

LAPORAN RESMI
MODUL I
KOTLIN DASAR
PEMROGRAMAN BERGERAK



NAMA	: IMAM ARISHANDI IRFANTO
N.R.P	: 220441100034
DOSEN	: ACHMAD ZAIN NUR, S.kom,M.T
ASISTEN	: MOCHAMMAD SYAHRUL ABIDIN
TGL PRAKTIKUM	: 22 MARET 2024

Disetujui : .. Maret 2024
Asisten

MOCHAMMAD SYAHRUL ABIDIN
21.04.411.00001



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

BAB I

PENDAHULUAN

1.1 Latar Belakang

Kotlin adalah bahasa pemrograman modern yang dirancang untuk berjalan pada mesin virtual Java (JVM), serta dapat dikompilasi menjadi kode sumber JavaScript atau kode sumber native. Dikembangkan oleh JetBrains, Kotlin menawarkan sintaksis yang ekspresif, aman, dan mudah dipahami, sehingga membuatnya menjadi pilihan yang menarik untuk pengembangan berbagai jenis aplikasi, mulai dari pengembangan aplikasi Android hingga pengembangan aplikasi web.

Sebagai bahasa yang interoperabel dengan Java, Kotlin memiliki dukungan yang luas dari komunitas pengembang, serta memiliki integrasi yang mulus dengan perangkat lunak dan framework yang ada dalam ekosistem Java. Hal ini membuatnya menjadi pilihan yang kuat bagi pengembang yang ingin meningkatkan produktivitas mereka tanpa harus meninggalkan investasi yang sudah ada dalam ekosistem Java.

Selain itu, keamanan merupakan aspek yang diutamakan dalam desain Kotlin. Kotlin menawarkan fitur-fitur seperti null safety, tipe data yang aman, dan interoperabilitas yang kuat dengan Java, sehingga membantu pengembang dalam mencegah kesalahan umum yang terjadi dalam pengembangan perangkat lunak. Dengan demikian, Kotlin tidak hanya memberikan produktivitas yang tinggi bagi pengembang, tetapi juga membantu meningkatkan kualitas dan keamanan dari aplikasi yang dikembangkan. Dalam ekosistem yang semakin kompleks dan berubah-ubah, Kotlin menjanjikan kesinambungan dan keandalan dalam pengembangan perangkat lunak untuk masa depan.

1.2 Tujuan

1. Mahasiswa mampu memahami konsep dasar kotlin.
2. Mahasiswa mampu menggunakan pengetahuan dasar kotlin untuk mengembangkan aplikasi sederhana.
3. Mahasiswa mampu beradaptasi dengan bahasa kotlin

BAB II

DASAR TEORI

2.1 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 gambar ini?



Kalo variabel adalah tempat yang kita gunakan untuk menyimpan berbagai macam tipe data, 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?

Anyway, di Kotlin kita bisa nulis berbagai macam tipe data 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")  
}
```

Kita bedah satu per satu kali, ya?

Char

Characters direpresentasikan menggunakan tipe Char. Tipe data Char hanya dapat kita gunakan untuk menyimpan karakter tunggal. Intinya sih, tipe data ini cuman bisa dipake buat nyimpen 1 karakter aja.

```
// 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 cuman 1 karakter aja kaya Char.

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

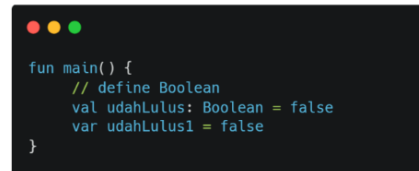


```
fun main() {  
    // define String  
    val nama: String = "Kevin"  
    var nama1 = "Kevin"  
}
```

Boolean

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

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

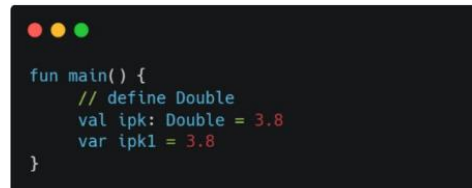


```
fun main() {  
    // define Boolean  
    val udahLulus: Boolean = false  
    var udahLulus1 = false  
}
```

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
```

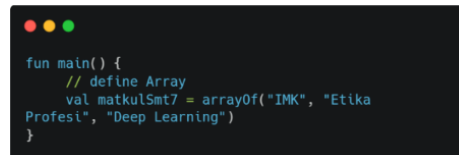


```
fun main() {  
    // define Double  
    val ipk: Double = 3.8  
    var ipk1 = 3.8  
}
```

Array

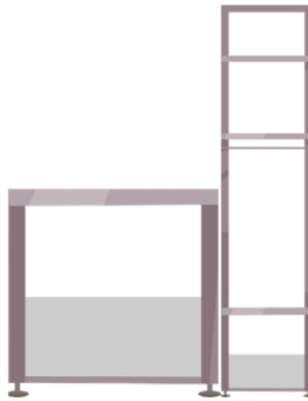
Array yakni tipe data yang memungkinkan kita untuk menyimpan beberapa objek di dalam sebuah variable.

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



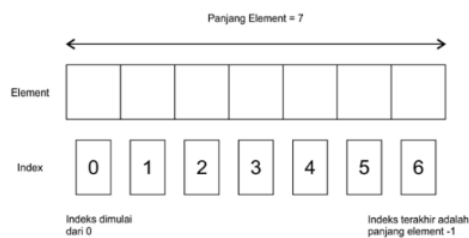
```
fun main() {  
    // define Array  
    val matkulSmt7 = arrayOf("IMK", "Etika  
Profesi", "Deep Learning")  
}
```

By the way, array ini cukup unik karena berbeda dengan tipe data lain yang Cuma bisa menyimpan 1 jenis data, sedangkan array ini bisa menyimpan berbagai macam data dengan jumlah banyak, ilustrasinya sih gini:



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.



2.2 If Else (Penyeleksian Kondisi)



Bayangin kamu mau nganterin 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 gampangny sih, if else itu konsep pengambilan keputusan, di mana kita akan

menentukan mana yang akan kita taruh di urutan paling atas, hingga urutan paling akhir. Nah untuk penulisan if else di kotlin tuh kaya gini:

```
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?")
    }
}
```

Anyway, ada versi lain dari if else, yaitu if – else if – else, bedanya apa sih kak dengan if else? Bedanya cuman di pengkondisiannya aja, jadi statement pengkondisiannya lebih banyak dari if else yang cuman 2 pilihan, contohnya kaya gini:

```
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")
}
```

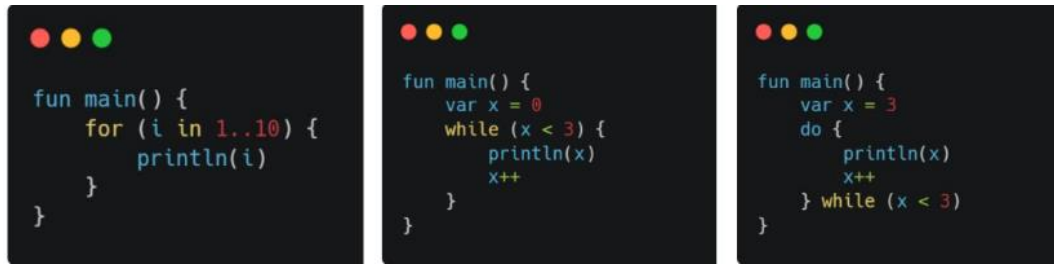


```
val semesterBerapa = 8
if (semesterBerapa == 6) {
    println("jangan lupa kp kak")
} else if (semesterBerapa == 7) {
    println("kak udah ngambil skripsi?")
} else if (semesterBerapa == 8) {
    println("kak gimana skripsinya?")
} else {
    println("kapan lulus kak? hehe")
}
```

2.3 Looping

Loop atau dalam bahasa Indonesia disebut putaran/perulangan adalah kondisi ketikakita 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:



```

fun main() {
    for (i in 1..10) {
        println(i)
    }
}

fun main() {
    var x = 0
    while (x < 3) {
        println(x)
        x++
    }
}

fun main() {
    var x = 3
    do {
        println(x)
        x++
    } while (x < 3)
}

```

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 udah tau pasti perulangannya bakal terjadi berapa kali, misalnya di kode tersebut tuh perulangannya terjadi 1 sampai 10.

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

Perulangan ketiga itu do while, sama aja kaya while tapi bedanya dia ngelakuin perintah dulu baru ngecek kondisi yang ada di dalam kurung while. Sedangkan kalau while dia ngecek kondisi dulu baru ngejalanin perintah.

2.4 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.



Supaya familiar dengan Function, coba kamu bayangin 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.

Anyway, 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 bisa bikin program lebih efisien.

Terus gimana kak cara buat function di Kotlin? Nah caranya gini...

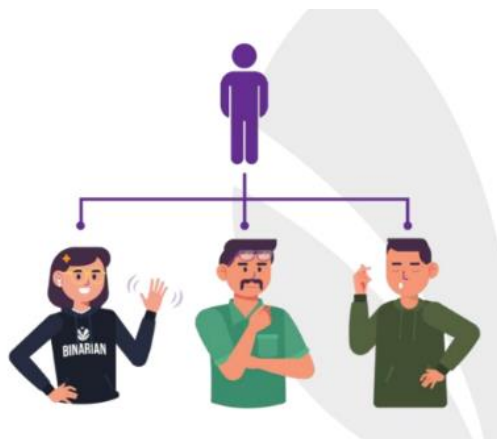
```
fun main() {  
    biodata()  
}  
  
fun biodata() {  
    val nama = "Kevin"  
    val universitas = "Universitas  
Trunojoyo"  
    print(nama + " dari " +  
    universitas)  
}
```



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

2.5 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 dengan tinggi 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.

Nah sekarang kita bakal bahas cara mendeklarasikan class di dalam Kotlin.

```
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 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 shows a main function and a class named 'manusia'. The main function calls 'manusia()' and 'manusia.eat()'. The 'manusia' class has properties 'nama' and 'tinggi', and a method 'eat'.

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

SR

BAB III

TUGAS PENDAHULUAN

3.1 Soal

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

3.2 Jawaban

1. Kotlin adalah bahasa pemrograman modern yang diciptakan JetBrains, Kotlin berjalan diatas mesin virtual java [JVM] dan juga dapat dikompilasi menjadi kode java script atau native. Kotlin juga menggunakan program fungsional dan OOP.
2. Kotlin dinamai berdasarkan pulau di Rusia bernama Kotlin JetBrains, pengembang bahasa tersebut, memilih nama tersebut karena pulau Kotlin terletak dengan St. Petersburg, tempat kantor pusat perusahaan ini berada.
3. Pendekatan Kotlin pertamakali dalam pengembangan android adalah fokus utama pada penggunaan bahasa pemrograman Kotlin daripada java untuk membangun aplikasi android. Ini menekankan manfaat Kotlin dalam meningkatkan produktifitas dan kualitas kode.

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!

Beberapa ide program yang bisa teman-teman buat

- Toko Online berbasis CLI
- Penyewaan lapangan futsal
- Dll

Contoh output program (hanya contoh dan tidak diwajibkan untuk ditiru)

```
Selamat Datang Di Warung Binar
=====
Our Menu :
1. Ayam Bakar = Rp50.000/Porsi
2. Ayam Goreng = Rp40.000/Porsi
3. Ayam Geprek = Rp40.000/Porsi
4. Kulit Ayam = Rp15.000/Porsi
5. Sate Usus Ayam = 5.000/Porsi

Kamu Mau Beli Apa ? 1/2/3/4/5
1
Kamu Memilih Menu 1
Nama Menu : Ayam Bakar
Harga : 50000.0
=====
Silahkan lakukan Pembayaran Atas Pemesanan Ayam Bakar !!!
50000
Total Harga : 50000.0
Pembayaran Gagal, Silahkan Input Pembayaran Ulang!!!
=====
Silahkan lakukan Pembayaran Atas Pemesanan Ayam Bakar !!!
50000
Total Harga : 50000.0
Pembayaran Berhasil
|
```

4.2 Source Code

```
fun main() {
    clearConsole()
    println("=====
=====")
    println("          Selamat datang di Toko Penyewaan Alat
Mendaki!          ")
    println("=====
=====")

    val databaseAlat = listOf(
        AlatMendaki("Tenda", 50000),
```

```

        AlatMendaki("Sleeping bag", 30000),
        AlatMendaki("Ransel", 20000),
        AlatMendaki("Sepatu gunung", 40000),
        AlatMendaki("Jaket gunung", 35000),
        AlatMendaki("Senter", 15000),
        AlatMendaki("Kompas", 10000),
        AlatMendaki("Peralatan masak", 25000),
        AlatMendaki("Kotak P3K", 10000)
    )

    val pilihanAlat: MutableList<Int> = mutableListOf()
    val jumlahHari: MutableList<Int> = mutableListOf()

    var isLanjutSewa = true

    while (isLanjutSewa) {
        // Menampilkan daftar alat mendaki
        tampilkanDaftarAlat(databaseAlat)

        // Memilih alat mendaki
        pilihAlat(pilihanAlat)

        // Meminta jumlah hari sewa
        print("Masukkan jumlah hari sewa untuk semua alat
yang dipilih: ")
        val jumlahHariSewa = readLine()?.toIntOrNull() ?: 0
        jumlahHari.addAll(List(pilihanAlat.size) {
            jumlahHariSewa })
        println("=====
=====")

        // Menghitung total harga sewa
        val totalHarga = hitungTotalHargaSewa(databaseAlat,
            pilihanAlat, jumlahHari)

        // Menampilkan daftar alat yang disewa
        println("Daftar alat yang disewa:")
        for (i in pilihanAlat.indices) {
            if (i < jumlahHari.size && pilihanAlat[i] - 1 <
                databaseAlat.size) {
                println("${databaseAlat[pilihanAlat[i] -
1].nama} - ${jumlahHari[i]} hari")
            }
        }
        println("=====
=====")
    }

```

```

        // Menampilkan total harga sewa
        println("Total harga sewa: Rp$totalHarga")
        println("=====
=====")

        // Meminta pembayaran dari pengguna
        var bayar = true

        while (bayar) {
            print("Masukkan jumlah pembayaran: Rp")
            val pembayaran = readLine()?.toIntOrNull() ?: 0

            // Menghitung kembalian
            val kembalian = pembayaran - totalHarga
            if (kembalian >= 0) {
                println("Kembalian: Rp$kembalian")
                bayar = false
            } else {
                println("Maaf, uang pembayaran tidak
mencukupi.")
                bayar = true
            }
        }
        resetRiwayatSewa(pilihanAlat, jumlahHari)

        // Menanyakan apakah ingin lanjut sewa
        print("Apakah ingin lanjut sewa? (y/n): ")
        val lanjutSewa = readLine()
        isLanjutSewa = lanjutSewa.equals("y", ignoreCase =
true)
        println("=====
=====")
    }

    println("                Terima kasih telah
berkunjung!")
    println("=====
=====")
}

fun tampilkanDaftarAlat(databaseAlat: List<AlatMendaki>) {
    for (i in databaseAlat.indices) {
        println("${i + 1}. ${databaseAlat[i].nama} -
Rp${databaseAlat[i].hargaSewa}/hari")
    }
}

```

```

fun pilihAlat(pilihanAlat: MutableList<Int>) {
    println("=====
=====")
    print("Pilih alat yang ingin disewa (pisahkan dengan
koma): ")
    val inputPilihanAlat = readLine()
    val alatTerpilih =
inputPilihanAlat?.split(",")?.mapNotNull {
it.trim().toIntOrNull() }
    alatTerpilih?.let { pilihanAlat.addAll(it) }
    println("=====
=====")
}
fun hitungTotalHargaSewa(databaseAlat: List<AlatMendaki>,
pilihanAlat: List<Int>, jumlahHari: List<Int>): Int {
    var totalHarga = 0
    for (i in pilihanAlat.indices) {
        if (i < jumlahHari.size) {
            totalHarga += databaseAlat[pilihanAlat[i] -
1].hargaSewa * jumlahHari[i]
        }
    }
    return totalHarga
}

data class AlatMendaki(val nama: String, val hargaSewa: Int)

fun resetRiwayatSewa(pilihanAlat: MutableList<Int>,
jumlahHari: MutableList<Int>) {
    pilihanAlat.clear()
    jumlahHari.clear()
}
fun clearConsole() {
    print("\u001b[H\u001b[2J")
    System.out.flush()
}

```

Penjelasan : kali ini saya membuat penyewaan alat dari yang berisi alat-alat yang digunakan untuk mendaki, dengan menggunakan class untuk mengelompokkan data barang yang di sewa, lalu ada beberapa fungsi yang saya gunakan salah satunya untuk mereset sewa yang berfungsi menghapus riwayat sewa Ketika user memilih menyewa Kembali, ada juga hitung harga sewa yang berfungsi untuk menghitung harga total barang yang di sewa, perulangan untuk menampilkan beberapa fungsi, dan terakhir perkondisian.

```
=====
Selamat datang di Toko Penyewaan Alat Mendaki!
=====
1. Tenda - Rp50000/hari
2. Sleeping bag - Rp30000/hari
3. Ransel - Rp20000/hari
4. Sepatu gunung - Rp40000/hari
5. Jaket gunung - Rp35000/hari
6. Senter - Rp15000/hari
7. Kompas - Rp10000/hari
8. Peralatan masak - Rp25000/hari
9. Kotak P3K - Rp10000/hari
=====
Pilih alat yang ingin disewa (pisahkan dengan koma): 1,2,3,6
=====
Masukkan jumlah hari sewa untuk semua alat yang dipilih: 2
=====
Daftar alat yang disewa:
Tenda - 2 hari
Sleeping bag - 2 hari
Ransel - 2 hari
Senter - 2 hari
=====
Total harga sewa: Rp230000
=====
Masukkan jumlah pembayaran: Rp250000
Kembalian: Rp20000
Apakah ingin lanjut sewa? (y/n): n
=====
Terima kasih telah berkunjung!
=====
```


BAB V

PENUTUP

5.1 Analisa

Dalam pemrograman, variabel itu seperti kotak penyimpanan untuk berbagai macam informasi. Misalnya, kotak untuk menyimpan nama, umur, atau nilai angka. Tipe data adalah jenis-jenis informasi yang bisa disimpan dalam kotak itu. Ada kotak khusus untuk menyimpan huruf, teks panjang, atau bahkan "ya" dan "tidak". Kalau kita mau bikin program memilih antara dua pilihan, misalnya "jika ini benar, lakukan ini, kalau tidak, lakukan itu", kita pakai yang namanya if else. Jika kita punya lebih dari dua pilihan, kita bisa pakai if-else if-else. Kalau kita mau program melakukan sesuatu berulang-ulang, misalnya mencetak hal yang sama berkali-kali, kita pakai yang namanya loop. Ada tiga jenis loop: for loop (untuk jika kita sudah tahu berapa kali perulangannya), while loop (jika kita belum pasti berapa kali), dan do-while loop (mirip while loop, tapi minimal dijalankan sekali). Fungsi adalah seperti resep atau instruksi yang berisi langkah-langkah tertentu. Misalnya, kita punya resep untuk mencetak "halo", setiap kali kita mau, tinggal ikuti instruksi yang ada di resep itu. Terakhir, class adalah seperti cetakan untuk membuat objek. Misalnya, jika kita ingin membuat objek "orang", kita bisa menggunakan cetakan yang sudah ada, dan kemudian mengisi detailnya seperti nama, umur, dll.

Jadi, intinya, dalam pemrograman itu seperti punya alat dan instruksi untuk membuat, menyimpan, dan melakukan berbagai hal dengan data dan informasi. Semoga membantu!

5.2 Kesimpulan

1. JFrame dan Komponen GUI Dasar Aplikasi ini menggunakan JFrame sebagai jendela utama.
2. Interaktivitas Melalui Antarmuka Grafis Pengguna dapat berinteraksi dengan program ini melalui antarmuka grafis yang intuitif.
3. membuat halaman sederhana menggunakan komponen Swing dengan tool visual editor.