



Fundamental Jetpack Compose

Praktikum Pemrograman Mobile - 03



Composable function

Semua penyebab masalah yang muncul pada pendekatan pembuatan UI menggunakan XML terdapat dalam cara Android View membangun state-nya dan menampilkan dirinya serta subkelasnya. Untuk menghindari masalah tersebut, kita perlu menggunakan blok program yang berbeda. Di Jetpack Compose, blok program tersebut disebut fungsi composable.

Berikut adalah contoh fungsi composable:

```
@Composable
fun MyComposableFunction() {
    // TODO
}
```

Kita perlu memberi anotasi `@Composable` pada fungsi tersebut. Fungsi apa pun yang dianotasi dengan cara ini disebut fungsi composable, karena kita dapat menyusunnya (compose) di dalam fungsi composable lainnya. Anotasi menyederhanakan kode dengan melampirkan metadata ke dalamnya. Javac, kompiler java, menggunakan kakas pemroses anotasi untuk memindai dan memproses anotasi pada saat kompilasi. Proses tersebut membuat file sumber baru dengan metadata yang ditambahkan. Singkatnya, dengan menggunakan anotasi, kita dapat menambahkan perilaku ke kelas dan menghasilkan kode yang bermanfaat, tanpa menulis banyak boilerplate. Anotasi khusus ini mengubah jenis fungsi tersebut menjadi composable, artinya hanya fungsi composable lainnya yang dapat memanggilnya.

Latihan 1

Pada latihan ini, kita akan mempelajari fungsi composable dan prinsip event-driven programming dengan sebuah form login. Hapus fungsi Greeting, kemudian buatlah fungsi composable `FormLogin` di `MainActivity` dengan kode seperti berikut.

```
@Composable
fun FormLogin() {
    val username = remember { mutableStateOf(TextFieldValue("")) }
    val password = remember { mutableStateOf(TextFieldValue("")) }

    Column(modifier = Modifier
        .padding(10.dp)
        .fillMaxWidth()) {
        Text(text = "Username", modifier =
            Modifier.padding(4.dp).fillMaxWidth())
        TextField(value = username.value, onChange = {
            username.value = it
        }, modifier = Modifier.padding(4.dp).fillMaxWidth())

        Text(text = "Password", modifier =
            Modifier.padding(4.dp).fillMaxWidth())
        TextField(value = password.value,
            visualTransformation = PasswordVisualTransformation(),
            onChange = {
                password.value = it
            }, keyboardOptions = KeyboardOptions(keyboardType =
                KeyboardType.Password),
            modifier = Modifier.padding(4.dp).fillMaxWidth())
    }
}
```

```

        val loginButtonColors = ButtonDefaults.buttonColors(
            backgroundColor = Purple700,
            contentColor = Teal200
        )

        val resetButtonColors = ButtonDefaults.buttonColors(
            backgroundColor = Teal200,
            contentColor = Purple700
        )

        Divider(modifier = Modifier.weight(1f).width(0.dp))

        Row(modifier = Modifier.padding(4.dp).fillMaxWidth()) {
            Button(modifier = Modifier.weight(5f), onClick = {

                }, colors = loginButtonColors) {
                Text(
                    text = "Login",
                    style = TextStyle(
                        color = Color.White,
                        fontSize = 18.sp
                    ), modifier = Modifier.padding(8.dp)
                )
            }

            Button(modifier = Modifier.weight(5f), onClick = {

                }, colors = resetButtonColors) {
                Text(
                    text = "Reset",
                    style = TextStyle(
                        color = Color.White,
                        fontSize = 18.sp
                    ), modifier = Modifier.padding(8.dp)
                )
            }
        }
    }
}

```

Warna Purple700 dan Teal200 berasal dari file Color di package ui.theme berikut.

```

package id.ac.unpas.composelogin.ui.theme

import androidx.compose.ui.graphics.Color

val Purple200 = Color(0xFFBB86FC)
val Purple500 = Color(0xFF6200EE)
val Purple700 = Color(0xFF3700B3)
val Teal200 = Color(0xFF03DAC5)

```

Warna-warna tersebut dapat diganti dengan warna apapun sesuai keinginan.

File MainActivity akan terlihat seperti ini.

```
package id.ac.unpas.composellogin

import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.compose.foundation.layout.*
import androidx.compose.foundation.text.KeyboardOptions
import androidx.compose.material.*
import androidx.compose.runtime.Composable
import androidx.compose.runtime.mutableStateOf
import androidx.compose.runtime.remember
import androidx.compose.ui.Modifier
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.text.TextStyle
import androidx.compose.ui.text.input.KeyboardType
import androidx.compose.ui.text.input.PasswordVisualTransformation
import androidx.compose.ui.text.input.TextFieldValue
import androidx.compose.ui.tooling.preview.Preview
import androidx.compose.ui.unit.dp
import androidx.compose.ui.unit.sp
import id.ac.unpas.composellogin.ui.theme.ComposeLoginTheme
import id.ac.unpas.composellogin.ui.theme.Purple700
import id.ac.unpas.composellogin.ui.theme.Teal200

class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContent {
            ComposeLoginTheme {
                // A surface container using the 'background' color from the
theme
                Surface(
                    modifier = Modifier.fillMaxSize(),
                    color = MaterialTheme.colors.background
                ) {
                    FormLogin()
                }
            }
        }
    }
}

@Composable
fun FormLogin() {
    val username = remember { mutableStateOf(TextFieldValue("")) }
    val password = remember { mutableStateOf(TextFieldValue("")) }

    Column(modifier = Modifier
        .padding(10.dp)
        .fillMaxWidth()) {
        Text(text = "Username", modifier =
Modifier.padding(4.dp).fillMaxWidth())
        TextField(value = username.value, onChange = {
            username.value = it
        }, modifier = Modifier.padding(4.dp).fillMaxWidth())
    }
}
```

```

        Text(text = "Password", modifier =
Modifier.padding(4.dp).fillMaxWidth())
        TextField(value = password.value,
            visualTransformation = PasswordVisualTransformation(),
            onChange = {
                password.value = it
            }, keyboardOptions = KeyboardOptions(keyboardType =
KeyboardType.Password),
            modifier = Modifier.padding(4.dp).fillMaxWidth())

        val loginButtonColors = ButtonDefaults.buttonColors(
            backgroundColor = Purple700,
            contentColor = Teal200
        )

        val resetButtonColors = ButtonDefaults.buttonColors(
            backgroundColor = Teal200,
            contentColor = Purple700
        )

        Divider(modifier = Modifier.weight(1f).width(0.dp))

        Row (modifier = Modifier.padding(4.dp).fillMaxWidth()) {
            Button(modifier = Modifier.weight(5f), onClick = {

                }, colors = loginButtonColors) {
                Text(
                    text = "Login",
                    style = TextStyle(
                        color = Color.White,
                        fontSize = 18.sp
                    ), modifier = Modifier.padding(8.dp)
                )
            }

            Button(modifier = Modifier.weight(5f), onClick = {

                }, colors = resetButtonColors) {
                Text(
                    text = "Reset",
                    style = TextStyle(
                        color = Color.White,
                        fontSize = 18.sp
                    ), modifier = Modifier.padding(8.dp)
                )
            }
        }
    }
}

@Preview(showBackground = true)
@Composable
fun DefaultPreview() {
    ComposeLoginTheme {
        FormLogin()
    }
}

```

```
}
}
```

Kode di atas baru membuat sebuah form login yang diatur dalam kolom dan baris seperti pada tangkapan layar berikut.

Column (kolom) dapat menampung komponen GUI dalam baris-baris secara vertikal, sedangkan Row (baris) dapat menampung komponen GUI dalam kolom-kolom secara horizontal. Kontainer Column dan Row tersebut dapat mengatur komponen berdasarkan tinggi (height) dan lebar (width), atau dengan bobot (weight). Pada contoh di atas, komponen Divider digunakan sebagai pemisah yang memiliki weight 1f (f untuk float seperti di Java), sedangkan komponen lain dalam Column tidak memiliki weight sehingga komponen Divider akan memenuhi sisa ruang dalam Column. Jika setiap komponen dalam Column atau Row memiliki weight, maka tinggi (untuk Column)/lebar (untuk Row) komponen-komponen tersebut akan diatur sesuai dengan persentase weight-nya. Contohnya pada contoh di atas, kedua Button memiliki weight 5f sehingga lebar masing-masing Button adalah 50% (5/10) dari lebar Row.

Selanjutnya, tambahkan fungsionalitas untuk form login yang kita buat dengan mengisi event handler onClick pada masing-masing button. Pertama, buat variabel context di fungsi FormLogin untuk mengambil context aplikasi (pemegang lifecycle).

```
val context = LocalContext.current
```

Kemudian, untuk Button dengan text Login, isi fungsi onClick dengan kode berikut.

```
if (username.value.text == "admin" && password.value.text == "admin") {  
    Toast.makeText(context, "Login Sukses", Toast.LENGTH_LONG).show()  
} else {  
    Toast.makeText(context, "Login Gagal", Toast.LENGTH_LONG).show()  
}
```

Sedangkan untuk Button dengan text Reset, isi fungsi onClick dengan kode berikut.

```
username.value = TextFieldValue("")  
password.value = TextFieldValue("")
```

Sekarang, kode MainActivity akan terlihat seperti ini.

```
package id.ac.unpas.composelogin  
  
import android.os.Bundle  
import android.widget.Toast  
import androidx.activity.ComponentActivity  
import androidx.activity.compose.setContent  
import androidx.compose.foundation.layout.*  
import androidx.compose.foundation.text.KeyboardOptions  
import androidx.compose.material.*  
import androidx.compose.runtime.Composable  
import androidx.compose.runtime.mutableStateOf  
import androidx.compose.runtime.remember  
import androidx.compose.ui.Modifier  
import androidx.compose.ui.graphics.Color  
import androidx.compose.ui.platform.LocalContext  
import androidx.compose.ui.text.TextStyle  
import androidx.compose.ui.text.input.KeyboardType  
import androidx.compose.ui.text.input.PasswordVisualTransformation  
import androidx.compose.ui.text.input.TextFieldValue  
import androidx.compose.ui.tooling.preview.Preview  
import androidx.compose.ui.unit.dp  
import androidx.compose.ui.unit.sp  
import id.ac.unpas.composelogin.ui.theme.ComposeLoginTheme  
import id.ac.unpas.composelogin.ui.theme.Purple700  
import id.ac.unpas.composelogin.ui.theme.Teal200  
  
class MainActivity : ComponentActivity() {  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContent {  
            ComposeLoginTheme {  
                // A surface container using the 'background' color from the  
theme  
                Surface(  
                    modifier = Modifier.fillMaxSize(),  
                    color = MaterialTheme.colors.background  
                ) {  
                    FormLogin()  
                }  
            }  
        }  
    }  
}
```



```

    }
}

@Composable
fun FormLogin() {
    val context = LocalContext.current
    val username = remember { mutableStateOf(TextFieldValue("")) }
    val password = remember { mutableStateOf(TextFieldValue("")) }

    Column(modifier = Modifier
        .padding(10.dp)
        .fillMaxWidth()) {
        Text(text = "Username", modifier =
Modifier.padding(4.dp).fillMaxWidth())
        TextField(value = username.value, onValueChange = {
            username.value = it
        }, modifier = Modifier.padding(4.dp).fillMaxWidth())

        Text(text = "Password", modifier =
Modifier.padding(4.dp).fillMaxWidth())
        TextField(value = password.value,
            visualTransformation = PasswordVisualTransformation(),
            onValueChange = {
                password.value = it
            }, keyboardOptions = KeyboardOptions(keyboardType =
KeyboardType.Password),
            modifier = Modifier.padding(4.dp).fillMaxWidth())

        val loginButtonColors = ButtonDefaults.buttonColors(
            backgroundColor = Purple700,
            contentColor = Teal200
        )

        val resetButtonColors = ButtonDefaults.buttonColors(
            backgroundColor = Teal200,
            contentColor = Purple700
        )

        Divider(modifier = Modifier.weight(1f).width(0.dp))

        Row (modifier = Modifier.padding(4.dp).fillMaxWidth()) {
            Button(modifier = Modifier.weight(5f), onClick = {
                if (username.value.text == "admin" && password.value.text ==
"admin") {
                    Toast.makeText(context, "Login Sukses",
Toast.LENGTH_LONG).show()
                } else {
                    Toast.makeText(context, "Login Gagal",
Toast.LENGTH_LONG).show()
                }
            }, colors = loginButtonColors) {
                Text(
                    text = "Login",
                    style = TextStyle(
                        color = Color.White,
                        fontSize = 18.sp

```



```

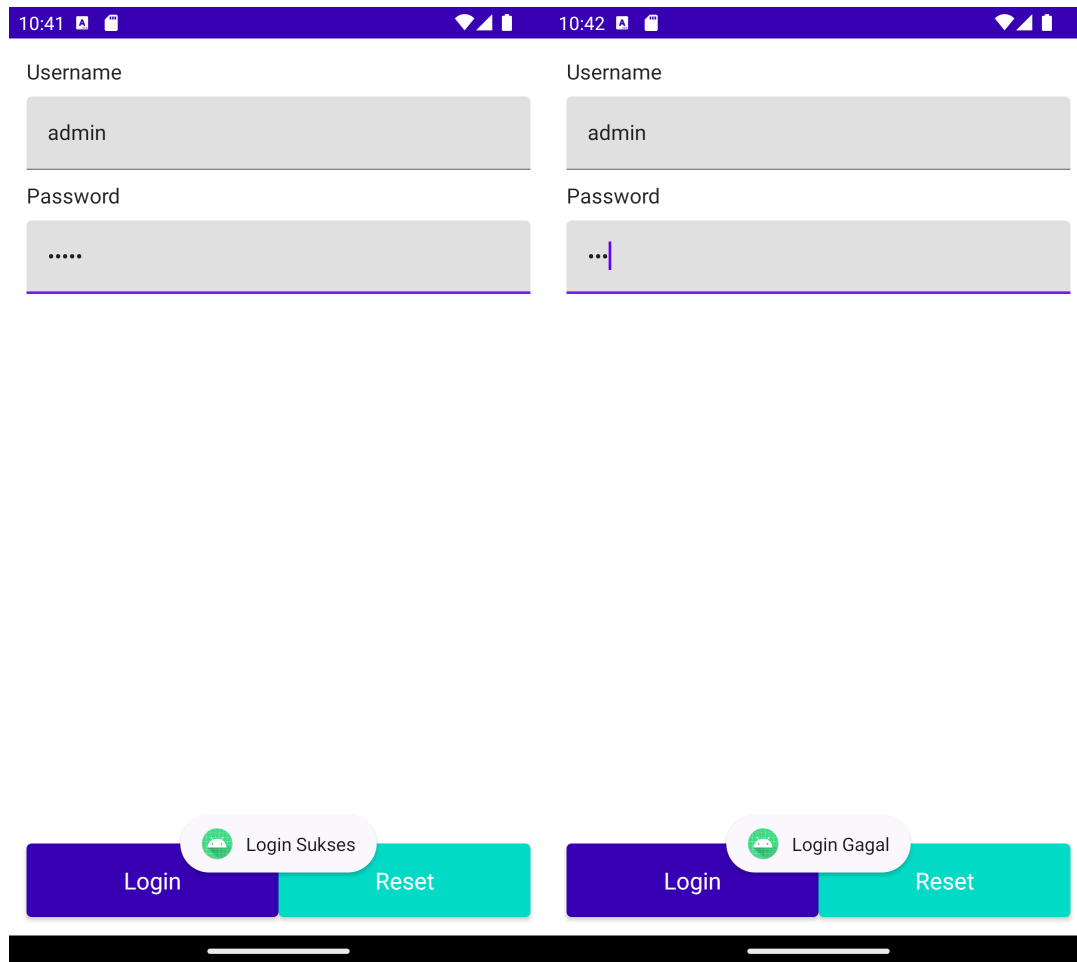
        ), modifier = Modifier.padding(8.dp)
    )
}

Button(modifier = Modifier.weight(5f), onClick = {
    username.value = TextFieldValue("")
    password.value = TextFieldValue("")
}, colors = resetButtonColors) {
    Text(
        text = "Reset",
        style = TextStyle(
            color = Color.White,
            fontSize = 18.sp
        ), modifier = Modifier.padding(8.dp)
    )
}
}
}

@Preview(showBackground = true)
@Composable
fun DefaultPreview() {
    ComposeLoginTheme {
        FormLogin()
    }
}

```

Jalankan aplikasi, kemudian coba masukkan input username dan password. Jika aplikasi menerima input username = admin dan password = admin, maka akan muncul pesan Login Sukses, jika selain itu akan muncul pesan Login Gagal. Tombol Reset dapat digunakan untuk menghapus inputan.



Latihan 2

Buatlah aplikasi baru untuk menerima form registrasi dengan isian nama, username, nomor telepon, email, dan alamat rumah. Cobalah eksplorasi `KeyboardOptions` yang cocok untuk nomor telepon dan email. Tambahkan dua buah tombol yaitu Simpan, yang jika semua inputan diisi akan memunculkan pesan Halo, \$nama. Jika ada inputan yang tidak diisi, aplikasi harus menampilkan pesan bahwa semua inputan harus diisi. Tombol kedua adalah tombol Reset yang dapat digunakan untuk menghapus semua inputan.