

LAPORAN PRAKTIKUM
Algoritma Pemrograman

MODUL 11

Switch-case



Disusun oleh:

Cofa Xavier Marvel

109082500001

S1IF-13-04

PROGRAM STUDI S1 INFORMATIKA
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2025

L

1. Guided 1

Source Code

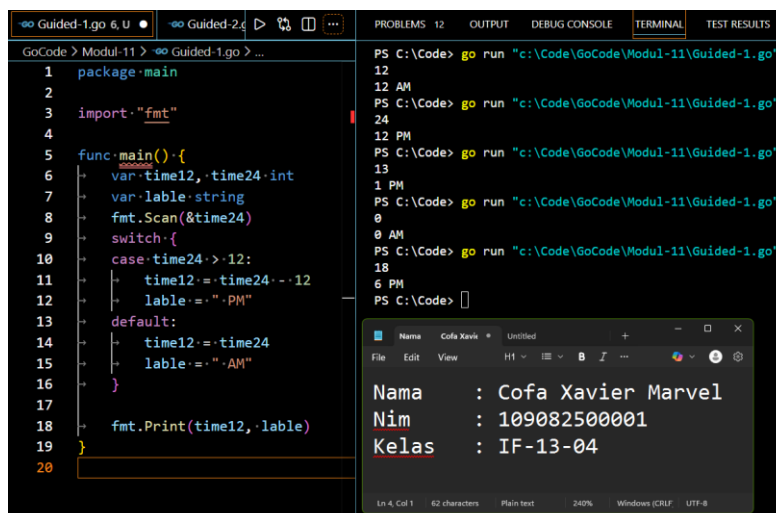
```
package main

import "fmt"

func main() {
    var time12, time24 int
    var lable string
    fmt.Scan(&time24)
    switch {
    case time24 > 12:
        time12 = time24 - 12
        lable = " PM"
    default:
        time12 = time24
        lable = " AM"
    }

    fmt.Print(time12, lable)
}
```

Screenshoot program



Deskripsi program

This is a program that uses a switch-case operator with a blank expression and one case. In the case where `time24 > 12` is true, `time12` is `time24 - 12` and the label is PM. In default, when `time24 > 12` is false, `time12` is `time24` and the label is AM.

2. Guided 2

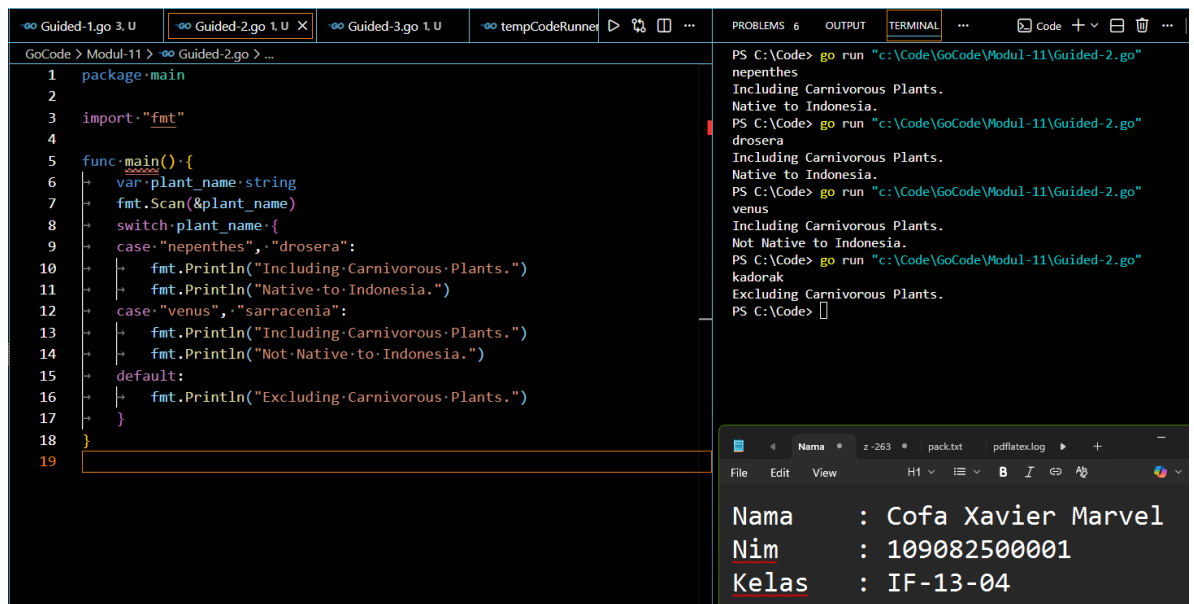
Source Code

```
package main

import "fmt"

func main() {
    var plant_name string
    fmt.Scan(&plant_name)
    switch plant_name {
    case "nepenthes", "drosera":
        fmt.Println("Including Carnivorous Plants.")
        fmt.Println("Native to Indonesia.")
    case "venus", "sarracenia":
        fmt.Println("Including Carnivorous Plants.")
        fmt.Println("Not Native to Indonesia.")
    default:
        fmt.Println("Excluding Carnivorous Plants.")
    }
}
```

Screenshoot program



The screenshot displays a Go code editor with the source code for a plant classifier. The code uses a switch statement to categorize plants based on their name. The output window shows the results of running the program for several inputs: 'nepenthes' and 'drosera' are classified as 'Including Carnivorous Plants.' and 'Native to Indonesia.'; 'venus' is classified as 'Including Carnivorous Plants.' and 'Not Native to Indonesia.'; and 'kadorak' is classified as 'Excluding Carnivorous Plants.'.

```
GoCode > Modul-11 > Guided-2.go > ...
1 package main
2
3 import "fmt"
4
5 func main() {
6     var plant_name string
7     fmt.Scan(&plant_name)
8     switch plant_name {
9     case "nepenthes", "drosera":
10         fmt.Println("Including Carnivorous Plants.")
11         fmt.Println("Native to Indonesia.")
12     case "venus", "sarracenia":
13         fmt.Println("Including Carnivorous Plants.")
14         fmt.Println("Not Native to Indonesia.")
15     default:
16         fmt.Println("Excluding Carnivorous Plants.")
17     }
18 }
19
```

PS C:\Code> go run "c:\Code\GoCode\Modul-11\Guided-2.go"

nepenthes
Including Carnivorous Plants.
Native to Indonesia.

PS C:\Code> go run "c:\Code\GoCode\Modul-11\Guided-2.go"

drosera
Including Carnivorous Plants.
Native to Indonesia.

PS C:\Code> go run "c:\Code\GoCode\Modul-11\Guided-2.go"

venus
Including Carnivorous Plants.
Not Native to Indonesia.

PS C:\Code> go run "c:\Code\GoCode\Modul-11\Guided-2.go"

kadorak
Excluding Carnivorous Plants.

PS C:\Code>

Nama : Cofa Xavier Marvel
Nim : 109082500001
Kelas : IF-13-04

Deskripsi program

This program is a simple classifier that checks whether a plant is a carnivore plant or not and if it is native to Indonesia or not the outputs the appropriate message based on the category

3. Guided 3

Source Code

```

package main

import "fmt"

func main() {
    var vehicle string
    var duration int
    var price int
    fmt.Print("Enter the type of vehicle
(Motorcycle/Car/Truck): ")
    fmt.Scan(&vehicle)
    fmt.Print("Enter the parking duration (in hours): ")
    fmt.Scan(&duration)
    switch {
        case vehicle == "Motorcycle" && duration >= 1 && duration
<= 2:
            price = 7000
        case vehicle == "Motorcycle" && duration > 2:
            price = 9000
        case vehicle == "Car" && duration >= 1 && duration <= 2:
            price = 15000
        case vehicle == "Car" && duration > 2:
            price = 20000
        case vehicle == "Truck" && duration >= 1 && duration <=
2:
            price = 25000
        case vehicle == "Truck" && duration > 2:
            price = 35000
        default:
            fmt.Println("Invalid vehicle type or parking
duration")
    }
    fmt.Printf("Parking rates: Rp %d\n", price)
}

```

Screenshoot program

```
Guided-1.go 3, U    Guided-2.go 1, U    Guided-3.go 1, U X    tempCodeRunner
code > Modul-11 > Guided-3.go > main
3  import "fmt"
4
5  func main(){
6      var vehicle string
7      var duration int
8      var price int
9      fmt.Print("Enter the type of vehicle (Motorcycle/Car/Truck): ")
10     fmt.Scan(&vehicle)
11     fmt.Print("Enter the parking duration (in hours): ")
12     fmt.Scan(&duration)
13     switch vehicle {
14     case "Motorcycle", "motorcycle":
15         if duration >= 1 && duration <= 2 {
16             price = 7000
17         } else if duration > 2 {
18             price = 9000
19         }
20     case "Car", "car":
21         if duration >= 1 && duration <= 2 {
22             price = 15000
23         } else if duration > 2 {
24             price = 20000
25         }
26     case "Truck", "truck":
27         if duration >= 1 && duration <= 2 {
28             price = 25000
29         } else if duration > 2 {
30             price = 35000
31         }
32     default:
33         fmt.Println("Invalid vehicle type or parking duration")
34     }
35     fmt.Printf("Parking rates: Rp.%d\n", price)
36 }
37
```

The image shows a screenshot of a Visual Studio Code (VS Code) interface. The top panel is the 'TERMINAL' tab, which displays the execution of a Go program. The program prompts the user to enter the type of vehicle (Motorcycle, Car, or Truck) and the parking duration in hours. It then calculates the parking rate based on the vehicle type and duration. The terminal output shows several test cases: a motorcycle parked for 2 hours (Rp 7000), a car parked for 4 hours (Rp 20000), a motorcycle parked for 3 hours (Rp 9000), a truck parked for 1 hour (Rp 25000), and an invalid vehicle type 'biycile' (Rp 0). The bottom panel is the 'Code' editor, showing a file named 'Nama' with the following content:

```
Nama      : Cofa Xavier Marvel
Nim       : 109082500001
Kelas    : IF-13-04
```

The status bar at the bottom of the code editor indicates the cursor is at line 4, column 1, with 62 characters, in plain text, at 240% zoom, using Windows (CRLF) line endings and UTF-8 encoding.

Deskripsi program

This program sets the price of parking based on the vehicle type it uses a switch-case to identify the vehicle and an if-else thing to find the amount owed.

TUGAS

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("Drinkable Water")
    case ph < 6.5 && ph > 8.6:
        fmt.Println("Undrinkable Water")
    case ph < 0 || ph > 14:
        fmt.Println("Invalid input, pH range 0 - 14")
    default:
        fmt.Println("Water not fit for drinking")
    }
}
```

Screenshoot program



The screenshot shows a GoCode IDE interface. At the top, there are tabs for 'Guided-1.go 4, U', 'Guided-2.go 1, U', 'Tugas-1.go 1, U' (which is selected and highlighted with a yellow border), and 'Guided-3.go 1, U'. Below the tabs, the breadcrumb path reads 'GoCode > Modul-11 > Tugas-1.go > ...'. The main editor area displays the Go source code from the 'Source code' block, with line numbers 1 through 20 on the left. The code is as follows:

```
1 package main
2
3 import "fmt"
4
5 func main() {
6     var ph float64
7     fmt.Scan(&ph)
8
9     switch {
10    case ph >= 6.5 && ph <= 8.6:
11        |> fmt.Println("Drinkable Water")
12    case ph < 6.5 && ph > 8.6:
13        |> fmt.Println("Undrinkable Water")
14    case ph < 0 || ph > 14:
15        |> fmt.Println("Invalid input, pH range 0 - 14")
16    default:
17        |> fmt.Println("Water not fit for drinking")
18    }
19 }
20
```

The screenshot shows the Visual Studio Code interface. The 'TERMINAL' tab is active, displaying the execution of a Go program. The command prompt shows the user running 'go run' on a file named 'Tugas-1.go'. The program outputs '8.6', 'Drinkable Water', '9', 'Water not fit for drinking', and '16', followed by an error message 'Invalid input, pH range 0 - 14'. Below the terminal, the editor shows a file named 'Nama' with the following text: 'Nama : Cofa Xavier Marvel', 'Nim : 109082500001', and 'Kelas : IF-13-04'. The status bar at the bottom indicates the cursor is at line 4, column 1, with 62 characters in plain text, at 240% zoom, using Windows (CRLF) line endings and UTF-8 encoding.

```
PS C:\Code> go run "c:\Code\GoCode\Modul-11\Tugas-1.go"
8.6
Drinkable Water
PS C:\Code> go run "c:\Code\GoCode\Modul-11\Tugas-1.go"
9
Water not fit for drinking
PS C:\Code> go run "c:\Code\GoCode\Modul-11\Tugas-1.go"
16
Invalid input, pH range 0 - 14
PS C:\Code>
```

Nama : Cofa Xavier Marvel
Nim : 109082500001
Kelas : IF-13-04

Deskripsi program

This program finds the drinkability of water depending on the pH level using a switch-case with no expression and 3 cases with a default

Tugas 2

Source code

```
package main

import "fmt"

func main() {
    var duration, price int
    var vehicle string

    fmt.Scan(&vehicle, &duration)

    if duration == 0 {
        duration = 1
    }
}
```



```

    }

    switch vehicle {
    case "Motorcycle", "motorcycle":
        price = 2000 * duration
    case "Car", "car":
        price = 5000 * duration
    case "Truck", "truck":
        price = 8000 * duration
    default:
        fmt.Println("Invalid vehicle type or parking
duration")
    }
    fmt.Printf("Parking rates: Rp %d\n", price)
}

```

Screenshoot program

The screenshot shows a Go program in a code editor and its execution output in a terminal window. The program calculates parking rates based on vehicle type and duration. The terminal output shows the program being run with various inputs and the resulting parking rates.

```

GoCode > Modul-11 > Tugas-2.go > ...
1 package main
2
3 import "fmt"
4
5 func main() {
6     var duration, price int
7     var vehicle string
8
9     fmt.Scan(&vehicle, &duration)
10
11     if duration == 0 {
12         duration = 1
13     }
14
15     switch vehicle {
16     case "Motorcycle", "motorcycle":
17         price = 2000 * duration
18     case "Car", "car":
19         price = 5000 * duration
20     case "Truck", "truck":
21         price = 8000 * duration
22     default:
23         fmt.Println("Invalid vehicle type or parking duration")
24     }
25     fmt.Printf("Parking rates: Rp %d\n", price)
26 }
27
PS C:\Code> go run "c:\Code\GoCode\Modul-11\Tugas-2.go"
motorcycle 2
Parking rates: Rp 4000
PS C:\Code> go run "c:\Code\GoCode\Modul-11\Tugas-2.go"
motorcycle 0
Parking rates: Rp 2000
PS C:\Code> go run "c:\Code\GoCode\Modul-11\Tugas-2.go"
Truck 12
Parking rates: Rp 96000
PS C:\Code> go run "c:\Code\GoCode\Modul-11\Tugas-2.go"
Car 1
Parking rates: Rp 5000
PS C:\Code> go run "c:\Code\GoCode\Modul-11\Tugas-2.go"
Mikecycle 69
Invalid vehicle type or parking duration
Parking rates: Rp 0
PS C:\Code>

```

Deskripsi program

This program is the same as Guided 3 but the way it calculates the price of parking is linear instead of constant.

Tugas 3

Source code

```

package main

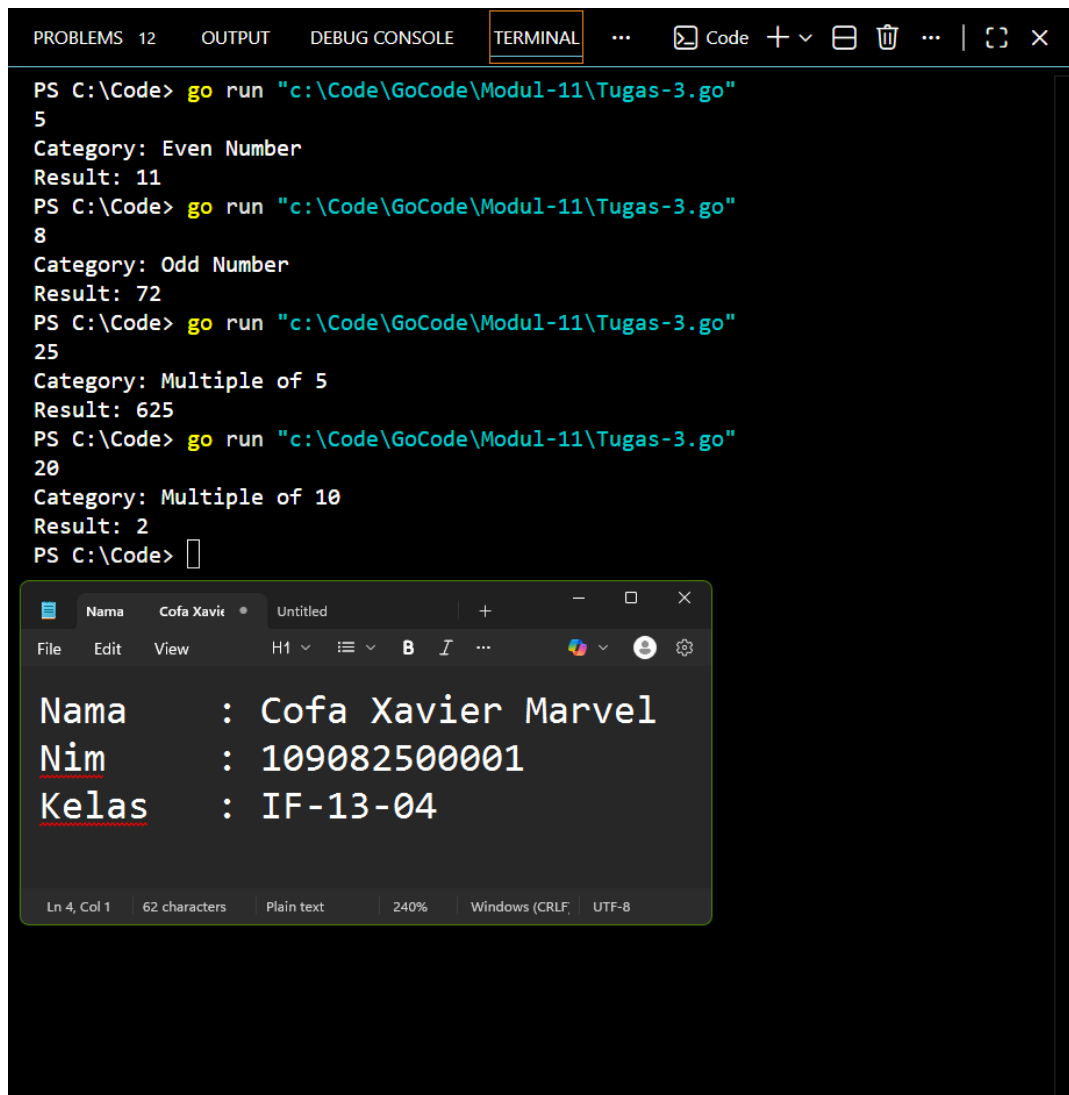
import "fmt"

```

```
func main() {  
    var num, num1 int  
    var Category string  
    fmt.Scan(&num)  
  
    switch {  
    case num%10 == 0 && num > 10:  
        Category = "Multiple of 10"  
        num1 = num / 10  
    case num%5 == 0 && num > 10:  
        Category = "Multiple of 5"  
        num1 = num * num  
    }  
    switch {  
    case num%2 != 0 && num < 10:  
        Category = "Even Number"  
        num1 = num + (num + 1)  
    case num%2 == 0 && num < 10:  
        Category = "Odd Number"  
        num1 = num * (num + 1)  
    }  
    fmt.Printf("Category: %s \nResult: %d", Category, num1)  
}
```

Screenshoot program

```
2.go 1, U  Tugas-1.go 1, U  Tugas-2.go 1, U  Tugas-3.go 1, U X  ▶ 🔍 📄 ...
GoCode > Modul-11 > Tugas-3.go > main
1  package main
2
3  import "fmt"
4
5  func main(){
6      var num, num1 int
7      var Category string
8      fmt.Scan(&num)
9
10     switch{
11     case num%10 == 0 && num > 10:
12         Category = "Multiple of 10"
13         num1 = num / 10
14     case num%5 == 0 && num > 10:
15         Category = "Multiple of 5"
16         num1 = num * num
17     }
18     switch{
19     case num%2 != 0 && num < 10:
20         Category = "Even Number"
21         num1 = num + (num + 1)
22     case num%2 == 0 && num < 10:
23         Category = "Odd Number"
24         num1 = num * (num + 1)
25     }
26     fmt.Printf("Category: %s\nResult: %d", Category, num1)
27 }
```



The image shows a screenshot of a Visual Studio Code interface. The top bar includes tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, and TERMINAL. The TERMINAL tab is active, displaying a PowerShell session where the command `go run "c:\Code\GoCode\Modul-11\Tugas-3.go"` is executed four times with different inputs: 5, 8, 25, and 20. The program outputs category and result for each input. An inset window shows a text editor with the following text:

```
Nama      : Cofa Xavier Marvel
Nim       : 109082500001
Kelas    : IF-13-04
```

The status bar at the bottom of the text editor indicates the cursor is at line 4, column 1, with 62 characters, in plain text, at 240% zoom, using Windows (CRLF) line endings and UTF-8 encoding.

Deskripsi program

This program uses two switch-case operations each with two cases neither have expressions, the first switch-case finds if the number given is a multiple of 10 or multiple of 5 and greater, greater than 10 exist to avoid the conflict that comes when the number is even or odd and is a multiple of 5 or 10, the second switch-case checks if the number given is odd or even then it Outputs the correct number using the appropriate formula depending on whether the number is a multiple of 10, a multiple of 5, Odd or Even and greater or lesser than 10.