

LAPORAN PRAKTIKUM
Algoritma Pemrograman

MODUL 11
SWITCH-CASE



Disusun oleh:

Akhmad Noval Annur

109082500100

S1IF-13-04

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)
}
```

Screenshot program

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

The screenshot shows a Windows desktop environment. In the center is a Microsoft Visual Studio Code (VS Code) window. The left sidebar displays a file tree under 'UNTITLED (WORKSPACE)' containing files like r.go, profit.go, parkir.go, akhirtahun.go, guided1.go, guided2.go, and guided3.go. The main editor area contains the following Go code:

```

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     switch {
10         case jam24 == 0:
11             jam12 = 12
12             label = "AM"
13         case jam24 < 12:
14             jam12 = jam24
15             label = "AM"
16         case jam24 == 12:
17             jam12 = 12
18             label = "PM"
19         case jam24 > 12:
20             jam12 = jam24 - 12
21             label = "PM"
22     }
23     fmt.Println(jam12, label)
24 }

```

To the right of the code editor is a terminal window showing command-line output:

```

PS C:\SEMESTER 1\modul11> go run "c:\SEMESTER 1\modul11\guided1.go"
13
1 PM
PS C:\SEMESTER 1\modul11> go run "c:\SEMESTER 1\modul11\guided1.go"
0
12 AM
PS C:\SEMESTER 1\modul11> go run "c:\SEMESTER 1\modul11\guided1.go"
12
12 PM
PS C:\SEMESTER 1\modul11>

```

Below the terminal is a small Notepad window titled 'Untitled - Notepad...' displaying student information:

```

NAMA : AKHMAD NOVAL ANNUR
NIM : 109082500100
KELAS : S1IF-13-04

```

The taskbar at the bottom shows various pinned icons, including Microsoft Edge, File Explorer, and File History. The system tray indicates the date as 30/11/2025 and the time as 17:17.

Deskripsi program

Program Guided 1 bertujuan untuk mengonversi format waktu dari sistem 24 jam menjadi sistem 12 jam. Program menerima input berupa bilangan bulat yang mewakili jam pada format 24 jam. Struktur switch-case digunakan untuk menentukan hasil konversi berdasarkan nilai jam yang dimasukkan. Jika bilangan jam bernilai 0, maka hasil konversi menjadi 12 AM karena jam 0 dalam sistem 24 jam ekuivalen dengan pukul 12 dini hari dalam sistem 12 jam. Jika jam lebih dari 0 dan kurang dari 12, program tetap menampilkan angka jam yang sama tetapi dengan label AM. Sementara itu, apabila jam bernilai 12, maka tetap ditampilkan sebagai 12 tetapi menggunakan label PM.

Untuk nilai jam lebih dari 12, program akan mengurangi input dengan 12 dan menambahkan label PM. Contohnya, jika pengguna memasukkan angka 15, maka dikonversi menjadi 3 PM. Program ini menunjukkan penerapan struktur kondisi bercabang menggunakan switch-case secara efektif dalam mengolah nilai numerik berdasarkan kasus tertentu. Dengan demikian, program ini bermanfaat untuk memahami mekanisme konversi waktu sekaligus implementasi percabangan dalam bahasa Go.

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

```

    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.")
        }
    }

```

TERMINAL

```

PS C:\SEMESTER 1\modull1> go run "c:\SEMESTER 1\modull1\guided2.go"
nepenthes
Termasuk Tanaman Karnivora.
Asli Indonesia.
PS C:\SEMESTER 1\modull1> go run "c:\SEMESTER 1\modull1\guided2.go"
venus
Termasuk Tanaman Karnivora.
Tidak Asli Indonesia.
PS C:\SEMESTER 1\modull1> go run "c:\SEMESTER 1\modull1\guided2.go"
karedok
Tidak termasuk Tanaman Karnivora.
PS C:\SEMESTER 1\modull1>

```

Untitled - Notepad

```

NAMA : AKHMAD NOVAL ANNUR
NIM : 109082500100
KELAS : S1IF-13-94

```

Deskripsi program

Program Guided 2 bertujuan untuk menentukan apakah nama tanaman yang dimasukkan termasuk dalam kategori tanaman karnivora atau tidak. Pengguna memasukkan input berupa nama tanaman dalam bentuk string. Melalui struktur switch-case, program membandingkan input tersebut dengan beberapa nama tanaman tertentu. Jika yang dimasukkan adalah “nepenthes” atau “drosera”, program menampilkan bahwa tanaman tersebut merupakan tanaman karnivora dan berasal dari Indonesia. Jika yang dimasukkan adalah “venus” atau “sarracenia”, tanaman tetap dikategorikan sebagai karnivora, namun bukan berasal dari Indonesia.

Apabila nama tanaman yang dimasukkan tidak cocok dengan salah satu pilihan, program akan menampilkan output bahwa tanaman tersebut bukan tergolong tanaman karnivora. Program ini memberikan gambaran penggunaan struktur switch-case dalam pencocokan nilai string dengan beberapa kasus sekaligus. Melalui pengujian sederhana, pengguna dapat memahami bagaimana percabangan digunakan untuk menghasilkan keputusan berdasarkan kesesuaian input dengan data yang telah ditentukan.

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 Windows desktop with a code editor window titled "Untitled (Workspace)". The editor displays a Go program named "guided3.go". The code prompts the user for the type of vehicle and parking duration, then calculates and prints the parking fee based on a switch-case structure. The terminal window to the right shows the command "go run" being used to execute the program, followed by the user's input and the program's output, which includes the user's name, ID, and class.

```

File Edit Selection View Go Run ... ⟲ ⟳ Untitled (Workspace)
EXPLORER ... ⟲ UNTITLED (WORKSPACE)
> gotraining
> tulis
> laprak2
> laprak3
> laprak modul 5...
> evaluasi
> tulisbahasalain
> modul9
> modul10
> modul11
    ⟲ guided1.go 2
    ⟲ guided2.go 1
    ⟲ guided3.go 1
modul11 > guided3.go > main
1 import "fmt"
2
3
4
5 func main() {
6     var kendaraan string
7     var durasi int
8     var tarif int
9
10    fmt.Print("Masukkan jenis kendaraan (Motor/Mobil/Truk): ")
11    fmt.Scan(&kendaraan)
12    fmt.Print("Masukkan durasi parkir (dalam jam): ")
13    fmt.Scan(&durasi)
14
15    switch {
16        case kendaraan == "Motor" && durasi >= 1 && durasi <= 2:
17            tarif = 7000
18        case kendaraan == "Motor" && durasi > 2:
19            tarif = 9000
20        case kendaraan == "Mobil" && durasi >= 1 && durasi <= 2:
21            tarif = 15000
22        case kendaraan == "Mobil" && durasi > 2:
23            tarif = 20000
24        case kendaraan == "Truk" && durasi >= 1 && durasi <= 2:
25            tarif = 25000
26        case kendaraan == "Truk" && durasi > 2:
27            tarif = 35000
28        default:
29            fmt.Println("Jenis kendaraan atau durasi parkir tidak valid")
30    }
31    fmt.Printf("Tarif Parkir: Rp %d\n", tarif)
}

```

Deskripsi program

Program Guided 3 berfungsi untuk menghitung biaya parkir berdasarkan jenis kendaraan serta durasi parkir. Program menerima dua masukan dari pengguna, yaitu jenis kendaraan (“Motor”, “Mobil”, atau “Truk”) dan jumlah waktu parkir dalam jam. Struktur switch-case digunakan untuk membandingkan input kendaraan dengan beberapa kemungkinan nilai. Tarif yang dikenakan berbeda-beda sesuai dengan jenis kendaraan dan lamanya durasi parkir.

Untuk kendaraan motor, pengguna dikenakan tarif sebesar Rp7.000 jika waktu parkir antara 1 hingga 2 jam, dan Rp9.000 jika waktu parkir lebih dari 2 jam. Kendaraan mobil dikenai tarif Rp15.000 untuk durasi 1–2 jam, dan Rp20.000 jika durasi lebih dari 2 jam. Sedangkan kendaraan truk dikenakan tarif Rp25.000 untuk durasi hingga 2 jam, dan Rp35.000 jika lebih dari 2 jam. Apabila pengguna memasukkan jenis kendaraan

atau durasi yang tidak sesuai dengan ketentuan, program akan menampilkan pesan input tidak valid.

Program ini menunjukkan penerapan percabangan majemuk menggunakan switch-case dengan kondisi gabungan (kombinasi antara jenis kendaraan dan perbandingan durasi). Dengan struktur tersebut, program mampu menghasilkan output yang sesuai berdasarkan variabel input yang beragam. Implementasi ini juga menggambarkan bagaimana logika perhitungan dapat diintegrasikan dengan struktur percabangan dalam pengembangan aplikasi sederhana.

\

TUGAS

1. Tugas 1

Source code

```
package main

import "fmt"

func main() {
    var ph float64
    fmt.Scan(&ph)

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

Screenshoot program

The screenshot shows a Windows desktop environment. In the center is a Microsoft Visual Studio Code (VS Code) window. The Explorer sidebar on the left shows a file tree with several Go files: uided2.go, guided3.go, soal1.go (selected), soal2.go, soal3.go, and modul11. The Terminal tab on the right displays command-line output in PowerShell (PS) and a Command Prompt (CMD). The output shows the execution of 'go run' commands for 'soal1.go' and 'soal2.go', with responses indicating whether the pH value is valid or not. Below the VS Code window is a Notepad window containing student information: NAMA : AKHMAD NOVAL ANNUR, NIM : 189682500100, and KELAS : SIIF-13-04. The taskbar at the bottom includes icons for File Explorer, Task View, Start, Taskbar settings, and various pinned applications like Edge, File Explorer, and File History. The system tray shows battery level (75.5%), temperature (28°C), date (30/11/2025), and time (18:15).

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

Deskripsi program

Program pada Tugas 1 digunakan untuk menentukan apakah air layak untuk diminum berdasarkan nilai pH yang dimasukkan oleh pengguna. Input berupa bilangan desimal (float) yang merepresentasikan nilai keasaman air. Struktur *switch-case* digunakan untuk melakukan pemeriksaan kondisi berdasarkan rentang nilai pH. Apabila nilai pH berada antara 6,5 hingga 8,6, program akan menampilkan bahwa air layak untuk diminum karena berada dalam rentang ideal. Sebaliknya, jika nilai pH berada di luar rentang ideal namun masih dalam batas valid (0–14), maka air dinyatakan tidak layak dikonsumsi.

Jika pengguna memasukkan nilai pH di luar batas 0 hingga 14, program akan memberikan pesan bahwa input tidak valid. Implementasi program ini menunjukkan penggunaan logika kondisi bersyarat dalam mengevaluasi data lingkungan berdasarkan standar ilmiah. Program ini juga menggambarkan penerapan struktur *switch-case* untuk mengevaluasi kondisi berdasarkan rentang nilai, bukan hanya nilai tunggal.

2. Tugas 2

Source code

```
package main

import "fmt"

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

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

    fmt.Scan(&kendaraan)

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

    if durasi < 1 {
        durasi = 1
    }

    switch kendaraan {
    case "motor":
        total = 2000 * durasi
    case "mobil":
        total = 5000 * durasi
    case "truk":
        total = 8000 * durasi
    default:
        fmt.Println("Jenis kendaraan tidak valid")
```

```

    }

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

}

```

Screenshoot program

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

- EXPLORER:** Shows files in the workspace, including `ed1.go`, `guided2.go`, `guided3.go`, `soal1.go`, and `soal2.go`.
- TERMINAL:** Displays command-line interactions:
 - `PS C:\SEMESTER 1\modul11> go run "c:\SEMESTER 1\modul11\soal2.go"`
 - Masukkan jenis kendaraan (motor/mobil/truk): `motor`
 - Masukkan durasi parkir (dalam jam): `3`
 - Tarif Parkir: `Rp 6000`
 - `PS C:\SEMESTER 1\modul11> go run "c:\SEMESTER 1\modul11\soal2.go"`
 - Masukkan jenis kendaraan (motor/mobil/truk): `mobil`
 - Masukkan durasi parkir (dalam jam): `1`
 - Tarif Parkir: `Rp 5000`
 - `PS C:\SEMESTER 1\modul11> go run "c:\SEMESTER 1\modul11\soal2.go"`
 - Masukkan jenis kendaraan (motor/mobil/truk): `truk`
 - Masukkan durasi parkir (dalam jam): `5`
 - Tarif Parkir: `Rp 40000`
 - `PS C:\SEMESTER 1\modul11>`
- NOTEBOOK:** A floating window titled "Untitled - Notepad" contains personal information:
 - NAMA : AKHMAD NOVAL ANNUR
 - NIM : 109082500100
 - KELAS : S1IF-13-04
- STATUS BAR:** Shows file path (`C:\SEMESTER 1\modul11\soal2.go`), line (Ln 27), column (Col 9), tab size (Tab Size: 4), encoding (UTF-8), and other system information.

Deskripsi program

Tugas 2 bertujuan untuk menghitung total biaya parkir berdasarkan jenis kendaraan dan durasi parkir. Program menerima input berupa jenis kendaraan (motor, mobil, atau truk) dan lamanya waktu parkir dalam jam. Jika durasi parkir kurang dari satu jam, program menetapkan durasi minimal yaitu satu jam. Kemudian, perhitungan tarif dilakukan berdasarkan jenis kendaraan yang dimasukkan oleh pengguna. Motor dikenakan tarif Rp2.000 per jam, mobil Rp5.000 per jam, dan truk Rp8.000 per jam.

Setelah perhitungan dilakukan, program akan menampilkan hasil akhir berupa total biaya parkir. Jika jenis kendaraan tidak sesuai dengan pilihan yang tersedia, program akan menampilkan pesan bahwa kendaraan tidak valid. Melalui program ini, pengguna dapat memahami penerapan struktur *switch-case* untuk memproses data dan menghasilkan keluaran berupa perhitungan tarif berdasarkan kondisi yang diberikan.

3. Tugas 3

Source code

```
package main
import "fmt"

func main() {
    var n int
    fmt.Scan(&n)

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

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

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

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

Screenshot program

The screenshot shows a Windows desktop environment. In the center is a Microsoft Visual Studio Code (VS Code) window titled "Untitled (Workspace)". The workspace contains several Go files: "modul11", "soal1.go", "soal2.go", and "soal3.go". The code in "soal3.go" is as follows:

```
modul11 > soal3.go > main
1 package main
2 import "fmt"
3
4 func main() {
5     var n int
6     fmt.Scan(&n)
7
8     switch {
9     case n%10 == 0:
10        fmt.Println("Kategori: Bilangan Kelipatan 10")
11        fmt.Printf("Hasil pembagian antara %d / 10 = %d\n", n, n/10)
12
13    case n%5 == 0 && n != 5:
14        fmt.Println("Kategori: Bilangan Kelipatan 5")
15        fmt.Printf("Hasil kuadrat dari %d^2 = %d\n", n, n*n)
16
17    case n%2 == 0:
18        fmt.Println("Kategori: Bilangan Genap")
19        fmt.Printf("Hasil perkalian dengan bilangan berikutnya %d * %d = %d\n", n, n+1, n*(n+1))
20
21    default:
22        fmt.Println("Kategori: Bilangan Ganjil")
23        fmt.Printf("Hasil penjumlahan dengan bilangan berikutnya %d + %d = %d\n", n, n+1, n+(n+1))
24    }
25 }
```

To the right of the code editor is a terminal window titled "TERMINAL". It shows the command "go run" being executed multiple times with different inputs, resulting in categorized outputs like "Bilangan Ganjil", "Bilangan Genap", and "Bilangan Kelipatan 5".

The taskbar at the bottom of the screen includes icons for File Explorer, Task View, Start, Taskbar settings, and various pinned applications. A system tray icon for "Activate Windows" is also visible.

Deskripsi program

Program Tugas 3 dibuat untuk mengidentifikasi jenis bilangan berdasarkan pola aritmatika dan kemudian melakukan operasi matematika sesuai kategori yang ditentukan. Program menerima satu input berupa bilangan bulat. Dengan menggunakan struktur percabangan switch-case, program mengevaluasi bilangan tersebut dengan urutan prioritas tertentu untuk menghindari bentrok logika. Jika bilangan merupakan kelipatan 10, program akan mengelompokkan bilangan tersebut sebagai “Bilangan Kelipatan 10” dan melakukan operasi pembagian dengan bilangan 10. Kondisi ini diberikan prioritas tertinggi karena untuk menghindari bilangan seperti 20 masuk ke kategori bilangan genap.

Jika bilangan merupakan kelipatan 5 tetapi bukan angka 5, maka dikategorikan sebagai “Bilangan Kelipatan 5” dan program menghitung hasil kuadrat dari bilangan tersebut. Penyusunan logika ini bertujuan untuk menghindari kasus khusus angka 5, yang dalam contoh modul dikategorikan sebagai bilangan ganjil. Jika bilangan genap dan bukan kelipatan 10, maka program menampilkan kategori “Bilangan Genap” dan menghitung hasil perkalian bilangan tersebut dengan bilangan berikutnya.

Jika semua kondisi sebelumnya tidak terpenuhi, maka bilangan dikategorikan sebagai “Bilangan Ganjil” dan program melakukan operasi penjumlahan antara bilangan tersebut dengan bilangan setelahnya. Melalui program ini, pengguna dapat memahami bagaimana menerapkan struktur switch-case secara tepat dengan

mempertimbangkan urutan prioritas kondisi. Program juga menunjukkan kombinasi logika matematika dan percabangan sehingga menghasilkan output yang sesuai dengan ketentuan modul.