

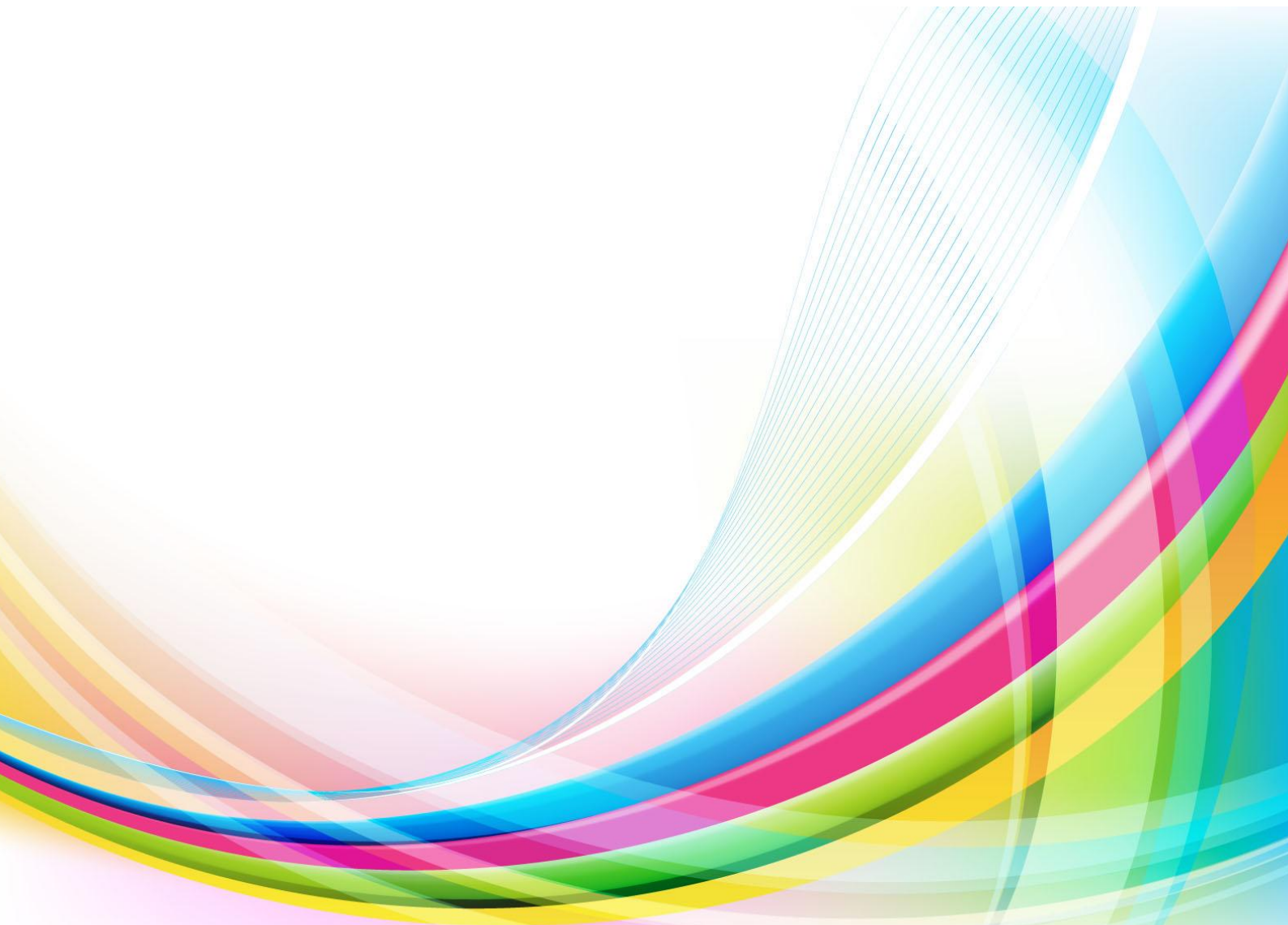


MODUL PRAKTIKUM 4

CLASS DAN OBJECT

BAHASA PEMROGRAMAN II S1 SI

STEVI EMA WIJAYANTI



CLASS DAN OBJECT IN ANDROID STUDIO

Overview :

Pada modul ini akan dibahas mengenai class dan juga object serta Activity dan Intent serta implementasinya dalam Android Studio.

Tujuan dari praktikum 4 :

1. Mengetahui definisi class dan contoh class
2. Mengetahui definisi object dan fungsi dari object
3. Mereview kembali terkait activity dan intent yang telah dibahas pada praktikum sebelumnya dan mengimplementasikan pada Android Studio
4. Mampu mengimplementasikan class dan object sederhana pada Android Studio

CLASS DAN OBJECT

Class

Dasar pemrograman berorientasi objek adalah class. Class merupakan suatu “blueprint” atau “cetakan” atau “prototype” atau rancangan yang mendefinisikan/menggambarkan atribut/property dan tingkah laku berbagai entitas. Class juga merupakan deskripsi tergeneralisir yang menggambarkan kumpulan objek yang sama. Class juga bisa disebut blok-blok bangun pada program Kotlin/Java. Class merupakan wadah yang digunakan untuk menciptakan suatu objek.

Sebuah benda (entitas) yang ada dapat disebut sebuah class apabila memiliki 2 hal, yaitu :

1. Atribut/property

Atribut adalah elemen, data field atau data item yang digunakan untuk menerangkan suatu data dari entitas. Attribute berlaku sebagai data, didefinisikan oleh class, individu, dan berbeda satu dengan lainnya. Attribute didefinisikan dalam class dengan menggunakan variabel.

Contoh class Mahasiswa, maka atribut/property yang dimiliki oleh entitas Mahasiswa antara lain nim, nama, jurusan, kelas, ipk dan lain sebagainya.

2. Behavior/tingkah laku

Behavior merupakan hal-hal yang bisa dilakukan oleh suatu entitas/class. Behavior berlaku sebagai method (operasi). Untuk menentukan behavior atau perilaku dari sebuah entity harus membuat methods. Method ini biasanya dapat digunakan untuk mengubah nilai atribut (variabel), menerima informasi dari class (object) lain dan mengirim informasi ke class (object) lain untuk melakukan task.

CONTOH :

Entity : Manusia

Attribute : 1. Nama
2. Gender
3. Tempat tanggal lahir
4. Alamat
5. dan sebagainya.

Behavior : 1. Berjalan
2. Makan
3. Tidur
4. dan sebagainya.

Karena entity Manusia memiliki attribute dan behavior, maka bisa dikatakan bahwa Manusia merupakan sebuah class di dalam Kotlin programming.

CONTOH LAIN :

Entity : Kursi (Bangku Kuliah)

Attribute : 1. Merk
2. Type
3. Dimesi
4. Harga
5. dan sebagainya.

Behavior : 1. Tidak ada

Karena entity Kursi hanya memiliki attribute dan tidak memiliki behavior, maka Kursi bukan sebuah class dalam bahasa pemrograman Java.

Class mendefinisikan atribut/properti atau biasa disebut dengan variabel dan perilaku (behavior) atau biasa disebut dengan method dari obyek yang dibuatnya.

Object

Object adalah instance dari class secara umum merepresentasikan (template) sebuah object. Object adalah representasi nyata dari class itu sendiri.

Jika pada class dicontohkan adalah class Manusia maka object dari class tersebut antara lain : ani, budi, joko, hanif, dan lain sebagainya.

Contoh lain, misalkan class Hewan maka object dari class hewan antara lain : sapi, ayam, ikan, ular, dan lain sebagainya.

Apabila digambarkan dalam contoh sederhana, misalkan pada saat pembuatan Kue. Maka cetakan kue adalah class, kemudian isi/adonan kue adalah object. Jika cetakan berbentuk bulat, maka kue yang dihasilkan juga akan berbentuk bulat mengikuti cetakannya. Itulah alasan bahwa object adalah representasi dari class.

IMPLEMENTASI CLASS DAN OBJECT

Silahkan buka kembali project (ProjectBPXXXX) yang telah anda kerjakan pada praktikum sebelumnya. Jika sebelumnya anda telah menambahkan activity Login pada aplikasi anda, dan telah ditambahkan intents agar saat button login di-klik akan diarahkan masuk ke activity main (homepage), maka pada praktikum ini agar diimplementasikan class dan object untuk activity login, langkah-langkahnya :

1. Silahkan buka kembali LoginActivity.kt. Tambahkan proses validasi login (username dan password bebas, bisa menggunakan nama masing-masing).

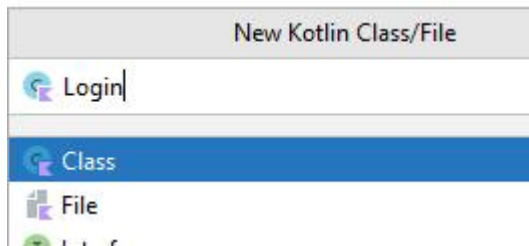
```
class LoginActivity : AppCompatActivity() {
    // Stevi Ema Wijayanti *
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_login)

        //instance
        val btnLogin:Button = findViewById(R.id.buttonLogin)
        val txtUsername:EditText = findViewById(R.id.editTextUsername)
        val txtPassword:EditText = findViewById(R.id.editTextPassword)

        //event saat button Login click
        btnLogin.setOnClickListener{ it: View!
            //login cek
            if(txtUsername.text.toString().equals("stevi") &&
                txtPassword.text.toString().equals("amikom")){
                //call homepage
                val intent = Intent( packageContext: this,HomeActivity::class.java)
                startActivity(intent)
            }else{
                Toast.makeText( context: this, text: "Login gagal, cek username dan password",
                    Toast.LENGTH_SHORT).show()
            }
        }
    }
}
```

Simpan dan jalankan program anda, pastikan proses validasi login berjalan dengan baik.

- Modifikasi program diatas dengan menerapkan class dan object sederhana. Silahkan tambahkan kotlin file baru, pilih Class, sehingga muncul dialog box seperti gambar berikut :



Berikan nama class "Login"

- Pada class **Login**, silahkan tambahkan baris kode seperti berikut ini :

```
class Login {
    var username = ""
    var password = ""

    new *
    fun loginCheck(): Boolean {
        if (username.equals("stevi") && password.equals("amikom")) {
            return true
        } else {
            return false
        }
    }
}
```

- Buka kembali LoginActivity dan modifikasi program agar dapat melakukan validasi login.

```
override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContentView(R.layout.activity_login)

    //instance
    val btnLogin: Button = findViewById(R.id.buttonLogin)
    val txtUsername: EditText = findViewById(R.id.editTextUsername)
    val txtPassword: EditText = findViewById(R.id.editTextPassword)
    val logObj = Login()

    //input
    logObj.username = txtUsername.text.toString()
    logObj.password = txtPassword.text.toString()
}
```

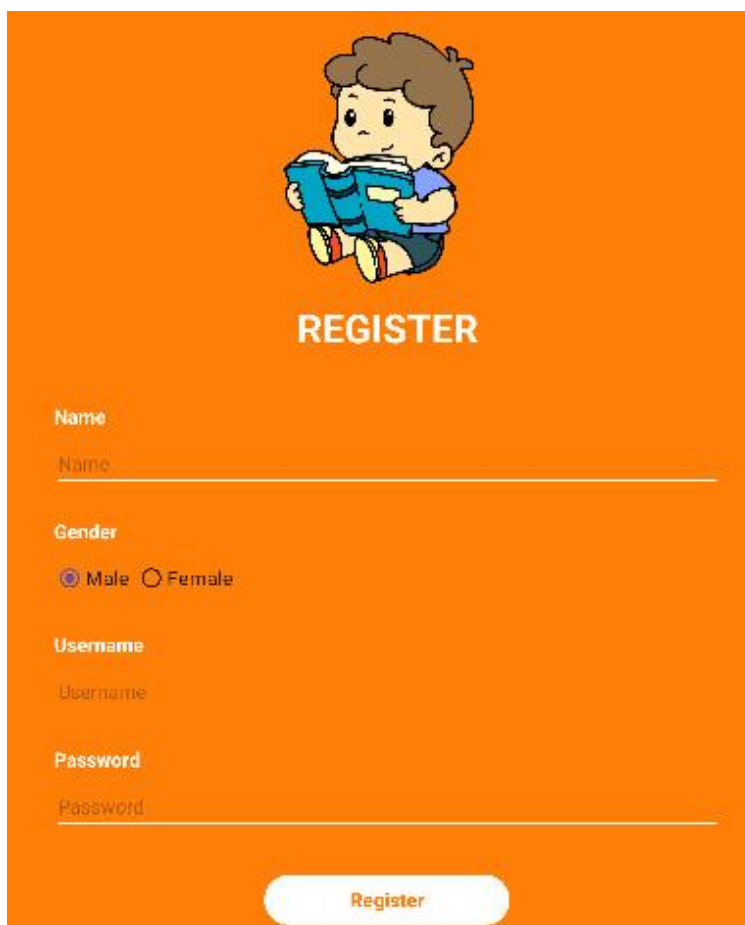
```
//event saat button login click
btnLogin.setOnClickListener{ it: View!
    //login cek
    if(logObj.loginCheck() == true){
        //call homepage
        val intent = Intent( packageContext: this,HomeActivity::class.java)
        startActivity(intent)
    }else{
        Toast.makeText( context: this, text: "Login gagal, cek username dan password",
            Toast.LENGTH_SHORT).show()
    }
}
```

5. Simpan dan jalankan program yang telah anda modifikasi, pastikan prosesnya tidak berubah.

TUGAS

Silahkan kerjakan kasus berikut ini :

1. Silahkan buat activity Register, dan atur layoutnya untuk inputan 4 data yaitu nama, jenis kelamin, username dan password. (Desain bebas)



2. Tambahkan intent untuk mengakses halaman register dari activity login saat text “Don’t have an account, Register Here” !
3. Implementasikan class dan object untuk activity yang anda buat.