

Mobile Programming (MP)

Pertemuan 4 : Fungsi dan Dasar OOP Kotlin

Alauddin Maulana Hirzan

Fakultas Teknologi Informasi dan Komunikasi
Universitas Semarang

Outline

Mobile Programming
(MP)

Alauddin
Maulana
Hirzan

Kotlin:
Functions

Kotlin:
*Object-Oriented
Programming*

Lastly

1 Kotlin: *Functions*

2 Kotlin: *Object-Oriented Programming*

3 Lastly

Definisi *Functions*

Mobile Programming
(MP)

Alauddin
Maulana
Hirzan

Kotlin:
Functions

Kotlin:
*Object-Oriented
Programming*

Lastly

Fungsi adalah blok kode yang dirancang untuk menjalankan tugas tertentu dan dapat dipanggil kembali kapan pun diperlukan dalam program.

Fungsi membantu memecah program besar menjadi bagian-bagian kecil yang lebih mudah dipahami dan dikelola.

Daripada menuliskan kode hingga ratusan hingga ribuan baris, programmer yang baik akan memecah kode sesuai tugas yang diberikan kepada bagian kode tersebut. Contoh:

- 1 Konversi Huruf Kapital
- 2 Verifikasi Password
- 3 Kalkulasi Statistik

Manfaat Penggunaan Fungsi

Mobile Programming
(MP)

Alauddin
Maulana
Hirzan

Kotlin:
Functions

Kotlin:
*Object-Oriented
Programming*

Lastly

Penggunaan fungsi dalam pemrograman memberikan beberapa keuntungan:

- 1 menghindari pengulangan kode (*code duplication*)
- 2 meningkatkan keterbacaan program (*readability*)
- 3 mempermudah pemeliharaan dan pengembangan kode (*maintainability*)
- 4 memungkinkan penggunaan kembali kode (*code reuse*)

Contoh Kode

Mobile Programming
(MP)

Alauddin
Maulana
Hirzan

Kotlin:
Functions

Kotlin:
*Object-
Oriented
Program-
ming*

Lastly

Tanpa Fungsi (Kotlin)

```
var rata_rata = (9 + 8 + 7 + 8) / 4
```

Dengan Fungsi (Kolin)

```
val nilai = [9, 8, 7, 8]  
var rata_rata = average(nilai) // Kalkulasi dilakukan oleh average()
```

Deskripsi

Tanpa fungsi, programmer akan kewalahan melakukan perhitungan apabila data sudah lebih dari 50

Struktur Dasar Fungsi

Struktur umum fungsi dalam Kotlin, perhatikan contoh berikut:

Contoh

```
fun namaFungsi(parameter: Tipe): TipeKembali {  
    // isi fungsi  
}
```

Penjelasan:

- **fun** → kata kunci untuk mendefinisikan fungsi
- **namaFungsi** → nama fungsi yang digunakan untuk pemanggilan
- **parameter** → nilai masukan ke fungsi
- **TipeKembali** → tipe data nilai yang dikembalikan

Contoh Fungsi Tanpa Parameter

Mobile Programming
(MP)

Alauddin
Maulana
Hirzan

Kotlin:
Functions

Kotlin:
*Object-
Oriented
Programming*

Lastly

Fungsi Tanpa Parameter

```
fun sapa() {  
    println("Halo Kotlin")  
}
```

```
sapa()
```

Hasil Kompilasi

```
Halo Kotlin
```

Contoh Fungsi Dengan Parameter

Mobile Programming
(MP)

Alauddin
Maulana
Hirzan

Kotlin:
Functions

Kotlin:
*Object-
Oriented
Programming*

Lastly

Fungsi Dengan Parameter

```
fun sapa(nama: String) {  
    println("Halo, $nama")  
}  
  
sapa("Maulana")
```

Hasil Kompilasi

```
Halo, Maulana
```

Penjelasan Kode

Mobile Programming
(MP)

Alauddin
Maulana
Hirzan

Kotlin:
Functions

Kotlin:
*Object-Oriented
Programming*

Lastly

Penjelasan:

- 1 nama adalah parameter bertipe String
- 2 nilai "Mahasiswa" dikirim ke fungsi sebagai argumen
- 3 fungsi menampilkan pesan berdasarkan nilai parameter

Parameter memungkinkan satu fungsi digunakan untuk berbagai kondisi berbeda tanpa menulis ulang kode.

Contoh Fungsi Dengan Nilai Kembali

Mobile Programming
(MP)

Alauddin
Maulana
Hirzan

Kotlin:
Functions

Kotlin:
*Object-
Oriented
Programming*

Lastly

Fungsi Dengan Nilai Kembali

```
fun tambah(a: Int, b: Int): Int {  
    return a + b  
}
```

```
val hasil = tambah(5, 3)  
print("Hasil Tambah 5 dan 3 = ${hasil} ")
```

Hasil Kompilasi

```
Hasil Tambah 5 dan 3 = 8
```

Penjelasan Kode

Mobile Programming
(MP)

Alauddin
Maulana
Hirzan

Kotlin:
Functions

Kotlin:
*Object-Oriented
Programming*

Lastly

Penjelasan:

- 1 fungsi menerima dua parameter bertipe Int
- 2 fungsi mengembalikan hasil penjumlahan
- 3 nilai hasil disimpan ke dalam variabel hasil

Fungsi dengan nilai kembali sering digunakan dalam perhitungan, pengolahan data, dan logika program.

Contoh Fungsi Ekspresi Tunggal

Mobile Programming
(MP)

Alauddin
Maulana
Hirzan

Kotlin:
Functions

Kotlin:
*Object-Oriented
Programming*

Lastly

Fungsi Ekspresi Tunggal

```
fun kali(a: Int, b: Int): Int = a * b
```

Fungsi Ekuivalensi

```
fun kali(a: Int, b: Int): Int {  
    return a * b  
}
```

Keuntungan Ekspresi Tunggal

Mobile Programming
(MP)

Alauddin
Maulana
Hirzan

Kotlin:
Functions

Kotlin:
*Object-
Oriented
Programming*

Lastly

Keuntungan:

- 1 kode lebih pendek dan bersih
- 2 cocok untuk fungsi sederhana
- 3 meningkatkan keterbacaan bagi fungsi dengan logika langsung

Definisi *Object-Oriented Programming*

Mobile Programming
(MP)

Alauddin
Maulana
Hirzan

Kotlin:
Functions

Kotlin:
*Object-Oriented
Programming*

Lastly

Object-Oriented Programming (OOP) adalah paradigma pemrograman yang berfokus pada penggunaan objek untuk merepresentasikan data dan perilaku dalam suatu sistem.

Dalam OOP, program tidak hanya terdiri dari fungsi, tetapi juga terdiri dari entitas yang memiliki atribut dan metode. Empat konsep utama dalam OOP:

- 1 *Class* : Cetak biru untuk membuat objek
- 2 *Object* : Instansi nyata dari class
- 3 *Encapsulation* : Mekanisme pembungkusan data dan metode
- 4 *Inheritance* : Pewarisan sifat dari class lain

Keuntungan OOP

Mobile Programming
(MP)

Alauddin
Maulana
Hirzan

Kotlin:
Functions

Kotlin:
*Object-
Oriented
Programming*

Lastly

Keuntungan menggunakan OOP:

- 1 Kode lebih terstruktur dan modular
- 2 Mempermudah pengembangan aplikasi besar
- 3 Mendukung reuse kode melalui pewarisan

Definisi *Class*

Mobile Programming
(MP)

Alauddin
Maulana
Hirzan

Kotlin:
Functions

Kotlin:
*Object-Oriented
Programming*

Lastly

Class adalah cetak biru (blueprint) yang digunakan untuk membuat objek.

Class mendefinisikan:

- 1 *Atribut (Properties)* → Data yang dimiliki objek
- 2 *Metode (Methods)* → Fungsi atau perilaku objek

Ilustrasi

Mobile Programming (MP)

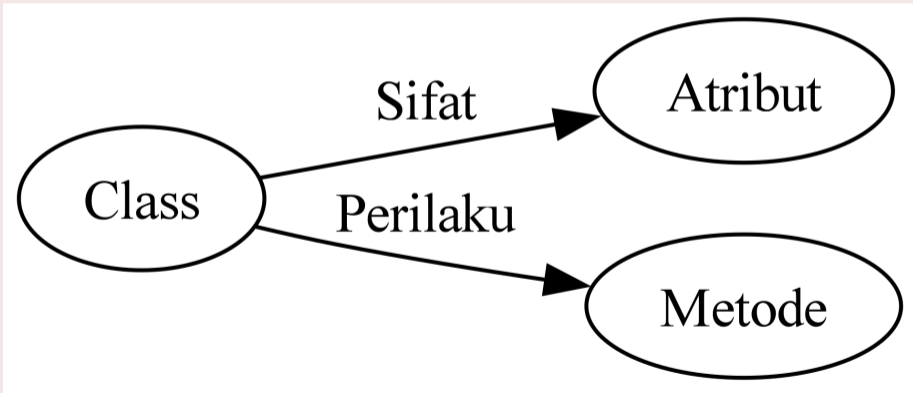
Alauddin Maulana Hirzan

Kotlin: Functions

Kotlin: Object-Oriented Programming

Lastly

Ilustrasi



Contoh Kode Kotlin

Mobile Programming
(MP)

Alauddin
Maulana
Hirzan

Kotlin:
Functions

Kotlin:
Object-Oriented Programming

Lastly

Kode Class Kotlin - Contoh Umum

```
class NamaClass {  
    // atribut  
    // metode  
}
```

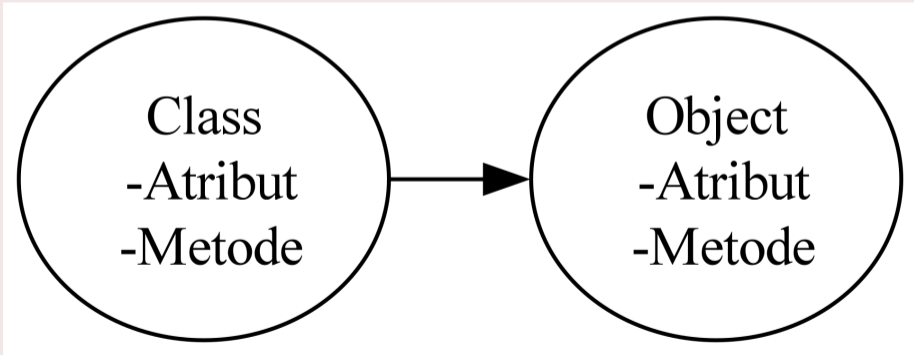
Kode Class Kotlin - Contoh Mahasiswa

```
class Mahasiswa {  
    var nama: String = ""  
    var alamat: String = ""  
}
```

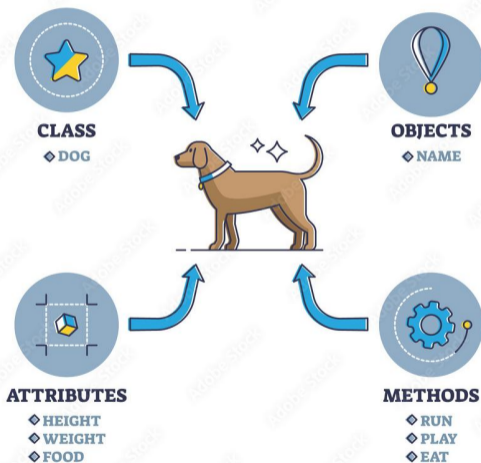
Definisi *Object*

Object adalah instansi nyata dari sebuah class. *Object* dibuat dari class dan dapat digunakan untuk mengakses atribut dan metode yang ada di dalamnya.

Ilustrasi



Contoh Objek



Mobile Programming (MP)

Alauddin Maulana Hirzan

Kotlin: Functions

Kotlin: Object-Oriented Programming

Lastly

Contoh Kode

Mobile Programming
(MP)

Alauddin
Maulana
Hirzan

Kotlin:
Functions

Kotlin:
*Object-
Oriented
Programming*

Lastly

Objek Mahasiswa

```
class Mahasiswa {  
    var nama: String = ""  
    var umur: Int = 0  
}  
  
fun main() {  
    val mhs = Mahasiswa()  
    mhs.nama = "Budi"  
    mhs.umur = 20  
  
    println("Selamat Datang, ${mhs.nama}")  
}
```

Hasil

Mobile Programming
(MP)

Alauddin
Maulana
Hirzan

Kotlin:
Functions

Kotlin:
Object-Oriented Programming

Lastly

Hasil Kompilasi

```
Selamat Datang, Budi
```

Penjelasan

Penjelasan kode:

- 1 `Mahasiswa()` → membuat object baru
- 2 `mhs.nama` → mengisi nilai atribut nama
- 3 `mhs.umur` → mengisi nilai atribut umur
- 4 `println(mhs.nama)` → menampilkan data dari object

Ilustrasi Konsep

Mobile Programming
(MP)

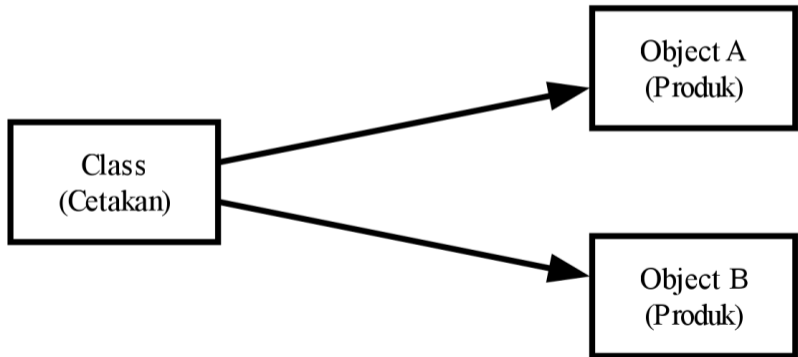
Alauddin
Maulana
Hirzan

Kotlin:
Functions

Kotlin:
*Object-Oriented
Programming*

Lastly

Ilustrasi Konsep



Konstruktor Objek

Mobile Programming
(MP)

Alauddin
Maulana
Hirzan

Kotlin:
Functions

Kotlin:
*Object-
Oriented
Programming*

Lastly

Constructor digunakan untuk menginisialisasi objek pada saat objek tersebut dibuat dari sebuah class.

Constructor memungkinkan pemberian nilai awal pada atribut tanpa harus mengisi satu per satu setelah objek dibuat.

Contoh

```
class Mahasiswa(var nama: String, var umur: Int)
```

Keuntungan Kontruktor

Mobile Programming
(MP)

Alauddin
Maulana
Hirzan

Kotlin:
Functions

Kotlin:
*Object-Oriented
Programming*

Lastly

Keuntungan penggunaan constructor:

- 1 Inisialisasi objek menjadi lebih cepat
- 2 Kode lebih singkat dan mudah dibaca
- 3 Mengurangi kemungkinan objek berada dalam kondisi tidak valid

Enkapsulasi *Class*

Encapsulation adalah konsep OOP untuk menyembunyikan data internal dan hanya menyediakan akses melalui metode tertentu. Tujuannya adalah untuk melindungi data agar tidak diubah secara langsung dari luar class.

Contoh Enkapsulasi

```
class Akun {  
    private var saldo: Int = 0  
    fun setor(jumlah: Int) {  
        saldo += jumlah  
    }  
    fun cekSaldo(): Int {  
        return saldo  
    }  
}
```

Penjelasan

Mobile Programming
(MP)

Alauddin
Maulana
Hirzan

Kotlin:
Functions

Kotlin:
*Object-
Oriented
Programming*

Lastly

Penjelasan:

- 1 `private var saldo` → tidak dapat diakses langsung dari luar class
- 2 Akses saldo hanya melalui fungsi `setor()` dan `cekSaldo()`

Keuntungan encapsulation:

- 1 Menjaga keamanan dan integritas data
- 2 Mencegah perubahan data yang tidak terkontrol
- 3 Mempermudah pengelolaan dan debugging program

Pewarisan (*Inheritance*)

Mobile Programming
(MP)

Alauddin
Maulana
Hirzan

Kotlin:
Functions

Kotlin:
Object-
Oriented
Programming

Lastly

Inheritance memungkinkan sebuah class baru mewarisi atribut dan metode dari class yang sudah ada. Class yang diwarisi disebut parent class atau superclass, sedangkan class yang mewarisi disebut child class atau subclass. Bersifat lebih umum dari *Class*

Contoh

```
open class Kendaraan {  
    fun jalan() {  
        println("Kendaraan berjalan")  
    }  
}  
  
class Mobil : Kendaraan()
```

Penjelasan

Mobile Programming
(MP)

Alauddin
Maulana
Hirzan

Kotlin:
Functions

Kotlin:
*Object-
Oriented
Programming*

Lastly

Penjelasan:

- 1 open diperlukan agar class dapat diwarisi
- 2 Mobil mewarisi semua fungsi dari Kendaraan
- 3 Objek Mobil dapat langsung menggunakan fungsi jalan()

Penggunaan *Inheritance*

Mobile Programming
(MP)

Alauddin
Maulana
Hirzan

Kotlin:
Functions

Kotlin:
*Object-
Oriented
Programming*

Lastly

Cara Penggunaan

```
fun main() {  
    val mobil = Mobil()  
    mobil.jalan()  
}
```

Contoh Program OOP Kotlin

Mobile Programming
(MP)

Alauddin
Maulana
Hirzan

Kotlin:
Functions

Kotlin:
*Object-
Oriented
Programming*

Lastly

Contoh

```
class Mahasiswa(var nama: String, var nilai: Int) {  
    fun cekKelulusan(): String {  
        return if (nilai >= 75) "Lulus" else "Tidak Lulus"  
    }  
}  
  
fun main() {  
    val mhs = Mahasiswa("Andi", 80)  
    println("${mhs.nama} : ${mhs.cekKelulusan()}")  
}
```

Hasil Kompilasi

Mobile Programming
(MP)

Alauddin
Maulana
Hirzan

Kotlin:
Functions

Kotlin:
*Object-
Oriented
Programming*

Lastly

Hasil Kompilasi

```
Andi : Lulus
```

Penjelasan

- 1 Class Mahasiswa memiliki atribut nama dan nilai
- 2 Constructor menginisialisasi data saat objek dibuat
- 3 Method cekKelulusan() menentukan status kelulusan
- 4 Objek mhs digunakan untuk memanggil method dan menampilkan hasil

Terima Kasih