
Modul Pemrograman Perangkat Bergerak

#009

Materi : Penambahan Widget melalui Java, Penggunaan Butter Knife dan Data Binding.

Penambahan Widget melalui Java (secara Programmatically)

Selain menambahkan Widget melalui XML kita juga dapat menambahkan Widget secara **Programmatically** melalui Java. Termasuk di dalamnya kita juga dapat melakukan pengaturan pada layout secara programmatically. Misalnya, kita dapat menambahkan Widget menggunakan method **addView** ataupun menghapus Widget menggunakan method **removeView**. Selain itu kita dapat mengatur layout menggunakan sub class **LayoutParams** dari class layout yang kita gunakan.

Link:

<https://developer.android.com/reference/android/view/ViewGroup.html>

<https://developer.android.com/reference/android/widget/LinearLayout.LayoutParams.html>

Butter Knife

Butter Knife adalah library yang dibuat oleh **Jake Wharton** untuk memudahkan pengaksesan Widget juga resources pada XML melalui Java (Field and method **binding**), salah satunya dengan mengurangi penulisan **findViewById** maupun **setOnClickListener** (diganti dengan **annotation**). Penjelasan lebih lengkap dapat dibaca pada link.

Link:

<http://jakewharton.github.io/butterknife/>

<https://github.com/JakeWharton/butterknife>

Data Binding

Data Binding adalah library yang dibuat oleh **Google** yang merupakan bagian dari **Android Support Library** yang bertujuan untuk memudahkan kita dalam menyatukan kode logika dari aplikasi kita dengan layout yang kita miliki (**bind** your application logic and layouts), salah satunya dengan cara memberikan kode khusus untuk data pada xml. Penjelasan lebih lengkap dapat dibaca pada link.

Link:

<https://developer.android.com/topic/libraries/data-binding/index.html>

Praktek Widget Lanjutan 1 (Add Widget Programmatically)

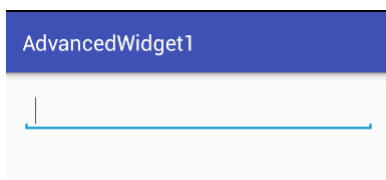
- 1) Buat Project Baru dengan Nama Aplikasi **AdvancedWidget1** dan Company Domain **learn.smktelkom-mlg.sch.id**, pilih template **Empty Activity** (yang lain default).
- 2) Aktifkan **Git**, add Project pada **Git**.
- 3) Buka **activity_main.xml** (res\layout\activity_main.xml).
- 4) Hapus **TextView** Hello World.
- 5) Rubah **RelativeLayout** menjadi **LinearLayout** vertikal.
- 6) Lakukan **commit** dengan message : **Initial commit**.
- 7) Kemudian beri id pada **LinearLayout**, yaitu **linearLayoutMain**.
- 8) Buka **MainActivity.java** (java\{nama package}\MainActivity).
- 9) Tambahkan deklarasi dan inisialisasi variable untuk **linearLayoutMain** di method **OnCreate** seperti di bawah ini.

```
LinearLayout llMain = (LinearLayout) findViewById(R.id.linearLayoutMain);
```

- 10) Selanjutnya kita akan menambahkan **EditText** untuk input nama anak ke dalam **linearLayoutMain** secara programmatically dengan menambahkan kode di method **OnCreate** seperti di bawah ini (buat EditText baru dan tambahkan ke layout dengan **addView**).

```
EditText etNama = new EditText(this);  
llMain.addView(etNama);
```

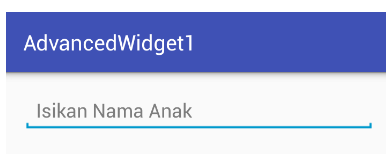
- 11) Jalankan Aplikasi pada Emulator/Device untuk melihat hasilnya.



- 12) Selanjutnya kita tambahkan **hint** "Isikan Nama Anak" pada EditText tersebut dengan menambahkan kode seperti di bawah ini.

```
etNama.setHint("Isikan Nama Anak");
```

- 13) Jalankan Aplikasi pada Emulator/Device untuk melihat hasilnya.



- 14) Selanjutnya kita akan menambahkan **EditText** untuk input umur anak ke dalam **linearLayoutMain** secara programmatically dengan menambahkan kode di method **OnCreate** dengan cara:
 - ❖ Copy kode EditText **etNama** yang sudah ditambahkan sebelumnya dan paste di bawahnya seperti di bawah ini.

```

EditText etNama = new EditText(this);
llMain.addView(etNama);
etNama.setHint("Isikan Nama Anak");

EditText etNama = new EditText(this);
llMain.addView(etNama);
etNama.setHint("Isikan Nama Anak");

```

- ❖ Klik pada tulisan **etNama** yang digaris merah (yang error karena nama variable sama dengan sebelumnya), dan tekan **Shift+F6** untuk melakukan **rename**.
- ❖ Ganti menjadi **etUmur** dan tekan Enter sehingga semua tulisan etNama yang ada di bawahnya juga otomatis terganti.

```

EditText etNama = new EditText(this);
llMain.addView(etNama);
etNama.setHint("Isikan Nama Anak");

EditText etUmur = new EditText(this);
llMain.addView(etUmur);
etUmur.setHint("Isikan Nama Anak");

```

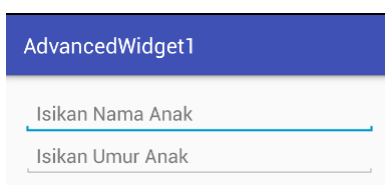
- 15) Kemudian rubah **hint** dan tambahkan kode untuk setting **tipe input** pada kode **etUmur** sehingga menjadi seperti di bawah ini.

```

EditText etUmur = new EditText(this);
llMain.addView(etUmur);
etUmur.setHint("Isikan Umur Anak");
etUmur.setInputType(InputType.TYPE_CLASS_NUMBER);

```

- 16) Jalankan Aplikasi pada Emulator/Device untuk melihat hasilnya.



- 17) Selanjutnya kita akan menambahkan **Button** dan **TextView** sehingga jika Button ditekan hasil isian akan tampil di TextView, dengan menambahkan kode di method **OnCreate** seperti di bawah ini (saat menambahkan kode pada method **onClick** di **setOnClickListener bProses** tulisan **final** akan otomatis ditambahkan pada deklarasi **etNama**, **etUmur**, dan **tvHasil**).

```

Button bProses = new Button(this);
bProses.setText("Proses");
llMain.addView(bProses);
final TextView tvHasil = new TextView(this);
llMain.addView(tvHasil);

```

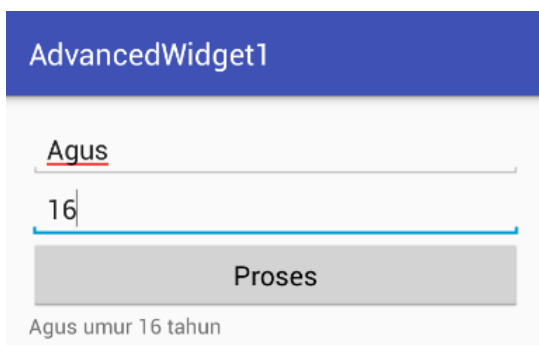
```

bProses.setOnClickListener(new View.OnClickListener()
{
    @Override
    public void onClick(View view)
    {
        String nama = etNama.getText().toString();
        String umur = etUmur.getText().toString();

        tvHasil.setText(nama + " umur " + umur + " tahun");
    }
});

```

- 18) Jalankan Aplikasi pada Emulator/Device untuk melihat hasilnya. Isi input dan klik pada Button untuk melakukan tes, pastikan hasil sesuai yang kita harapkan.



- 19) Lakukan **commit** dengan message: **Widget Programming Part 1**.
- 20) Selanjutnya untuk lebih mempermudah maintenance kode kita buat method baru **addEditText** pada class **MainActivity** (gunakan autocomplete) dan pindahkan kode EditText **etNama** dan **etUmur** ke sana, sehingga kode pada method **OnCreate** menjadi seperti di bawah ini.

```

LinearLayout llMain = (LinearLayout) findViewById(R.id.LinearLayoutMain);

addEditText(llMain);

Button bProses = new Button(this);

```

Dan kode pada method **addEditText** menjadi seperti di bawah ini.

```

private void addEditText(LinearLayout llMain)
{
    final EditText etNama = new EditText(this);
    llMain.addView(etNama);
    etNama.setHint("Isikan Nama Anak");

    final EditText etUmur = new EditText(this);
    llMain.addView(etUmur);
    etUmur.setHint("Isikan Umur Anak");
    etUmur.setInputType(InputType.TYPE_CLASS_NUMBER);
}

```

- 21) Dapat kita perhatikan bahwa kode pada method **onClick** menjadi **error**, sehingga perlu kita perbaiki menjadi seperti di bawah ini (tulisan **final** akan otomatis ditambahkan pada deklarasi **llMain**).

```
bProses.setOnClickListener(new View.OnClickListener()
{
    @Override
    public void onClick(View view)
    {
        EditText etNama = (EditText) llMain.getChildAt(0);
        EditText etUmur = (EditText) llMain.getChildAt(1);

        String nama = etNama.getText().toString();
        String umur = etUmur.getText().toString();

        tvHasil.setText(nama + " umur " + umur + " tahun");
    }
});
```

- 22) Jalankan Aplikasi pada Emulator/Device untuk melihat hasilnya. Isi input dan klik pada Button untuk melakukan tes, pastikan hasil tetap sesuai yang kita harapkan.
- 23) Selanjutnya kita akan membuat inputan untuk dapat diisi **5** orang **anak**, sehingga kita perlu menambahkan **4** lagi **EditText** masing-masing untuk **nama** dan **umur**. Untuk mempermudah melakukannya kita dapat menggunakan **perulangan** pada method **addEditText** sehingga kodenya menjadi seperti di bawah ini.

```
private void addEditText(LinearLayout llMain)
{
    for (int i = 1; i <= 5; i++)
    {
        EditText etNama = new EditText(this);
        llMain.addView(etNama);
        etNama.setHint("Isikan Nama Anak ke-" + i);

        EditText etUmur = new EditText(this);
        llMain.addView(etUmur);
        etUmur.setHint("Isikan Umur Anak ke-" + i);
        etUmur.setInputType(InputType.TYPE_CLASS_NUMBER);
    }
}
```

- 24) Jalankan Aplikasi pada Emulator/Device untuk melihat hasilnya. Bisa dilihat bahwa sekarang kita sudah memiliki inputan untuk 5 orang anak. Tetapi jika kita klik Button Proses, kita masih menampilkan data 1 orang saja, maka selanjutnya kita perlu merubah agar kita bisa menampilkan data 5 orang.

AdvancedWidget1

Isikan Nama Anak ke-1

Isikan Umur Anak ke-1

Isikan Nama Anak ke-2

Isikan Umur Anak ke-2

Isikan Nama Anak ke-3

Isikan Umur Anak ke-3

Isikan Nama Anak ke-4

Isikan Umur Anak ke-4

Isikan Nama Anak ke-5

Isikan Umur Anak ke-5

Proses

- 25) Untuk menampilkan data 5 orang , terlebih dahulu kita perlu membuat method baru **doProses** pada class **MainActivity** (gunakan autocomplete) dan memindahkan kode pada method **onClick** pada method tersebut untuk memudahkan maintenance sehingga kode pada method **onClick** menjadi seperti di bawah ini.

```
bProses.setOnClickListener(new View.OnClickListener()
{
    @Override
    public void onClick(View view)
    {
        doProses(llMain);
    }
});
```

Dan kode pada method **doProses** seperti di bawah ini.

```
private void doProses(LinearLayout llMain)
{
    EditText etNama = (EditText) llMain.getChildAt(0);
    EditText etUmur = (EditText) llMain.getChildAt(1);

    String nama = etNama.getText().toString();
    String umur = etUmur.getText().toString();

    tvHasil.setText(nama + " umur " + umur + " tahun");
}
```

- 26) Bisa dilihat bahwa pada method **doProses** masih terdapat **error**. Selain itu kita juga perlu menyesuaikan agar dapat menampilkan data **5** orang **anak** yang diinputkan, juga hanya menampilkan data anak jika nama diisi, dan jika umur tidak diisi maka otomatis ditampilkan 0, maka kode perlu kita rubah menjadi seperti di bawah ini.

```

private void doProses(LinearLayout llMain)
{
    String hasil="";
    for (int i = 0; i < 5; i++)
    {
        EditText etNama = (EditText) llMain.getChildAt(i*2);
        EditText etUmur = (EditText) llMain.getChildAt((i*2)+1);

        String nama = etNama.getText().toString().trim();
        String umur = etUmur.getText().toString();

        if(umur.isEmpty())
            umur="0";
        if(!nama.isEmpty())
            hasil += "Anak ke-"+(i+1)+": " + nama + " umur "+umur+" tahun\n";
    }

    TextView tvHasil = (TextView) llMain.getChildAt(11);
    tvHasil.setText(hasil);
}

```

27) Jalankan Aplikasi pada Emulator/Device untuk melihat hasilnya. Isi input dan klik pada Button untuk melakukan tes, pastikan hasil sesuai yang kita harapkan.

AdvancedWidget1

Agus

20

Rudi

18

Ucok

17

Bagus

12

Alice

10

Proses

Anak ke-1: Agus umur 20 tahun
 Anak ke-2: Rudi umur 18 tahun
 Anak ke-3: Ucok umur 17 tahun
 Anak ke-4: Bagus umur 12 tahun
 Anak ke-5: Alice umur 10 tahun

AdvancedWidget1

Agus

20

Isikan Nama Anak ke-2

Isikan Umur Anak ke-2

Ucok

17

Isikan Nama Anak ke-4

12

Alice

Isikan Umur Anak ke-5

Proses

Anak ke-1: Agus umur 20 tahun
 Anak ke-3: Ucok umur 17 tahun
 Anak ke-5: Alice umur 0 tahun

28) Lakukan **commit** dengan message: **Widget Programming Part 2.**

29) Selanjutnya kita akan merubah jumlah isian nama dan umur yang ditampilkan berdasarkan pilihan user melalui **Spinner** juga melakukan perombakan terhadap tampilan, dan melakukan penambahan Widget secara programmatically hanya untuk isian nama dan umur saja. Untuk itu kita perlu mengembalikan kondisi Project ke kondisi awal yaitu pada **Initial commit**.

Untuk melakukannya, klik kanan pada file **MainActivity.java** kemudian pilih **Git → Compare with...** pilih commit yang **Initial commit** dan kembalikan kode ke kondisi **Initial commit** (lihat Modul 4 jika lupa caranya).

Lakukan juga pada file **activity_main.xml** dengan cara yang sama.

- 30) Kemudian tambahkan 1 **ScrollView**, 1 **Spinner**, 1 **LinearLayout vertical**, 1 **Button**, dan 1 **TextView** pada **activity_main.xml**, lalu rubah kodenya menjadi seperti di bawah ini.

```
<?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:layout_width="match_parent"
    android:layout_height="match_parent">
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:paddingBottom="@dimen/activity_vertical_margin"
        android:paddingLeft="@dimen/activity_horizontal_margin"
        android:paddingRight="@dimen/activity_horizontal_margin"
        android:paddingTop="@dimen/activity_vertical_margin"
        android:orientation="vertical"
        tools:context="id.sch.smktelkom_mlg.learn.advancedwidget1.MainActivity">

        <Spinner
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:id="@+id/spinnerJumlahAnak"/>

        <LinearLayout
            android:id="@+id/linearLayoutAnak"
            android:orientation="vertical"
            android:layout_width="match_parent"
            android:layout_height="wrap_content">

            </LinearLayout>

            <Button
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:text="Proses"
                android:id="@+id/buttonProses"/>

            <TextView
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                tools:text="Hasil"
                android:id="@+id/textViewHasil"/>

        </LinearLayout>
    </ScrollView>
```


- 31) Kemudian buat layout baru dengan nama **layout_anak** dan rubah isinya menjadi seperti di bawah ini. Jangan lupa menambahkan dependencies **Android Support Design**.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:weightSum="1"
    android:padding="4dp"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <android.support.design.widget.TextInputLayout
        android:layout_weight="0.8"
        android:layout_width="0dp"
        android:layout_height="wrap_content">

        <android.support.design.widget.TextInputEditText
            android:id="@+id/editTextNama"
            android:hint="Nama Anak"
            android:layout_width="match_parent"
            android:layout_height="match_parent"/>

    </android.support.design.widget.TextInputLayout>

    <android.support.design.widget.TextInputLayout
        android:layout_weight="0.2"
        android:layout_width="0dp"
        android:layout_height="wrap_content">

        <android.support.design.widget.TextInputEditText
            android:id="@+id/editTextUmur"
            android:hint="Umur"
            android:maxLength="2"
            android:layout_width="match_parent"
            android:layout_height="match_parent"/>

    </android.support.design.widget.TextInputLayout>

</LinearLayout>
```

- 32) Buka **MainActivity.java** dan rubahlah menjadi seperti di bawah ini.

```
public class MainActivity extends AppCompatActivity
{
    Spinner spJumlah;
    LinearLayout llAnak;
    TextView tvHasil;

    @Override
    protected void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
```

```

        setContentView(R.layout.activity_main);

        spJumlah = (Spinner) findViewById(R.id.spinnerJumlahAnak);
        Integer [] arJumlah = new Integer[10];
        for (int i = 0; i < 10; i++)
        {
            arJumlah[i]=i+1;
        }
        ArrayAdapter adapter = new ArrayAdapter<>(this,
            android.R.layout.simple_spinner_item, arJumlah);
        adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
        spJumlah.setAdapter(adapter);

        llAnak = (LinearLayout) findViewById(R.id.LinearLayoutAnak);
        tvHasil = (TextView) findViewById(R.id.textViewHasil);

        findViewById(R.id.buttonProses).setOnClickListener(new View.OnClickListener()
        {
            @Override
            public void onClick(View view)
            {
                doProses();
            }
        });

        spJumlah.setOnItemClickListener(new AdapterView.OnItemClickListener()
        {
            @Override
            public void onItemClick(AdapterView<?> adapterView, View view, int i, long l)
            {
                addEditText((int)spJumlah.getSelectedItem());
            }

            @Override
            public void onNothingSelected(AdapterView<?> adapterView)
            {
                //
            }
        });

        private void addEditText(int jumlah)
        {

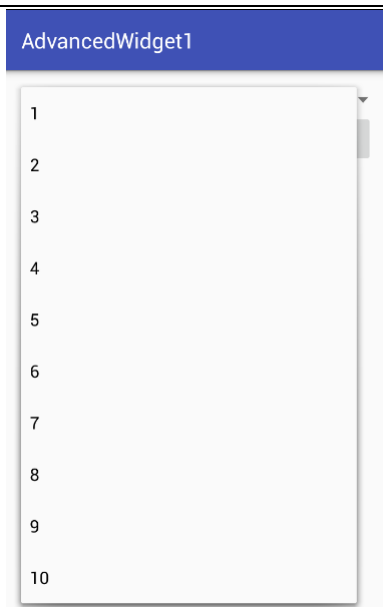
        }

        private void doProses()
        {

        }
    }

```

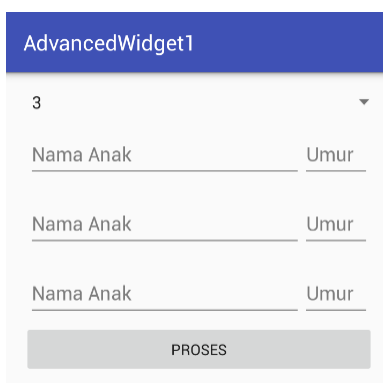
33) Jalankan Aplikasi pada Emulator/Device untuk melihat hasilnya. Klik pada Spinner untuk memastikan Spinner telah terisi.



34) Kemudian isi kode pada method **addEditText** menjadi seperti di bawah ini.

```
private void addEditText(int jumlah)
{
    llAnak.removeAllViews();
    for (int i = 1; i <= jumlah; i++)
    {
        View v = LayoutInflater.from(this).inflate(R.layout.layout_anak, llAnak, false);
        v.setTag("Anak"+i);
        llAnak.addView(v);
    }
}
```

35) Jalankan Aplikasi pada Emulator/Device untuk melihat hasilnya. Klik pada Spinner dan pilih jumlah yang lain untuk memastikan jumlah isian sesuai dengan pilihan pada Spinner.



36) Selanjutnya untuk mengfungsikan tombol agar menampilkan hasil input, isi kode pada method **doProses** menjadi seperti di bawah ini.

```

private void doProses()
{
    int jumlah = (int) spJumlah.getSelectedItem();
    String hasil = "";
    for (int i = 1; i <= jumlah; i++)
    {
        LinearLayout llNow = (LinearLayout) llAnak.findViewById("Anak"+i);

        EditText etNama = (EditText) llNow.findViewById(R.id.editTextNama);
        EditText etUmur = (EditText) llNow.findViewById(R.id.editTextUmur);

        String nama = etNama.getText().toString().trim();
        String umur = etUmur.getText().toString();

        if(umur.isEmpty())
            umur="0";
        if(!nama.isEmpty())
            hasil += "Anak ke-" + i + ": " + nama + " umur " + umur + " tahun\n";
    }

    tvHasil.setText(hasil);
}

```

- 37) Jalankan Aplikasi pada Emulator/Device untuk melihat hasilnya. Lakukan pengujian dengan klik pada Spinner dan pilih jumlah yang lain, isi input yang ada, lalu klik pada tombol Proses, pastikan hasilnya sesuai.

AdvancedWidget1

5

Nama Anak	Umur
Agus	20
Rudi	18
Ucok	16
Bagus	12
Alice	9

PROSES

Anak ke-1: Agus umur 20 tahun
 Anak ke-2: Rudi umur 18 tahun
 Anak ke-3: Ucok umur 16 tahun
 Anak ke-4: Bagus umur 12 tahun
 Anak ke-5: Alice umur 9 tahun

AdvancedWidget1

5

Nama Anak	Umur
Agus	20
Ucok	16
Bagus	12
Alice	0

PROSES

Anak ke-1: Agus umur 20 tahun
 Anak ke-3: Ucok umur 16 tahun
 Anak ke-5: Alice umur 0 tahun

- 38) Lakukan **commit** dengan message: **Widget Programming Final Part.**

Praktek Widget Lanjutan 2 (Butter Knife)

- 1) Buat Project Baru dengan Nama Aplikasi **AdvancedWidget2** dan Company Domain **learn.smktelkom-mlg.sch.id**, pilih template **Empty Activity** (yang lain default).
- 2) Buka **activity_main.xml** (res\layout\activity_main.xml).
- 3) Rubah **RelativeLayout** menjadi **LinearLayout** vertikal.
- 4) Lakukan **commit** dengan message : **Initial commit**.
- 5) Tambahkan 1 **EditText**, 1 **Button**, dan 1 **TextView**, kemudian rubah dan atur sehingga kodenya seperti di bawah ini.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:paddingBottom="16dp"
    android:paddingLeft="16dp"
    android:paddingRight="16dp"
    android:paddingTop="16dp"
    tools:context="id.sch.smktelkom_mlg.learn.advancedwidget2.MainActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Nama"/>

    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/editTextNama"/>

    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Proses"
        android:id="@+id/buttonProses"/>

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        tools:text="Hasil"
        android:id="@+id/textViewHasil"/>

</LinearLayout>
```

- 6) Selanjutnya kita akan menggunakan library **Butter Knife** untuk membantu kita mengakses widget yang ada pada XML melalui Java. Terlebih dahulu kita perlu menambahkan **dependencies Butter Knife** pada Project kita, dengan cara :

- ❖ Buka **build.gradle** yang **Project** dan tambahkan kode classpath pada bagian **dependencies** seperti di bawah ini.

```
dependencies {  
    classpath 'com.android.tools.build:gradle:2.1.3'  
    classpath 'com.neenbedankt.gradle.plugins:android-apt:1.8'  
    // NOTE: Do not place your application dependencies here; they belong  
    // in the individual module build.gradle files  
}
```

- ❖ Kemudian buka **build.gradle** yang **app** dan tambahkan kode **apply plugin** pada bagian atas seperti di bawah ini.

```
apply plugin: 'com.android.application'  
apply plugin: 'android-apt'
```

Dan kode **compile** juga **apt** Butter Knife pada bagian **dependencies** seperti di bawah ini.

```
dependencies {  
    compile fileTree(dir: 'libs', include: ['*.jar'])  
    testCompile 'junit:junit:4.12'  
    compile 'com.android.support:appcompat-v7:24.2.1'  
    compile 'com.jakewharton:butterknife:8.4.0'  
    apt 'com.jakewharton:butterknife-compiler:8.4.0'  
}
```

- ❖ Lakukan **Sync Now** (perlu koneksi internet) dan tunggu hingga proses selesai.

- 7) Buka **MainActivity.java** dan tambahkan kode seperti di bawah ini.

```
public class MainActivity extends AppCompatActivity  
{  
    @BindView(R.id.editTextNama)  
    EditText etNama;  
    @BindView(R.id.textViewHasil)  
    TextView tvHasil;  
    @BindColor(R.color.colorAccent)  
    int colorAccent;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState)  
    {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
  
        ButterKnife.bind(this);  
  
        tvHasil.setTextColor(colorAccent);  
    }  
}
```

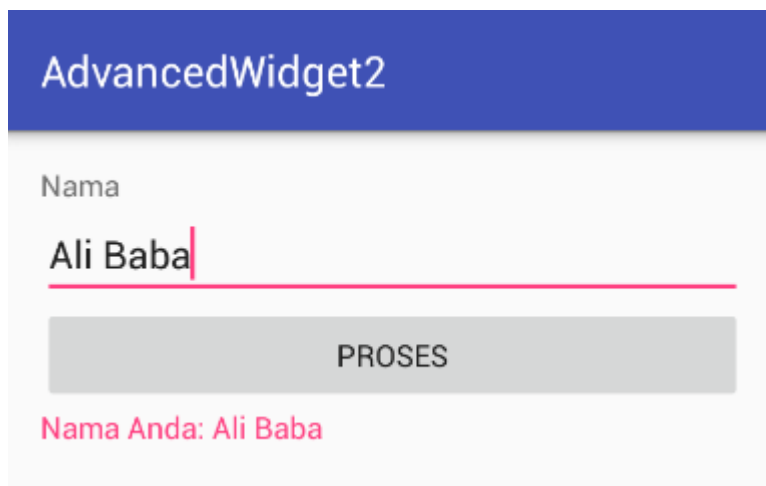
```

    @OnClick(R.id.buttonProses)
    public void doProcess()
    {
        tvHasil.setText("Nama Anda: "+etNama.getText());
    }
}

```

Dapat dilihat di atas kita menggunakan **annotation @BindView** mengakses Widget, **@BindColor** untuk mengakses resources color, dan **@OnClick** untuk membuat tombol dapat diklik. Yang tidak boleh dilupakan adalah kita perlu menambahkan kode **ButterKnife.bind(this)** ataupun yang sejenisnya agar annotation Butter Knife dapat bekerja sesuai yang seharusnya.

- 8) Jalankan Aplikasi pada Emulator/Device untuk melihat hasilnya. Lakukan pengetesan dengan mengisi input dan klik pada tombol Proses, pastikan tombol dapat bekerja dan hasilnya sesuai.



- 9) Lakukan **commit** dengan message: **Basic Butter Knife Final**.

Praktek Widget Lanjutan 3 (Data Binding)

- 1) Buat Project Baru dengan Nama Aplikasi **AdvancedWidget3** dan Company Domain **learn.smktelkom-mlg.sch.id**, pilih template **Empty Activity** (yang lain default).
- 2) Buka **activity_main.xml** (res/layout/activity_main.xml).
- 3) Rubah **RelativeLayout** menjadi **LinearLayout** vertikal.
- 4) Lakukan **commit** dengan message : **Initial commit**.
- 5) Tambahkan 1 **EditText** dan 1 **TextView**, kemudian rubah dan atur sehingga kodenya seperti di bawah ini.

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"

```

```

        android:orientation="vertical"
        android:paddingBottom="16dp"
        android:paddingLeft="16dp"
        android:paddingRight="16dp"
        android:paddingTop="16dp"
        tools:context="id.sch.smktelkom_mlg.learn.advancedwidget3.MainActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Nama"/>

    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/editTextNama"/>

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        tools:text="Hasil"
        android:id="@+id/textViewHasil"/>

</LinearLayout>

```

- 6) Untuk memulai menggunakan **Data Binding**, terlebih dahulu kita perlu mengaktifkan Data Binding terlebih dahulu, dengan cara:

- ❖ Buka **build.gradle** yang **app** dan tambahkan kode untuk mengaktifkan Data Binding pada bagian android seperti di bawah ini.

```

android {
    compileSdkVersion 24
    buildToolsVersion "24.0.2"

    dataBinding {
        enabled = true
    }
}

```

- ❖ Lakukan **Sync Now** dan tunggu hingga proses selesai.

- 7) Selanjutnya kita tambahkan element **data** pada **activity_main.xml** sehingga kita perlu rubah kode yang ada menjadi seperti di bawah ini.

```

<?xml version="1.0" encoding="utf-8"?>
<layout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools">

    <data>
        <variable name="nama" type="String"/>
    </data>

```

```

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical"
        android:paddingBottom="16dp"
        android:paddingLeft="16dp"
        android:paddingRight="16dp"
        android:paddingTop="16dp"
        tools:context="id.sch.smktelkom_mlg.learn.advancedwidget3.MainActivity">

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Nama"/>

        <EditText
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="@{nama}"
            android:id="@+id/editTextNama"/>

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@{nama==null||nama.equals("") ? "" : "Nama Anda: "+nama}"/>

    </LinearLayout>
</layout>

```

- 8) Kemudian kita perlu menambahkan kode pada **MainActivity.java** sehingga menjadi seperti di bawah ini.

```

public class MainActivity extends AppCompatActivity
{
    String nama;
    ActivityMainBinding binding;

    @Override
    protected void onCreate(Bundle savedInstanceState)
    {
        {
            super.onCreate(savedInstanceState);
            binding = DataBindingUtil.setContentView(this, R.layout.activity_main);
            binding.setNama(nama);
        }
    }
}

```

- 9) Jalankan Aplikasi pada Emulator/Device untuk melihat hasilnya. Lakukan pengetesan dengan mengisi input, ketika input diisi tulisan pada hasil juga otomatis langsung berubah, dan jika input kosong, maka tulisan tidak akan tampil. Pastikan hasilnya sesuai.

AdvancedWidget3	AdvancedWidget3
Nama <input type="text"/>	Nama <input type="text" value="Ali Rudiansa"/> Nama Anda: Ali Rudiansa

10) Lakukan **commit** dengan message: **Simple Data Binding Final**.

----- Selamat Mengerjakan -----