

**LAPORAN PRAKTIKUM ALGORITMA
DAN PEMROGRAMAN 1**

MODUL 10

ELSE-IF



Disusun oleh:

Ahmad Malik Arrayyan

109082500116

S1IF-13-02

Asisten Praktikum

Adithana dharma putra

Alfin Ilham Berlianto

PROGRAM STUDI S1 INFORMATIKA

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2025

LATIHAN KELAS – GUIDED

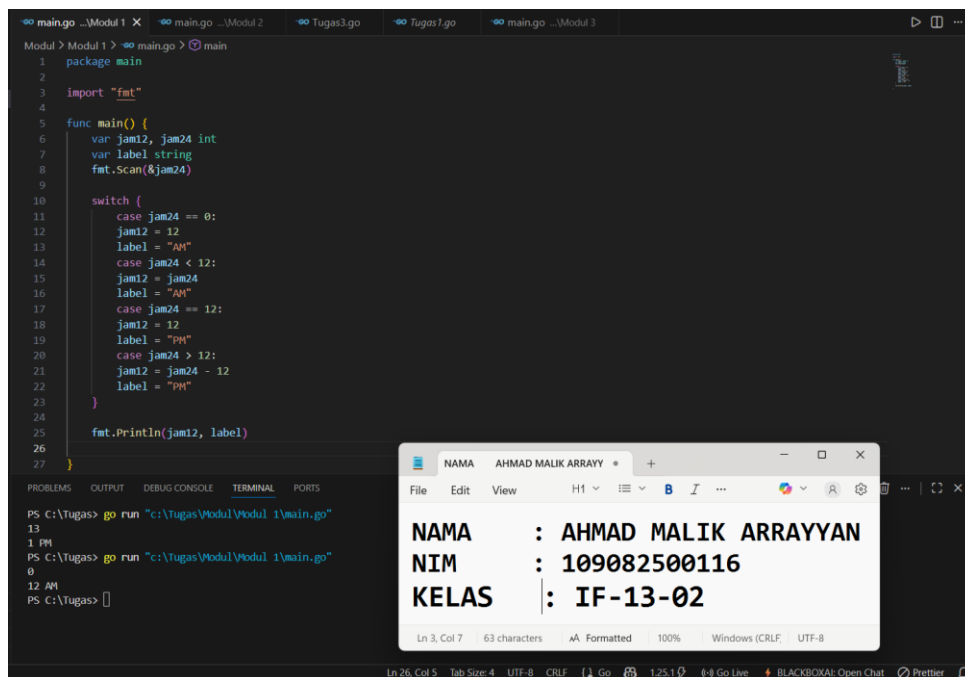
1. Guided 1 Source Code

```
package main

import "fmt"

func main() {
    var jam12, jam24 int
    var label string
    fmt.Scan(&jam24)
    switch {
        case jam24 == 0:
            jam12 = 12
            label = "AM"
        case jam24 < 12:
            jam12 = jam24
            label = "AM"
        case jam24 == 12:
            jam12 = 12
            label = "PM"
        case jam24 > 12:
            jam12 = jam24 - 12
            label = "PM"
    }
    fmt.Println(jam12, label)
}
```

Screenshoot program



The screenshot shows a Go program in a VS Code editor. The program reads an integer input 'jam24' and uses a switch statement to convert it to a 12-hour format 'jam12' and a label 'AM' or 'PM'. The terminal window shows the program being run twice: first with input 13, resulting in '1 PM', and then with input 0, resulting in '12 AM'.

```
Modul > Modul 1 > main
1 package main
2
3 import "fmt"
4
5 func main() {
6     var jam12, jam24 int
7     var label string
8     fmt.Scan(&jam24)
9
10    switch {
11    case jam24 == 0:
12        jam12 = 12
13        label = "AM"
14    case jam24 < 12:
15        jam12 = jam24
16        label = "AM"
17    case jam24 == 12:
18        jam12 = 12
19        label = "PM"
20    case jam24 > 12:
21        jam12 = jam24 - 12
22        label = "PM"
23    }
24
25    fmt.Println(jam12, label)
26
27 }
```

```
PS C:\Tugas> go run "c:\Tugas\Modul\Modul 1\main.go"
13
1 PM
PS C:\Tugas> go run "c:\Tugas\Modul\Modul 1\main.go"
0
12 AM
PS C:\Tugas>
```

Deskripsi program

Program di atas membaca sebuah angka bernama jam24 dari input. Kemudian, program menggunakan switch tanpa kondisi untuk mengecek beberapa aturan waktu dalam format 24 jam. Jika nilai jam24 sama dengan 0, maka jam12 diubah menjadi 12 dan label waktu diisi "AM". Jika jam24 kurang dari 12, maka jam12 tetap sama seperti jam24 dan labelnya juga "AM". Jika jam24 sama dengan 12, maka jam12 tetap 12 tetapi labelnya berubah menjadi "PM". Jika jam24 lebih dari 12, maka jam12 dihitung dengan jam24 dikurangi 12 dan labelnya menjadi "PM". Jadi, program ini pada dasarnya mengubah waktu format 24 jam menjadi format 12 jam, lalu mencetak hasil konversi tersebut berupa jam12 dan label "AM" atau "PM".

2. Guided 2

Source Code

```
package main

import "fmt"

func main() {
    var nama_tanaman string
    fmt.Scan(&nama_tanaman)

    switch nama_tanaman {
        case "nepenthes", "drosera":
            fmt.Println("Termasuk Tanaman Karnivora.")
            fmt.Println("Asli Indonesia.")
        case "venus", "sarracenia":
            fmt.Println("Termasuk Tanaman Karnivora.")
            fmt.Println("Tidak Asli Indonesia.")
        default:
            fmt.Println("Tidak termasuk Tanaman
Karnivora.")
    }
}
```

Screenshoot program

The screenshot shows a Go program in a code editor and its execution output in a terminal. The program uses a switch statement to categorize plants based on their name. The terminal shows the results of running the program for three different inputs: 'nepenthes', 'venus', and 'karedok'.

```
2
3 import "fmt"
4
5 func main() {
6     var nama_tanaman string
7     fmt.Scan(&nama_tanaman)
8
9     switch nama_tanaman {
10        case "nepenthes", "drosera":
11            fmt.Println("Termasuk Tanaman Karnivora.")
12            fmt.Println("Asli Indonesia.")
13        case "venus", "sarracenia":
14            fmt.Println("Termasuk Tanaman Karnivora.")
15            fmt.Println("Tidak Asli Indonesia.")
16        default:
17            fmt.Println("Tidak termasuk Tanaman Karnivora.")
18        }
19    }
```

Terminal Output:

```
nepenthes
Termasuk Tanaman Karnivora.
Asli Indonesia.
PS C:\Tugas> go run "c:\Tugas\Modul 2\main.go"
venus
Termasuk Tanaman Karnivora.
Tidak Asli Indonesia.
PS C:\Tugas> go run "c:\Tugas\Modul 2\main.go"
karedok
Tidak termasuk Tanaman Karnivora.
PS C:\Tugas>
```

Output Window Content:

```
NAMA      : AHMAD MALIK ARRAYYAN
NIM       : 109082500116
KELAS    : IF-13-02
```

Deskripsi program

Program ini membaca input berupa nama tanaman, lalu menggunakan switch untuk menentukan apakah tanaman ini termasuk tanaman karnivora dan apakah berasal dari Indonesia atau tidak. Jika nama tanaman yang dimasukkan adalah "nepenthes" atau "drosera", program menampilkan bahwa tanaman ini adalah tanaman karnivora dan merupakan tanaman asli Indonesia. Sebaliknya, jika yang dimasukkan adalah "venus" atau "sarracenia", program akan tetap menyatakan bahwa tanaman ini adalah karnivora; tapi bukan tanaman asli Indonesia. Jika nama tanaman tidak masuk salah satu daftar tersebut, maka program menampilkan bahwa tanaman tersebut bukanlah tanaman karnivora. Sejatinya, program ini berfungsi sebagai pengelompokan tanaman berdasarkan jenis dan asalnya.

3. Guided 3

Source Code

```
package main

import "fmt"

func main() {
    var kendaraan string
    var durasi int
    var tarif int

    fmt.Print("Masukkan jenis kendaraan\n(Motor/Mobil/Truk): ")
    fmt.Scan(&kendaraan)

    fmt.Print("Masukkan durasi parkir (dalam jam): ")
    fmt.Scan(&durasi)

    switch {
    case kendaraan == "Motor" && durasi >= 1 && durasi <= 2:
        tarif = 7000
    case kendaraan == "Motor" && durasi > 2:
        tarif = 9000
    case kendaraan == "Mobil" && durasi >= 1 && durasi <= 2:
        tarif = 15000
    case kendaraan == "Mobil" && durasi > 2:
        tarif = 20000
    case kendaraan == "Truk" && durasi >= 1 && durasi <= 2:
        tarif = 25000
    case kendaraan == "Truk" && durasi > 2:
        tarif = 35000
    default:
```

```

        fmt.Println("Jenis kendaraan atau durasi parkir
tidak valid")
    }

    fmt.Printf("Tarif Parkir: Rp %d\n", tarif)
}

```

Screenshoot program

```

Modul 3 > Modul 3 > main
1 package main
2
3 import "fmt"
4
5 func main() {
6     var kendaraan string
7     var durasi int
8     var tarif int
9     fmt.Print("Masukkan jenis kendaraan (Motor/Mobil/Truk): ")
10    fmt.Scan(&kendaraan)
11    fmt.Print("Masukkan durasi parkir (dalam jam): ")
12    fmt.Scan(&durasi)
13
14    switch {
15    case kendaraan == "Motor" && durasi >= 1 && durasi <= 2:
16        tarif = 7000
17    case kendaraan == "Motor" && durasi > 2:
18        tarif = 9000
19    case kendaraan == "Mobil" && durasi >= 1 && durasi <= 2:
20        tarif = 15000
21    case kendaraan == "Mobil" && durasi > 2:
22        tarif = 20000
23    case kendaraan == "Truk" && durasi >= 1 && durasi <= 2:
24        tarif = 25000
25    case kendaraan == "Truk" && durasi > 2:
26        tarif = 35000
27    default:
28
29    }
30    fmt.Printf("Tarif Parkir: Rp %d\n", tarif)
31 }

```

```

PS C:\Tugas> go run "C:\Tugas\Modul\Modul 3\main.go"
Masukkan jenis kendaraan (Motor/Mobil/Truk): Motor
Masukkan durasi parkir (dalam jam): 2
Tarif Parkir: Rp 7000
PS C:\Tugas> go run "C:\Tugas\Modul\Modul 3\main.go"
Masukkan jenis kendaraan (Motor/Mobil/Truk): Mobil
Masukkan durasi parkir (dalam jam): 4
Tarif Parkir: Rp 20000
PS C:\Tugas>

```

Deskripsi program

Program ini digunakan untuk menghitung tarif parkir tergantung dari jenis kendaraan dan lamanya parkir dalam jam. Program ini akan menginstruksikan pengguna untuk mengetikkan jenis kendaraan dan lamanya parkir. Kemudian, program ini menggunakan struktur switch yang mengidentifikasi tarif sesuai dengan aturan-aturan yang sudah ditentukan untuk setiap jenis kendaraan: setiap jenis kendaraan memiliki jenis tarif untuk durasi 1–2 jam dan lainnya jika durasinya lebih dari 2 jam. Sebagai contoh, motor akan dikenakan Rp 9.000 untuk durasi 1–2 jam dan Rp 12.000 jika lebih dari 2 jam, mobil sebesar Rp 15.000 atau Rp 20.000, begitu juga truk Rp 20.000 atau Rp 35.000. Setelah kondisi yang sesuai ditemukan, program menampilkan hasil akhir berupa total tarif parkir yang harus dibayar oleh pengguna.

TUGAS

1. Tugas 1

Source code

```
package main

import "fmt"

func main() {
    var ph float64

    fmt.Print("Masukkan nilai pH: ")
    fmt.Scan(&ph)

    switch {
    case ph < 0 || ph > 14:
        fmt.Println("Nilai pH tidak valid. Nilai pH harus antara 0 dan 14.")
    case ph >= 6.5 && ph <= 8.6:
        fmt.Println("Air layak minum")
    default:
        fmt.Println("Air tidak layak minum")
    }
}

    fmt.Printf("Detail biaya: Rp. %d + Rp. %d\n", biayaKg, biayaSisa)
    fmt.Printf("Total biaya: Rp. %d\n", biayaKg+biayaSisa)
}
```


Screenshoot program

The screenshot shows a Go program in a code editor and its execution output in a terminal. The program is a simple pH checker. It prompts the user to enter a pH value. If the value is outside the range 0-14, it says 'Nilai pH tidak valid. Nilai pH harus antara 0 dan 14.' If the value is between 6.5 and 8.6, it says 'Air layak minum'. Otherwise, it says 'Air tidak layak minum'.

```
1 package main
2
3 import "fmt"
4
5 func main() {
6     var ph float64
7     fmt.Print("Masukkan nilai pH: ")
8     fmt.Scan(&ph)
9
10    switch {
11    case ph < 0 || ph > 14:
12        fmt.Println("Nilai pH tidak valid. Nilai pH harus antara 0 dan 14.")
13    case ph >= 6.5 && ph <= 8.6:
14        fmt.Println("Air layak minum")
15    default:
16        fmt.Println("Air tidak layak minum")
17    }
18 }
19
```

Terminal Output:

```
PS C:\Tugas> go run "c:\Tugas\Soal\Tugas 1\Tugas1.go"
Masukkan nilai pH: 8.6
Air layak minum
PS C:\Tugas> go run "c:\Tugas\Soal\Tugas 1\Tugas1.go"
Masukkan nilai pH: 9
Air tidak layak minum
PS C:\Tugas> go run "c:\Tugas\Soal\Tugas 1\Tugas1.go"
Masukkan nilai pH: 16
Nilai pH tidak valid. Nilai pH harus antara 0 dan 14.
PS C:\Tugas>
```

Deskripsi program

Program ini digunakan untuk menentukan apakah air layak diminum berdasarkan nilai pH yang dimasukkan oleh pengguna. Saat program dijalankan, pengguna diminta mengetikkan angka pH, kemudian program memeriksa nilai tersebut menggunakan struktur switch dengan beberapa kondisi. Jika pH kurang dari 0 atau lebih dari 14, program akan memberi tahu bahwa nilai tersebut tidak valid karena pH hanya berada dalam rentang 0–14. Jika nilai pH berada pada rentang 6.5 sampai 8.6, maka air tersebut layak diminum. Untuk nilai pH lain yang masih dalam batas valid tetapi berada di luar rentang ideal, program menyatakan air tidak layak diminum. Program ini membantu mengevaluasi kualitas air dengan logika sederhana berdasarkan standar pH normal.

2. Tugas 2

Source code

```
package main

import "fmt"

func main() {
    var jenis string
    var durasi int
    var tarif int

    fmt.Print("Masukkan jenis kendaraan (motor/mobil/truk): ")
    fmt.Scan(&jenis)

    fmt.Print("Masukkan durasi parkir (jam): ")
    fmt.Scan(&durasi)

    if durasi < 1 {
        durasi = 1
    }

    switch jenis {
    case "motor":
        tarif = 2000
    case "mobil":
        tarif = 5000
    case "truk":
        tarif = 8000
    default:
        fmt.Println("Jenis kendaraan tidak valid.")
        return
    }

    total := tarif * durasi

    fmt.Printf("Total biaya parkir: Rp %d\n", total)
}
```

```
}
```

Screenshoot program

The screenshot shows a Go program in a code editor and its execution in a terminal. The program is a parking fee calculator. It prompts the user for the vehicle type (motor, mobil, or truk) and the parking duration in hours. It then calculates the total fee based on the vehicle type and duration, with a minimum of 1 hour for duration.

```
1 package main
2
3 import "fmt"
4
5 func main() {
6     var jenis string
7     var durasi int
8     var tarif int
9     fmt.Println("Masukkan jenis kendaraan (motor/mobil/truk): ")
10    fmt.Scan(&jenis)
11
12    fmt.Println("Masukkan durasi parkir (jam): ")
13    fmt.Scan(&durasi)
14
15    if durasi < 1 {
16        durasi = 1
17    }
18
19
20
21    switch jenis {
22    case "motor":
23        tarif = 2000
24    case "mobil":
25        tarif = 5000
26    case "truk":
27        tarif = 8000
28    }
29
30    totalBiaya := durasi * tarif
31    fmt.Println("Total biaya parkir: Rp", totalBiaya)
32 }
```

Terminal Output:

```
PS C:\Tugas> go run "c:\Tugas\Soal\Tugas 2\Tugas2.go"
Masukkan jenis kendaraan (motor/mobil/truk): mobil
Masukkan durasi parkir (jam): 1
Total biaya parkir: Rp 5000
PS C:\Tugas> go run "c:\Tugas\Soal\Tugas 2\Tugas2.go"
Masukkan jenis kendaraan (motor/mobil/truk): truk
Masukkan durasi parkir (jam): 5
Total biaya parkir: Rp 40000
PS C:\Tugas>
```

Deskripsi program

Program ini digunakan untuk menghitung total biaya parkir berdasarkan jenis kendaraan dan durasi parkir dalam jam. Diawali dengan meminta pengguna memasukkan jenis kendaraan, kemudian diikuti memasukkan durasi parkir. Kemudian program mengecek durasi; jika kurang dari 1 jam, maka akan secara otomatis dihitung 1 jam. Setelah itu, program menentukan tarif per jam sesuai jenis kendaraan menggunakan struktur switch: motor Rp 2.000/jam, mobil Rp 5.000/jam, dan truk Rp 8.000/jam. Jika sudah ketemu, program mengalikan tarif tersebut dengan jumlah jam parkir untuk mendapatkan total biaya. Terakhir, hasilnya berupa total biaya parkir akan ditampilkan ke pengguna. Program ini membantu memberikan perhitungan tarif yang sederhana dan langsung sesuai jenis kendaraan yang digunakan.

3. Tugas 3

Source code

```
package main

import "fmt"

func main() {
    var n int
    fmt.Print("Masukkan bilangan: ")
    fmt.Scan(&n)

    switch {
    case n%10 == 0:
        hasil := n / 10
        fmt.Println("Kategori: Bilangan Kelipatan 10")
        fmt.Printf("Hasil pembagian antara %d / 10 = %d\n", n, hasil)

    case n%5 == 0:
        hasil := n * n
        fmt.Println("Kategori: Bilangan Kelipatan 5")
        fmt.Printf("Hasil kuadrat dari %d ^ 2 = %d\n", n, hasil)

    case n%2 == 0:
        next := n + 1
        hasil := n * next
        fmt.Println("Kategori: Bilangan Genap")
        fmt.Printf("Hasil perkalian dengan bilangan berikutnya %d * %d = %d\n", n, next, hasil)

    default:
        next := n + 1
        hasil := n + next
        fmt.Println("Kategori: Bilangan Ganjil")
        fmt.Printf("Hasil penjumlahan dengan bilangan berikutnya %d + %d = %d\n", n, next, hasil)
    }
}
```

Screenshoot program

The screenshot shows a Go program in a code editor and its execution output in a terminal. The program uses a switch statement to categorize a number and perform operations based on that category. The terminal output shows the program being run with the number 8, which is categorized as 'Bilangan Genap' (Even Number), and the result of the operation $8 * 9 = 72$ is displayed.

```
Soal > Tugas3 > main
5 func main() {
9
10     switch {
11     case n%10 == 0:
12         hasil := n / 10
13         fmt.Println("Kategori: Bilangan Kelipatan 10")
14         fmt.Printf("Hasil pembagian antara %d / 10 = %d\n", n, hasil)
15
16     case n%5 == 0:
17         hasil := n * n
18         fmt.Println("Kategori: Bilangan Kelipatan 5")
19         fmt.Printf("Hasil kuadrat dari %d ^ 2 = %d\n", n, hasil)
20
21     case n%2 == 0:
22         next := n + 1
23         hasil := n * next
24         fmt.Println("Kategori: Bilangan Genap")
25         fmt.Printf("Hasil perkalian dengan bilangan berikutnya %d * %d = %d\n", n, next, hasil)
26
27     default:
28         next := n + 1
29         hasil := n + next
30         fmt.Println("Kategori: Bilangan Ganjil")
31         fmt.Printf("Hasil penjumlahan dengan bilangan berikutnya %d + %d = %d\n", n, next, hasil)
32     }
33 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
Kategori: Bilangan Genap
Hasil perkalian dengan bilangan berikutnya 8 * 9 = 72
PS C:\Tugas> go run "c:\Tugas\Soal\Tugas 3\Tugas3.go"
Masukkan bilangan: 25
Kategori: Bilangan Kelipatan 5
Hasil kuadrat dari 25 ^ 2 = 625
PS C:\Tugas> go run "c:\Tugas\Soal\Tugas 3\Tugas3.go"
Masukkan bilangan: 20
Kategori: Bilangan Kelipatan 10
Hasil pembagian antara 20 / 10 = 2
PS C:\Tugas> 
```

Deskripsi program

Program ini akan mengelompokkan suatu bilangan ke dalam kategori tertentu, seperti kelipatan 10, kelipatan 5, bilangan genap, atau bilangan ganjil, dan kemudian melakukan operasi perhitungan sesuai kategori. Setelah pengguna memasukkan sebuah angka, program menggunakan struktur switch dengan beberapa kondisi: jika bilangan tersebut habis dibagi 10, maka termasuk kategori kelipatan 10, dan program membagi angka itu dengan 10; jika habis dibagi 5, maka termasuk kelipatan 5 dan program menghitung kuadrat dari angka tersebut; jika angka bukan kelipatan 5 atau 10 tetapi genap, maka masuk kategori bilangan genap dan program mengalikan angka itu dengan bilangan setelahnya; jika tidak memenuhi semua kondisi di atas, maka angka dianggap bilangan ganjil dan program menjumlahkannya dengan bilangan berikutnya. Setelah dilakukan perhitungan, hasilnya ditampilkan pada pengguna sesuai dengan kategori dan operasi masing-masing.