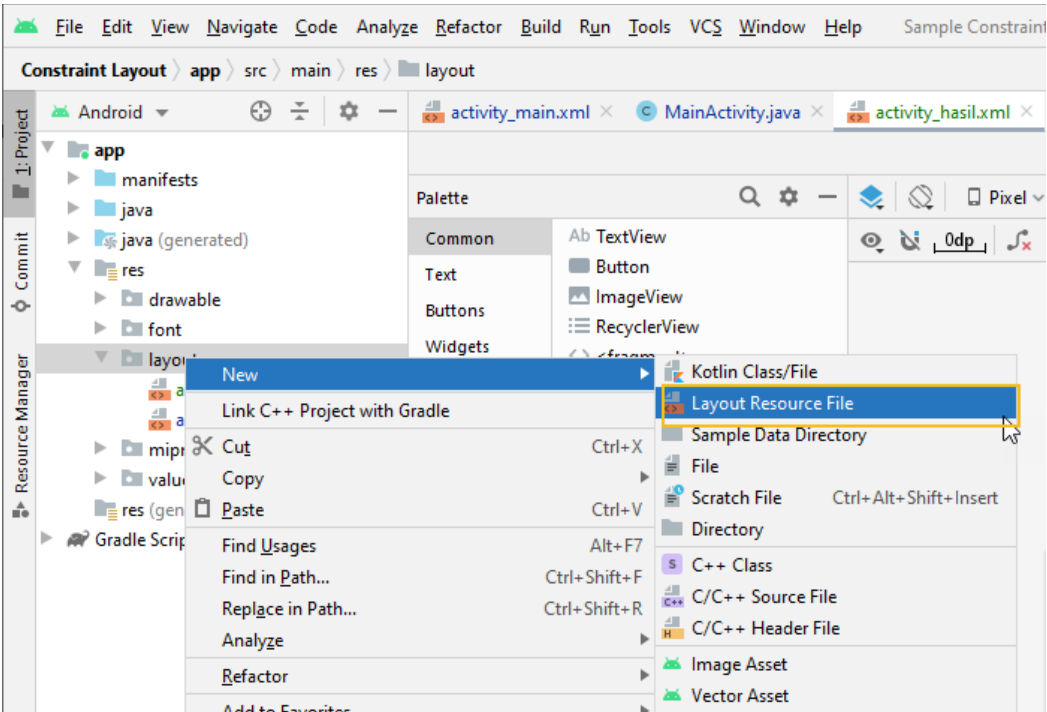
```
36 //membuat fungsi onClik pada button btnLogin
37 btnLogin.setOnClickListener(new View.OnClickListener() {
38     @Override
39     public void onClick(View view) {
40         //menyimpan input user di editText password kedalam variabel nama
41         nama = edemail.getText().toString();
42
43         //menyimpan input user di editText password kedalam variabel password
44         password = edpassword.getText().toString();
45
46         //menengeset email yang benar
47         String email = "admin@mail.com";
48
49         //mengeset password yang benar
50         String pass = "123";
51
52         //mengecek apakah editText email dan password terdapat isi atau tidak
53         if (nama.isEmpty() || password.isEmpty()) {
54             //membuat variabel toast dan menampilkan pesan "edittext tidak boleh kosong"
55             Toast t = Toast.makeText(getApplicationContext(),
56                 "Email dan password wajib diisi!!!",
57                 Toast.LENGTH_LONG);
58             //menampilkan toast
59             t.show();
60         } else {
61
62             //mengecek apakah isi dari eamial dan password sudah sama dengan email dan
63             // password yang sudah diset
64             if (nama.equals(email) && password.equals(pass)) {
65                 //membuat variabel toast dan menampilkan pesan "Login sukses"
66                 Toast t = Toast.makeText(getApplicationContext(),
67                     "Login Sukses",
68                     Toast.LENGTH_LONG);
69
70                 //menampilkan toast
71                 t.show();
72
73                 //Membuat objek bundle
74                 Bundle b = new Bundle();
75
76                 //memsukkan value dari variabel nama dengan kunci "a"
77                 // dan dimasukkan kedalam bundle
78                 b.putString("a", nama.trim());
79
80                 //memsukkan value dari variabel password dengan kunci "b"
81                 // dan dimasukkan kedalam bundle
82                 b.putString("b", password.trim());
83
84                 //membuat objek intent berpindah activity dari mainactivity ke ActivityHasil
85                 Intent i = new Intent(getApplicationContext(), ActivityHasil.class);
86
87                 //memasukkan bundle kedalam intent untuk dikirimkan ke ActivityHasil
88                 i.putExtras(b);
89
90                 //berpindah ke ActivityHasil
91                 startActivity(i);
92             } else {
93                 //membuat variabel toast dan membuat toast dan menampilkan pesan "login gagal"
94
95                 Toast t = Toast.makeText(getApplicationContext(),
96                     "Login Gagal", Toast.LENGTH_LONG);
97
98                 //menampilkan toast
99                 t.show();
100             }
101         }
102     });
```

3. Buka strings.xml → tambahkan beberapa string resources seperti pada gambar dibawah ini.

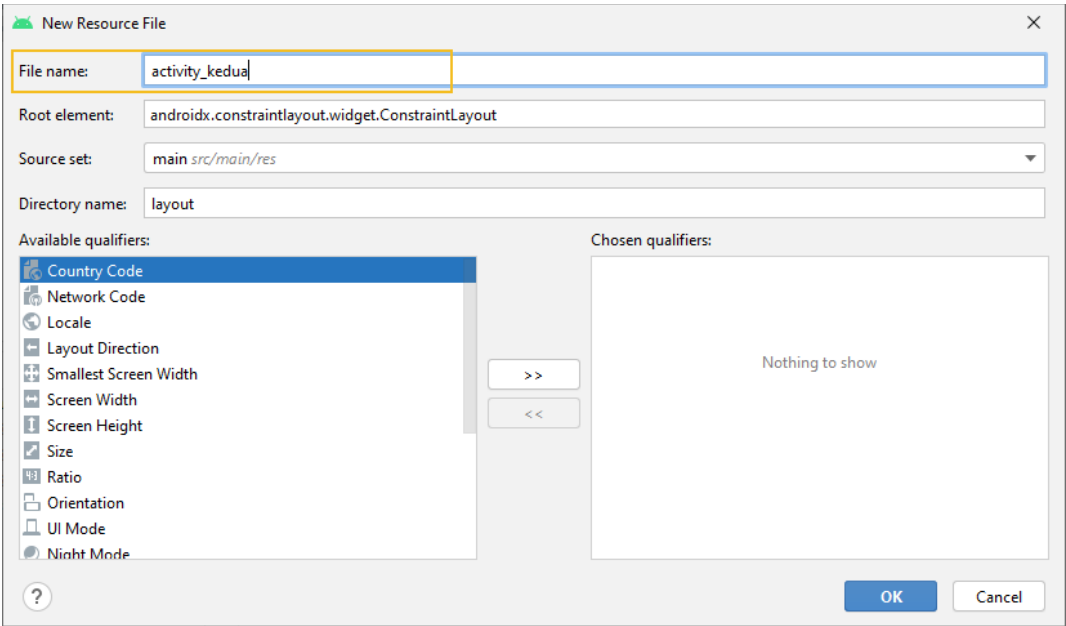


```
1 <resources>
2   <string name="app_name">Sample Constraint Layout</string>
3   <string name="var1">ini pesan</string>
4   <string name="bsignin">Sign-In</string>
5   <string name="tLogin">LOGIN</string>
6   <string name="tData">Data Anda</string>
7   <string name="tEmail">Email: </string>
8   <string name="tPassword">Password: </string>
9   <string name="txemail">Ini hasil Email</string>
10  <string name="txpassword">Ini hasil Password</string>
11  <string name="label2">Login Berhasil</string>
12 </resources>
```

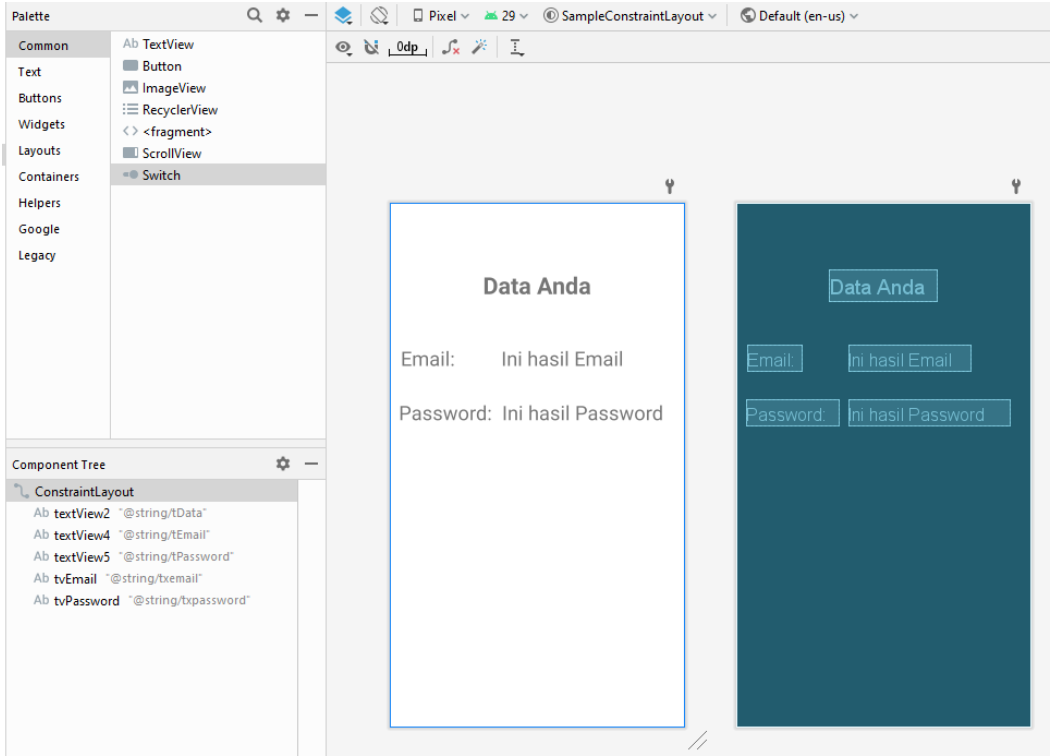
4. Tambah activity baru. Terdapat dua cara untuk menambahkan activity baru, yaitu:
- a. Menambahkan layout dan java class secara terpisah kemudian mendaftarkan activity kedalam android manifest
 - **Langkah pertama**
 - Klik kanan pada folder layout → new → Pilih Layout Resources File



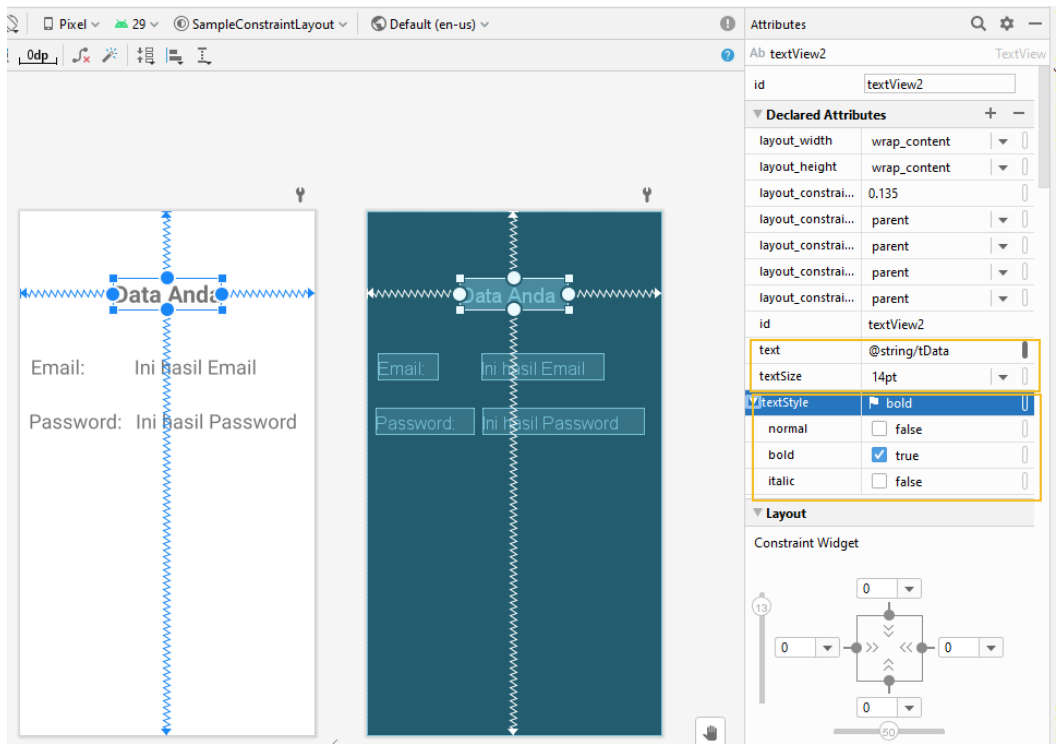
- **Langkah kedua**
 - Beri nama file layout untuk activity kedua (note: penamaan layout wajib menggunakan huruf kecil) → jika sudah klik “OK”.



- **Langkah ketiga**
 - Tambahkan 5 buah textview sehingga tampilan akan menjadi seperti dibawah ini



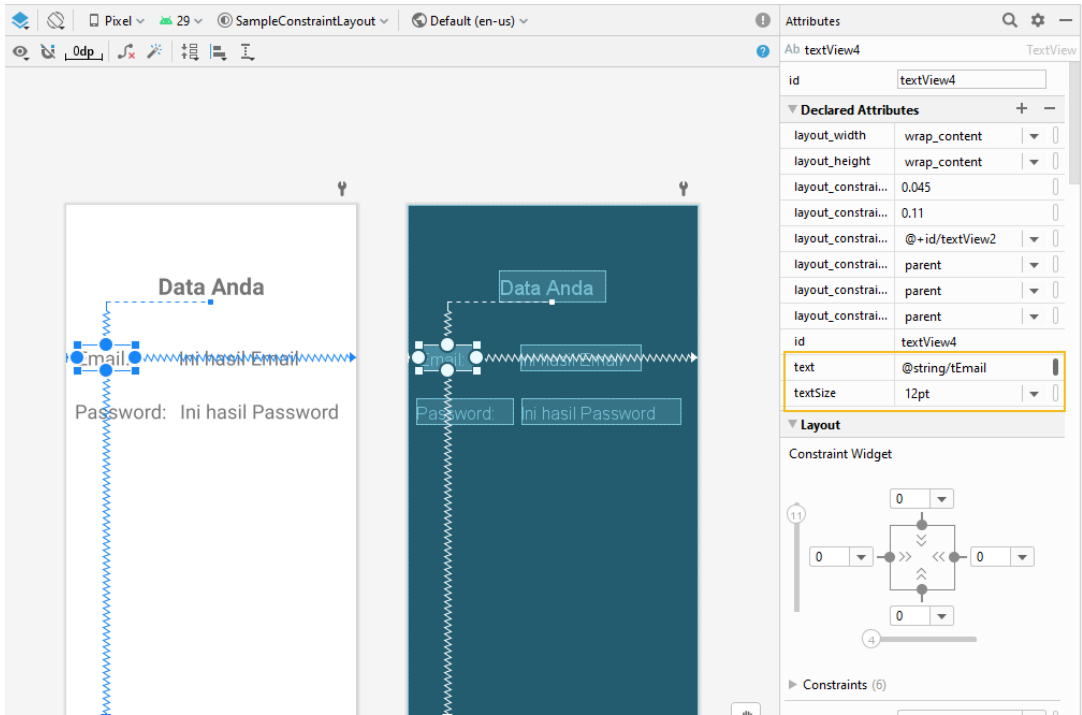
- Klik textview pertama → pada bagian atribut → cari textStyle → centang bagian “bold” → cari textSize → ubah ukuran font menjadi “14pt” → kemudian pada bagian text → isi dengan string resources dengan nama variabel resources string “tData”. Sehingga tampilan atribut dari textview yang pertama akan seperti pada gambar dibawah.



Untuk code xml dari textview pertama akan menjadi seperti gambar dibawah.

```
<TextView
    android:id="@+id/textView2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/tData"
    android:textSize="14pt"
    android:textStyle="bold"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.135" />
```

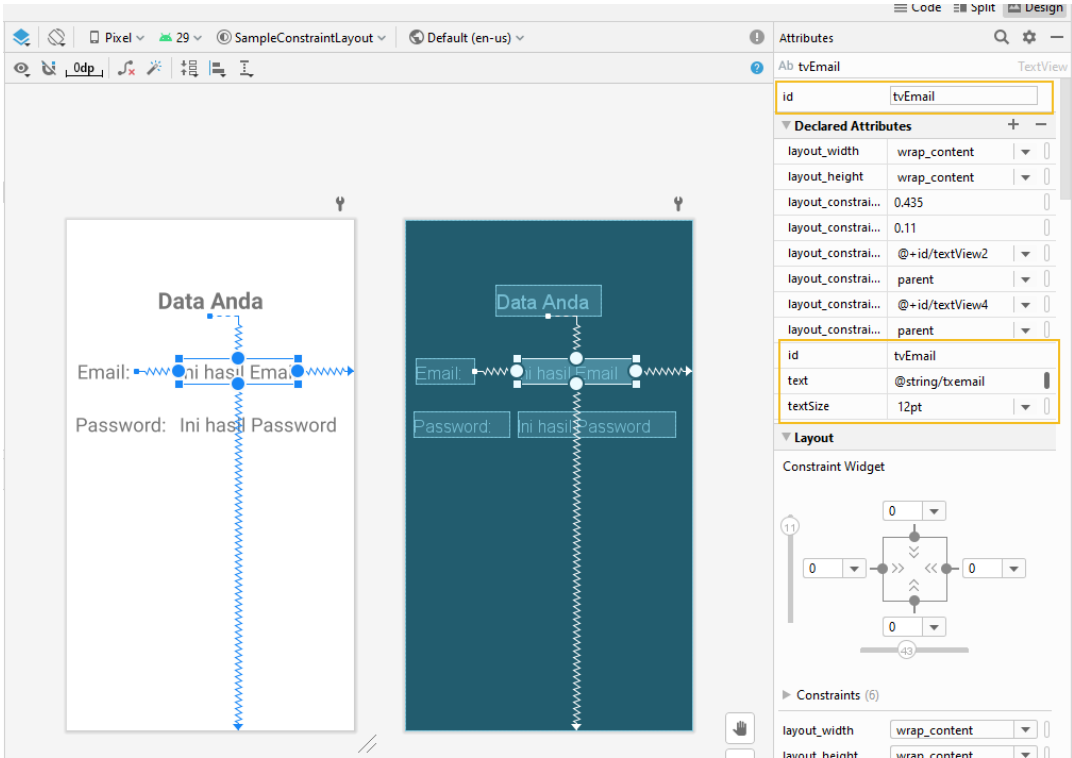
- Klik textview kedua → pada bagian atribut → cari textSize → ubah ukuran font menjadi “12pt” → kemudian pada bagian text → isi dengan string resources dengan nama variabel resources string “tEmail”. Sehingga tampilan atribut dari textview yang kedua akan seperti pada gambar dibawah.



Untuk code xml dari textview kedua akan menjadi seperti gambar dibawah.

```
<TextView
    android:id="@+id/textView4"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/tEmail"
    android:textSize="12pt"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.045"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/textView2"
    app:layout_constraintVertical_bias="0.11" />
```

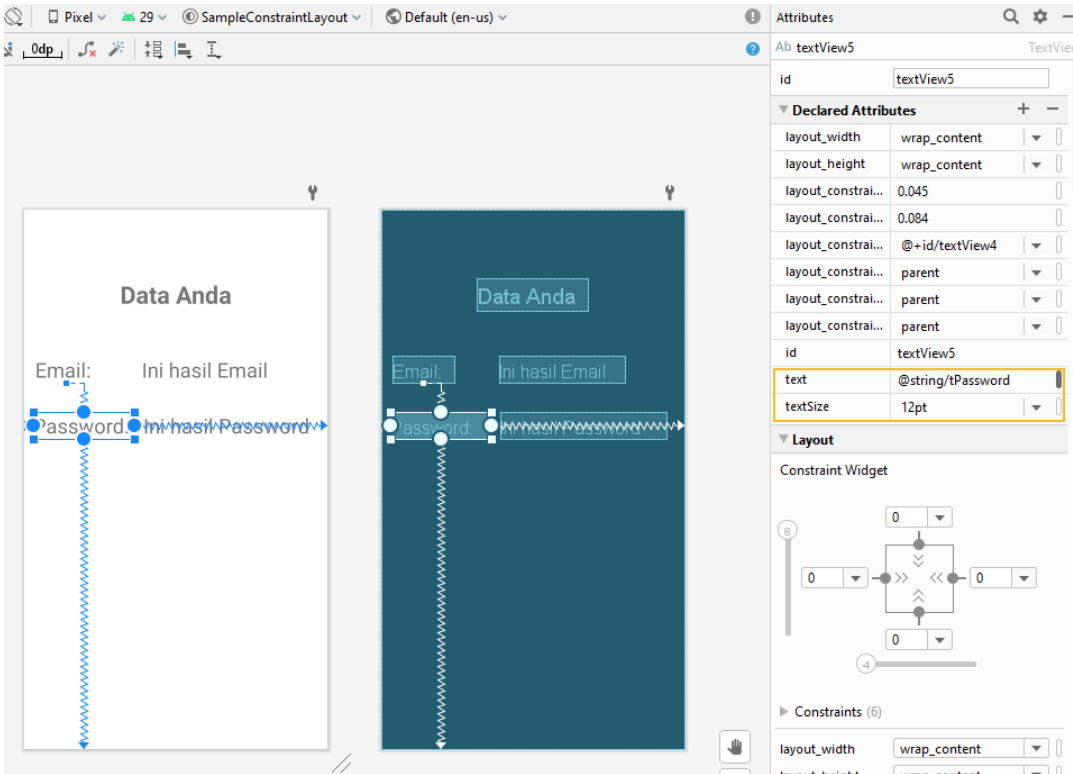
- Klik textview ketiga → pada bagian atribut → cari textSize → ubah ukuran font menjadi “12pt” → kemudian pada bagian text → isi dengan string resources dengan nama variabel resources string “txemail” → kemudian pada bagian “id” → ubah id dari textview ketiga menjadi “tvEmail”. Sehingga tampilan atribut dari textview yang ketiga akan seperti pada gambar dibawah.



Untuk code xml dari textview ketiga akan menjadi seperti gambar dibawah.

```
<TextView
    android:id="@+id/tvEmail"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/txemail"
    android:textSize="12pt"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.435"
    app:layout_constraintStart_toEndOf="@+id/textView4"
    app:layout_constraintTop_toBottomOf="@+id/textView2"
    app:layout_constraintVertical_bias="0.11" />
```

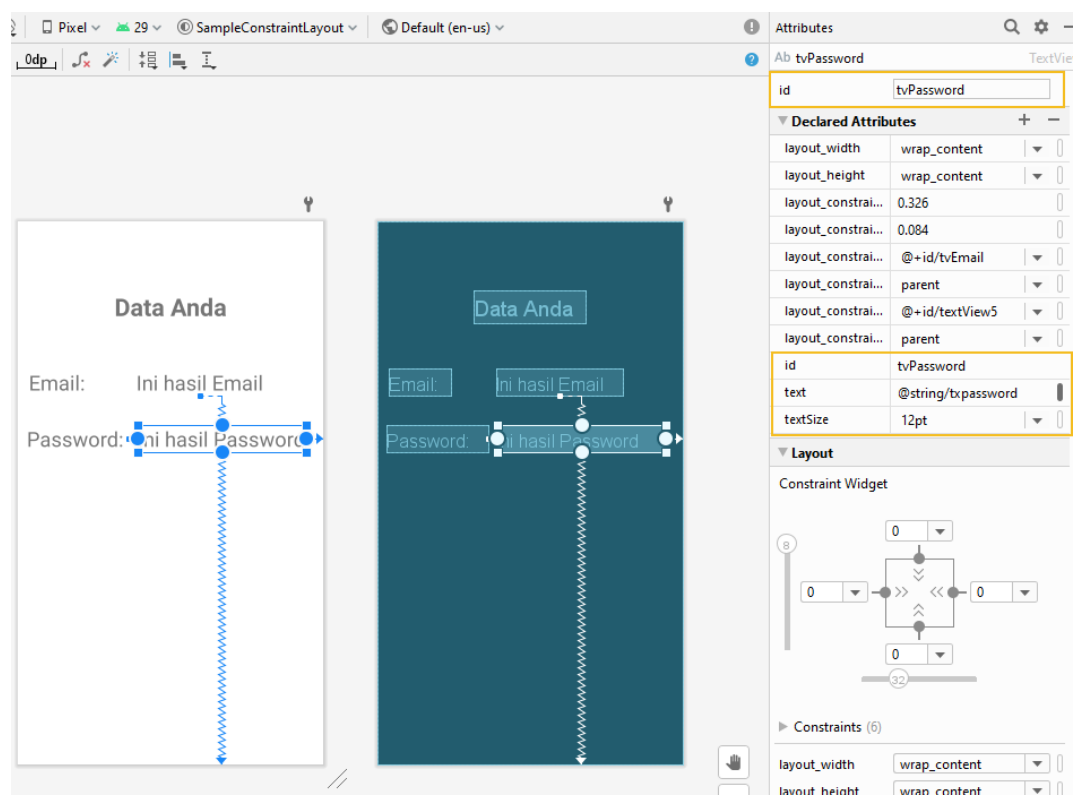
- Klik textview keempat → pada bagian atribut → cari textSize → ubah ukuran font menjadi “12pt” → kemudian pada bagian text → isi dengan string resources dengan nama variabel resources string “tPassword”. Sehingga tampilan atribut dari textview yang keempat akan seperti pada gambar dibawah.



Untuk code xml dari textview keempat akan menjadi seperti gambar dibawah.

```
<TextView
    android:id="@+id/textView5"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/tPassword"
    android:textSize="12pt"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.045"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/textView4"
    app:layout_constraintVertical_bias="0.084" />
```

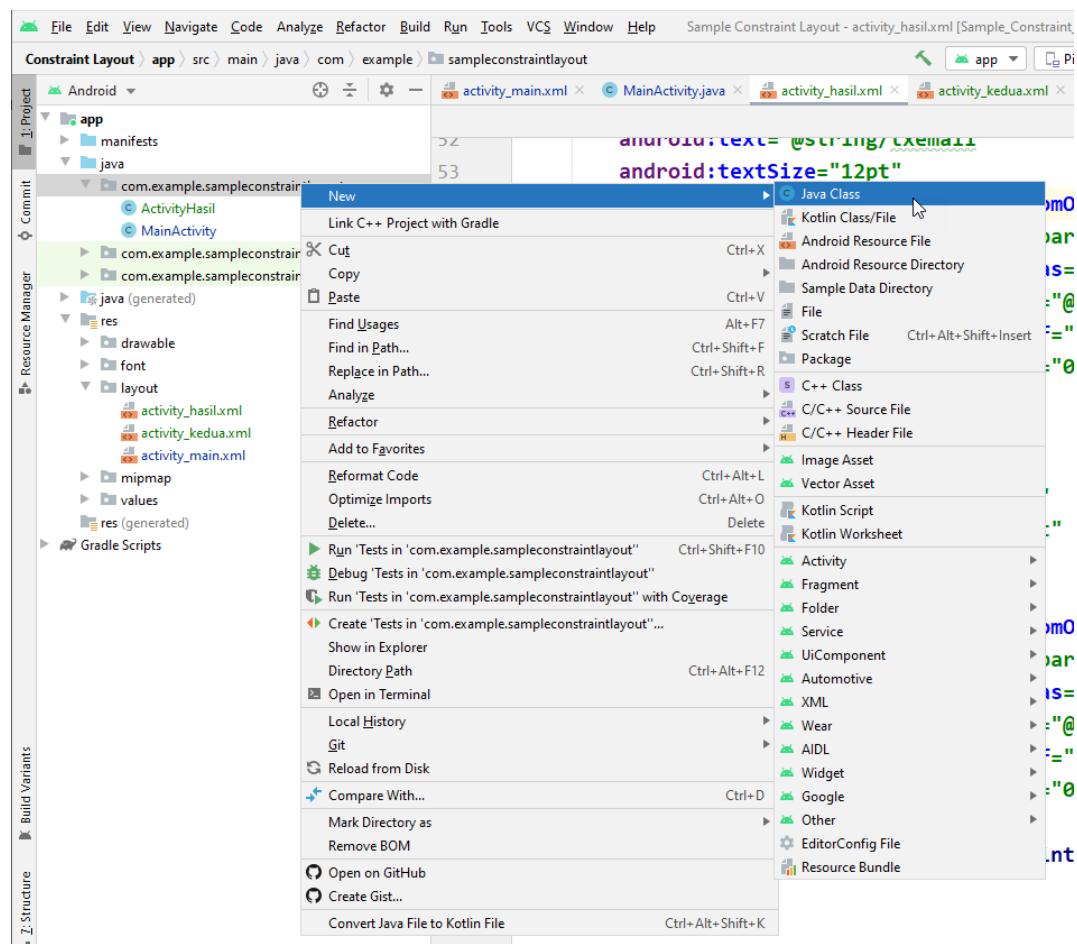

- Klik textview kelima → pada bagian atribut → cari textSize → ubah ukuran font menjadi “12pt” → kemudian pada bagian text → isi dengan string resources dengan nama variabel resources string “txpassword” → cari bagian “id” → ubah id dari textview kelima menjadi “tvPassword”. Sehingga tampilan atribut dari textview yang kelima akan seperti pada gambar dibawah.



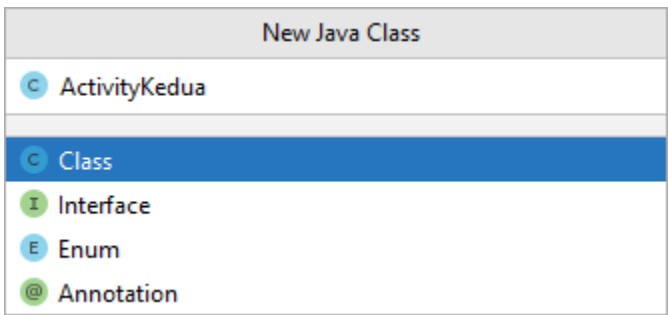
Untuk code xml dari textview kelima akan menjadi seperti gambar dibawah.

```
<TextView
    android:id="@+id/tvPassword"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/txpassword"
    android:textSize="12pt"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.326"
    app:layout_constraintStart_toEndOf="@+id/textView5"
    app:layout_constraintTop_toBottomOf="@+id/tvEmail"
    app:layout_constraintVertical_bias="0.084" />
```

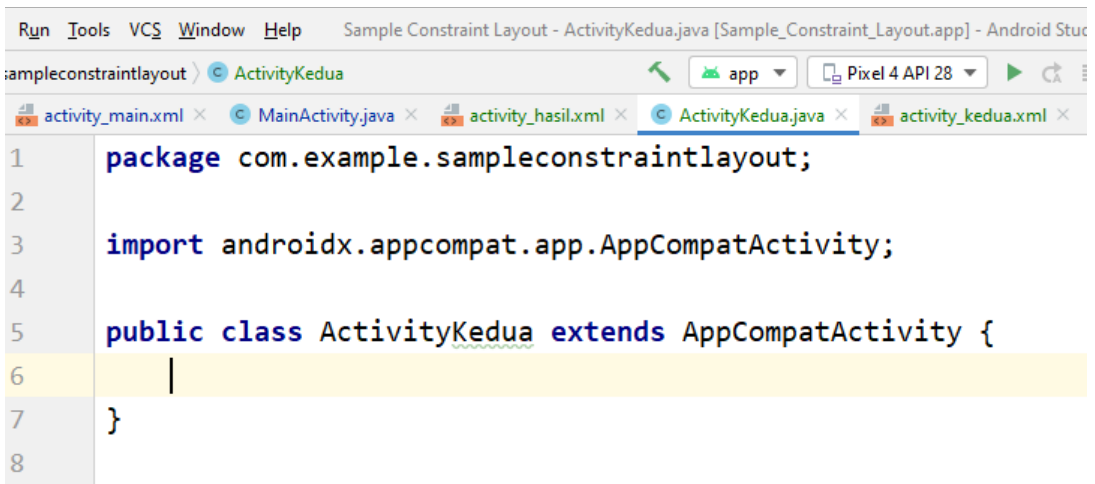
- Langkah keempat
 - Membuat java class baru → klik kanan pada nama package pada folder java → klik new → pilih Java Class



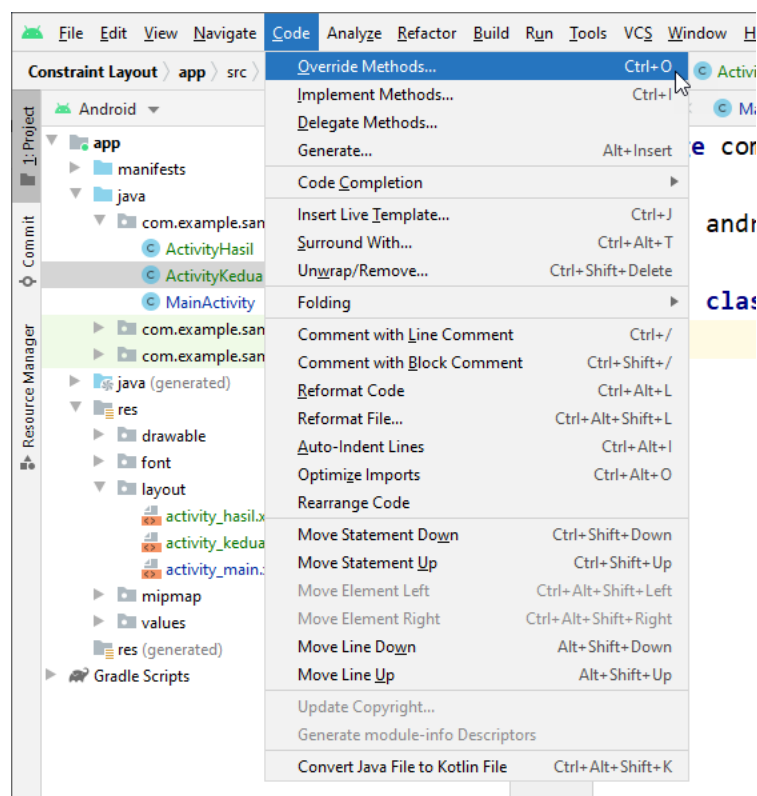
- Beri nama java class “ActivityKedua” → pilih “class” → tekan “enter”



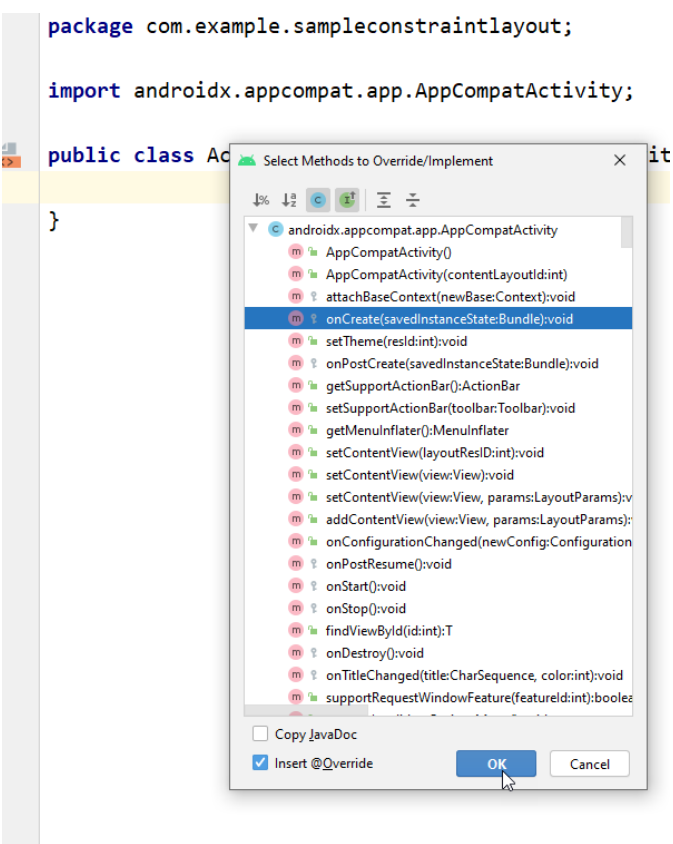
- Buka ActivityKedua.java → tambahkan keyword “extends AppCompatActivity” setelah nama class. Sehingga akan menjadi seperti pada gambar dibawah.



- Didalam class ActivityKedua → klik “Code” → klik “Override Methods...”.



- Pilih “onCreate(savedInstanceState:Bundle):void” → kemudian klik “OK”



- Hapus “import androidx.annotation.Nullable;” dan “@Nullable”.

```
package com.example.sampleconstraintlayout;

import android.os.Bundle;

import androidx.appcompat.app.AppCompatActivity;

public class ActivityKedua extends AppCompatActivity {
    @Override
    protected void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
    }
}
```

- Tambahkan method “setContentView()” untuk menampilkan layout “activity_kedua”. Sehingga tampilannya akan menjadi seperti gambar dibawah

```
package com.example.sampleconstraintlayout;

import android.os.Bundle;
import androidx.appcompat.app.AppCompatActivity;

public class ActivityKedua extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_kedua);
    }
}
```

- Deklarasikan variabel dengan typedata TextView yang akan digunakan untuk menghubungkan textview didalam java class dan textview didalam layout didalam class “ActivityKedua”

```
package com.example.sampleconstraintlayout;

import android.os.Bundle;
import android.widget.TextView;
import androidx.appcompat.app.AppCompatActivity;

public class ActivityKedua extends AppCompatActivity {
    //Mendeklarasikan variabel dengan tipe data TextView
    TextView txEmail, txPassword;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_kedua);
    }
}
```

- Hubungkan textview yang ada didalam java class dengan textview yang ada pada layout dengan menggunakan method “findViewById”. Sehingga sintak code menjadi seperti pada gambar dibawah.

@Override

```
protected void onCreate( Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_kedua);

    //Menghubungkan variabel txEmail dengan componen Textview pada layout
    txEmail = findViewById(R.id.tvEmail);

    //Menghubungkan variabel txEmail dengan componen Textview pada layout
    txPassword = findViewById(R.id.tvPassword);
```

- Mendeklarasikan bundle yang akan digunakan untuk mengambil pesan atau nilai yang dikirimkan melalui intent dari activity pertama ke activity kedua. (penulisan method tersebut dilakukan didalam method onCreate)

```
//mendeklarasikan variabel bundle yang akan digunakan untuk mengambil
// pesan yang dikirimkan melalui method intent
Bundle bundle = getIntent().getExtras();
```

- Buat sintak yang berfungsi untuk mengambil nilai yang dikirimkan dengan menggunakan intent dari activity pertama ke activity kedua. Method yang digunakan adalah method “bundle.getString(“key”)” dan kemudian simpan nilai tersebut kedalam variabel dengan tipe data string. (penulisan method tersebut dilakukan didalam method onCreate)

```
//membuat variabel string yang digunakan untuk menyimpan data yang
// dikirimkan dari activity sebelumnya dengan kunci "a"
String email = bundle.getString( key: "a");
```

```
//membuat variabel string yang digunakan untuk menyimpan data yang
// dikirimkan dari activity sebelumnya dengan kunci "b"
String pass = bundle.getString( key: "b");
```

- Tampilkan nilai tersebut kedalam textview dengan menggunakan method “TextView.setText(value)”. (penulisan method tersebut dilakukan didalam method onCreate)

```
//menampilkan value dari variabel email kedalam txEmail
txEmail.setText(email);
```

```
//menampilkan value dari variabel pass kedalam txPassword
txPassword.setText(pass);
```

- Sehingga tampilan dari class ActivityKedua akan menjadi seperti pada gambar dibawah.

```
package com.example.sampleconstraintlayout;

import android.os.Bundle;
import android.widget.TextView;
import androidx.appcompat.app.AppCompatActivity;

public class ActivityKedua extends AppCompatActivity {
    //Mendeklarasikan variabel dengan tipe data TextView
    TextView txEmail, txPassword;

    @Override
    protected void onCreate( Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_kedua);

        //Menghubungkan variabel txEmail dengan componen Textview pada layout
        txEmail = findViewById(R.id.tvEmail);

        //Menghubungkan variabel txEmail dengan componen Textview pada layout
        txPassword = findViewById(R.id.tvPassword);

        //mendeklarasikan variabel bundle yang akan digunakan untuk mengambil
        // pesan yang dikirimkan melalui method intent
        Bundle bundle = getIntent().getExtras();

        //membuat variabel string yang digunakan untuk menyimpan data yang
        // dikirimkan dari activity sebelumnya dengan kunci "a"
        String email = bundle.getString( key: "a");

        //membuat variabel string yang digunakan untuk menyimpan data yang
        // dikirimkan dari activity sebelumnya dengan kunci "b"
        String pass = bundle.getString( key: "b");

        //menampilkan value dari variabel email kedalam txEmail
        txEmail.setText(email);

        //menampilkan value dari variabel pass kedalam txPassword
        txPassword.setText(pass);
    }
}
```

- **Langkah Kelima**

- Buka AndroidManifest.xml → daftar activity kedua didalam manifest
→ pada bagian "<application ... " tambahkan sintak berikut

```
<activity android:name=".ActivityKedua"
    android:parentActivityName=".MainActivity"
    android:label="@string/label2"></activity>
```

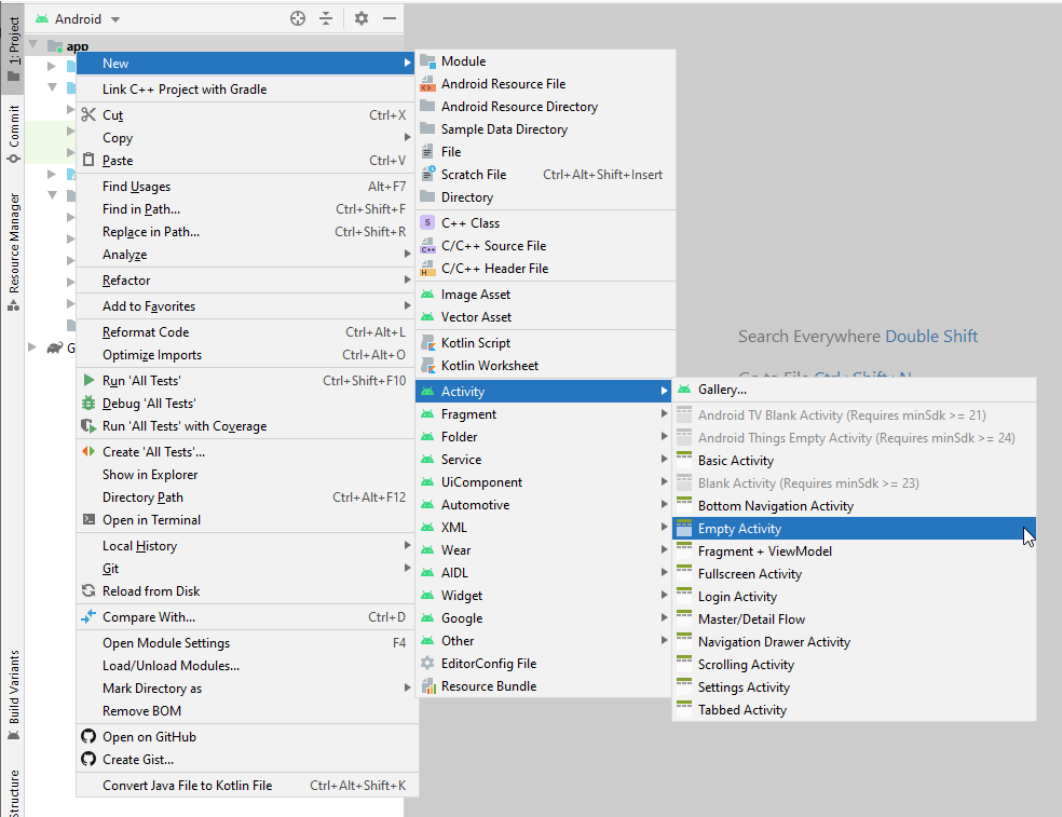
sehingga tampilan dari AndroidManifest.xml akan menjadi seperti pada gambar dibawah.

```
<application
    android:allowBackup="true"
    android:icon="@mipmap/ic_launcher"
    android:label="@string/app_name"
    android:roundIcon="@mipmap/ic_launcher_round"
    android:supportsRtl="true"
    android:theme="@style/Theme.SampleConstraintLayout">
    <activity android:name=".ActivityKedua"
        android:parentActivityName=".MainActivity"
        android:label="@string/label2"></activity>
    <activity android:name=".MainActivity">
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />

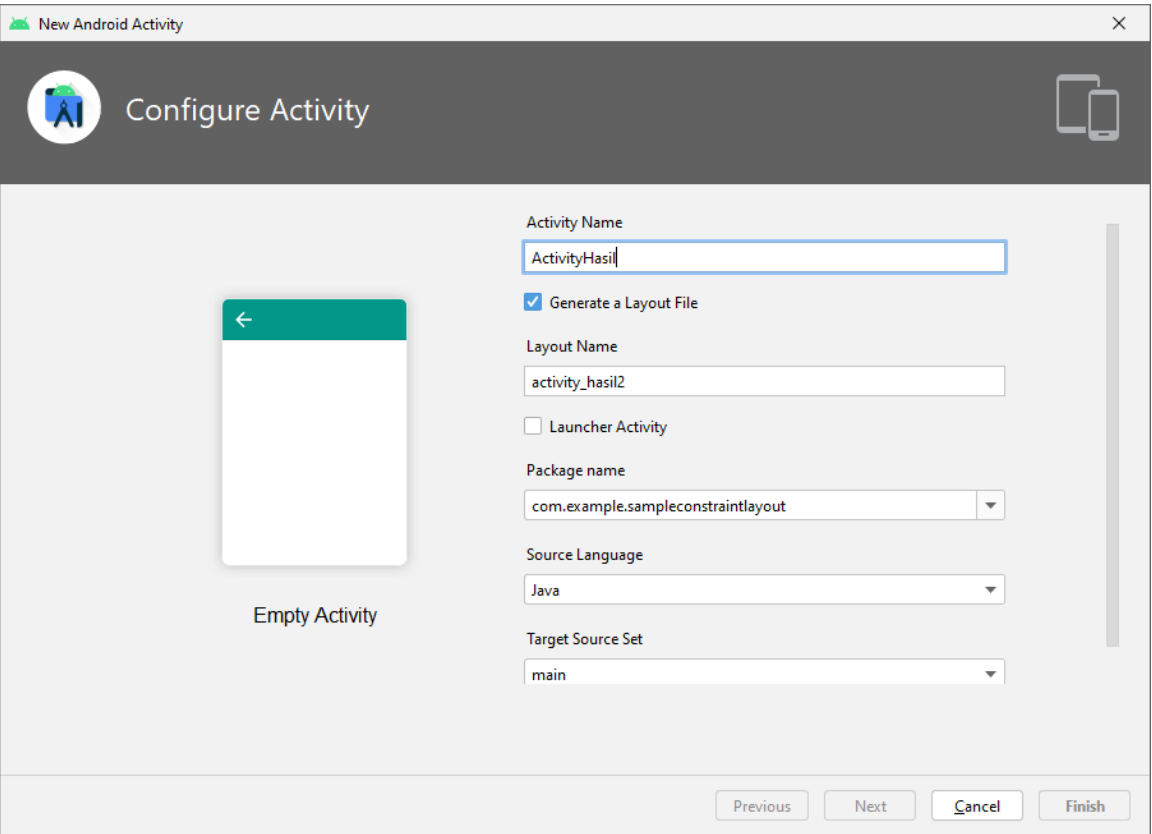
            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>

    <meta-data
        android:name="preloaded_fonts"
        android:resource="@array/preloaded_fonts" />
</application>
```

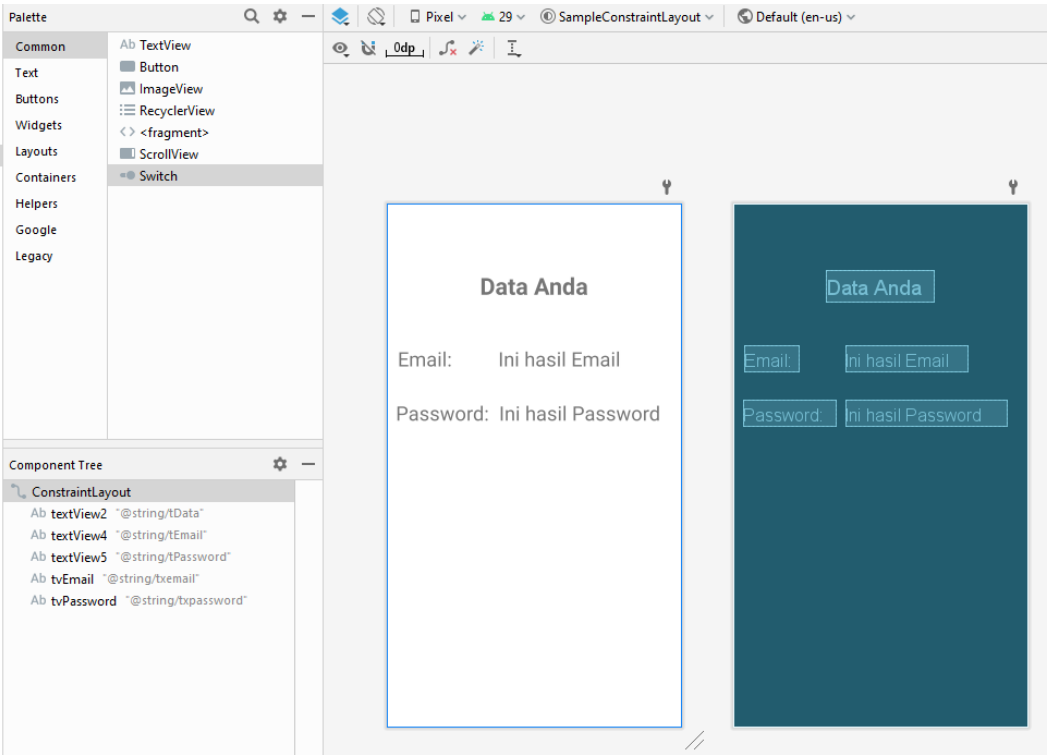

- b. Menambahkan activity yang sekaligus menambahkan java class dan otomatis terdaftar didalam android manifest.
- **Langkah Pertama**
 - Klik kana pada folder “app” → new → Activity → pilih “Empty Activity”



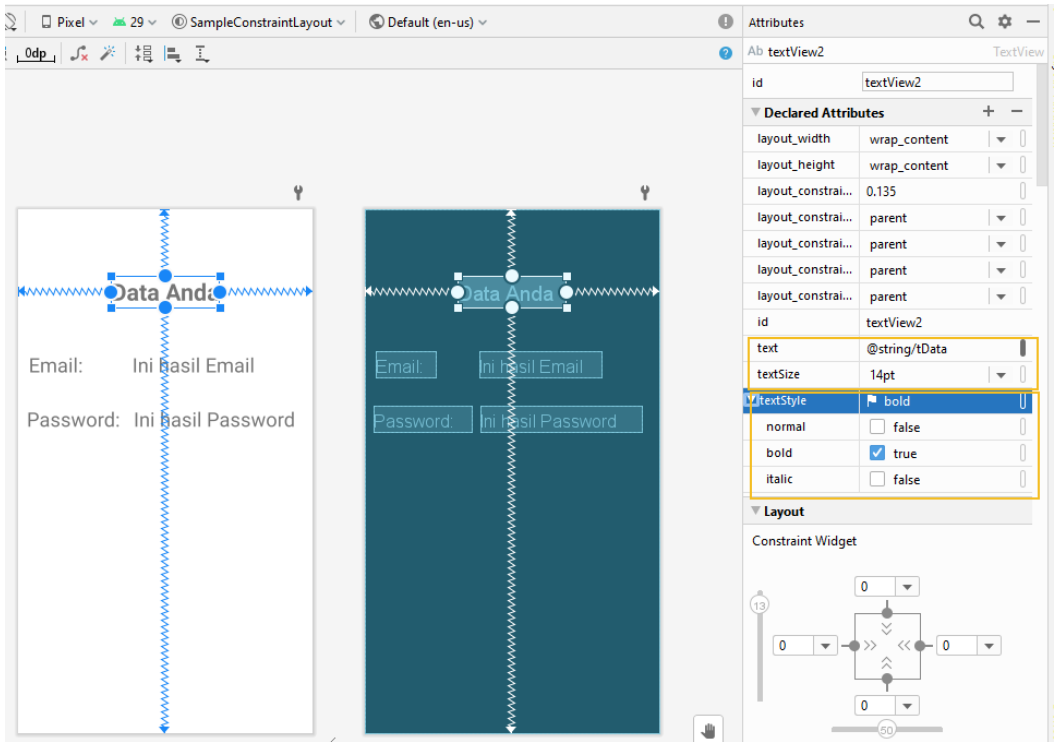
- Tulis “Activity Name” menjadi “ActivityHasil”, untuk nama layout akan berubah secara otomatis mengikuti “Activity Name” → pilih Bahasa yang akan digunakan menjadi “java” → kemudian pilih “target source set” menjadi “main” → Klik “Finish”



- **Langkah Kedua**
 - Tambahkan 5 buah textview sehingga tampilan akan menjadi seperti dibawah ini



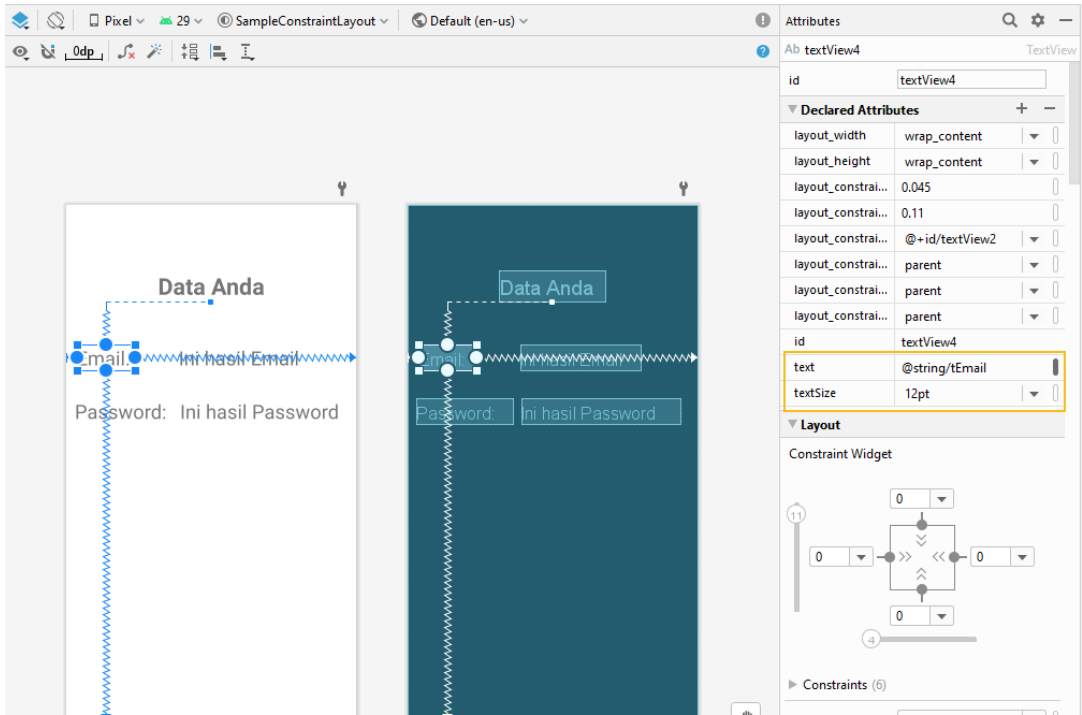
- Klik textview pertama → pada bagian atribut → cari textStyle → centang bagian “bold” → cari textSize → ubah ukuran font menjadi “14pt” → kemudian pada bagian text → isi dengan string resources dengan nama variabel resources string “tData”. Sehingga tampilan atribut dari textview yang pertama akan seperti pada gambar dibawah.



Untuk code xml dari textview pertama akan menjadi seperti gambar dibawah.

```
<TextView
    android:id="@+id/textView2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/tData"
    android:textSize="14pt"
    android:textStyle="bold"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.135" />
```

- Klik textview kedua → pada bagian atribut → cari textSize → ubah ukuran font menjadi “12pt” → kemudian pada bagian text → isi dengan string resources dengan nama variabel resources string “tEmail”. Sehingga tampilan atribut dari textview yang kedua akan seperti pada gambar dibawah.

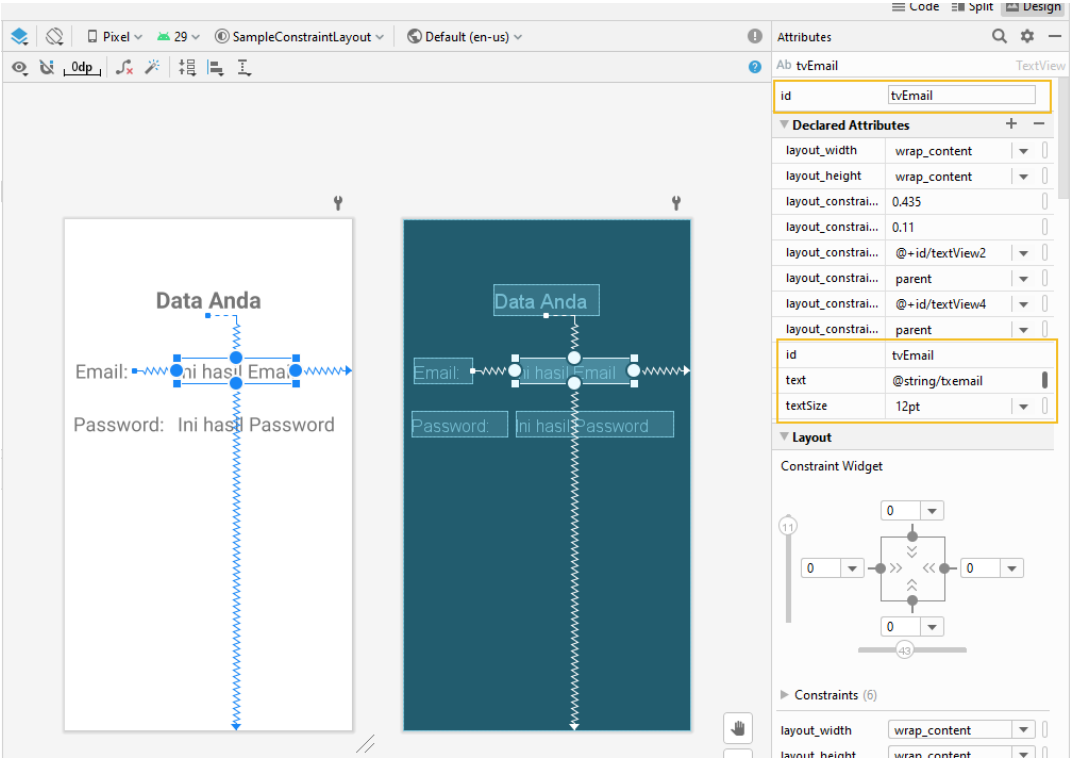


Untuk code xml dari textview kedua akan menjadi seperti gambar dibawah.

<TextView

```
android:id="@+id/textView4"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="@string/tEmail"
android:textSize="12pt"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.045"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/textView2"
app:layout_constraintVertical_bias="0.11" />
```

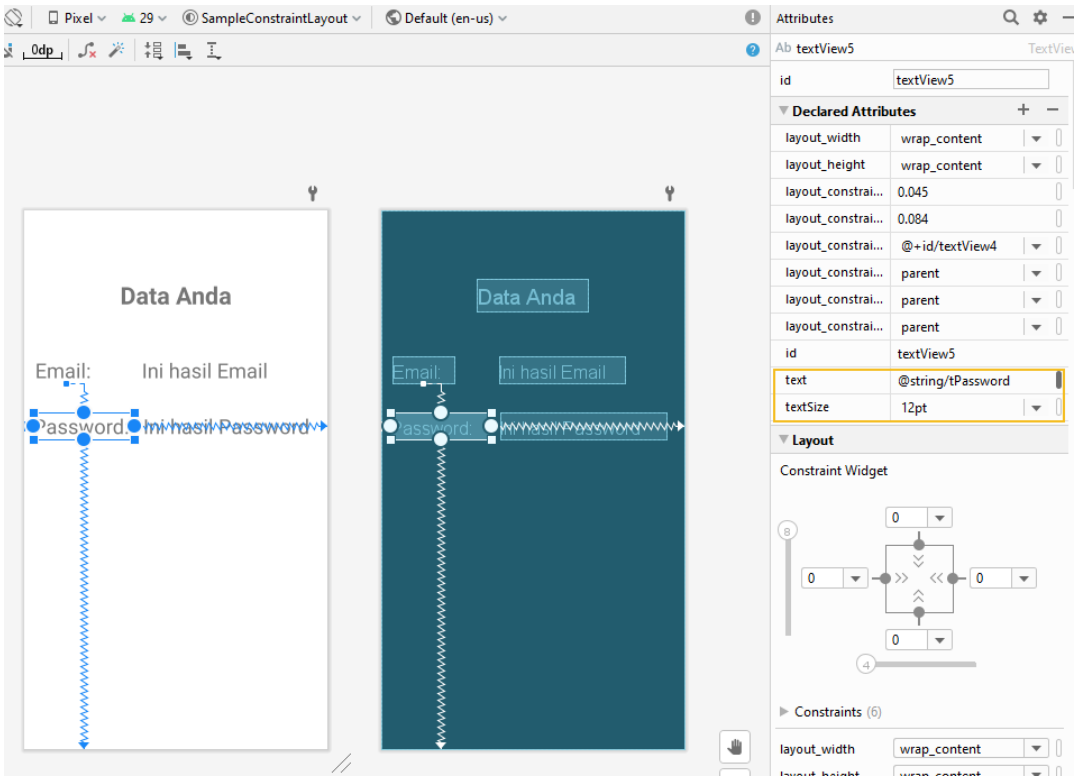
- Klik textview ketiga → pada bagian atribut → cari textSize → ubah ukuran font menjadi “12pt” → kemudian pada bagian text → isi dengan string resources dengan nama variabel resources string “txemail” → kemudian pada bagian “id” → ubah id dari textview ketiga menjadi “tvEmail”. Sehingga tampilan atribut dari textview yang ketiga akan seperti pada gambar dibawah.



Untuk code xml dari textview ketiga akan menjadi seperti gambar dibawah.

```
<TextView
    android:id="@+id/tvEmail"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/txemail"
    android:textSize="12pt"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.435"
    app:layout_constraintStart_toEndOf="@+id/textView4"
    app:layout_constraintTop_toBottomOf="@+id/textView2"
    app:layout_constraintVertical_bias="0.11" />
```

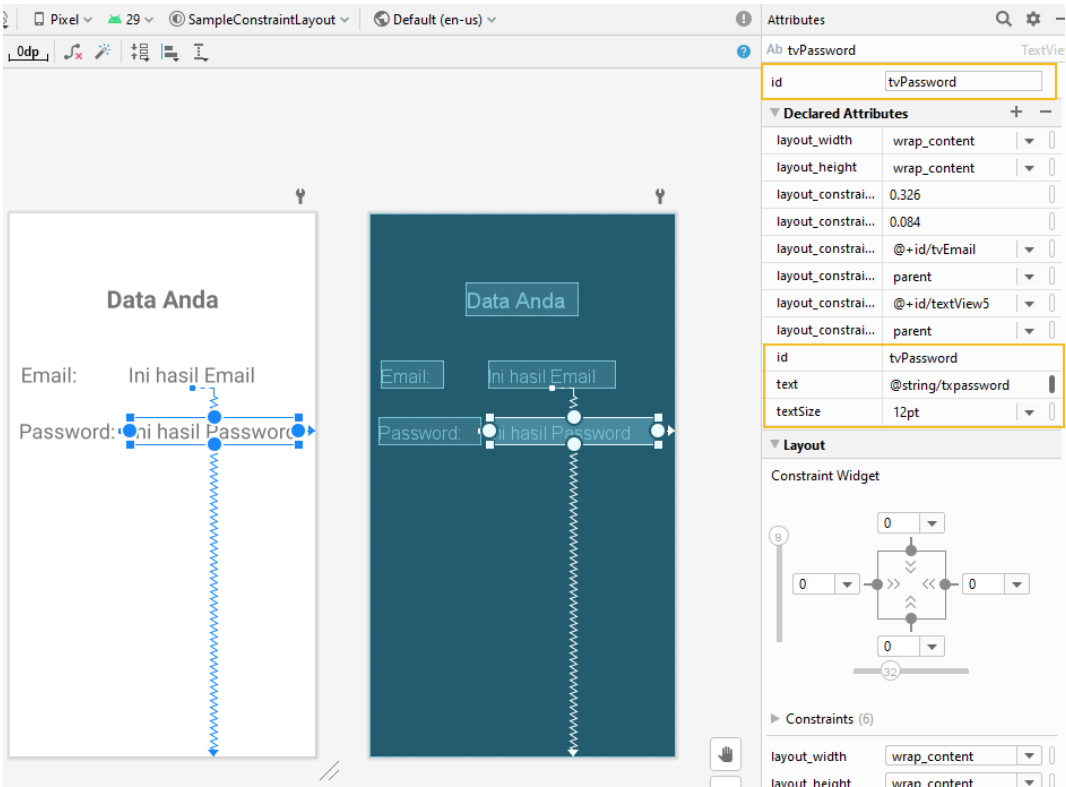
- Klik textview keempat → pada bagian atribut → cari textSize → ubah ukuran font menjadi “12pt” → kemudian pada bagian text → isi dengan string resources dengan nama variabel resources string “tPassword”. Sehingga tampilan atribut dari textview yang keempat akan seperti pada gambar dibawah.



Untuk code xml dari textview keempat akan menjadi seperti gambar dibawah.

```
<TextView
    android:id="@+id/textView5"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/tPassword"
    android:textSize="12pt"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.045"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/textView4"
    app:layout_constraintVertical_bias="0.084" />
```

- Klik textview kelima → pada bagian atribut → cari textSize → ubah ukuran font menjadi “12pt” → kemudian pada bagian text → isi dengan string resources dengan nama variabel resources string “txpassword” → cari bagian “id” → ubah id dari textview kelima menjadi “tvPassword”. Sehingga tampilan atribut dari textview yang kelima akan seperti pada gambar dibawah.



Untuk code xml dari textview kelima akan menjadi seperti gambar dibawah.

```
<TextView
    android:id="@+id/tvPassword"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/txpassword"
    android:textSize="12pt"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.326"
    app:layout_constraintStart_toEndOf="@+id/textView5"
    app:layout_constraintTop_toBottomOf="@+id/tvEmail"
    app:layout_constraintVertical_bias="0.084" />
```

- Langkah Ketiga

- Buka ActivityHasil.java → tambahkan sintak berikut
- Deklarasikan variabel dengan tipe data TextView yang akan digunakan untuk menghubungkan textview didalam java class dan textview didalam layout didalam class "ActivityKedua"

```
public class ActivityHasil extends AppCompatActivity {
    //Mendeklarasikan variabel dengan tipe data TextView
    TextView txEmail, txPassword;
```

@Override

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_hasil);
```

- Hubungkan textview yang ada didalam java class dengan textview yang ada pada layout dengan menggunakan method "findViewById". Sehingga sintak code menjadi seperti pada gambar dibawah.

@Override

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_hasil);

    //Menghubungkan variabel txEmail dengan componen Textview pada Layout
    txEmail = findViewById(R.id.txEmail);

    //Menghubungkan variabel txEmail dengan componen Textview pada Layout
    txPassword = findViewById(R.id.txPassword);
```

- Mendeklarasikan bundle yang akan digunakan untuk mengambil pesan atau nilai yang dikirimkan melalui intent dari activity pertama ke activity kedua. (penulisan method tersebut dilakukan didalam method onCreate)

```
//mendeklarasikan variabel bundle yang akan digunakan untuk mengambil
// pesan yang dikirimkan melalui method intent
Bundle bundle = getIntent().getExtras();
```

- Buat sintak yang berfungsi untuk mengambil nilai yang dikirimkan dengan menggunakan intent dari activity pertama ke activity kedua. Method yang digunakan adalah method “bundle.getString(“key”)” dan kemudian simpan nilai tersebut kedalam variabel dengan tipe data string. (penulisan method tersebut dilakukan didalam method onCreate)

```
//membuat variabel string yang digunakan untuk menyimpan data yang
// dikirimkan dari activity sebelumnya dengan kunci "a"
String email = bundle.getString( key: "a");
```

```
//membuat variabel string yang digunakan untuk menyimpan data yang
// dikirimkan dari activity sebelumnya dengan kunci "b"
String pass = bundle.getString( key: "b");
```

- Tampilkan nilai tersebut kedalam textview dengan menggunakan method “TextView.setText(value)”. (penulisan method tersebut dilakukan didalam method onCreate)

```
//menampilkan value dari variabel email kedalam txEmail
txEmail.setText(email);
```

```
//menampilkan value dari variabel pass kedalam txPassword
txPassword.setText(pass);
```

- Sehingga tampilan dari class ActivityHasil akan menjadi seperti pada gambar dibawah.

```
package com.example.sampleconstraintlayout;

import ...

public class ActivityHasil extends AppCompatActivity {
    //Mendeklarasikan variabel dengan tipe data TextView
    TextView txEmail, txPassword;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_hasil);

        //Menghubungkan variabel txEmail dengan componen Textview pada layout
        txEmail = findViewById(R.id.tvEmail);

        //Menghubungkan variabel txEmail dengan componen Textview pada layout
        txPassword = findViewById(R.id.tvPassword);

        //membuat objek bundle dan mengambil data yang dikirimkan dari activity sebelumnya
        Bundle bundle = getIntent().getExtras();

        //membuat variabel string yang digunakan untuk menyimpan data yang
        // dikirimkan dari activity sebelumnya dengan kunci "a"
        String email = bundle.getString( key: "a");

        //membuat variabel string yang digunakan untuk menyimpan data yang
        // dikirimkan dari activity sebelumnya dengan kunci "b"
        String pass = bundle.getString( key: "b");

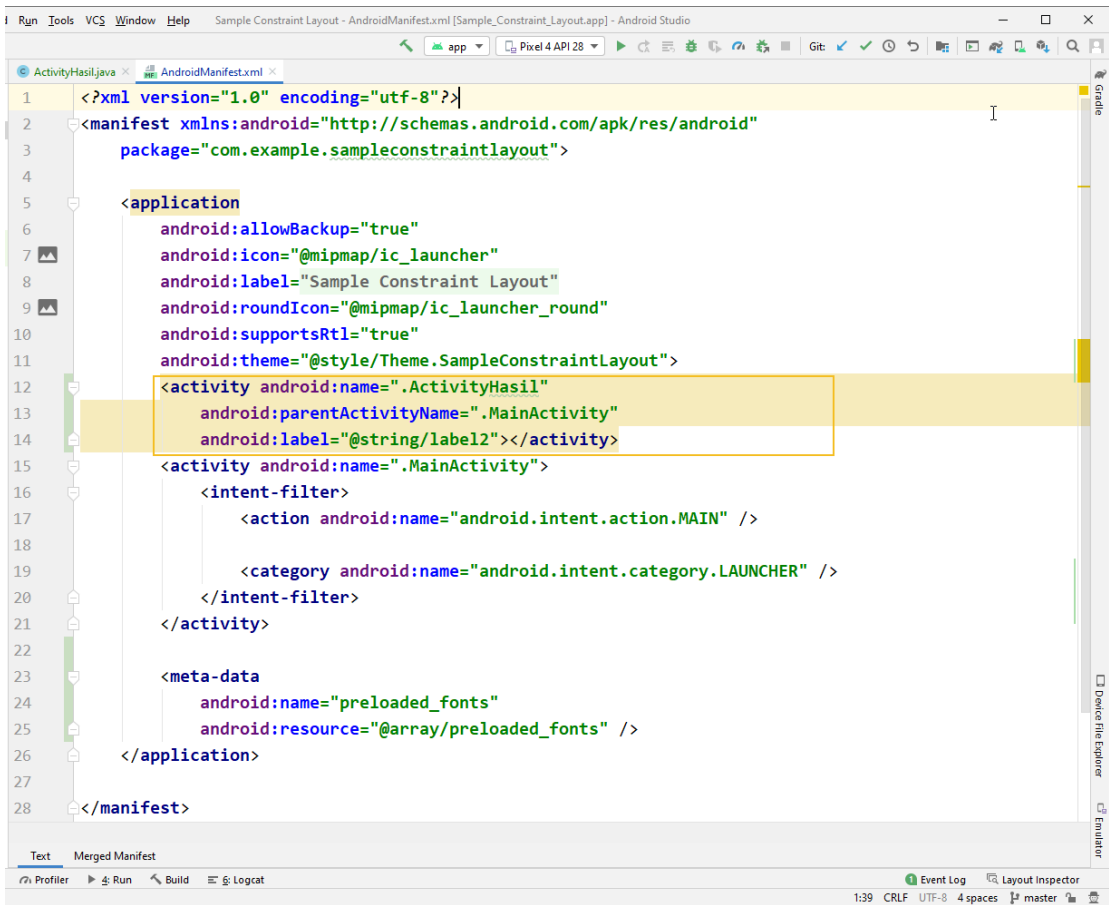
        //menampilkan value dari variabel email kedalam txEmail
        txEmail.setText(email);

        //menampilkan value dari variabel pass kedalam txPassword
        txPassword.setText(pass);
    }
}
```

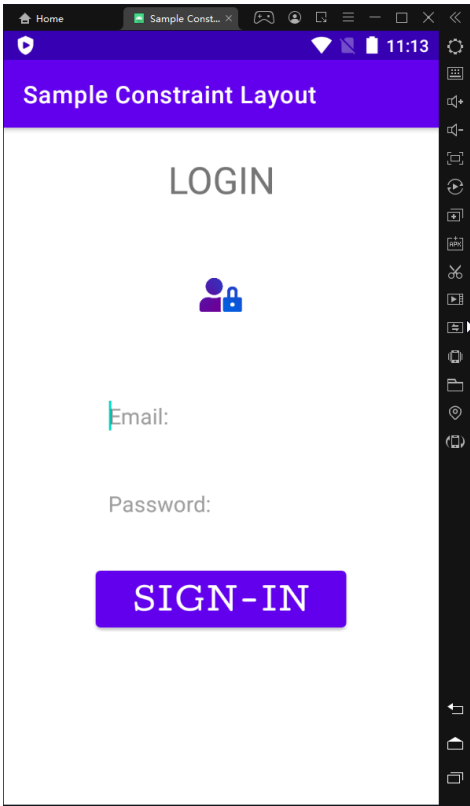
- Langkah Keempat
 - Buka AndroidManifest → tambahkan “android:parentactivityname” → sehingga menjadi seperti berikut

```
<activity android:name=".ActivityHasil"
    android:parentActivityName=".MainActivity"
    android:label="@string/label2"></activity>
```

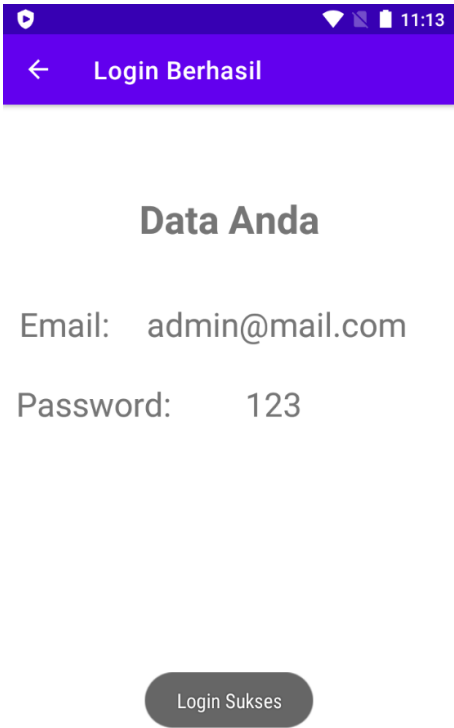

- Tampilan dari AndroidManifest.xml akan menjadi seperti gambar dibawah.



5. Jalankan aplikasi android tersebut melalui emulator atau smartphone android anda. Sehingga tampilannya akan menjadi seperti ini.



Gambar 1. Tampilan dari Activity 1



Gambar 2. Tampilan dari Activity 2

Activity 3

Buatlah sebuah aplikasi android yang berisi dua buah activity dan menggunakan intent dan passing parameter seperti pada modul diatas. Silahkan memilih salah satu dari cara untuk menambahkan activity baru. (Gunakan projek yang sudah dibuat sebelumnya)