

**LAPORAN PRAKTIKUM
ALGORITMA PEMROGRAMAN 2
MODUL II
REVIEW STRUKTUR CONTROL**



Oleh:

ALIFATUS SHABRINA AMALIA

NIM:

2311102225

**S1 TEKNIK INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2024**

I. DASAR TEORI

Struktur program Go terdiri dari paket (package), impor pustaka (import), fungsi utama (func main), dan definisi fungsi atau variabel lain, yang semuanya disusun secara terstruktur untuk memudahkan pengorganisasian dan pembacaan kode.

Untuk menulis, mengompilasi, dan mengeksekusi program Go, Anda cukup membuat file dengan ekstensi `.go`, menggunakan perintah `go build` untuk mengompilasinya menjadi file eksekusi, dan menjalankannya dengan perintah `./nama_file` di terminal, atau langsung menggunakan `go run nama_file.go`.

Dalam Go, tipe data dasar mencakup `int`, `float64`, `bool`, `string`, dan struktur data kompleks seperti `array`, `slice`, `map`, serta instruksi dasar seperti pernyataan variabel (`var`), pernyataan kondisi (`if`, `switch`), perulangan (`for`), dan pemanggilan fungsi (`func`), yang bersama-sama membentuk fondasi untuk pengembangan aplikasi.

Struktur kontrol dalam Go mencakup pernyataan `if`, `switch`, dan `defer`, sedangkan perulangan dilakukan dengan pernyataan `for`, yang dapat digunakan dalam berbagai bentuk untuk mengendalikan alur eksekusi program secara efisien. Sedangkan percabangan yang dilakukan melalui pernyataan `if`, `else`, dan `switch`, yang memungkinkan pengembang untuk mengatur alur eksekusi program berdasarkan kondisi yang ditentukan.

II. GUIDED

1. Guided 1

Source code

```
package main

import "fmt"

func main() {

    var (
        satu, dua, tiga string
        temp string
    )

    fmt.Print("Masukan input string: ")

    fmt.Scanln(&satu)

    fmt.Print("Masukan input string: ")

    fmt.Scanln(&dua)

    fmt.Print("Masukan input string: ")

    fmt.Scanln(&tiga)

    fmt.Println("Output awal = " + satu + " " + dua + " " + tiga)

    temp = satu

    satu = dua

    dua = tiga

    tiga = temp

    fmt.Println("Output akhir = " + satu + " " + dua + " " + tiga)

}
```

Screenshoot program

```
PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS D:\Praktikum\Alpro2\Modul2\guided1> go run "d:\Praktikum\Alpro2\Modul2\guided1\guided1.go"
Masukan input string: 2
Masukan input string: 4
Masukan input string: 6
Output awal = 2 4 6
Output akhir = 4 6 2
PS D:\Praktikum\Alpro2\Modul2\guided1> |
```

Deskripsi program

Program tersebut meminta pengguna memasukkan tiga string, kemudian menukar nilai string tersebut menggunakan variabel sementara, sehingga urutan nilai berubah, dan menampilkan hasil penukarannya.

2. Guided 2

Source code

```
package main

import "fmt"

func main() {
    year := 0
    fmt.Scan(&year)
    fmt.Println(year%4 == 0 && year%100 != 0 || year%400 == 0)
}
```

Screenshoot program

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS D:\Praktikum\Alpro2\Modul2\Guided2> go run "d:\Praktikum\Alpro2\Modul2\Guided2\guided2.go"
2020
true
PS D:\Praktikum\Alpro2\Modul2\Guided2> |
```

Deskripsi program

- Tahun kabisat adalah tahun yang habis dibagi 4, kecuali jika habis dibagi 100, kecuali jika habis dibagi 400.
- Kondisi $\text{year} \% 4 == 0 \ \&\& \ \text{year} \% 100 != 0 \ || \ \text{year} \% 400 == 0$ adalah syarat yang tepat untuk menentukan apakah suatu tahun adalah kabisat.

3. Guided 3

Source code

```
#include <iostream>

#include <iomanip>

using namespace std;

string simpul[7] = { "Ciamis",
                    "Bandung",
                    "Bekasi",
                    "Tasikmalaya",
                    "Cianjur",
                    "Purwokerto",
                    "Yogyakarta" };

int busur[7][7] =
{
    {0,7,8,0,0,0,0},
    {0,0,5,0,0,15,0},
    {0,6,0,0,5,0,0},
    {0,5,0,0,2,4,0},
    {23,0,0,10,0,0,8},
    {0,0,0,0,7,0,3},
    {0,0,0,0,9,4,0} };
```

```

void tampilGraph()
{
    for (int baris = 0; baris < 7; baris++) {
        cout << " " << setiosflags(ios::left) << setw(15) <<
simpul[baris] << " : ";

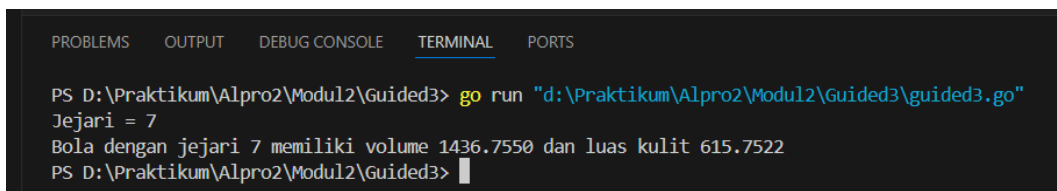
        for (int kolom = 0; kolom < 7; kolom++) {
            if (busur[baris][kolom] != 0) {
                cout << " " << simpul[kolom] << "("
                    << busur[baris][kolom] << ") ";
            }
        }
        cout << endl;
    }
}

int main()
{
    tampilGraph();

    return 0;
}

```

Screenshoot program



```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS D:\Praktikum\Alpro2\Modul2\Guided3> go run "d:\Praktikum\Alpro2\Modul2\Guided3\guided3.go"
Jejari = 7
Bola dengan jejari 7 memiliki volume 1436.7550 dan luas kulit 615.7522
PS D:\Praktikum\Alpro2\Modul2\Guided3>

```

Deskripsi program

- Program ini menggunakan paket math untuk melakukan operasi pangkat (math.Pow).

- Menggunakan konstanta $\pi = 3.1415926535$ untuk perhitungan volume dan luas permukaan bola.
- Input jari-jari yang dimasukkan oleh pengguna disimpan dalam variabel `jejari`.
- Formula volume bola adalah $(4/3) * \pi * r^3$ dan luas permukaan bola adalah $4 * \pi * r^2$, di mana r adalah jari-jari.
- Program ini akan secara otomatis menghitung volume dan luas kulit bola dengan jari-jari yang diberikan.

III. UNGUIDED

1. Unguided 1

Source code

```
package main

import (
    "fmt"
)

func main() {
    var celsius float64

    fmt.Print("Temperatur Celsius: ")
    fmt.Scanln(&celsius)

    fahrenheit := (celsius * 9 / 5) + 32

    reamur := celsius * 4 / 5

    kelvin := celsius + 273.15

    fmt.Printf("Derajat Reamur: %.2f\n", reamur)
    fmt.Printf("Derajat Fahrenheit: %.2f\n", fahrenheit)
    fmt.Printf("Derajat Kelvin: %.2f\n", kelvin)
}
```


Screenshoot program

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS D:\Praktikum\Alpro2\Modul2\Unguided1> go run "d:\Praktikum\Alpro2\Modul2\Unguided1\unguided1.go"
Temperatur Celsius: 20
Derajat Reamur: 16.00
Derajat Fahrenheit: 68.00
Derajat Kelvin: 293.15
PS D:\Praktikum\Alpro2\Modul2\Unguided1> █
```

Deskripsi program

- Input: Program meminta pengguna untuk memasukkan temperatur dalam derajat Celsius, yang disimpan dalam variabel celsius.
- Konversi:
 1. Fahrenheit: Dihitung menggunakan rumus $(Celsius \times \frac{9}{5}) + 32$.
 2. Reamur: Dihitung menggunakan rumus $Celsius \times \frac{4}{5}$.
 3. Kelvin: Dihitung dengan menambahkan 273.15 ke nilai Celsius.
- Output: Program menampilkan temperatur dalam derajat Reamur, Fahrenheit, dan Kelvin dengan dua angka di belakang koma.
- Program ini menyediakan hasil konversi dengan presisi yang baik sesuai dengan format yang diinginkan.

2. Unguided 2

Source code

```
package main

import (
    "fmt"
)

func main() {
    var integers [5]int
```

```
var chars [3]rune

fmt.Print("Masukkan 5 buah data integer (antara 32 hingga 127):")
for i := 0; i < 5; i++ {
    fmt.Scan(&integers[i])
}

fmt.Print("Masukkan 3 karakter (tanpa spasi): ")
for i := 0; i < 3; i++ {
    fmt.Scanf("%c", &chars[i])

    fmt.Scanf("%c")
}

fmt.Print("Hasil karakter dari data integer: ")
for _, val := range integers {
    fmt.Printf("%c", val)
}

fmt.Println()

fmt.Print("Karakter setelah karakter input: ")
for _, char := range chars {
    fmt.Printf("%c", char+1)
}

fmt.Println()
}
```

Screenshoot program

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS D:\Praktikum\Alpro2\Modul2\Unguided2> go run "d:\Praktikum\Alpro2\Modul2\Unguided2\unguided2.go"
Masukkan 5 buah data integer (antara 32 hingga 127): 65 66 67 68 69
Masukkan 3 karakter (tanpa spasi): ABC
Hasil karakter dari data integer: ABCDE
Karakter setelah karakter input:
BC
PS D:\Praktikum\Alpro2\Modul2\Unguided2> |
```

Deskripsi program

- Input Integer: Program meminta pengguna untuk memasukkan 5 buah integer dalam rentang 32 hingga 127. Nilai-nilai ini disimpan dalam array integers.
- Input Karakter: Program meminta pengguna untuk memasukkan 3 karakter. Karena kita menggunakan `fmt.Scanf("%c", &chars[i])` untuk membaca karakter, kita juga perlu membaca karakter newline setelah setiap karakter untuk menghindari pembacaan karakter kosong.
- Mencetak Karakter dari Integer: Program mencetak karakter yang sesuai dengan nilai integer yang dimasukkan.
- Mencetak Karakter Berikutnya: Program mencetak karakter yang berada satu posisi setelah setiap karakter yang dimasukkan dengan menambahkan 1 ke nilai ASCII karakter tersebut.

Program ini menggunakan `fmt.Printf` untuk format output yang tepat dan `fmt.Scanf` untuk input karakter dengan cara yang sesuai.

3. Unguided 3

Source code

```
package main

import (
    "fmt"
)

func main() {
```

```

var warna [4]string

var berhasil bool = true


for i := 1; i <= 5; i++ {

    fmt.Printf("Percobaan %d: ", i)

    fmt.Scan(&warna[0], &warna[1], &warna[2], &warna[3])


    if warna[0] != "merah" || warna[1] != "kuning" || warna[2]
!= "hijau" || warna[3] != "ungu" {

        berhasil = false

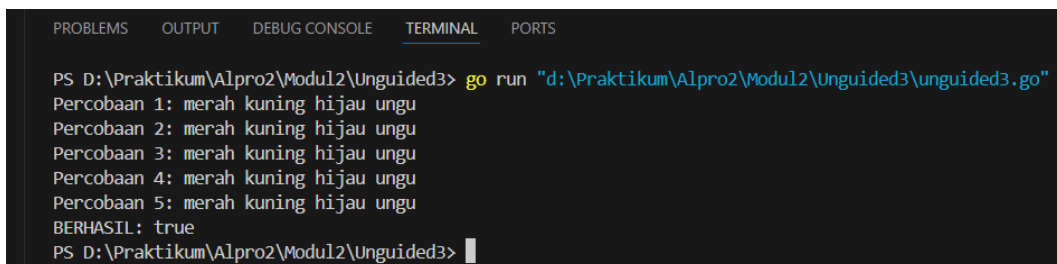
    }

}

fmt.Printf("BERHASIL: %t\n", berhasil)
}

```

Screenshoot program

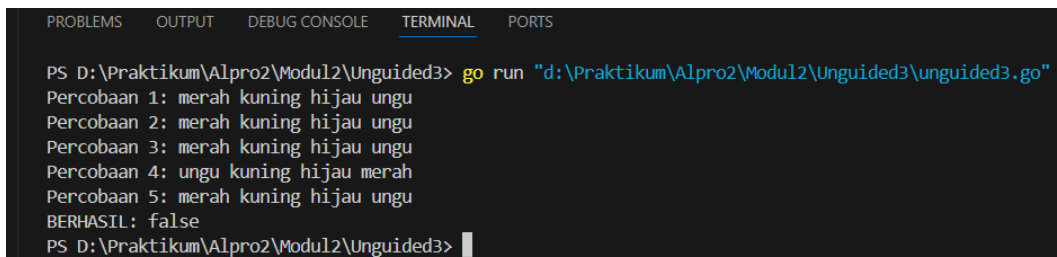


```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS D:\Praktikum\Alpro2\Modul2\Unguided3> go run "d:\Praktikum\Alpro2\Modul2\Unguided3\unguided3.go"
Percobaan 1: merah kuning hijau ungu
Percobaan 2: merah kuning hijau ungu
Percobaan 3: merah kuning hijau ungu
Percobaan 4: merah kuning hijau ungu
Percobaan 5: merah kuning hijau ungu
BERHASIL: true
PS D:\Praktikum\Alpro2\Modul2\Unguided3>

```



```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS D:\Praktikum\Alpro2\Modul2\Unguided3> go run "d:\Praktikum\Alpro2\Modul2\Unguided3\unguided3.go"
Percobaan 1: merah kuning hijau ungu
Percobaan 2: merah kuning hijau ungu
Percobaan 3: merah kuning hijau ungu
Percobaan 4: ungu kuning hijau merah
Percobaan 5: merah kuning hijau ungu
BERHASIL: false
PS D:\Praktikum\Alpro2\Modul2\Unguided3>

```

Deskripsi program

- Deklarasi Variabel: Program mendeklarasikan array `warna` untuk menyimpan warna dari 4 tabung reaksi dan variabel `berhasil` yang diinisialisasi dengan `true`.
- Loop untuk Input: Menggunakan loop untuk meminta input dari pengguna sebanyak 5 kali (untuk 5 percobaan).
- Input Warna: Program membaca 4 warna sekaligus menggunakan `fmt.Scan(&warna[0], &warna[1], &warna[2], &warna[3])`.
- Pemeriksaan Urutan: Program memeriksa apakah urutan warna sesuai dengan yang diinginkan ('merah', 'kuning', 'hijau', 'ungu'). Jika tidak, variabel `berhasil` diubah menjadi `false`.
- Output: Setelah semua percobaan selesai, program mencetak apakah percobaan berhasil (`true`) atau tidak (`false`).

Program ini akan memeriksa setiap percobaan dan memberikan hasil yang sesuai berdasarkan urutan warna yang diinputkan.

4. Unguided 4

Source code

```
package main

import (
    "fmt"
    "strings"
)

func main() {
    var pita string
    var count int

    var N int

    fmt.Print("N: ")

    fmt.Scanln(&N)
```

```
if N > 0 {  
    for i := 1; i <= N; i++ {  
        var bunga string  
        fmt.Printf("Bunga %d: ", i)  
        fmt.Scanln(&bunga)  
        pita += bunga + " - "  
    }  
    fmt.Println("Pita:", strings.TrimSpace(pita))  
} else {  
    fmt.Println("Pita: ")  
}  
  
pita = ""  
count = 0  
  
for {  
    var bunga string  
    fmt.Print("Bunga: ")  
    fmt.Scanln(&bunga)  
  
    if strings.ToUpper(bunga) == "SELESAI" {  
        break  
    }  
  
    pita += bunga + " - "
```

```

        count++

    }

    fmt.Println("Pita:", strings.TrimSpace(pita))

    fmt.Printf("Bunga: %d\n", count)

}

```

Screenshoot program

The screenshot shows a terminal window with the following output:

```

PS D:\Praktikum\Alpro2\Modul2\Unguided4> go run "d:\Praktikum\Alpro2\Modul2\Unguided4\unguided4.go"
N: 3
Bunga 1: Kertas
Bunga 2: Mawar
Bunga 3: Tulip
Pita: Kertas - Mawar - Tulip -
Bunga: 

```

Deskripsi program

- Deklarasi Variabel: Program mendeklarasikan variabel pita untuk menyimpan nama-nama bunga dan count untuk menghitung jumlah bunga.
- Input Bilangan Bulat Positif N: Program meminta input bilangan bulat positif N dan memeriksa apakah N lebih besar dari 0.
- Pengulangan Input Nama Bunga: Program menggunakan loop untuk meminta nama bunga sebanyak N kali. Setiap nama bunga yang dimasukkan akan digabungkan dengan pemisah " - ".
- Menampilkan Isi Pita: Setelah proses input selesai, program menampilkan isi pita.
- Modifikasi Program untuk Input hingga "SELESAI": Program mengosongkan pita dan count sebelum memulai proses input baru. Input akan berlanjut hingga pengguna mengetik "SELESAI".
- Menampilkan Hasil Akhir: Program menampilkan isi pita dan jumlah bunga yang dimasukkan setelah "SELESAI" diketik

5. Unguided 5

Source code

```
package main

import (
    "fmt"
)

func main() {
    var kantong1, kantong2 float64

    for {
        fmt.Print("Masukan berat belanjaan di kedua kantong: ")
        fmt.Scan(&kantong1, &kantong2)

        if kantong1 < 0 || kantong2 < 0 {
            fmt.Println("Proses selesai.")
            break
        }

        selisih := kantong1 - kantong2

        if selisih < 0 {
            selisih = -selisih
        }

        if selisih >= 9 {
            fmt.Println("Sepeda motor pak Andi akan oleng: true")
        } else {
            fmt.Println("Sepeda motor pak Andi akan oleng: false")
        }
    }
}
```



```

    }

    totalBerat := kantong1 + kantong2

    if totalBerat > 150 {

        fmt.Println("Proses selesai.")

        break

    }

}
}

```

Screenshoot program

```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
PS D:\Praktikum\Alpro2\Modul2\Unguided5> go run "d:\Praktikum\Alpro2\Modul2\Unguided5\unguided5.go"
Masukan berat belanjaan di kedua kantong: 5 10
Sepeda motor pak Andi akan oleng: false
Masukan berat belanjaan di kedua kantong: 55.6 70.2
Sepeda motor pak Andi akan oleng: true
Masukan berat belanjaan di kedua kantong: 72.3 66.9
Sepeda motor pak Andi akan oleng: false
Masukan berat belanjaan di kedua kantong: 59.5 98.7
Sepeda motor pak Andi akan oleng: true
Proses selesai.
PS D:\Praktikum\Alpro2\Modul2\Unguided5>

```

Deskripsi program

- Deklarasi Variabel: Program mendeklarasikan dua variabel kantong1 dan kantong2 untuk menyimpan berat belanjaan di kedua kantong.
- Loop Input: Program menggunakan loop for untuk meminta input berat dari kedua kantong berulang kali.
- Input Berat: Menggunakan `fmt.Scan(&kantong1, &kantong2)` untuk membaca input berat dari pengguna.
- Negatif Check: Jika salah satu kantong berisi berat negatif, program akan menghentikan proses dengan mencetak "Proses selesai."
- Selisih Berat: Menghitung selisih berat antara kantong1 dan kantong2. Jika selisihnya negatif, maka diambil nilai absolutnya.
- Menampilkan Oleng: Program memeriksa apakah selisih berat lebih dari atau sama dengan 9 kg dan mencetak hasilnya.

- Total Berat Check: Jika total berat dari kedua kantong melebihi 150 kg, program akan menghentikan proses dengan mencetak "Proses selesai."

Program ini dirancang untuk mengikuti semua spesifikasi yang Anda berikan dan akan memberikan output yang diharapkan berdasarkan input yang diberikan.

6. Unguided 6

Source code

```
package main

import (
    "fmt"
)

func main() {
    var k float64

    fmt.Print("Nilai K= ")
    fmt.Scanln(&k)

    fK := (4 * k) / (k + k + 3)

    fmt.Printf("Nilai f(K) = %.10f\n", fK)
}
```

```
package main
```

```

import (

    "fmt"

    "math"

)

func main() {

    var K int

    fmt.Print("Nilai K= ")

    fmt.Scanln(&K)

    akar2 := (1 + float64(K) + 3) / (1 + 2)

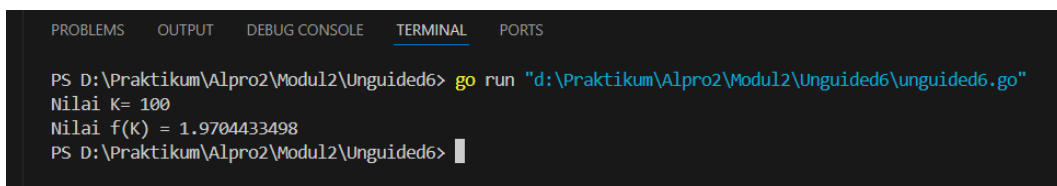
    fmt.Printf("Nilai akar2 = %.10f\n", akar2)

    fmt.Printf("Nilai akar 2 = %.10f\n", math.Sqrt(2))

}

```

Screenshoot program

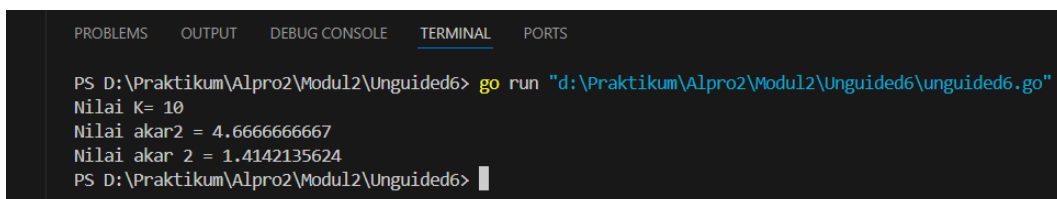


PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```

PS D:\Praktikum\Alpro2\Modul2\Unguided6> go run "d:\Praktikum\Alpro2\Modul2\Unguided6\unguided6.go"
Nilai K= 100
Nilai f(K) = 1.9704433498
PS D:\Praktikum\Alpro2\Modul2\Unguided6>

```



PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```

PS D:\Praktikum\Alpro2\Modul2\Unguided6> go run "d:\Praktikum\Alpro2\Modul2\Unguided6\unguided6.go"
Nilai K= 10
Nilai akar2 = 4.6666666667
Nilai akar 2 = 1.4142135624
PS D:\Praktikum\Alpro2\Modul2\Unguided6>

```

Deskripsi program

Penjelasan Program Pertama:

- Input K: Program meminta pengguna untuk memasukkan nilai K.
- Perhitungan: Menghitung nilai $f(K)$ menggunakan rumus $f(k) = \frac{4k}{k+k+3}$.
- Output: Menampilkan hasil $f(K)$ dengan format 10 angka di belakang koma.

Penjelasan Program Kedua:

- Input K: Program meminta pengguna untuk memasukkan nilai integer K.
- Perhitungan Akar 2: Menghitung $\sqrt{2}$ menggunakan rumus yang telah dimodifikasi dan menampilkan hasil dengan ketelitian 10 angka di belakang koma.
- Output: Menggunakan `math.Sqrt(2)` untuk menghitung akar 2 dan menampilkannya dengan format yang ditentukan.

7. Unguided 7

Source code

```
package main

import (
    "fmt"
)

func main() {
    var beratParsel int

    const biayaPerKg = 10000

    const tambahanBiaya500gr = 5

    const tambahanBiayaKurang500gr = 15

    fmt.Print("Berat parsel (gram): ")

    fmt.Scanln(&beratParsel)
```

```
beratKg := beratParsel / 1000

sisagram := beratParsel % 1000

fmt.Printf("Detail berat: %d kg + %d gr\n", beratKg, sisagram)

totalBiaya := beratKg * biayaPerKg

if beratKg > 10 {

    fmt.Printf("Detail biaya: Rp. %d + Rp. 0\n", totalBiaya)

    totalBiaya += 0

} else if sisagram >= 500 {

    tambahanBiaya := sisagram * tambahanBiaya500gr

    fmt.Printf("Detail biaya: Rp. %d + Rp. %d\n", totalBiaya,
tambahanBiaya)

    totalBiaya += tambahanBiaya

} else {

    tambahanBiaya := sisagram * tambahanBiayaKurang500gr

    fmt.Printf("Detail biaya: Rp. %d + Rp. %d\n", totalBiaya,
tambahanBiaya)

    totalBiaya += tambahanBiaya

}

fmt.Printf("Total biaya: Rp. %d\n", totalBiaya)

}
```

Screenshoot program

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS D:\Praktikum\Alpro2\Modul2\Unguided7> go run "d:\Praktikum\Alpro2\Modul2\Unguided7\unguided7.go"
Berat parcel (gram): 8500
Detail berat: 8 kg + 500 gr
Detail biaya: Rp. 80000 + Rp. 2500
Total biaya: Rp. 82500
PS D:\Praktikum\Alpro2\Modul2\Unguided7> |
```

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS D:\Praktikum\Alpro2\Modul2\Unguided7> go run "d:\Praktikum\Alpro2\Modul2\Unguided7\unguided7.go"
Berat parcel (gram): 9250
Detail berat: 9 kg + 250 gr
Detail biaya: Rp. 90000 + Rp. 3750
Total biaya: Rp. 93750
PS D:\Praktikum\Alpro2\Modul2\Unguided7> |
```

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS D:\Praktikum\Alpro2\Modul2\Unguided7> go run "d:\Praktikum\Alpro2\Modul2\Unguided7\unguided7.go"
Berat parcel (gram): 11750
Detail berat: 11 kg + 750 gr
Detail biaya: Rp. 110000 + Rp. 0
Total biaya: Rp. 110000
PS D:\Praktikum\Alpro2\Modul2\Unguided7> |
```

Deskripsi program

- Input Berat Parcel: Program meminta pengguna untuk memasukkan berat parcel dalam gram.
- Menghitung Berat dalam Kg dan Sisa: Berat dalam kg dihitung dengan membagi berat parcel dengan 1000, dan sisa berat dalam gram dihitung dengan sisa pembagian berat parcel dengan 1000.
- Menampilkan Detail Berat: Menampilkan detail berat dalam format "x kg + y gr".
- Menghitung Biaya:
 1. Biaya dasar dihitung dengan mengalikan berat dalam kg dengan biaya per kg.
 2. Jika berat lebih dari 10 kg, sisa berat digratiskan.
 3. Jika sisa berat 500 gram atau lebih, biaya tambahan dihitung dengan mengalikan sisa berat dengan Rp. 5.
 4. Jika sisa berat kurang dari 500 gram, biaya tambahan dihitung dengan mengalikan sisa berat dengan Rp. 15.

- Menampilkan Total Biaya: Menampilkan total biaya kirim berdasarkan perhitungan.

8. Unguided 8

A. Keluaran Program untuk NAM = 80.1

Jika nam diberikan nilai 80.1, maka keluaran dari program yang tidak benar (sebelum perbaikan) akan mengarah pada error atau keluaran yang tidak sesuai karena kode di dalam program tidak terstruktur dengan benar. Dengan program yang cacat tersebut, program tidak akan mengembalikan nilai nmk yang benar.

Keluaran seharusnya untuk nilai nam = 80.1 adalah 'A', tetapi program tersebut tidak akan menghasilkan keluaran yang sesuai.

B. Kesalahan dalam Program

1. Struktur Kontrol yang Salah: Dalam program asli, struktur if dan else if tidak diatur dengan benar. Kode yang ada tidak mengikuti struktur yang tepat dan mengandung kesalahan sintaks.
2. Penggunaan Nama Variabel: Variabel nmk tidak digunakan dengan benar, program juga tidak menyimpan nilai penilaian yang benar.
3. Kesalahan dalam Logika Penentuan Nilai: Kode saat ini tidak mengikuti urutan logika yang seharusnya, sehingga beberapa kondisi tidak akan pernah terpenuhi.

Alur Program yang Seharusnya:

- Menerima input nilai nam.
- Memeriksa rentang nilai dan menetapkan nilai nmk berdasarkan rentang yang telah ditentukan.
- Menampilkan nilai akhir mata kuliah.

C. Perbaikan Program

Berikut adalah program Go yang sudah diperbaiki:

Source code

```
package main

import "fmt"

func main() {

    var nam float64

    var nmk string

    fmt.Print("Nilai akhir mata kuliah: ")

    fmt.Scanln(&nam)

    if nam > 80 {

        nmk = "A"

    } else if nam > 72.5 {

        nmk = "AB"

    } else if nam > 65 {

        nmk = "B"

    } else if nam > 57.5 {

        nmk = "BC"

    } else if nam > 50 {

        nmk = "C"

    } else if nam > 40 {

        nmk = "D"

    } else {

        nmk = "E"

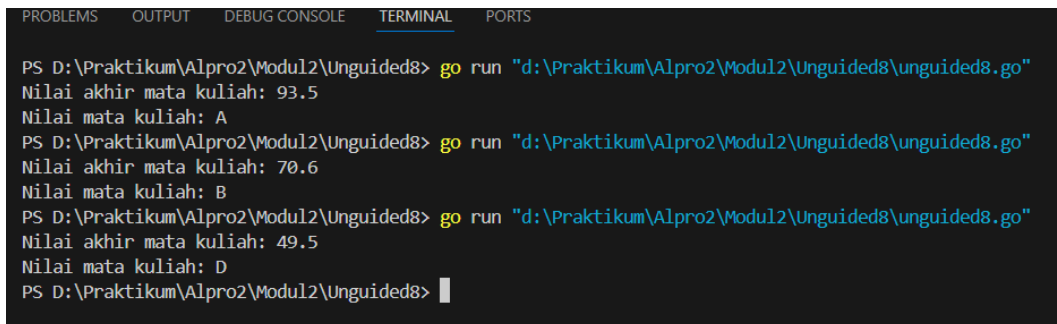
    }

}
```



```
fmt.Println("Nilai mata kuliah:", nmk)
}
```

Screenshoot program



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS D:\Praktikum\Alpro2\Modul2\Unguided8> go run "d:\Praktikum\Alpro2\Modul2\Unguided8\unguided8.go"
Nilai akhir mata kuliah: 93.5
Nilai mata kuliah: A
PS D:\Praktikum\Alpro2\Modul2\Unguided8> go run "d:\Praktikum\Alpro2\Modul2\Unguided8\unguided8.go"
Nilai akhir mata kuliah: 70.6
Nilai mata kuliah: B
PS D:\Praktikum\Alpro2\Modul2\Unguided8> go run "d:\Praktikum\Alpro2\Modul2\Unguided8\unguided8.go"
Nilai akhir mata kuliah: 49.5
Nilai mata kuliah: D
PS D:\Praktikum\Alpro2\Modul2\Unguided8> |
```

Uji Coba Program

Sekarang kita bisa menguji program dengan input yang diberikan:

1. Input: 93.5
 - o Output: Nilai mata kuliah: A
2. Input: 70.6
 - o Output: Nilai mata kuliah: B
3. Input: 49.5
 - o Output: Nilai mata kuliah: D

Deskripsi program

Dengan perbaikan yang dilakukan, program kini berjalan sesuai spesifikasi yang diberikan. Setiap input nilai akhir mata kuliah (NAM) menghasilkan nilai mata kuliah (NMK) yang tepat berdasarkan ketentuan yang telah ditetapkan.

9. Unguided 9

Source code

```
package main

import (
```

```
"fmt"

)

func main() {

    var b int

    fmt.Print("Bilangan: ")

    fmt.Scanln(&b)

    if b <= 1 {

        fmt.Println("Silakan masukkan bilangan bulat b > 1.")

        return

    }

    fmt.Print("Faktor: ")

    isPrima := true

    for i := 1; i <= b; i++ {

        if b%i == 0 {

            fmt.Print(i, " ")

            if i > 1 && i < b {

                isPrima = false

            }

        }

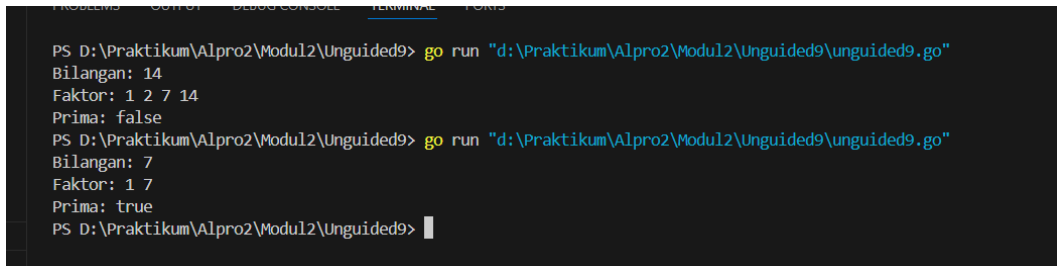
    }

    fmt.Println()

    fmt.Printf("Prima: %t\n", isPrima)
```

```
}
```

Screenshoot program



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  TESTS
PS D:\Praktikum\Alpro2\Modul2\Unguided9> go run "d:\Praktikum\Alpro2\Modul2\Unguided9\unguided9.go"
Bilangan: 14
Faktor: 1 2 7 14
Prima: false
PS D:\Praktikum\Alpro2\Modul2\Unguided9> go run "d:\Praktikum\Alpro2\Modul2\Unguided9\unguided9.go"
Bilangan: 7
Faktor: 1 7
Prima: true
PS D:\Praktikum\Alpro2\Modul2\Unguided9> |
```

Deskripsi program

- Input Bilangan: Program meminta pengguna untuk memasukkan bilangan bulat b.
- Pengecekan Validitas: Program memeriksa apakah b lebih besar dari 1. Jika tidak, program akan meminta pengguna untuk memasukkan nilai yang benar.
- Mencari dan Menampilkan Faktor: Program menggunakan loop untuk mencari dan mencetak semua faktor dari b.
 1. Jika i adalah faktor, maka akan dicetak ke layar.
 2. Jika ditemukan faktor lain selain 1 dan b, maka isPrima diset ke false.
- Menampilkan Hasil: Program menampilkan hasil apakah b adalah bilangan prima.