

LAPORAN RESMI
MODUL I
KOTLIN DASAR
PEMROGRAMAN BERGERAK



NAMA : HILMIS SHOLEH
N.R.P : 210441100116
DOSEN : ACHMAD ZAIN NUR, S.KOM., M.T.
ASISTEN : DWI ANGGA VIRGI
TGL PRAKTIKUM : 22 MARET 2024

Disetujui : Maret 2024
Asisten

DWI ANGGA VIRGI
20.04.411.00174



LABORATORIUM BISNIS INTELIJEN SISTEM
PRODI SISTEM INFORMASI
JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS TRUNOJOYO MADURA

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dunia pengembangan aplikasi mobile kian berkembang pesat. Berbagai bahasa pemrograman bermunculan untuk memenuhi kebutuhan para developer dalam menciptakan aplikasi yang inovatif dan fungsional. Di antara bahasa pemrograman yang kian populer adalah Kotlin. Kotlin menawarkan berbagai keunggulan yang menjadikannya pilihan menarik bagi para developer, baik pemula maupun berpengalaman.

Dibandingkan dengan bahasa pemrograman lain seperti Java, Kotlin memiliki sintaks yang lebih ringkas dan mudah dibaca. Hal ini tentunya memudahkan proses belajar dan pengembangan aplikasi bagi programmer pemula. Selain itu, Kotlin menawarkan fitur keamanan dan interoperabilitas yang baik. Fitur keamanan yang dimiliki Kotlin dapat membantu developer dalam membuat aplikasi yang lebih kokoh dan terhindar dari celah keamanan dengan fitur interoperabilitas memungkinkan developer untuk menggunakan library Java yang sudah ada dalam proyek Kotlin mereka, sehingga proses pengembangan aplikasi menjadi lebih efisien.

Keunggulan-keunggulan inilah yang membuat Kotlin semakin banyak diminati oleh para developer di seluruh dunia. Oleh karena itu, mempelajari dasar-dasar pemrograman Kotlin menjadi penting untuk mempersiapkan diri dalam menghadapi perkembangan dunia aplikasi mobile yang semakin dinamis

1.2 Tujuan

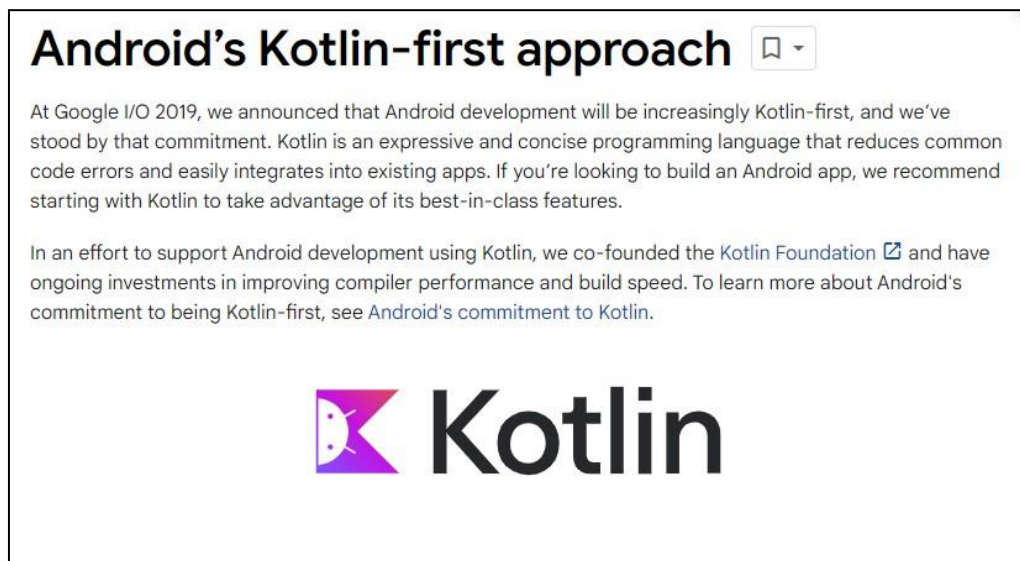
- Mahasiswa mampu memahami apa itu kotlin pada Bahasa pemrograman android.
- Mahasiswa mampu memahami konsep-konsep yang ada di kotlin dengan baik.
- Mahasiswa mampu menerapkan serta mengembangkan pengetahuan Bahasa pemrograman android.

BAB II

DASAR TEORI

2.1 Pengenalan Kotlin

Pada tahun 2010 lalu JetBrains memulai sebuah proyek open-sources baru. Proyek ini merupakan sebuah bahasa pemrograman statically typed yang menargetkan JVM, Android, JavaScript dan Native. Kotlin -nama sebuah pulau di Rusia- menjadi inspirasi oleh tim pengembang untuk ditetapkan sebagai nama bahasa pemrograman tersebut. Mungkin karena pulau tersebut terletak tidak jauh dari markas tim programer JetBrains, yaitu Saint Petersburg, Rusia. Bahasa Kotlin pertama kali dirilis pada bulan Februari 2016 dengan versi 1.0. Saat ini sudah mencapai versi 1.5.21 per rilis 14 Juli 2021. Kotlin dapat digunakan untuk berbagai macam pengembangan aplikasi, baik itu server atau backend, website, maupun mobile Android. Bahkan saat ini tengah dikembangkan Kotlin/Native. Apa uniknya? Kotlin/Native memungkinkan developer untuk menggunakannya sebagai bahasa pemrograman dalam pengembangan aplikasi di platform lain seperti embedded system, desktop, macOS, dan iOS. Bahkan tak menutup kemungkinan Kotlin juga bisa digunakan untuk data science dan machine learning. Menarik, bukan? Dukungan tools untuk Kotlin, sangat kuat. Kita bisa dengan mudah menggunakan Kotlin pada IDE seperti IntelliJ IDEA, Android Studio, Eclipse, dan NetBeans.



Gambar 1. Kotlin First Approach

Biarpun Kotlin bisa kita gunakan untuk berbagai macam pengembangan aplikasi, tapi kali ini kita belajar Kotlin sebagai Bahasa Pemrograman untuk **pengembangan aplikasi Android**, ya!

So tunggu apa lagi, let's go!

Persiapan sebelum belajar Kotlin:

Sebelum kita belajar Kotlin, teman-teman silahkan buka website <https://play.kotlinlang.org/> untuk menjalankan kode Kotlin yang nantinya akan kita buat!

1.1. Fundamental Kotlin

1.1.1. Variabel Dan Tipe data



Setelah kita liat gambar di atas, harusnya kita udah tau gasi apa itu variabel? Iya variable, tempat yang biasa kita gunakan untuk menyimpan berbagai macam tipe data dalam Bahasa Pemrograman, di Kotlin kita bisa mendeklarasikan sebuah variable sebagai berikut:

```
var hello: String = "Hello World"

val hello: String = "Hello World"
```

Anyway, ada yang notice ga penulisan var dan val di situ, kira-kira apa sih bedanya?

Setelah variabel, sekarang kita masuk ke tipe data yaa, sebelum masuk ke tipe data, coba liat gambar ini dan tebak kira-kira tipe data apa aja yang dipake di

dalam gambarini?.



Kalo variabel adalah tempat yang kita gunakan untuk menyimpan berbagai macam tipedata, berarti tipe data adalah jenis data yang disimpan di dalam variabel, contohnya kaya gambar di bawah ini:



Gambar di atas kita sebut sebagai variabel, misalkan variabel lemari dan rakSepatu. Variable lemari punya tipe data pakaian dan rakSepatu punya tipe data sepatu. Jadi kalo disimpulkan tuh:

- Variabel : lemari dan rakSepatu
- Tipe Data : pakaian dan sepatu

Sampe sini paham kan apa bedanya variabel dan tipe data?

berbagai macam tipe data pada kotlin sebagai berikut:

```
fun main() {  
    // define Char  
    val hurufDepan: Char = 'A'  
  
    // define String  
    val nama = "kevin"  
  
    // define Int  
    val umur = 20  
  
    // define double  
    val ipk = 3.8  
  
    // define boolean  
    val udahLulus = false  
  
    // define array  
    val matkulSmt7 = arrayOf("IMK", "Etika Profesi", "Deep Learning")  
}
```

- Char

Characters direpresentasikan menggunakan tipe Char. Tipe data Char hanya dapat kita gunakan untuk menyimpan karakter tunggal. Pada intinya, tipe data ini cuma bisa dipakai untuk menyimpan 1 data saja.

```
// define Char  
val hurufDepan: Char = 'A'  
val hurufDepan1 = 'K'
```

```
fun main() {  
    // define Char  
    val hurufDepan: Char = 'K'  
    var hurufDepan1 = 'K'  
}
```

- String

String merupakan tipe data yang mirip dengan Char. Ia dapat digunakan untuk menyimpan nilai berupa teks. Perbedaannya, String bisa menampung beberapa karakter di dalamnya, jadi ga cuma 1 karakter aja kaya Char.

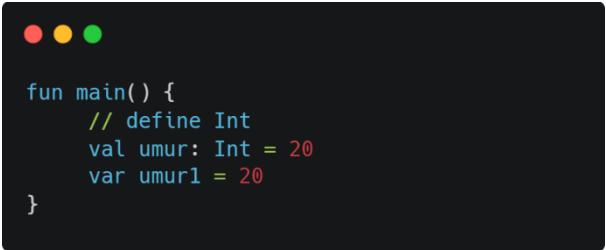
```
// define String  
val nama: String = "Kevin"  
val nama1 = "Kevin"
```

```
fun main() {  
    // define String  
    val nama: String = "Kevin"  
    var nama1 = "Kevin"  
}
```

- Integer

Int adalah tipe data yang umumnya digunakan untuk menyimpan nilai numerik. Int dapat menyimpan data dari range minimum -2147483648 dan maksimum 2147483647

```
// define Int  
val umur: Int = 20  
val umur1 = 20
```

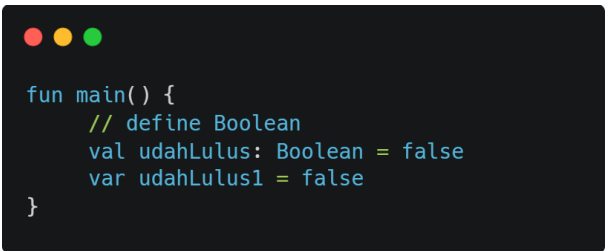


```
fun main() {  
    // define Int  
    val umur: Int = 20  
    var umur1 = 20  
}
```

- Boolean

Boolean adalah sebuah tipe data yang hanya memiliki dua nilai, yaitu true dan false.

```
// define Boolean  
val udahLulus: Boolean = false  
val udahLulus1 = false
```

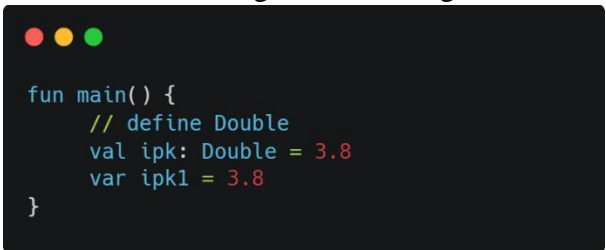


```
fun main() {  
    // define Boolean  
    val udahLulus: Boolean = false  
    var udahLulus1 = false  
}
```

- Double

Double adalah tipe data yang umumnya digunakan untuk menyimpan nilai desimal. Double digunakan untuk menyimpan nilai numerik pecahan sampai dengan maksimal 15-16 angka di belakang koma.

```
// define Double  
val ipk: Double = 3.8  
val ipk1 = 3.8
```



```
fun main() {  
    // define Double  
    val ipk: Double = 3.8  
    var ipk1 = 3.8  
}
```

- Array

Array yakni tipe data yang memungkinkan kita untuk menyimpan

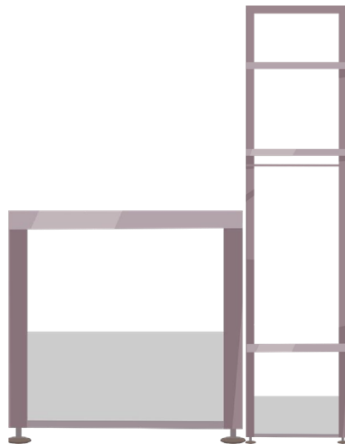
beberapa objek didalam sebuah variable:

```
// define Array
```

```
val matkulSmt7 =  
arrayOf("IMK", "Etika  
Profesi", "Skripsi")
```

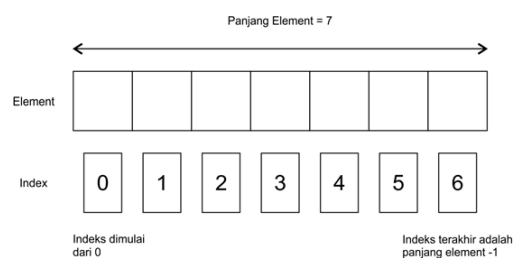
```
fun main() {  
    // define Array  
    val matkulSmt7 = arrayOf("IMK", "Etika  
Profesi", "Deep Learning")  
}
```

Perlu diketahui, array ini cukup unik karena berbeda dengan tipe data lain yang cuman bisa menyimpan 1 jenis data, sedangkan array ini bisa menyimpan berbagai macam data dengan jumlah banyak, ilustrasinya seperti:



Kalau kita mendeklarasikan variabel dengan tipe data int, kita itu kayak menyediakan lemari dengan 1 rak, kayak rak di atas ini. **Tapi** kalau kita mendeklarasikan variabel pakai tipe data array, kita tuh kayak beli lemari yang punya beberapa rak kayak gambar di atas ini. Jadi mending mana, array atau variabel biasa?

Terakhir, array juga memiliki urutan data atau yang biasa kita sebut sebagai **Index**. Index ini dimulai dari urutan 0, dan index terakhir berisi panjang elemen dikurangi satu. Jadi misal panjang elemennya 7, ya index terakhirnya 6.



1.1.2.If Else (Penyelesaian Kondisi)



Bayangkan kamu mau mengantarkan nasi syukuran ke rumah tetangga yang jumlahnya 3, pasti kita bakal menentukan tetangga mana yang rumahnya bakal kita kunjungin duluan. Nah secara ga langsung temen-temen udah menggunakan konsep if else!

Jadi gampangnya, if else itu konsep pengambilan keputusan, di mana kita akan menentukan mana yang akan kita taruh di urutan paling atas, hingga urutan paling akhir. Sedangkan untuk penulisan if else di kotlin seperti:

```
val udahLulus = false
if (udahLulus == true) {
    println("sinjay gasi?")
} else {
    println("kapan lulus kak? hehe")
}
```

```
fun main() {
    val udahLulus = false
    if (udahLulus == true) {
        print("Sinjay gasi?")
    } else {
        print("Kapan lulusnya kak?")
    }
}
```

Namun, ada versi lain dari if else, yaitu if – else if – else, bedanya apa dengan if else? Bedanya hanya di pengkondisiannya saja, jadi statement pengkondisiannya lebih banyak dari if else yang cuman 2 pilihan, contohnya seperti:

```
val semesterBerapa = 8
if (semesterBerapa == 6) {
    println("jangan lupa kp kak")
} else if (semesterBerapa == 7) {
    println("kak udah ngambil")
}
```

```
val semesterBerapa = 8
if (semesterBerapa == 6) {
    println("jangan lupa kp kak")
} else if (semesterBerapa == 7) {
    println("kak udah ngambil skripsi?")
} else if (semesterBerapa == 8) {
    println("kak gimana skripsinya?")
} else {
    println("kapan lulus kak? hehe")
}
```

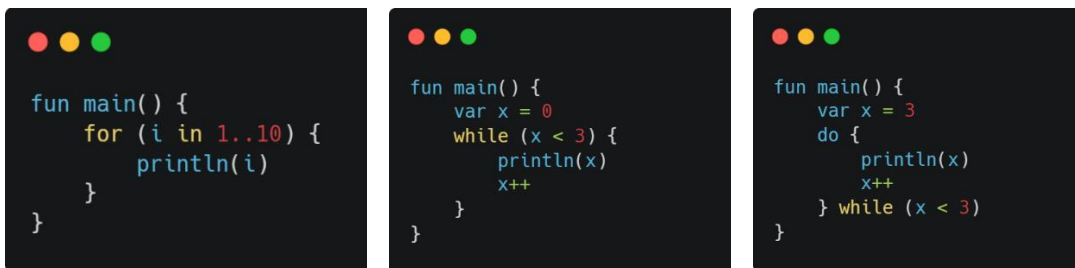
```

        skripsi?")
    } else if (semesterBerapa ==
        8) { println("kak gimana
        skripsinya?")
    } else {
        println("kapan lulus kak? hehe")
    }
}

```

1.1.3. Looping

Loop atau dalam bahasa Indonesia disebut putaran/perulangan adalah kondisi ketika kita akan menampilkan data yang banyak berulang kali. Bayangin ketika kita perlu mencetak kalimat 100x pada program. Bisa aja kita deklarasikan sampai 100x, tapi nanti jari kita yang jadi keriting. Kotlin punya 3 versi looping atau perulangan, yaitu:



Ada yang tau bedanya perulangan pertama, kedua dan ketiga (dibaca dari kiri ke kanan)?

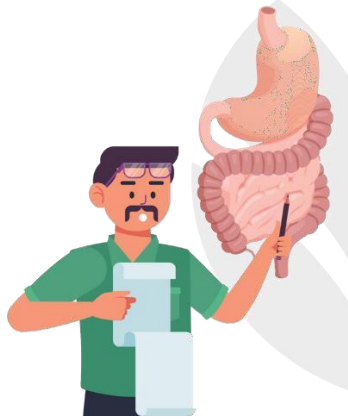
Perulangan pertama itu for loop, fungsinya untuk melakukan perulangan di mana kita sudah tau pasti perulangannya akan terjadi berapa kali, misalnya di kode tersebut perulangannya terjadi 1 sampai 10.

Perulangan kedua itu while, fungsinya untuk melakukan perulangan di mana kita belum tau pasti perulangannya bakal terjadi berapa kali. bisa saja perulangan itu hanya terjadi 1x, tapi di sisi lain bisa saja perulangannya terjadi 1000x, semua tergantung dari kondisi yang dipakai di dalam kurung while.

Perulangan ketiga itu do while, sama saja seperti while tetapi bedanya dia melakukan perintah dulu kemudian mengecek kondisi yang ada di dalam kurung while. Sedangkan while mengecek kondisi terlebih dahulu dan kemudian menjalankan perintah

1.1.4. Function

Function merupakan blok kode yang ditulis buat melakukan tugas tertentu. Pada tingkat yang lebih tinggi, suatu function, mengambil beberapa input tambahan yang disebut sebagai parameter, yang melakukan tugas tertentu pada input dan akhirnya bisa mengembalikan nilai.



Agar kita familiar dengan Function, coba kamu bayangkan sistem pencernaan manusia. Sistem pencernaan terdiri dari berbagai organ yang berpartisipasi dalam proses pencernaan. Setiap organ melakukan function spesifik, artinya mereka memiliki tugas yang berbeda-beda sesuai dengan kebutuhannya.

Disisi lain, blok function, yang merupakan bagian dari sistem, digunakan buat menangani tugas tertentu yang diperlukan untuk kelancaran function seluruh sistem. Function/method bisa memecah program menjadi sub-sub program, sehingga kita bisabikin program lebih efisien.

Cara membuat function di Kotlin caranya sebagai berikut:

```
fun main() {  
    biodata()  
}  
  
fun biodata() {  
    val nama = "Kevin"  
    val universitas =  
        "UniversitasTrunojoyo"
```

```
fun main() {  
    biodata()  
}  
  
fun biodata() {  
    val nama = "Kevin"  
    val universitas = "Universitas Trunojoyo"  
    print(nama + " dari " + universitas)  
}
```

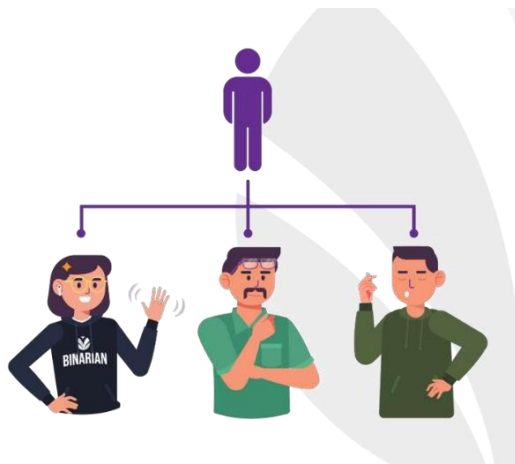
```

        print(na
ma + " dari " +
universitas)
}

```

1.1.5. Class

Manusia bisa kita ibaratkan sebagai sebuah CLASS. Manusia itu ada banyak karakter yang kita kenal. Misalnya, bestie kita ada Sabrina, Mas Gun, David.



Misal nih, dari 3 manusia itu, kita pilih sabrina untuk kita buat biodatanya. Nama dia Sabrina, perempuan dengantinggi badan 160 cm dan berat badan 54 kg. Kalau di Kotlin, nama, jenis kelamin, tinggi badan, dan berat badan Sabrina ini yang kita sebut sebagai **PROPERTY**.

Dalam aktivitas kesehariannya, Sabrina ini demen banget jogging dan berenang. Nah, berbagai aktivitas yang dilakukan sama Sabrina ini disebut sebagai **Method/Function**.

Jadi, apa itu class?

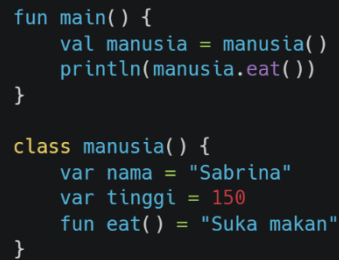


Class adalah blueprint (cetak biru/tempat) yang bisa kita gunakan untuk membuat sebuah objek. Contohnya kaya di atas, kita bikin objek yang

bernama sabrina. Nah di dalam class itu ada yang namanya property, kelas ini bisa menentukan property dan method apa aja yang dimiliki oleh sebuah class. Contoh dari property dan metode sudah dijelaskan di atas.

Adapun cara mendeklarasikan Class dalam Kotlin sebagai berikut:

```
fun main() {  
    val manusia = manusia()  
    println(manusia.eat())  
}  
  
class manusia() {  
    var nama = "Sabrina"  
    var tinggi = 158  
    fun eat() = "Suka makan"  
}
```

A screenshot of a code editor window with a dark background and three colored window control buttons (red, yellow, green) in the top-left corner. The code is written in Kotlin and is syntax-highlighted. It defines a class named 'manusia' with two properties, 'nama' and 'tinggi', and a method 'eat'. The 'main' function creates an instance of 'manusia' and calls the 'eat' method.

```
fun main() {  
    val manusia = manusia()  
    println(manusia.eat())  
}  
  
class manusia() {  
    var nama = "Sabrina"  
    var tinggi = 150  
    fun eat() = "Suka makan"  
}
```

Di sini kita mendeklarasikan class manusia, class manusia memiliki property namadan tinggi, selain itu dia juga memiliki method eat yang bisa kita panggil jika kita butuhkan.

BAB III

TUGAS PENDAHULUAN

3.1. Soal

1. Apa itu kotlin?
2. Mengapa diberi nama kotlin?
3. Apa maksud dari android kotlin first approach?

3.2. Jawaban

1. Kotlin adalah bahasa pemrograman modern yang bersifat open-source dan statis, berjalan di atas platform Java Virtual Machine (JVM). Bahasa ini menawarkan berbagai keunggulan dibandingkan dengan Java, seperti sintaks yang lebih ringkas, mudah dibaca, dan aman. Kotlin juga kompatibel dengan Java, sehingga developer dapat menggunakan library Java yang sudah ada dalam proyek Kotlin mereka.
2. Nama Kotlin berasal dari Pulau Kotlin, sebuah pulau di dekat St. Petersburg, Rusia. Pulau ini terkenal dengan bentengnya yang kokoh dan aman. Pemberian nama Kotlin pada bahasa pemrograman ini dimaksudkan untuk mencerminkan karakteristiknya yang aman dan terpercaya.
3. Android Kotlin First Approach merupakan sebuah inisiatif dari Google untuk menjadikan Kotlin sebagai bahasa pemrograman utama untuk pengembangan aplikasi Android. Hal ini didorong oleh berbagai keunggulan yang dimiliki Kotlin, seperti yang telah disebutkan sebelumnya. Dengan menggunakan Kotlin, developer dapat mengembangkan aplikasi Android dengan lebih mudah, efisien, dan aman.

BAB IV

IMPLEMENTASI

4.1 Soal

Buatlah sebuah program (program dibebaskan apa saja) yang memuat variabel, if else, looping, class, dan function di dalam program tersebut!

4.2 Source Code

```
open class Mobil(val nama: String, val biayaSewa: Int)

fun main() {
    // Daftar mobil yang tersedia
    val mobilTersedia = listOf(
        Mobil("Avanza", 200000),
        Mobil("Xenia", 200000),
        Mobil("Mobilio", 250000),
        Mobil("Innova", 350000),
        Mobil("Brio", 195000),
        Mobil("Fortuner", 450000),
        Mobil("Pajero", 450000),
        Mobil("Alphard", 550000),
        Mobil("Hiace", 650000)
    )

    // Variabel untuk menyimpan pilihan mobil
    var pilihanMobil: Mobil?

    // Variabel untuk menyimpan lama sewa
    var lamaSewa: Int?

    // Variabel untuk menyimpan total biaya
    var totalBiaya = 0

    // Menampilkan daftar mobil yang tersedia
    println("Daftar mobil yang tersedia:")
    for (mobil in mobilTersedia) {
        println("- ${mobil.nama}")
    }

    // Meminta pilihan mobil dari user
    println("Pilih mobil yang ingin Anda sewa (masukkan nama mobil):")
    val inputMobil = readLine()!!.trim()
```

```

// Mencari mobil yang dipilih user
pilihanMobil = mobilTersedia.find { it.nama == inputMobil }

// Menampilkan pesan jika mobil tidak ditemukan
if (pilihanMobil == null) {
    println("Mobil tidak ditemukan!")
    return
}

// Meminta lama sewa dari user
println("Masukkan lama sewa (hari):")
val inputLamaSewa = readLine()!!.toInt()

// Memastikan lama sewa tidak kurang dari 1
if (inputLamaSewa < 1) {
    println("Lama sewa minimal 1 hari!")
    return
}

lamaSewa = inputLamaSewa

// Menghitung total biaya
for (i in 1..lamaSewa) {
    totalBiaya += pilihanMobil.biayaSewa
}

// Fungsi untuk mencetak detail mobil
fun printDetailMobil(mobil: Mobil, lamaSewa: Int,
totalBiaya: Int) {
    println("Nama mobil: ${mobil.nama}")
    println("Biaya sewa per hari: Rp${mobil.biayaSewa}")
    println("Lama sewa: $lamaSewa hari")
    println("Total biaya: Rp$totalBiaya")
}

// Memproses pilihan user
if (pilihanMobil != null && lamaSewa != null) {
    printDetailMobil(pilihanMobil, lamaSewa, totalBiaya)
}
}

```


4.3. Output

```
PROBLEMAS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
PS D:\Cooleyah\perograman\gitclone\perograman-bergerak-44-2024> cd "d:\Cooleyah\perograman\gitclone\perograman-bergerak-44-2024\modul-1\210441100116-HilmiSholeh\" ; if ($?) { kotlinc TugasModul1.kt -include-runtime -d Tu
gashModul1.jar } ; if ($?) { kotlin TugasModul1.jar }
TugasModul1.kt:71:9: warning: condition 'pilihantMobil != null' is always 'true'
    if (pilihantMobil != null && lamaSewa != null) {
        ^
TugasModul1.kt:71:33: warning: condition 'lamaSewa != null' is always 'true'
    if (pilihantMobil != null && lamaSewa != null) {
        ^
Daftar mobil yang tersedia:
- Avanza
- Xenia
- Mobilio
- Innova
- Brio
- Fortuner
- Pajero
- Alphard
- Hilace
Pilih mobil yang ingin Anda sewa (masukkan nama mobil):
Hilace
Masukkan lama sewa (hari):
4
Nama mobil: Hilace
Biaya sewa per hari: Rp650000
Lama sewa: 4 hari
Total biaya: Rp2600000
PS D:\Cooleyah\perograman\gitclone\perograman-bergerak-44-2024\modul-1\210441100116-HilmiSholeh> []
```

4.4. Penjelasan Program

Program di atas adalah contoh program Kotlin untuk menyewa mobil. Program ini memungkinkan user untuk memilih mobil yang ingin disewa dan lama sewanya. Program kemudian akan menghitung total biaya sewa dan menampilkan detail mobil, lama sewa, dan total biaya.

Berikut adalah penjelasan dari program di atas:

- **Kelas Mobil:** Kelas Mobil mendefinisikan dua properti: nama (nama mobil) dan biayaSewa (biaya sewa per hari).
- **Fungsi main():** Fungsi main() adalah fungsi utama program. Fungsi ini:
 - Mendefinisikan daftar mobil yang tersedia (mobilTersedia).
 - Meminta user untuk memilih mobil yang ingin disewa (inputMobil).
 - Mencari mobil yang dipilih user dalam daftar mobilTersedia (pilihanMobil).
 - Meminta user untuk memasukkan lama sewa (inputLamaSewa).
 - Memastikan lama sewa tidak kurang dari 1.
 - Menghitung total biaya sewa (totalBiaya).
 - Mencetak detail mobil, lama sewa, dan total biaya (printDetailMobil).
- **Fungsi printDetailMobil():** Fungsi printDetailMobil mencetak detail mobil, lama sewa, dan total biaya.

BAB V

PENUTUP

5.1 Analisa

Materi Kotlin dasar telah mengantarkan kita pada pemahaman konsep-konsep fundamental yang penting untuk memulai pengembangan aplikasi dengan Kotlin. Konsep-konsep ini meliputi struktur dasar program, variabel, tipe data, operator, percabangan, dan perulangan. Dengan mempelajari materi ini, kita telah mendapatkan fondasi yang kokoh untuk melangkah lebih jauh dalam dunia pemrograman Kotlin.

Keunggulan Kotlin dibandingkan bahasa pemrograman lain seperti Java menjadikannya pilihan menarik bagi para developer. Sintaksnya yang ringkas dan mudah dibaca memudahkan proses belajar dan pengembangan aplikasi bagi pemula. Fitur keamanannya yang baik membantu developer dalam membuat aplikasi yang lebih kokoh dan terhindar dari celah keamanan. Interoperabilitasnya dengan Java memungkinkan developer untuk menggunakan library Java yang sudah ada dalam proyek Kotlin mereka, sehingga proses pengembangan aplikasi menjadi lebih efisien.

5.2 Kesimpulan

1. Struktur dasar program Kotlin, seperti package, class, object, function, dan lain sebagainya.
2. Berbagai tipe data yang tersedia di Kotlin, seperti Int, String, Boolean, dan lain sebagainya.
3. Operator-operator yang digunakan dalam Kotlin, seperti operator aritmatika, operator logika, dan lain sebagainya.
4. Cara membuat percabangan dalam program Kotlin dengan menggunakan statement if, else, dan when.
5. Cara membuat perulangan dalam program Kotlin dengan menggunakan statement for, while, dan do-while.