

LAPORAN PRAKTIKUM ALGORITMA

DAN PEMROGRAMAN 1

MODUL 11 – SWITCH-CASE

ALGORITMA DAN PEMOGRAMAN 1



Disusun oleh:

NAMA : PRIMATAMA SIGALINGGING

NIM : 109082500076

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)
}
```

Screenshot program

The screenshot shows a code editor interface with a dark theme. On the left, the code for `guided1.go` is displayed:

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

Below the code editor is a terminal window showing the command `go run guided1.go` and its output:

```
PS C:\alproo\go\minggu 11> go run guided1.go
13
1 PM
PS C:\alproo\go\minggu 11>
```

To the right of the terminal is a preview window showing the converted time: "Nama : Primatama Sigalingging NIM: 109082500076".

Deskripsi program

➤ Tujuan

Program ini bertujuan mengubah format jam dari sistem 24 jam menjadi sistem 12 jam lengkap dengan penanda AM atau PM. Ini membantu pengguna yang lebih terbiasa menggunakan format jam 12.

➤ Proses

Pengguna memasukkan angka yang mewakili jam pada format 24 jam. Program kemudian menyesuaikan waktu berdasarkan kondisi berikut:

- Jam 0 → dikonversi menjadi 12 AM
- Jam 1–11 → tampil apa adanya dengan label AM
- Jam 12 → menjadi 12 PM
- Jam 13–23 → dikurangi 12 dan diberi label PM

Setelah proses konversi selesai, program menampilkan waktu pada format 12 jam.

➤ Kesimpulan

Program ini mempermudah pengguna melihat waktu dalam format 12 jam tanpa harus menghitung sendiri. Dengan langkah-langkah yang sederhana, konversi dapat dilakukan secara cepat dan akurat.

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 code editor with the file `guided2.go` open. The code defines a `main` function that reads a plant name from standard input and prints whether it's a carnivorous plant from Indonesia or not. A terminal window below shows the command `go run guided2.go` being run, followed by the output: "nepenthes", "Termasuk Tanaman Karnivora.", "Asli Indonesia.". The terminal also shows the user's name and NIM.

```
-∞ guided2.go
1 package main
2 import "fmt"
3 func main() {
4     var nama_tanaman string
5     fmt.Scan(&nama_tanaman)
6     switch nama_tanaman {
7         case "nepenthes", "drosera":
8             fmt.Println("Termasuk Tanaman Karnivora.")
9             fmt.Println("Asli Indonesia.")
10        case "venus", "sarracenia":
11            fmt.Println("Termasuk Tanaman Karnivora.")
12            fmt.Println("Tidak Asli Indonesia.")
13        default:
14            fmt.Println("Tidak termasuk Tanaman Karnivora.")
15    }
16 }
```

```
PS C:\alproo\go\minggu 11> go run guided2.go
nepenthes
Termasuk Tanaman Karnivora.
Asli Indonesia.
PS C:\alproo\go\minggu 11> ]
```

Deskripsi program

➤ **Tujuan**

Tujuan utama program ini adalah mengetahui apakah sebuah tanaman tergolong tanaman karnivora. Jika iya, program juga menginformasikan apakah tanaman tersebut berasal dari Indonesia atau bukan.

➤ **Proses**

Pengguna memasukkan nama tanaman sebagai input.

Program lalu membandingkan nama tersebut dengan daftar tanaman yang sudah ditetapkan:

- nepenthes dan drosera → karnivora & asli Indonesia
- venus dan sarracenia → karnivora tetapi bukan asli Indonesia
-

Jika nama tanaman tidak cocok dengan daftar tersebut, program menyatakan bahwa tanaman itu bukan tanaman karnivora.

Output diberikan dalam dua baris: status karnivora dan status asal daerah.

➤ **Kesimpulan**

Program ini menyediakan cara cepat untuk mengidentifikasi jenis tanaman berdasarkan namanya. Dengan logika sederhana, pengguna dapat mengetahui apakah tanaman itu karnivora dan bagaimana asal-usulnya.

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 = 10000
        case kendaraan == "Mobil" && durasi >= 1 && durasi <= 2:
            tarif = 10000
        case kendaraan == "Mobil" && durasi > 2:
            tarif = 15000
        case kendaraan == "Truk" && durasi >= 1 && durasi <= 2:
            tarif = 15000
        case kendaraan == "Truk" && durasi > 2:
            tarif = 20000
    }
    fmt.Println("Total Biaya Parkir: ", tarif)
}
```

```
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)
}

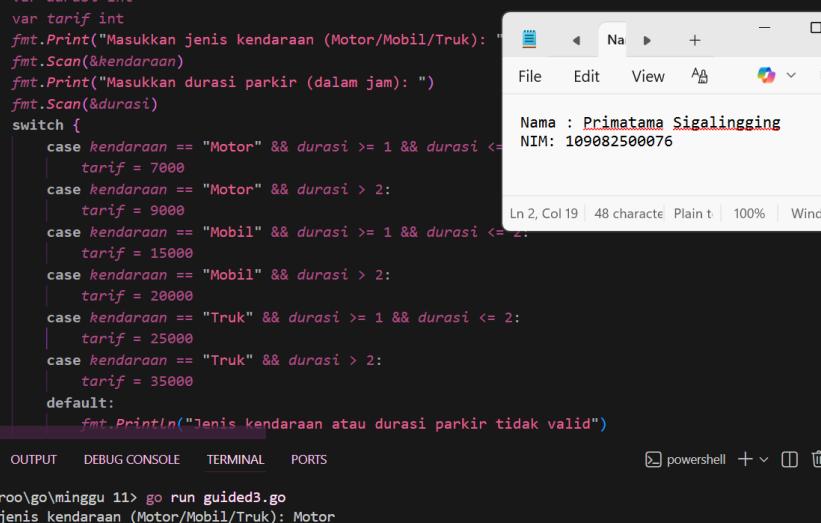
hasil = hasil + v1

}

fmt.Println(hasil)

}
```

Screenshot program



The screenshot shows a Go code editor with a file named `guided3.go`. The code defines a `main()` function that prompts the user for the type of vehicle and parking duration, then calculates and prints the fee based on a switch statement. A terminal window below shows the execution of the program, inputting "Motor" and "2", and outputting a fee of 7000.

```
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")
    }
}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\alproo\go\minggu 11> go run guided3.go
Masukkan jenis kendaraan (Motor/Mobil/Truk): Motor
Masukkan durasi parkir (dalam jam): 2
Tarif Parkir: Rp 7000
PS C:\alproo\go\minggu 11>

Deskripsi program

➤ **Tujuan**

Program ini dibuat untuk menentukan berapa biaya parkir yang harus dibayar berdasarkan jenis kendaraan dan lamanya waktu parkir. Dengan aturan tarif yang berbeda-beda, program membantu pengguna mengetahui nominal yang tepat secara otomatis.

➤ **Proses**

Pengguna terlebih dahulu memasukkan jenis kendaraan yang digunakan (Motor, Mobil, atau Truk).

Setelah itu, pengguna mengisi durasi parkir dalam satuan jam.

Program kemudian membandingkan input tersebut dengan ketentuan tarif:

- Motor → Rp 7.000 (1–2 jam), Rp 9.000 (>2 jam)
- Mobil → Rp 15.000 (1–2 jam), Rp 20.000 (>2 jam)
- Truk → Rp 25.000 (1–2 jam), Rp 35.000 (>2 jam)
-

Jika jenis kendaraan tidak ada dalam daftar, program akan memberi tahu bahwa data tidak valid. Setelah pengecekan selesai, tarif akhir ditampilkan kepada pengguna.

➤ **Kesimpulan**

Program ini memudahkan proses pengecekan biaya parkir tanpa harus menghitung secara manual. Dengan memanfaatkan logika percabangan, tarif dapat ditentukan dengan cepat sesuai aturan yang berlaku.

TUGAS

1. Tugas 1

Source code

```
package main

import (
    "fmt"
)

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

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

Screenshot program

Deskripsi program

➤ Tujuan

Program ini dirancang untuk menilai apakah suatu sampel air aman dikonsumsi berdasarkan nilai pH yang diberikan oleh pengguna. Dengan pengukuran ini, pengguna dapat mengetahui kualitas air tanpa melakukan analisis manual.

➤ Proses

1. Pengguna menginput nilai pH dalam bentuk angka.
 2. Program mengevaluasi angka tersebut melalui beberapa kondisi:
 - Jika pH berada di bawah 0 atau melebihi 14 → input dianggap tidak sah.
 - Jika pH berada pada kisaran 6.5 sampai 8.6 → air dinilai layak diminum.Selain rentang tersebut → air dikategorikan tidak layak.
 3. Setelah pemeriksaan selesai, program menampilkan status kelayakan air sesuai hasil evaluasi.

➤ Kesimpulan

Melalui program ini, pengguna dapat memperoleh penilaian kelayakan air dengan cepat dan otomatis. Struktur logika yang sederhana memastikan hasil yang konsisten tanpa perlu melakukan pengecekan pH secara manual.

2. Tugas 2

Source code

```
package main

import (
    "fmt"
)

func main() {
    var kendaraan string
    var durasi, tarif int
    fmt.Scan(&kendaraan, &durasi)

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

    if durasi < 1 {
        durasi = 1
    }
    biaya := tarif * durasi
    fmt.Printf("Rp %d\n", biaya)
}
```

Screenshot program

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

- Code Editor:** A code editor window titled "soal2.go" containing Go code for calculating parking fees based on vehicle type and duration.
- Terminal:** A terminal window showing the command "go run soal2.go" being run and the output "motor\n3\nRp 6000".
- Application Window:** A window titled "Na" (likely "Nama") displaying user input: "Nama : Primatama Sigalingging" and "NIM: 109082500076".

Deskripsi program

➤ Tujuan

Tujuan program ini adalah menghitung biaya parkir berdasarkan jenis kendaraan dan lama parkir. Perbedaan tarif tiap kendaraan diolah secara otomatis sehingga pengguna langsung mendapatkan total pembayaran yang benar.

➤ Proses

1. Pengguna memilih jenis kendaraan: motor, mobil, atau truk.
2. Pengguna memasukkan lama parkir dalam jam.
3. Program menetapkan tarif berdasarkan kendaraan:
 - Motor → Rp 2.000
 - Mobil → Rp 5.000
 - Truk → Rp 8.000
4. Jika durasi kurang dari 1 jam, program menetapkan durasi minimum menjadi 1 jam.
5. Total biaya dihitung dengan mengalikan tarif dan durasi.
6. Hasil akhir berupa total biaya parkir ditampilkan otomatis.

➤ Kesimpulan

Program ini membantu proses perhitungan biaya parkir menjadi lebih cepat, tepat, dan bebas kesalahan. Pengguna tidak perlu menghitung manual karena tarif dan durasi sudah dianalisis secara otomatis.

3. Tugas 3

Source code

```
package main
import (
    "fmt"
)
func main() {
    var input int
    fmt.Print("Masukkan bilangan bulat: ")
    fmt.Scan(&input)
    switch {
    case input%10 == 0:
        hasil := input / 10
        fmt.Printf("Kategori: Bilangan Kelipatan 10\nHasil pembagian antara %d / 10 = %d\n", input, hasil)
    case input%5 == 0:
        hasil := input * input
        fmt.Printf("Kategori: Bilangan Kelipatan 5\nHasil kuadrat dari %d ^ 2 = %d\n", input, hasil)
    case input%2 == 0:
        nambah1 := input + 1
        hasil := input * nambah1
        fmt.Printf("Kategori: Bilangan Genap\nHasil perkalian dengan bilangan berikutnya %d * %d = %d\n", input, nambah1, hasil)
    default:
        nambah1 := input + 1
        hasil := input + nambah1
        fmt.Printf("Kategori: Bilangan Ganjil\nHasil penjumlahan dengan bilangan berikutnya %d + %d = %d\n",
            input, nambah1, hasil)
    }
}
```

Screenshot program

```
soal3.go
5 func main() {
6     var input int
7     fmt.Print("Masukkan bilangan bulat: ")
8     fmt.Scan(&input)
9     switch {
10         case input%10 == 0:
11             hasil := input / 10
12             fmt.Printf("Kategori: Bilangan Kelipatan 10\nHasil pembagian antara %d / 10 = %d\n", input, hasil)
13         case input%5 == 0:
14             hasil := input * input
15             fmt.Printf("Kategori: Bilangan Kelipatan 5\nHasil kuadrat dari %d ^ 2 = %d\n", input, hasil)
16         case input%2 == 0:
17             nambah1 := input + 1
18             hasil := input * nambah1
19             fmt.Printf("Kategori: Bilangan Genap\nHasil perkalian dengan bilangan berikutnya %d * %d = %d", input, nambah1, hasil)
20         default:
21             nambah1 := input + 1
22             hasil := input + nambah1
23             fmt.Printf("Kategori: Bilangan Ganjil\nHasil penjumlahan dengan bilangan berikutnya %d + %d = %d", input, nambah1, hasil)
24     }
25 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\alproo\go\minggu 11> go run soal3.go
Masukkan bilangan bulat: 20
Kategori: Bilangan Kelipatan 10
Hasil pembagian antara 20 / 10 = 2
PS C:\alproo\go\minggu 11>

Nama : Pramatama Sigalingging
NIM: 10908250076 |

Deskripsi program

➤ Tujuan

Program ini dibuat untuk menentukan kategori dari sebuah bilangan bulat, serta melakukan operasi matematika tertentu berdasarkan kategori tersebut, seperti kelipatan 10, kelipatan 5, genap, atau ganjil.

➤ Proses

1. Pengguna memasukkan satu bilangan bulat.
2. Program memeriksa bilangan melalui beberapa aturan:
 - Jika bilangan habis dibagi 10 → dilakukan operasi pembagian dengan 10.
 - Jika bilangan habis dibagi 5 → dilakukan operasi pemangkatan (bilangan kuadrat).
 - Jika bilangan merupakan bilangan genap → dihitung hasil perkalian bilangan dengan angka setelahnya.
 - Jika bilangan ganjil → dihitung penjumlahan bilangan dengan angka berikutnya.
3. Setelah menentukan kategori, hasil perhitungan ditampilkan sesuai kondisi yang berlaku.

➤ Kesimpulan

Program ini memudahkan pengguna mengidentifikasi karakteristik suatu bilangan dan melihat hasil operasinya secara otomatis. Dengan logika percabangan yang runtut, proses klasifikasi dilakukan secara efisien dan hasilnya diberikan dengan jelas.