

**LAPORAN PRAKTIKUM
PEMROGRAMAN BERORIENTASI OBJEK**

**MODUL 2
REVIEW STRUKTUR KONTROL**



Oleh:

DAMARA GALUH PEMBAYUN

2311102110

IF-11-02

**S1 TEKNIK INFORMATIKA
INSTITUT TEKNOLOGI TELKOM PURWOKERTO
2024**

I. DASAR TEORI

Meskipun program Golang memiliki struktur yang sederhana, mereka sangat efektif. Setiap program Golang biasanya terdiri dari satu atau lebih paket. Paket adalah cara mengatur kode menjadi modul yang lebih kecil yang dapat digunakan kembali.

Dalam program ini, golang selalu memiliki program utama yaitu package buat isi program utama, dan func main buat kode utama program Go.

Ada beberapa tipe data golang. Seperti integer, float, string, Boolean, dan lain lain. Jadi dapat di simpulkan bahwa Struktur program Golang sangat sederhana dan mudah dipahami. Dengan pemahaman yang baik tentang paket, fungsi, variabel, dan kontrol aliran, Anda dapat mulai membangun program Golang yang lebih kompleks.

Contoh code :

```
package main

import "fmt"

func main() {
    var nama string
    nama = "Damara"
    fmt.Println("Hello, ", nama)
}
```

Keterangan :

- package main: Menyatakan bahwa program ini adalah program utama.
- import "fmt": Mengimpor paket fmt yang berisi fungsi-fungsi untuk input/output.
- var nama string: Mendeklarasikan variabel nama dengan tipe data string.
- nama = "Damara": Memberikan nilai "Damara" ke variabel nama.
- fmt.Println("Hello, ", nama): Mencetak pesan "Hello, John Doe" ke layar.

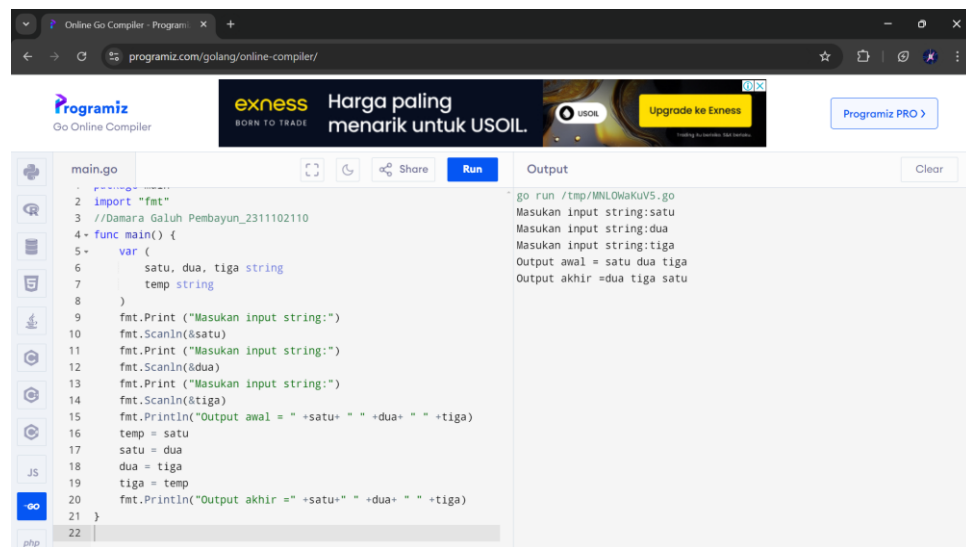
II. GUIDED

Guided 1

SourchCode

```
package main
import "fmt"
//Damara Galuh Pembayun_2311102110
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)
}
```

Hasil ScreenShoot



Deskripsi

Kode ini meminta pengguna untuk memasukkan tiga string, kemudian menggunakan variabel sementara temp untuk menukar posisi ketiga string tersebut, dan kemudian mencetak hasil penukaran.

Guided 2

SourchCode

```
package main
//Damara Galuh Pembayun 2311102110
import (

    "fmt"
    "math"
)

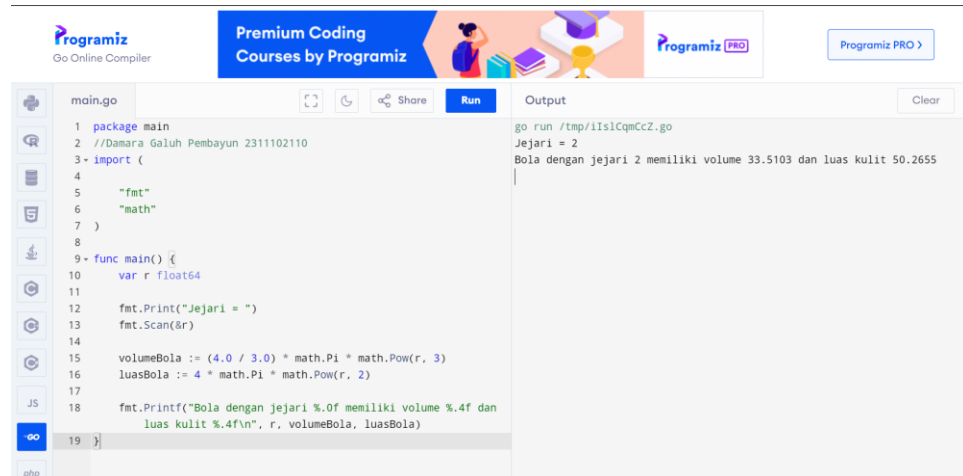
func main() {
    var r float64

    fmt.Print("Jejari = ")
    fmt.Scan(&r)

    volumeBola := (4.0 / 3.0) * math.Pi * math.Pow(r, 3)
    luasBola := 4 * math.Pi * math.Pow(r, 2)

    fmt.Printf("Bola dengan jejari %.0f memiliki volume
    %.4f dan luas kulit %.4f\n", r, volumeBola, luasBola)
}
```

Hasil ScreenShoot



Deskripsi

Program ini menerima input berupa jari-jari sebuah bola, kemudian menghitung dan menampilkan volume serta luas permukaan bola tersebut. Jika jari jari 2cm maka volumenya 33,5103 dan luas kulitnya 50.2655.

Guided 3

SourchCode

```
package main

import (
    "fmt"
```

```

)

func main() {
    var celsius float64

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

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

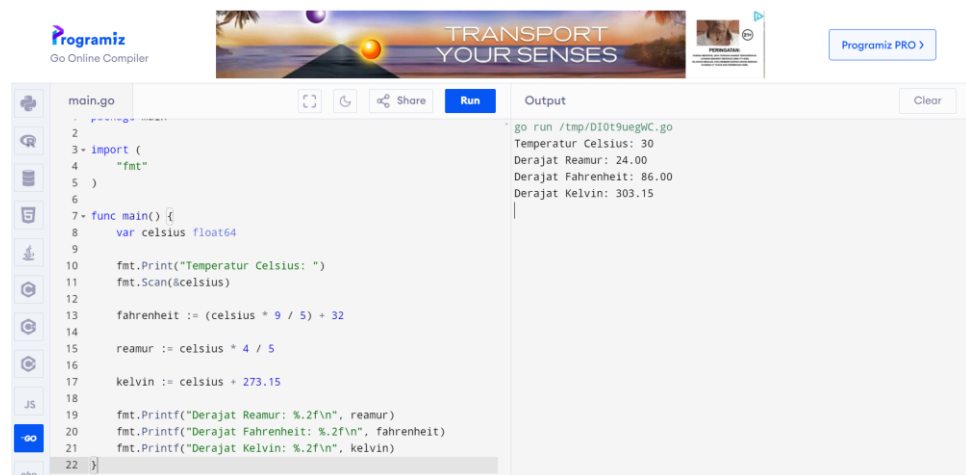
    reamur := celsius * 4 / 5

    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)
}

```

Hasil ScreenShoot



Deskripsi

Program ini menerima input suhu dalam derajat Celsius, kemudian mengkonversi suhu tersebut ke satuan suhu lainnya (Reamur, Fahrenheit, dan Kelvin) dan menampilkan hasilnya. Jika memasukan 30 derajat celcius akan menjadi 24 reamur, 86 fahrenheit, dan 303,15 kelvin.

III. UNGUIDED

Unguided 1 (2B no.1)

SourchCode

```

package main
//Damara Galuh Pembayun_2311102110

import "fmt"

func main() {

```

```

        var warnaTujuan = []string{"merah", "kuning",
"hijau", "ungu"}
        var percobaan int = 5

        for i := 1; i <= percobaan; i++ {
            fmt.Printf("Percobaan %d: ", i)
            var warnaInput [4]string
            fmt.Scanln(&warnaInput[0], &warnaInput[1],
&warnaInput[2], &warnaInput[3])

            var berhasil bool = true
            for j := 0; j < len(warnaTujuan); j++ {
                if warnaInput[j] != warnaTujuan[j]
{
                    berhasil = false
                    break
                }
            }

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

```

Hasil ScreenShoot

The screenshot shows the Programiz Online Go Compiler interface. On the left, the code editor displays the Go code from the previous block. On the right, the 'Output' panel shows the results of running the program. The output consists of five lines of input prompts and their corresponding success/failure status.

```

go run /tmp/m14xYd0I39.go
Percobaan 1: merah kuning hijau ungu
BERHASIL: true
Percobaan 2: merah kuning hijau ungu
BERHASIL: true
Percobaan 3: merah kuning hijau ungu
BERHASIL: true
Percobaan 4: ungu kuning hijau merah
BERHASIL: false
Percobaan 5: merah kuning hijau ungu
BERHASIL: true

```

Deskripsi

Program ini dirancang untuk melakukan beberapa eksperimen di mana pengguna diminta untuk memasukkan empat warna berbeda dalam setiap eksperimen. Setelah itu, program akan membandingkan warna-warna yang dimasukkan dengan daftar warna tujuan yang telah ditentukan, dan jika semua warna cocok dalam urutan yang sama, eksperimen dianggap berhasil.

Unguided 2 (2B no.2)

SourceCode

```
package main

import "fmt"

func main() {
    var n int
    var bunga string
    var pita string

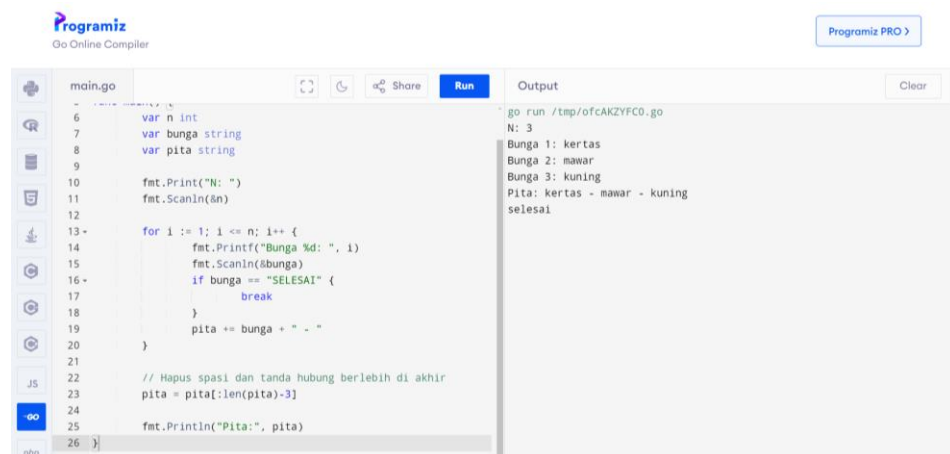
    fmt.Print("N: ")
    fmt.Scanln(&n)

    for i := 1; i <= n; i++ {
        fmt.Printf("Bunga %d: ", i)
        fmt.Scanln(&bunga)
        if bunga == "SELESAI" {
            break
        }
        pita += bunga + " - "
    }

    // Hapus spasi dan tanda hubung berlebih di akhir
    pita = pita[:len(pita)-3]

    fmt.Println("Pita:", pita)
}
```

Hasil ScreenShoot



Deskripsi

Program ini dirancang untuk mengumpulkan nama-nama bunga yang diinputkan oleh pengguna hingga pengguna memasukkan kata "SELESAI". Semua nama bunga yang diinputkan kemudian akan digabungkan menjadi satu string dan ditampilkan.

Unguided 3 (2B no.3)

SourceCode

```

package main
//Damara Galuh Pembayun_2311102110

import "fmt"

func main() {
    var beratKiri, beratKanan float64
    var totalBerat float64

    for {
        fmt.Print("Masukkan berat belanjaan di
kedua kantong: ")
        fmt.Scan(&beratKiri, &beratKanan)

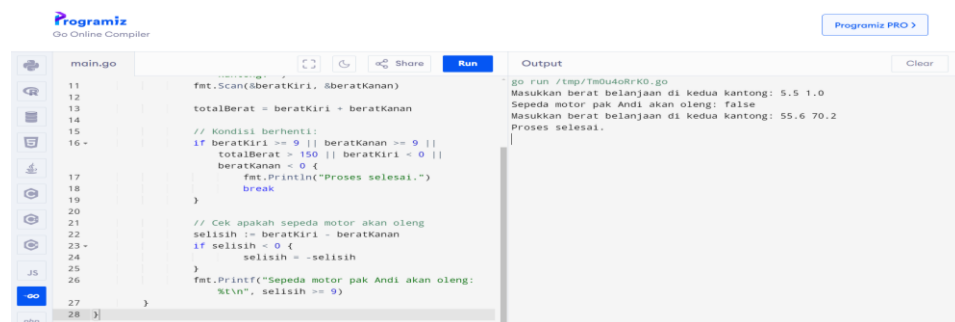
        totalBerat = beratKiri + beratKanan

        // Kondisi berhenti:
        if beratKiri >= 9 || beratKanan >= 9 ||
totalBerat > 150 || beratKiri < 0 || beratKanan < 0 {
            fmt.Println("Proses selesai.")
            break
        }

        // Cek apakah sepeda motor akan oleng
        selisih := beratKiri - beratKanan
        if selisih < 0 {
            selisih = -selisih
        }
        fmt.Printf("Sepeda motor pak Andi akan
oleng: %t\n", selisih >= 9)
    }
}

```

Hasil ScreenShoot



Deskripsi

Code ini dirancang untuk mensimulasikan proses pemuatan belanjaan ke dalam dua kantong yang digantung di sepeda motor. Program akan terus meminta pengguna untuk memasukkan berat belanjaan di kedua kantong hingga kondisi tertentu terpenuhi, yaitu:

- **Beban Berlebih:** Salah satu atau kedua kantong melebihi berat maksimum yang ditentukan (dalam kasus ini, 9 kg).

- **Beban Total Berlebih:** Total berat kedua kantong melebihi batas maksimum (dalam kasus ini, 150 kg).
- **Input Tidak Valid:** Pengguna memasukkan nilai berat yang negatif.
Selain itu, program juga akan memeriksa keseimbangan beban antara kedua kantong untuk menentukan apakah sepeda motor akan oleng.

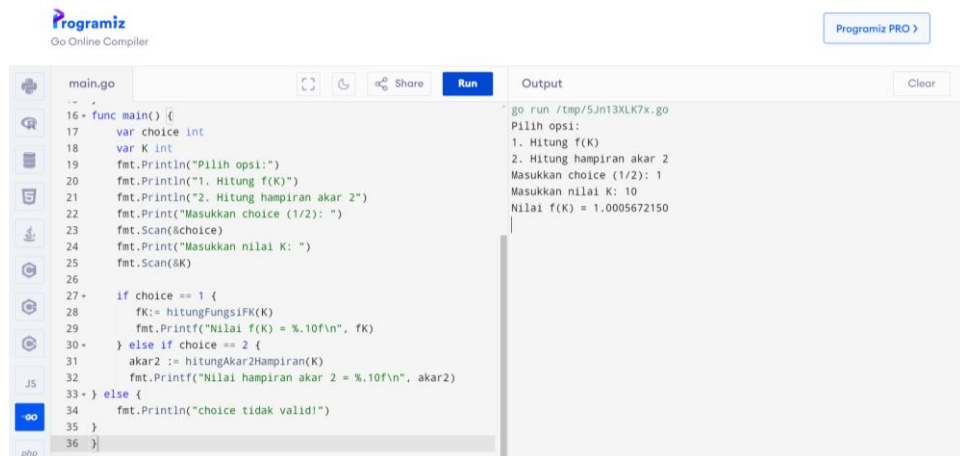
Unguided 4 (2B no.4)

SourchCode

```
package main
import "fmt"
//Damara Galuh Pembayun _2311102110
func hitungFungsiFK(K int) float64 {
    return float64((4*K+2)*(4*K+2)) /
float64((4*K+1)*(4*K+3))
}
func hitungAkar2Hampiran(K int) float64 {
    akar2 := 1.0
    for x := 0; x <= K; x++ {
        akar2 *= float64((4*x+2)*(4*x+2)) /
float64((4*x+1)*(4*x+3))
    }
    return akar2
}
func main() {
    var choice int
    var K int
    fmt.Println("Pilih opsi:")
    fmt.Println("1. Hitung f(K)")
    fmt.Println("2. Hitung hampiran akar 2")
    fmt.Print("Masukkan choice (1/2): ")
    fmt.Scan(&choice)
    fmt.Print("Masukkan nilai K: ")
    fmt.Scan(&K)

    if choice == 1 {
        fK:= hitungFungsiFK(K)
        fmt.Printf("Nilai f(K) = %.10f\n", fK)
    } else if choice == 2 {
        akar2 := hitungAkar2Hampiran(K)
        fmt.Printf("Nilai hampiran akar 2 = %.10f\n", akar2)
    } else {
        fmt.Println("choice tidak valid!")
    }
}
```

Hasil ScreenShoot



The screenshot shows the Programiz Go Online Compiler interface. On the left, there's a sidebar with icons for various programming languages. The main editor displays a Go program named 'main.go'. The program prompts the user to choose between calculating f(K) or the square root of 2. It then takes input for the choice and K, and outputs the result. The output window on the right shows the execution results for choice 1 and K=10.

```
16- func main() {
17-     var choice int
18-     var K int
19-     fmt.Println("Pilih opsi:")
20-     fmt.Println("1. Hitung f(K)")
21-     fmt.Println("2. Hitung hampiran akar 2")
22-     fmt.Print("Masukkan choice (1/2): ")
23-     fmt.Scan(&choice)
24-     fmt.Print("Masukkan nilai K: ")
25-     fmt.Scan(&K)
26-
27-     if choice == 1 {
28-         fK := hitungFungsiFK(K)
29-         fmt.Printf("Nilai f(K) = %.10f\n", fK)
30-     } else if choice == 2 {
31-         akar2 := hitungAkar2Hampiran(K)
32-         fmt.Printf("Nilai hampiran akar 2 = %.10f\n", akar2)
33-     } else {
34-         fmt.Println("choice tidak valid!")
35-     }
36- }
```

Output:

```
go run ./tmp/5Jn13XLK7x.go
Pilih opsi:
1. Hitung f(K)
2. Hitung hampiran akar 2
Masukkan choice (1/2): 1
Masukkan nilai K: 10
Nilai f(K) = 1.0005672150
```

Deskripsi

Program ini memberi pengguna pilihan antara dua hal: menghitung $f(k)$ atau menghitung hampiran akar 2. Metode iteratif menghasilkan serangkaian perkiraan untuk menyelesaikan sebuah masalah.

Unguided 5 (2C no.1)

SourchCode

```
package main
//Damara Galuh Pembayun_2311102110

import "fmt"

func main() {
    var beratParsel int

    fmt.Print("Berat parsel (gram): ")
    fmt.Scanln(&beratParsel)

    // Konversi ke kilogram dan gram
    kg := beratParsel / 1000
    gram := beratParsel % 1000

    // Hitung biaya dasar
    biayaDasar := kg * 10000

    // Hitung biaya tambahan
    var biayaTambahan int
    if gram >= 500 && kg <= 10 {
        biayaTambahan = gram * 15
    } else if gram >= 500 {
        biayaTambahan = gram * 5
    }

    // Hitung total biaya
    totalBiaya := biayaDasar + biayaTambahan

    // Tampilkan hasil
```

```

        fmt.Printf("Detail berat: %d kg + %d gr\n", kg,
gram)
        fmt.Printf("Detail biaya: Rp %d + Rp %d\n",
biayaDasar, biayaTambahan)
        fmt.Printf("Total biaya: Rp %d\n", totalBiaya)
}

```

Hasil ScreenShoot

```

main.go
15
16 // Hitung biaya dasar
17 biayaDasar := kg * 10000
18
19 // Hitung biaya tambahan
20 var biayaTambahan int
21 if gram >= 500 && kg <= 10 {
22     biayaTambahan = gram * 15
23 } else if gram >= 500 {
24     biayaTambahan = gram * 5
25 }
26
27 // Hitung total biaya
28 totalBiaya := biayaDasar + biayaTambahan
29
30 // Tampilkan hasil
31 fmt.Printf("Detail berat: %d kg + %d gr\n", kg, gram)
32 fmt.Printf("Detail biaya: Rp %d + Rp %d\n", biayaDasar,
    biayaTambahan)
33 fmt.Printf("Total biaya: Rp %d\n", totalBiaya)
34 }

```

```

go run /tmp/T9gyx5JNg2.go
Berat parsel (gram): 8500
Detail berat: 8 kg + 500 gr
Detail biaya: Rp 80000 + Rp 7500
Total biaya: Rp 87500

```

Deskripsi

Program ini dimaksudkan untuk menghitung biaya pengiriman untuk setiap parsel berdasarkan beratnya. Biaya pengiriman terdiri dari dua komponen: biaya dasar berdasarkan berat dalam kilogram dan biaya tambahan berdasarkan berat dalam gram. Untuk memudahkan perhitungan, program juga mengkonversi dari gram ke kilogram.

Unguided 6 (2C no.2)

SourchCode

```

package modul
import "fmt"
//Damara Galuh Pembayun_2311102110

func main () {
    var nam float 64
    var nmk string
    fmt.Print ("Nilai akhir mata kuliah: ")
    fmt.Scanln (&nam)
    if nam > 80 {
        nam = "A"
    }
    if nam > 72.5 {
        nam = "AB"
    }
    if nam > 65 {
        nam = "B"
    }
    if nam > 57.5 {

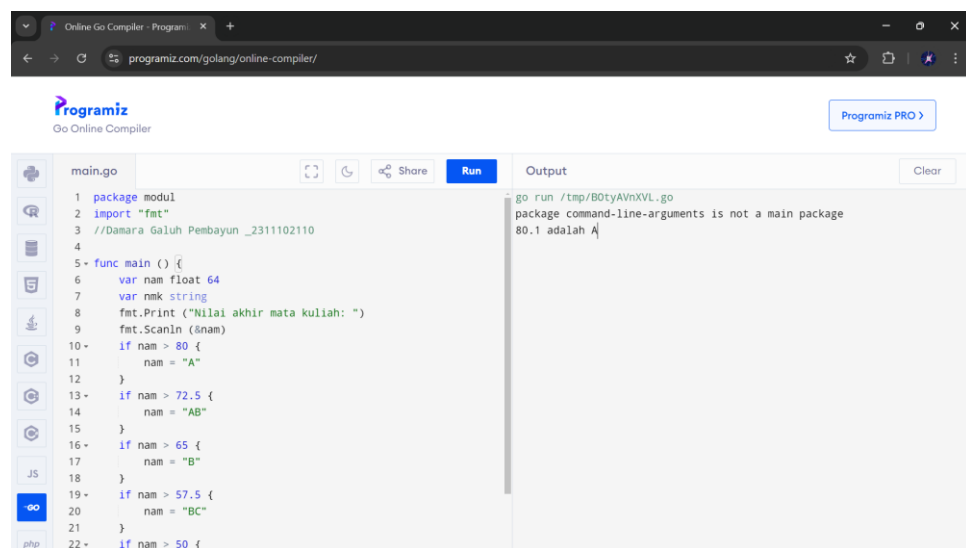
```

```

        nam = "BC"
    }
    if nam > 50 {
        nam = "C"
    }
    if nam > 40 {
        nam = "D"
    }else if nam <= 40 {
        nam = "E"
    }
    fmt.Println ("Nilai mata kuliah : " , nmk)
}

```

Hasil ScreenShoot



Deskripsi

Nilai mata kuliah yang muncul di output jika nam 80.1 adalah A. eksekusi program sesuai dengan spesifikasi soal. Variable nmk dideklarasikan tetapi tidak pernah digunakan. yang dimana variable ini seharusnya digunakan untuk menyimpan nilai akhir. Kemudian ada bug pada perkondisian if-else karena program akan mengecek semua kondisi. Namun, jika kita menggunakan else-if dan kondisi else untuk menggabungkan semua kondisi, program akan berjalan tanpa masalah.

Unguided 7 (2C no.3)

SourchCode

```

package main
//Damara Galuh Pembayun_2311102110

import "fmt"

func main() {

```

```

var bilangan int
var jumlahFaktor int = 0

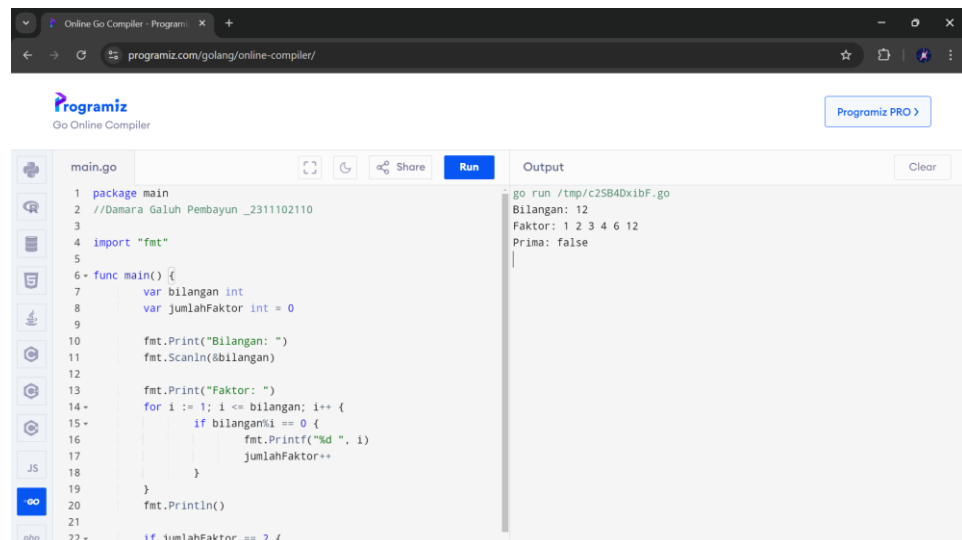
fmt.Print("Bilangan: ")
fmt.Scanln(&bilangan)

fmt.Print("Faktor: ")
for i := 1; i <= bilangan; i++ {
    if bilangan%i == 0 {
        fmt.Printf("%d ", i)
        jumlahFaktor++
    }
}
fmt.Println()

if jumlahFaktor == 2 {
    fmt.Println("Prima: true")
} else {
    fmt.Println("Prima: false")
}
}

```

Hasil ScreenShoot



Deskripsi

Program ini dimaksudkan untuk mengetahui apakah angka yang dimasukkan oleh pengguna adalah angka prima atau bukan. Bilangan prima adalah angka asli yang lebih besar dari 1, dengan faktor pembagi hanya 1 dan angka itu sendiri.