

LAPORAN PRAKTIKUM ALGORITMA

DAN PEMROGRAMAN 1

MODUL 11

SWITCH CASE



Disusun oleh:

NAUFAL BINTANG P

109082500096

S1IF-13-07

Asisten Praktikum

Adithana dharma putra

Apri pandu wicaksono

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

//tambahkan tangkapan layar dari program (boleh lebih dari 1 jika diperlukan)
CONTOH TANGKAPAN LAYAR:

The screenshot shows a Microsoft Visual Studio Code (VS Code) interface with the following details:

- File Explorer:** Shows a tree view with "Welcome" at the root, followed by "guide1mod11.go", "lat1mod11.go", "lat2mod11.go", "lat3mod11.go", and "guide1mod11.go" again.
- Editor:** Displays a Go program named "guide1mod11.go". The code uses a switch statement to determine whether a given hour is AM or PM based on its value relative to 12.
- Terminal:** Shows the output of running the program with the command "go run .\guide1mod11\guide1mod11.go". The output is:

```
PS D:\go\modul11> go run .\guide1mod11\guide1mod11.go
13
1 PM
PS D:\go\modul11> go run .\guide1mod11\guide1mod11.go
0
12 AM
PS D:\go\modul11> go run .\guide1mod11\guide1mod11.go
12
12 PM
PS D:\go\modul11>
```
- Output Panel:** Shows the name "NAUFAL BINTANG PRATAMA" and student ID "S11TF-13-07 109082500096".
- Status Bar:** Shows the current file is "guide1mod11.go", line 12, column 23, with 46 characters, in plain text mode, at 100% zoom, with Wind and UTF-8 encoding.

Deskripsi program

Program ini berfungsi untuk **mengubah input waktu dalam format 24 jam (0–23) menjadi format 12 jam yang lebih umum digunakan, serta menambahkan keterangan AM atau PM berdasarkan periode waktu tertentu**. Dengan memanfaatkan struktur percabangan switch-case, program menentukan apakah waktu tersebut berada pada rentang tengah malam, pagi, siang, atau malam. Setelah dilakukan proses konversi, hasil berupa angka jam dan label AM/PM ditampilkan sebagai output. Program ini membantu pengguna agar lebih mudah memahami waktu dalam format yang sering dipakai pada kehidupan sehari-hari, seperti pada jam dinding, jadwal kegiatan, dan tampilan waktu digital.

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.")

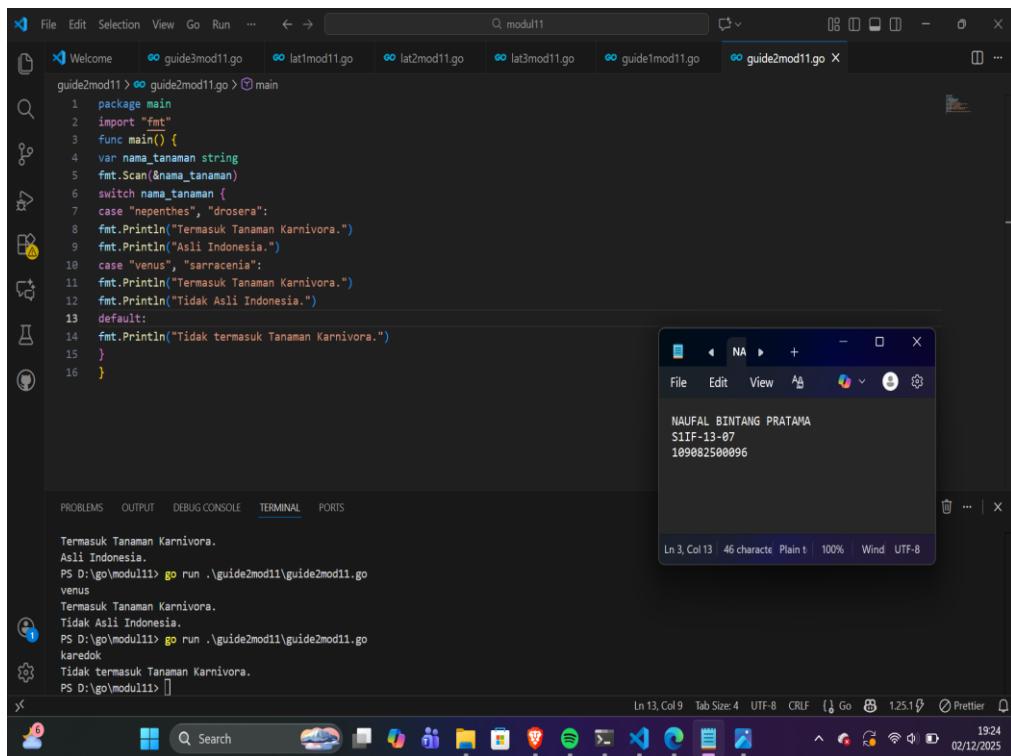
        case "venus", "sarracenia":
            fmt.Println("Termasuk Tanaman Karnivora.")

            fmt.Println("Tidak Asli Indonesia.")

        default:
            fmt.Println("Tidak termasuk Tanaman Karnivora.")
    }
}

```

Screenshot program



Deskripsi program

Program ini berfungsi untuk **mengidentifikasi jenis tanaman berdasarkan nama yang dimasukkan oleh pengguna, serta menentukan apakah tanaman tersebut termasuk dalam kategori tanaman karnivora dan apakah berasal dari Indonesia atau bukan.** Program menerima input berupa nama tanaman dalam bentuk string, kemudian menggunakan struktur percabangan **switch-case** untuk melakukan pengecekan. Jika pengguna memasukkan nama tertentu seperti *nepenthes* atau *drosera*, program akan menampilkan bahwa tanaman tersebut merupakan tanaman karnivora asli Indonesia. Untuk nama seperti *venus* atau *sarracenia*, program menampilkan bahwa tanaman tersebut merupakan tanaman karnivora tetapi bukan asli Indonesia. Apabila

nama tanaman tidak cocok dengan pilihan yang tersedia, program akan memberikan output bahwa tanaman tersebut tidak termasuk dalam tanaman karnivora. Dengan demikian, program ini membantu pengguna mengenali jenis tanaman berdasarkan klasifikasinya.

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
(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)

}

```

Screenshot program

The screenshot shows a code editor with a Go file named `guide3mod11.go`. The code defines a `func main()` function that prompts the user for vehicle type and parking duration, then calculates and prints the parking fee based on a switch statement. A terminal window is open below the editor, showing the execution of the program and its output. The output includes the user input ('Mobil' and '1'), the calculated fee ('Rp 20000'), and the program's name ('guide3mod11').

```

File Edit Selection View Go Run ...
guide3mod11 > guide3mod11.go > main
3 func main() {
6     var tarif int
7     fmt.Print("Masukkan jenis kendaraan (Motor/Mobil/Truk): ")
8     fmt.Scan(&kendaraan)
9     fmt.Print("Masukkan durasi parkir (dalam jam): ")
10    fmt.Scan(&durasi)
11    switch {
12        case kendaraan == "Motor" && durasi >= 1 && durasi <= 2:
13            tarif = 7000
14        case kendaraan == "Motor" && durasi > 2:
15            tarif = 9000
16        case kendaraan == "Mobil" && durasi >= 1 && durasi <= 2:
17            tarif = 15000
18        case kendaraan == "Mobil" && durasi > 2:
19            tarif = 20000
20        case kendaraan == "Truk" && durasi >= 1 && durasi <= 2:
21            tarif = 25000
22        case kendaraan == "Truk" && durasi > 2:
23            tarif = 35000
24        default:
25            fmt.Println("Jenis kendaraan atau durasi parkir tidak valid")
26    }
}

Terminal output:
Tarif Parkir: Rp 7000
PS D:\go\modul11> go run .\guide3mod11\guide3mod11.go
Masukkan jenis kendaraan (Motor/Mobil/Truk): Mobil
Masukkan durasi parkir (dalam jam): 4
Tarif Parkir: Rp 20000
PS D:\go\modul11> go run .\guide3mod11\guide3mod11.go
Masukkan jenis kendaraan (Motor/Mobil/Truk): Truk
Masukkan durasi parkir (dalam jam): 1
Tarif Parkir: Rp 25000
PS D:\go\modul11>

```

Deskripsi program

Program ini berfungsi untuk **menghitung dan menampilkan tarif parkir berdasarkan jenis kendaraan dan durasi parkir dalam satuan jam**. Program meminta pengguna memasukkan jenis kendaraan, seperti Motor, Mobil, atau Truk, serta lama parkir dalam jam. Setelah menerima input, program menggunakan struktur percabangan `switch` yang memeriksa kombinasi jenis kendaraan dan durasi untuk menentukan besarnya tarif yang harus dibayar. Tarif dibedakan berdasarkan kategori waktu, yaitu durasi parkir 1–2 jam dan lebih dari 2 jam. Apabila data yang dimasukkan tidak sesuai dengan pilihan yang tersedia, program akan menampilkan pesan bahwa input tidak valid. Hasil akhir berupa tarif parkir kemudian ditampilkan kepada pengguna dalam format rupiah.

TUGAS

1. Tugas 1

Source code

```
package main
```

```

import "fmt"

func main() {
    var ph float64

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

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

```

Screenshot program

The screenshot shows a Go development environment with the following details:

- Code Editor:** The code for `latimod11.go` is displayed. It includes imports, a package declaration, a main function, and logic to validate a pH value (between 0 and 14) and print whether it's suitable for drinking.
- Terminal:** The terminal window shows the execution of the program. It prompts for input, receives a pH value (8.6), and prints that the water is suitable for drinking. It then asks for another input (9) and prints that it is not suitable. Finally, it asks for a third input (16) and prints that the pH is invalid (between 0 and 14).
- Output Panel:** The output panel displays the user's name, student ID, and NIM.
- System Status:** The taskbar at the bottom shows various system icons and the date/time (01/12/2025, 12:22).

Deskripsi program

Program ini berfungsi untuk **menentukan kelayakan air untuk dikonsumsi berdasarkan nilai pH yang diinput oleh pengguna**. Program akan meminta pengguna memasukkan angka pH air dalam bentuk bilangan desimal. Kemudian, program melakukan pengecekan menggunakan struktur percabangan **if-else** untuk memastikan apakah nilai pH tersebut berada dalam rentang valid yaitu 0 sampai 14. Jika nilai pH di luar rentang tersebut, program menampilkan pesan bahwa nilai pH tidak valid. Namun, apabila nilai pH berada pada kisaran 6,5 hingga 8,6, program akan menampilkan bahwa air tersebut layak diminum karena berada dalam standar kualitas air minum. Sedangkan air yang memiliki pH di luar rentang tersebut dinyatakan tidak layak minum karena dapat membahayakan kesehatan. Program ini membantu pengguna memahami kelayakan air berdasarkan parameter kimia pH secara sederhana.

2. Tugas 2

Source code

```
package main

import (
    "fmt"
    "strings"
)

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

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

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

    if durasi < 1 {
```

```
durasi = 1

}

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

total = tarif * durasi

fmt.Println("Total biaya parkir: Rp", total)
}
```

Screenshot program

```

File Edit Selection View Go Run ... ← → Q: modul11
Welcome guide3mod11.go lat1mod11.go lat2mod11.go
1 lat2mod11 > 00 lat2mod11.go > main
2 import (
3     "fmt"
4     "strings"
5 )
6
7
8 func main() {
9     var kendaraan string
10    var durasi, tarif, total int
11
12    fmt.Println("Masukkan jenis kendaraan (motor/mobil/truk): ")
13    fmt.Scan(&kendaraan)
14    kendaraan = strings.ToLower(kendaraan)
15
16    fmt.Println("Masukkan durasi parkir (jam): ")
17    fmt.Scan(&durasi)
18
19    if durasi < 1 {
20        durasi = 1
21    }
22
23    switch kendaraan {
24    case "motor":
25        tarif = 2000
26    case "mobil":
27        tarif = 5000
28    case "truk":
29        tarif = 8000
30    }
31
32    total = tarif * durasi
33
34    fmt.Println("Total biaya parkir: Rp", total)
35
36 PS D:\go\modul11> go run .\lat2mod11\lat2mod11.go
37 Masukkan jenis kendaraan (motor/mobil/truk): mobil
38 Masukkan durasi parkir (jam): 1
39 Total biaya parkir: Rp 5000
40 PS D:\go\modul11> go run .\lat2mod11\lat2mod11.go
41 Masukkan jenis kendaraan (motor/mobil/truk): truk
42 Masukkan durasi parkir (jam): 5
43 Total biaya parkir: Rp 40000
44 PS D:\go\modul11>

```

NAUFAL BINTANG PRATAMA
S11F-13-07
109082500096

Ln 3, Col 13 | 46 character Plain t 100% Wind UTF-8

powershell + - ×

Ln 38, Col 2 Tab Size: 4 UTF-8 CRLF { } Go 🗑️ 1.25.1 Prettier 11:29 02/12/2025

Deskripsi program

Program ini berfungsi untuk **menghitung total biaya parkir berdasarkan jenis kendaraan dan durasi parkir dalam jam**. Pengguna diminta memasukkan jenis kendaraan seperti motor, mobil, atau truk, kemudian program akan mengubah input menjadi huruf kecil agar proses pengecekan lebih akurat. Setelah itu pengguna memasukkan durasi parkir, dan jika durasi kurang dari satu jam maka program otomatis menetapkan durasi minimum satu jam. Program menggunakan struktur **switch** untuk menentukan tarif per jam sesuai jenis kendaraan, yaitu Rp 2.000 untuk motor, Rp 5.000 untuk mobil, dan Rp 8.000 untuk truk. Jika jenis kendaraan tidak sesuai dengan pilihan yang tersedia, program menampilkan pesan kesalahan dan menghentikan proses. Setelah tarif ditentukan, total biaya dihitung dengan mengalikan tarif per jam dengan durasi parkir dan hasilnya ditampilkan kepada pengguna. Program ini membantu memberikan perhitungan biaya parkir secara cepat, praktis, dan akurat berdasarkan kategori kendaraan.

3. Tugas 3

Source code

```

package main

import "fmt"

func main() {
    var bilangan int
    fmt.Scan(&bilangan)
    switch {

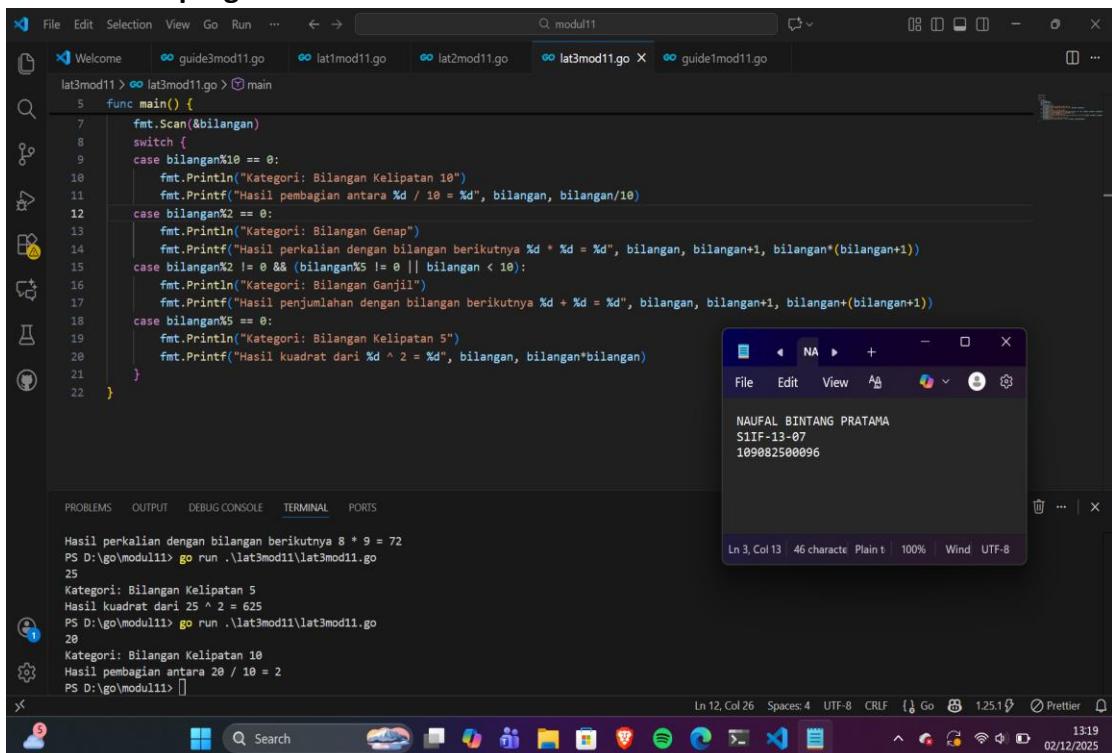
```

```

case bilangan%10 == 0:
    fmt.Println("Kategori: Bilangan Kelipatan 10")
    fmt.Printf("Hasil pembagian antara %d / 10 = %d", bilangan,
bilangan/10)
case bilangan%2 == 0:
    fmt.Println("Kategori: Bilangan Genap")
    fmt.Printf("Hasil perkalian dengan bilangan berikutnya %d * %d = %d",
bilangan, bilangan+1, bilangan*(bilangan+1))
case bilangan%2 != 0 && (bilangan%5 != 0 || bilangan < 10):
    fmt.Println("Kategori: Bilangan Ganjil")
    fmt.Printf("Hasil penjumlahan dengan bilangan berikutnya %d + %d =
%d", bilangan, bilangan+1, bilangan+(bilangan+1))
case bilangan%5 == 0:
    fmt.Println("Kategori: Bilangan Kelipatan 5")
    fmt.Printf("Hasil kuadrat dari %d ^ 2 = %d", bilangan,
bilangan*bilangan)
}
}

```

Screenshoot program



Deskripsi program

Program ini berfungsi untuk **mengklasifikasikan sebuah bilangan berdasarkan karakteristiknya, serta melakukan operasi matematika tertentu sesuai kategori yang dimasukkan oleh pengguna**. Setelah pengguna memasukkan sebuah bilangan, program menggunakan struktur percabangan `switch` untuk mengevaluasi kondisi bilangan tersebut. Jika bilangan merupakan kelipatan 10, program akan menampilkan hasil pembagian bilangan tersebut dengan angka 10. Jika bilangan genap namun bukan kelipatan 10, program menampilkan hasil perkalian antara bilangan dengan

bilangan berikutnya. Untuk bilangan ganjil yang bukan kelipatan 5 atau bilangan ganjil kurang dari 10, program menampilkan hasil penjumlahan dengan bilangan berikutnya. Sementara itu, apabila bilangan termasuk kelipatan 5, program menampilkan hasil kuadrat dari bilangan tersebut. Dengan proses ini, program membantu mengidentifikasi kategori bilangan dan memberikan hasil perhitungan yang sesuai secara otomatis.