

# MOBILE PROGRAMMING



Oleh :

CHALIFA CHAZAR



## MATERI 4

### CONDITIONAL STATEMENT

**Tujuan:**

Mahasiswa memahami dan mampu menggunakan struktur kondisi **IF..THEN..ELSE..** untuk mendukung pembuatan suatu aplikasi Android.

**Pustaka:**

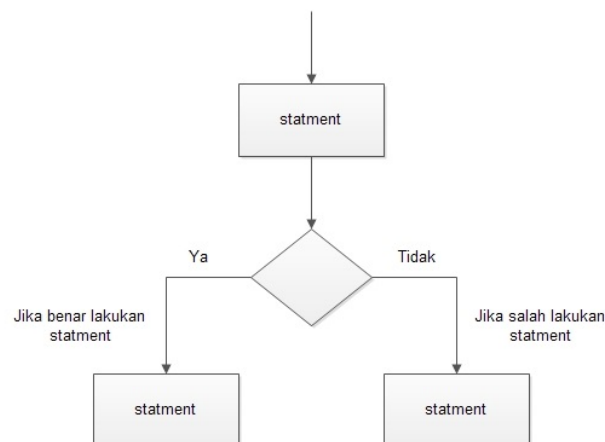
Abdul Kadir, 2013: Pemograman Aplikasi Android. Andi Yogyakarta.

[Http://www.gits.co.id](http://www.gits.co.id)

Dalam membuat suatu aplikasi yang memiliki suatu sifat pengambilan keputusan, digunakan Statement kondisi. Kondisi diilustrasikan dengan menggunakan ekspresi **Boolean**. Tipe data boolean berisi nilai **TRUE** dan **FALSE**.

#### 1. Statement **IF..THEN..ELSE..**

Secara singkat Statement **IF..THEN..ELSE..** dapat diartikan sebagai apa yang akan terjadi jika suatu kondisi terpenuhi dan apa yang akan terjadi bila suatu kondisi tidak terpenuhi. Apabila sebuah Statement kondisi digambarkan dengan menggunakan flowchart adalah sebagai berikut.





Bentuk penulisan **IF..THEN..ELSE..** dalam Android adalah sebagai berikut.

## Statement If..

```
If (Condition)
{
Statement condition = True
}
```

## Statement If..Else..

```
If (Condition)
{
Statement condition = True
} Else
{
Statement condition = False
}
```

## Statement If..Else If..Else

```
If (Condition 1)
{
Statement condition = True
} Else If (Condition 2)
{
Statement condition = True
} Else If (Condition 3)
{
Statement condition = True
} Else
{
Statement condition = False
}
```

## Statement Nested If..Else..

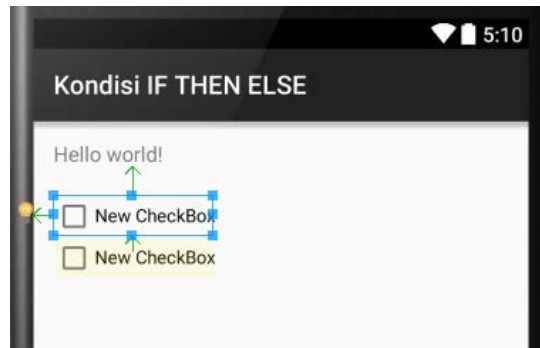
```
If (Condition 1)
{ Statement condition 1 = True
  If (Condition 2)
  { Statement condition 2 = True
  }
}
```

## 2. Membuat Statement IF..THEN..ELSE.. Pada Aplikasi Android

Berikut ini kita akan membuat sebuah aplikasi Android untuk merubah format sebuah Text menjadi **cetak tebal (bold)** atau **cetak miring (italic)** dengan menggunakan **CheckList**. Lakukan langkah-langkah sebagai berikut.



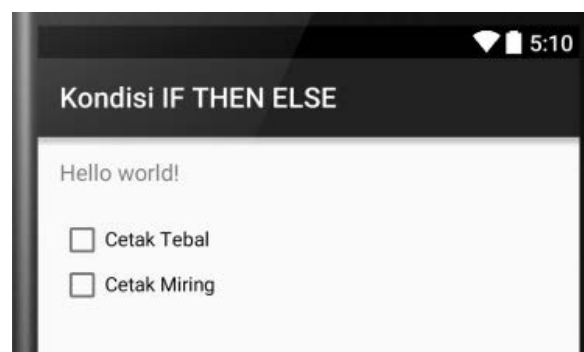
1. Pilih menu **File** dan kemudian klik **Close Project**, untuk menutup semua file project yang terbuka.
2. Buat sebuah project Android baru.
3. Beri nama project pada kolom Application Name.
4. Klik tombol **Next**. Langkah ini dilakukan sebanyak tiga kali.
5. Klik tombol **Finish**.
6. Lakukan sedikit perubahan pada tampilan project dengan menambahkan 2 buah **CheckBox** sehingga menghasilkan tampilan seperti berikut:



7. Melalui panel properties, lakukan perubahan **Id** dan **Text** pada **TextView** dan **CheckBox** seperti berikut ini:

Komponen	Property	Value Baru
textView	id	textViewLatihan
checkBox	id	checkBoxTebal
	text	Cetak Tebal
checkBox2	id	checkBoxMiring
	text	Cetak Miring

8. Berikut ini hasil akhir dari desain layar.



Selanjutnya kita akan mulai menuliskan kode pada `ActivityMain.java`. Terlebih dahulu buka file `MainActivity.java`. Lakukan perubahan sehingga isinya menjadi sebagai berikut ini:



```
package example.com.latihan4;

import android.graphics.Typeface;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;
import android.widget.CheckBox;
import android.widget.CompoundButton;
import android.widget.TextView;

public class MainActivity extends AppCompatActivity {

    boolean miring = false;
    boolean tebal = false;

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

        CheckBox kotakCekTebal = (CheckBox) findViewById(R.id.checkBoxTebal);
        CheckBox kotakCekMiring = (CheckBox) findViewById(R.id.checkBoxMiring);
        final TextView tampilanTextView = (TextView)
findViewById(R.id.textViewLatihan); // Nilai TextView adalah final, tidak dapat
diubah

        kotakCekMiring.setOnCheckedChangeListener(
            new CompoundButton.OnCheckedChangeListener() {
                @Override
                public void onCheckedChanged(CompoundButton buttonView, boolean
isChecked) {
                    miring = isChecked;

                    if (miring==true && tebal==true)
                    {
                        tampilanTextView.setTypeface(null, Typeface.BOLD_ITALIC);
                    } else if (miring==true && tebal==false)
                    {
                        tampilanTextView.setTypeface(null, Typeface.ITALIC);
                    } else if (miring==false && tebal==true)
                    {
                        tampilanTextView.setTypeface(null, Typeface.BOLD);
                    } else
                    {
                        tampilanTextView.setTypeface(null, Typeface.NORMAL);
                    }
                }
            });

        kotakCekTebal.setOnCheckedChangeListener(
            new CompoundButton.OnCheckedChangeListener() {
                @Override
                public void onCheckedChanged(CompoundButton buttonView, boolean
isChecked) {
                    tebal = isChecked;

                    if (miring==true && tebal==true)
                    {
                        tampilanTextView.setTypeface(null, Typeface.BOLD_ITALIC);
                    } else if (miring==true && tebal==false)
                    {
                        tampilanTextView.setTypeface(null, Typeface.ITALIC);
                    }
                }
            });
    }
}
```



```
        } else if (miring==false && tebal==true)
        {
            tampilanTextView.setTypeface(null, Typeface.BOLD);
        } else
        {
            tampilanTextView.setTypeface(null, Typeface.NORMAL);
        }
    }
});
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar if it is present.
    getMenuInflater().inflate(R.menu.menu_main, menu);
    return true;
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    // Handle action bar item clicks here. The action bar will
    // automatically handle clicks on the Home/Up button, so long
    // as you specify a parent activity in AndroidManifest.xml.
    int id = item.getItemId();

    //noinspection SimplifiableIfStatement
    if (id == R.id.action_settings) {
        return true;
    }

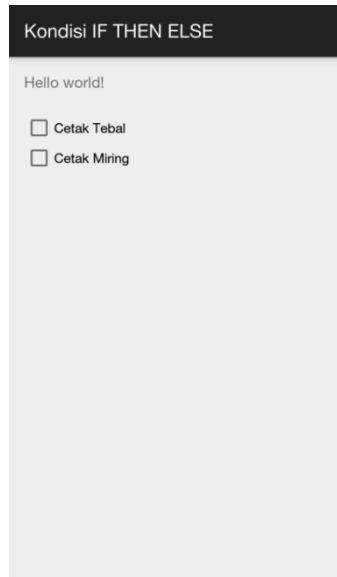
    return super.onOptionsItemSelected(item);
}
}
```

## Catatan!

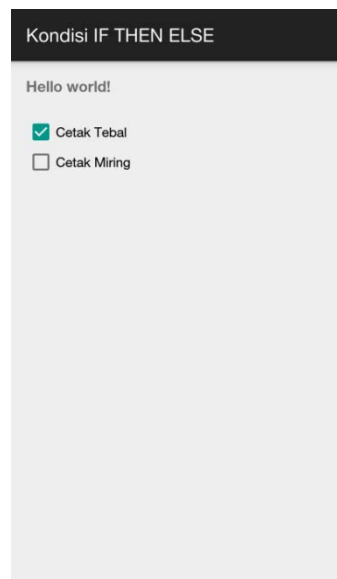
CompoundButton adalah direct sub class dari CheckBox, RadioButton, Switch, SwitchCompat, ToggleButton. Merupakan bentuk tombol yang memiliki 2 buah kondisi, **checked** dan **unchecked**.



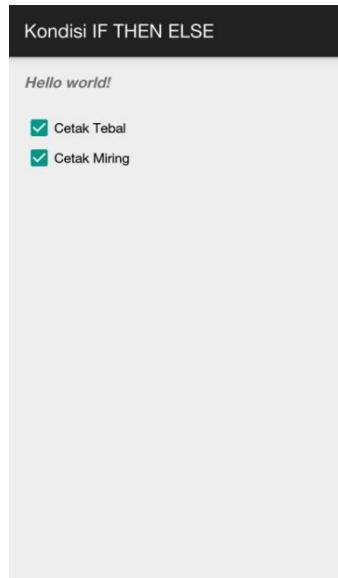
Berikut ini adalah tampilan keadaan awal ketika aplikasi dijalankan.



Apabila kotak cek miring dalam keadaan checked, maka berikut ini adalah tampilan aplikasi.



Dan apabila kotak cek miring dan kotak cek tebal dalam keadaan checked, maka berikut ini adalah tampilan aplikasi.



### 3. Statement SWITCH

Statement **SWITCH** memungkinkan suatu variable untuk diuji terhadap beberapa list nilai. Bentuk penulisan **SWITCH** dalam Android adalah sebagai berikut.

```
Statement SWITCH

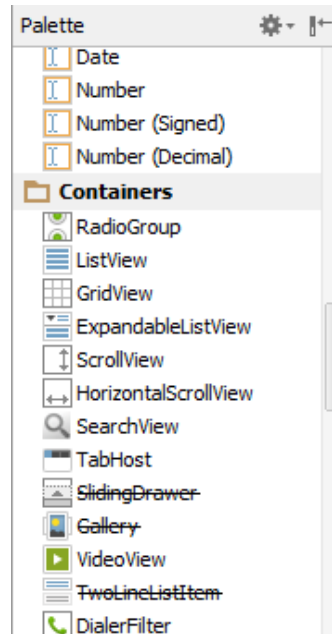
Switch (expresion) {
Case value :
//statement
Break; //optional
Case value :
//statement
Break; //optional
Default : //optional
//statement
}
```

### 4. Membuat Statement Switch Pada Aplikasi Android

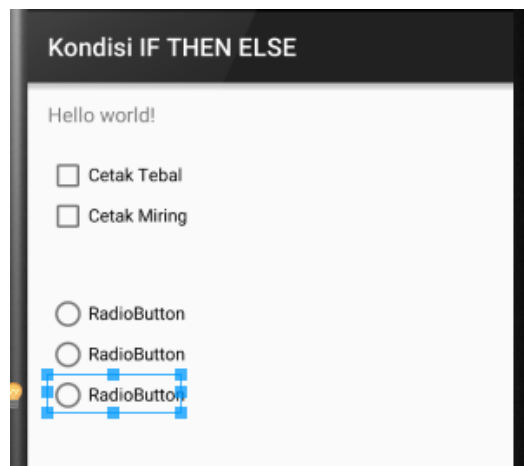
Pada latihan aplikasi sebelumnya, kita akan menambahkan sebuah **RadioGroup**, yang akan memiliki fungsi untuk mengatur penempatan posisi Text. Lakukan langkah-langkah sebagai berikut.

1. Buka project Android pada latihan sebelumnya.
2. Pada panel **Containers**, tambahkan **RadioGroup** ke dalam layar desain.





- Selanjutnya tambahkan 3 buah **RadioButton** ke dalam layar desain. sehingga menghasilkan tampilan seperti berikut:

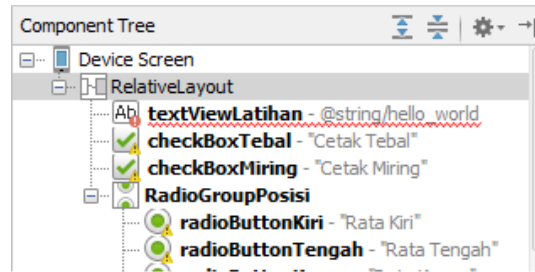


- Melalui panel **Properties**, lakukan perubahan **Id** dan **Text** pada **RadioGroup** dan **RadioButton** seperti berikut ini:

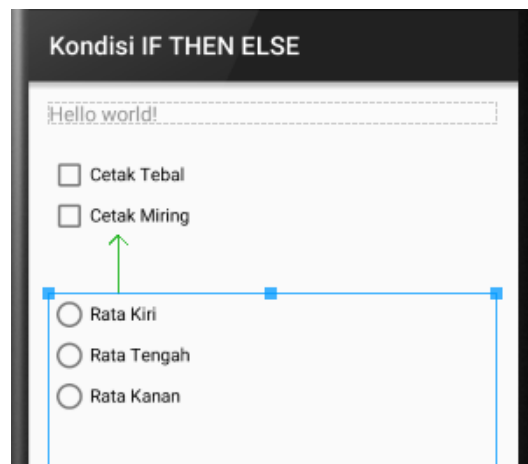
Komponen	Property	Value Baru
RadioGroup	id	RadioGroupPosisi
radioButton	id	radioButtonKiri
	text	Rata Kiri
radioButton2	id	radioButtonTengah
	text	Rata tengah
radioButton3	id	radioButtonKanan
	text	Rata Kanana



5. Berikut ini hasil panel Structure.



6. Berikut ini hasil akhir dari desain layar.

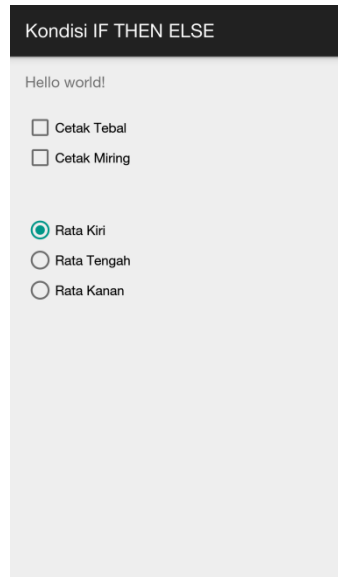


## Latihan!

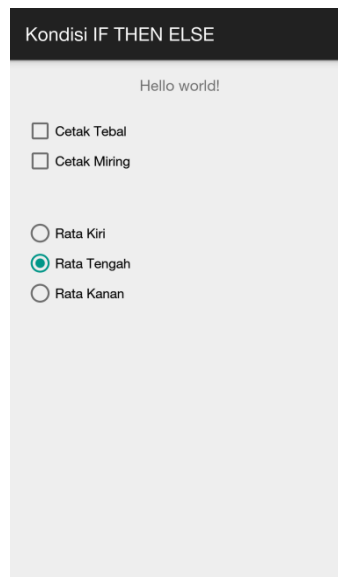
Buatlah fungsi dengan menggunakan statement **SWITCH** pada **RadioBox** untuk mengatur penempatan posisi Text, sehingga ketika **RadioBox Rata Kiri** di-klik maka Text akan berada di posisi **kiri** layar, jika **Rata Tengah** maka Text akan berada di posisi **tengah** layar, dan jika **Rata Kanan** maka Text akan berada di posisi **kanan** layar.



Berikut ini adalah tampilan keadaan awal ketika aplikasi dijalankan.

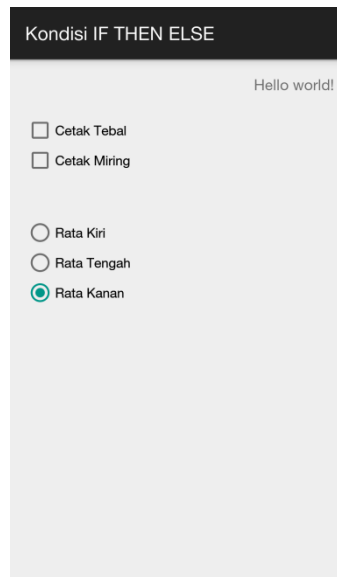


Apabila radio button **Rata Tengah** di-klik atau dalam keadaan checked, maka berikut ini adalah tampilan aplikasi.





Dan apabila radio button **Rata Kanan** di-klik atau dalam keadaan checked, maka berikut ini adalah tampilan aplikasi.



## 5. Conditional Operators

Java menyediakan kumpulan operator untuk memanipulasi data. Beberapa jenis operator dibutuhkan untuk penulisan kondisi **IF..THEN..ELSE**. Operator-operator tersebut dapat dikelompokkan menjadi beberapa kelompok yaitu:

### » Operator Aritmatika

Simbol	Nama Operator	Contoh
+	Penjumlahan	$n = n+1;$
-	Pengurangan	$n = n-1;$
*	Perkalian	$n = n*1;$
/	Pembagian	$n = n/1;$
%	Sisa pembagian	$n = n\%1;$
++	Increment	$n = ++n$
--	Decrement	$n = --n$

### » Operator Relational

Simbol	Operasi	Contoh
==	Sama dengan	$(a==b);$
!=	Tidak sama dengan	$(a!=b);$
>	Lebih besar	$(a>b);$



<	Lebih kecil	(a<b);
>=	Lebih besar sama dengan	(a>=b);
<=	Lebih kecil sama dengan	(a<=b);

## » Operator Bitwise

Simbol	Operasi	Contoh
&	AND	(A & B);
	OR	(A   B);
^	XOR	(A ^ B);
~	Binary complement operator	(~A)
<<	Binary left shift operator	A << 2
>>	Binary right shift operator	A >> 2
>>>	Shift right zero fill operator	A >>> 2

## » Operator Logika

Simbol	Operasi	Contoh
&&	Kondisi AND	(a && b);
	Kondisi OR	(a    b);
!	Kondisi NOT	!(a && b)

## » Operator Penugasan

Simbol	Operasi	Contoh
=	Simpel assignment operator, Assigns value	C = A+B
+=	Add AND assignmet operator	C += A
-=	Substract AND assignment operator	C -= A
*=	Multiply AND assignment operator	C *=A
/=	Devide AND operator	C /=A
%=	Modulus AND assignment operator	C %=A
<<=	Left shift AND assignment operator	C <<=2
>>=	Right shift AND assignment operator	C >>=2
&=	Bitwise AND assignment operator	C &=2
^=	Bitwise exclusive OR and assignment operator	C ^=2
=	Bitwise inclusive OR and assignment operator	C  =2