

**LAPORAN PRAKTIKUM
PEMROGRAMAN BERORIENTASI OBJEK**

MODUL X

MATERI



Oleh:

ANANDA BASKORO PUTRA

2311102187

IF 11 02

**S1 TEKNIK INFORMATIKA
INSTITUT TEKNOLOGI TELKOM PURWOKERTO**

2024

I. DASAR TEORI

2.1 Struktur Program Go

Dalam kerangka program yang ditulis dalam bahasa pemrograman Go, program utama selalu mempunyai dua komponen berikut:

`package main` merupakan penanda bahwa file ini berisi program utama `func main()` berisi kode utama dari sebuah program Go. Komentar, bukan bagian dari kode program dan dapat ditulis dimana saja di dalam program:

satu baris teks yang diawali dengan garis miring ganda (`//`) s.d. akhir baris, atau beberapa baris teks yang dimulai dengan pasangan karakter `'/'` dan di akhiri dengan `'/'`.

1. Koding, Kompilasi, dan Eksekusi Go

Koding

Tidak berbeda dengan penulisan program sumber dalam bahasa lain, program Go harus dibuat dengan menggunakan penyuntingan teks dan disimpan dalam format teks, bukan dalam format dokumen (doc, docx, atau lainnya). Setiap program go disimpan dalam file teks dengan ekstensi `*.go`, dengan nama bebas. Sebaiknya nama file adalah nama untuk program tersebut

Setiap satu program lengkap Go disimpan dalam satu folder tersendiri. Nama folder merupakan nama program tersebut. Karena itu secara prinsip, satu program Go dapat dipecah dalam beberapa file dengan ekstensi `*.go` selama disimpan dalam folder yang sama.

Kompilasi

Beberapa bahasa pemrograman dirancang untuk diimplementasikan sebagai interpreter dan lainnya sebagai kompilator. Interpreter akan membaca setiap baris instruksi dan kemudian langsung mengeksekusinya, dengan hanya sedikit pemeriksaan apakah penulisan keseluruhan program sudah benar atau belum. Kompilator akan memeriksa keseluruhan program sumber dan kemudian mengubahnya menjadi program eksekutabel, sehingga konsistensi penulisan (seperti penggunaan tipe data) sudah diperiksa sebelum dieksekusi. Selain itu karena program dibuat menjadi eksekutabel lebih dahulu, proses optimasi dapat dilakukan sehingga program menjadi sangat efisien. Go diimplementasikan sebagai kompilator. Berikut adalah contoh sesi yang biasa dilakukan saat mengkompilasi dan mengeksekusi program dalam bahasa Go:

2.2 Tipe Data dan Instruksi Dasar

1. Data dan Variabel

Variabel adalah nama dari suatu lokasi di memori, yang data dengan tipe tertentu dapat disimpan. Nama variabel dimulai dengan huruf dan dapat diikuti dengan sejumlah huruf, angka, atau garis bawah. Tipe data yang umum tersedia adalah integer, real, boolean, karakter, dan string. Lihat tabel berikut ini untuk variasi tipe data yang disediakan dalam bahasa Go. Nilai data yang tersimpan dalam variabel dapat diperoleh dengan menyebutkan langsung nama variabelnya contoh: menyebutkan nama `found` akan mengambil nilai tersimpan dalam memori untuk variabel `found`, pastinya.

Informasi alamat atau lokasi dari variabel dapat diperoleh dengan menambahkan prefiks `&` di depan nama variabel tersebut contoh: `&found` akan mendapatkan alamat memori untuk menyimpan data pada `found`. Jika variabel berisi alamat memori, prefiks `*` pada variabel tersebut akan memberikan nilai yang tersimpan dalam memori yang lokasinya disimpan dalam variabel tersebut contoh: `*mem` akan mendapatkan data di memori yang alamatnya tersimpan di `mem`, karenanya `*(&found)` akan mendapatkan data dari lokasi memori variabel `found` berada, alias sama saja dengan menyebutkan langsung `found` 8=)

Operasi yang dapat dilakukan terhadap tipe data diatas adalah Bahasa Go menganut kesesuaian tipe data yang ketat. Tipe data yang berbeda tidak boleh dicampur dalam satu ekspresi, bahkan tipe data masih yang sejenis, misalnya masih sama-sama integer (`int` dan `int32`). Untuk menyesuaikan tipe data, ada beberapa cara yang dapat dilakukan.

Casting, tipe (data), mengubah tipe dari data yang diberikan ke tipe yang diinginkan. Memanfaatkan fungsi `Sprint` dan `Sscan` dari paket `fmt`. Memanfaatkan fungsi-fungsi dalam paket `strconv`, seperti `Atol`, `Itoa`, dan `ParseBool`. Lihat lampiran berikut untuk contoh penggunaan.

Variabel harus dideklarasikan dulu sebelum digunakan. Variabel juga harus diinisialisasi dulu (diisi data) agar nilai yang tersimpan diketahui dengan jelas dan eksekusi algoritma menjadi terprediksi. Dalam bahasa Go, variabel yang tidak diinisialisasi lebih dulu otomatis diisi dengan nilai default yang ekuivalen dengan bit 0.

2.3 Struktur Kontrol Perulangan

Go hanya mempunyai kata kunci `for` untuk semua jenis perulangan yang kita pelajari dalam kondisi notasi algoritma. Dua bentuk yang kita gunakan disini adalah struktur `while-loop` dan `repeat-until`. Screenshot 2024-10-06 180629 Dalam konsep pemrograman terstruktur, setiap rancangan algoritma harus memenuhi syarat satu pintu masuk dan satu pintu keluar. Karena itu tidaklah diperkenankan untuk membuat program sumber yang mempunyai struktur loop yang mempunyai pintu keluar lebih dari satu.:

I. GUIDED

Source Code + Screenshot hasil program beserta penjelasan

Guided 1

```
package main

import "fmt"

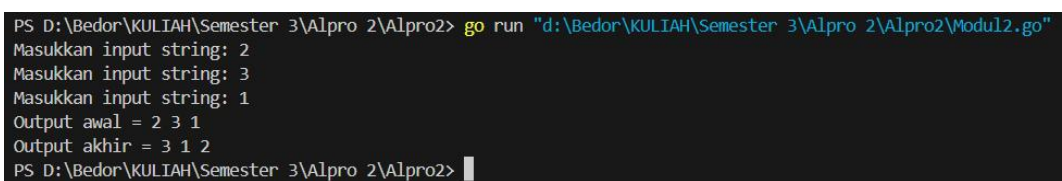
func main() {
    var satu, dua, tiga string
    var temp string

    fmt.Print("Masukkan input string: ")
    fmt.Scanln(&satu)
    fmt.Print("Masukkan input string: ")
    fmt.Scanln(&dua)
    fmt.Print("Masukkan 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 Output



```
PS D:\Bedor\KULIAH\Semester 3\Alpro 2\Alpro2> go run "d:\Bedor\KULIAH\Semester 3\Alpro 2\Alpro2\Modul2.go"
Masukkan input string: 2
Masukkan input string: 3
Masukkan input string: 1
Output awal = 2 3 1
Output akhir = 3 1 2
PS D:\Bedor\KULIAH\Semester 3\Alpro 2\Alpro2> |
```

Deskripsi Program

Program ini adalah program sederhana dalam bahasa Go yang meminta pengguna untuk memasukkan tiga string secara berurutan. Setelah menerima input, program mencetak output awal yang menunjukkan ketiga string yang dimasukkan. Selanjutnya, program melakukan pertukaran nilai antara variabel string, di mana nilai dari satu dipindahkan ke dua, nilai dua dipindahkan ke tiga, dan nilai tiga dipindahkan ke temp. Setelah proses pertukaran, program mencetak output akhir yang menunjukkan urutan string yang baru setelah pertukaran.

Guided 2

```
package main

import "fmt"

func main() {

    var tahun int

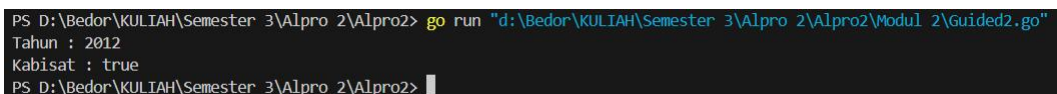
    fmt.Print("Tahun : ")

    fmt.Scan(&tahun)

    fmt.Printf("Kabisat : %t \n", (tahun%4 == 0 && (tahun%100 != 0 || tahun%400
== 0)))

}
```

Screenshoot Output



```
PS D:\Bedor\KULIAH\Semester 3\Alpro 2\Alpro2> go run "d:\Bedor\KULIAH\Semester 3\Alpro 2\Alpro2\Modul 2\Guided2.go"
Tahun : 2012
Kabisat : true
PS D:\Bedor\KULIAH\Semester 3\Alpro 2\Alpro2>
```

Deskripsi Program

Program digunakan untuk menentukan apakah suatu tahun adalah tahun kabisat atau tidak. Setelah meminta pengguna untuk memasukkan tahun, program akan melakukan perhitungan berdasarkan aturan tahun kabisat: tahun dianggap kabisat jika dapat dibagi 4, tetapi tidak boleh dibagi 100 kecuali juga dapat dibagi 400. Hasilnya dicetak dalam format boolean (true atau false), menunjukkan apakah tahun yang dimasukkan adalah tahun kabisat.

Guided 3

```
package main

import (
    "fmt"
    "math"
)

func main() {
    var jari, volume, luasKulit float64

    fmt.Print("Jejari = ")

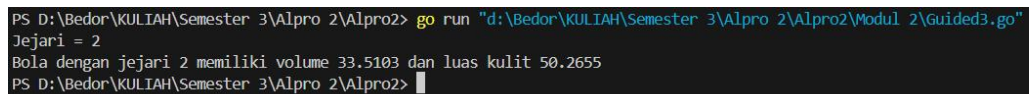
    fmt.Scan(&jari)

    volume = math.Pi * (4.0 / 3.0) * math.Pow(jari, 3)

    luasKulit = 4 * math.Pi * math.Pow(jari, 2)

    fmt.Printf("Bola dengan jejari %v memiliki volume %.4f dan luas kulit %.4f \n",
        jari, volume, luasKulit)
}
```

Screenshoot Output



```
PS D:\Bedor\KULIAH\Semester 3\Alpro 2\Alpro2> go run "d:\Bedor\KULIAH\Semester 3\Alpro 2\Alpro2\Modul 2\Guided3.go"
Jejari = 2
Bola dengan jejari 2 memiliki volume 33.5103 dan luas kulit 50.2655
PS D:\Bedor\KULIAH\Semester 3\Alpro 2\Alpro2> █
```

Deskripsi Program

Program ini berfungsi untuk menghitung volume dan luas kulit bola berdasarkan jejari yang dimasukkan oleh pengguna. Setelah meminta pengguna untuk memasukkan nilai jejari. Hasil perhitungan volume dan luas kulit kemudian dicetak dengan format yang menunjukkan dua angka desimal, memberikan informasi tentang ukuran bola yang sesuai dengan jejari yang dimasukkan.

II. UNGUIDED

Source Code + Screenshot hasil program beserta penjelasan

Unguided 2b_1

```
package main

import (
    "fmt"
    "strings"
)

func main() {
    var ujiKimia = [5][4]string{}

    var warna = [4]string{"merah", "kuning", "hijau", "ungu"}

    var beda bool = true

    for i := 0; i < 5; i++ {
        fmt.Print("Percobaan ", i+1, ": ")

        fmt.Scan(&ujiKimia[i][0], &ujiKimia[i][1], &ujiKimia[i][2], &ujiKimia[i][3])
    }

    for i := 0; i < 5; i++ {
        for j := 0; j < 4; j++ {
            beda = strings.ToLower(ujiKimia[i][j]) == warna[j]

            if !beda {
                i = 5
                break
            }
        }
    }
}
```



```

    }

    fmt.Println("Berhasil : ", beda)
}

```

Screenshoot Output

```

PS D:\Bedor\KULIAH\Semester 3\Alpro 2\Alpro2> go run "d:\Bedor\KULIAH\Semester 3\Alpro 2\Alpro2\Modul 2\Unguided2b_1.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:\Bedor\KULIAH\Semester 3\Alpro 2\Alpro2>

```

```

PS D:\Bedor\KULIAH\Semester 3\Alpro 2\Alpro2> go run "d:\Bedor\KULIAH\Semester 3\Alpro 2\Alpro2\Modul 2\Unguided2b_1.go"
Percobaan 1: merah kuning hijau ungu
Percobaan 2: merah kuning hijau ungu
Percobaan 3: ungu merah hijau kuning
Percobaan 4: merah kuning hijau ungu
Percobaan 5: merah kuning hijau ungu
Berhasil : false
PS D:\Bedor\KULIAH\Semester 3\Alpro 2\Alpro2>

```

Deskripsi Program

Program ini bertujuan untuk memeriksa kesesuaian hasil percobaan kimia dengan warna yang diharapkan. Pengguna diminta untuk memasukkan warna dari lima percobaan, masing-masing terdiri dari empat warna. Program kemudian membandingkan setiap warna yang dimasukkan dengan daftar warna yang ditetapkan (merah, kuning, hijau, ungu). Jika semua warna dalam satu percobaan sesuai dengan yang diharapkan, variabel beda tetap true; jika ada ketidakcocokan, beda di-set menjadi false, dan pemeriksaan dihentikan. Hasil akhir menunjukkan apakah semua percobaan berhasil atau tidak. Namun, terdapat beberapa kekurangan dalam logika program yang bisa mempengaruhi hasilnya.

Unguided 2b_2

```

package main

import "fmt"

func main() {

    var pita string

    var banyakBunga int

```

```

    for {

        var bunga string

        fmt.Print("Bunga (ketik 'SELESAI' untuk berhenti): ")

        fmt.Scanln(&bunga)

        if bunga == "SELESAI" {

            break

        }

        if pita != "" {

            pita += " - "

        }

        pita += bunga

        banyakBunga++

    }

    fmt.Println("Pita:", pita)

    fmt.Println("Banyaknya bunga:", banyakBunga)

}

```

Screenshoot Output

```

PS D:\Bedor\KULIAH\Semester 3\Alpro 2\Alpro2> go run "d:\Bedor\KULIAH\Semester 3\Alpro 2\Alpro2\Modul 2\Unguided2b_2.go"
Bunga (ketik 'SELESAI' untuk berhenti): kertas
Bunga (ketik 'SELESAI' untuk berhenti): mawar
Bunga (ketik 'SELESAI' untuk berhenti): tulip
Bunga (ketik 'SELESAI' untuk berhenti): SELESAI
Pita: kertas - mawar - tulip
Banyaknya bunga: 3

```

Deskripsi Program

Program yang meminta pengguna untuk memasukkan nama bunga secara berulang, kemudian menyimpan dan menampilkan daftar bunga tersebut dalam bentuk pita yang dipisahkan oleh tanda "-". Program akan terus meminta input bunga sampai pengguna mengetikkan "SELESAI", yang akan menghentikan perulangan. Setelah perulangan berakhir, program menampilkan daftar bunga yang telah diinput beserta jumlah total bunga yang dimasukkan oleh pengguna.

Unguided 2b_3

```
package main

import "fmt"

func main() {
    for {
        var berat1, berat2 float64

        fmt.Print("Masukkan berat belanjaan di kedua kantong: ")

        _, err := fmt.Scan(&berat1, &berat2)

        if err != nil {
            fmt.Println("Input tidak valid.")
            return
        }

        if berat1+berat2 > 150 || berat1 < 0 || berat2 < 0 {
            fmt.Println("Proses selesai.")
            break
        }

        selisih := berat1 - berat2

        if selisih < 0 {
            selisih = -selisih
        }
    }
}
```

```

    }

    akanOlang := selisih >= 9

    fmt.Printf("Sepeda motor pak Andi akan oleng: %t\n", akanOlang)

}

}

```

Screenshoot Output

```

PS D:\Bedor\KULIAH\Semester 3\Alpro 2\Alpro2> go run "d:\Bedor\KULIAH\Semester 3\Alpro 2\Alpro2\Modul 2\Unguided2b_3.go"
Masukkan berat belanjaan di kedua kantong: 5 10
Sepeda motor pak Andi akan oleng: false
Masukkan berat belanjaan di kedua kantong: 55.6 70.2
Sepeda motor pak Andi akan oleng: true
Masukkan berat belanjaan di kedua kantong: 72.3
66.9
Sepeda motor pak Andi akan oleng: false
Masukkan berat belanjaan di kedua kantong: 59.5 98.7
Proses selesai.
PS D:\Bedor\KULIAH\Semester 3\Alpro 2\Alpro2>

```

Deskripsi Program

Program tersebut meminta pengguna memasukkan berat belanjaan di dua kantong secara berulang. Jika total berat kedua kantong melebihi 150 kg atau salah satu berat bernilai negatif, program berhenti dengan pesan "Proses selesai". Program kemudian menghitung selisih berat antara dua kantong dan menentukan apakah sepeda motor Pak Andi akan oleng berdasarkan apakah selisih tersebut lebih besar atau sama dengan 9 kg. Hasilnya ditampilkan dalam bentuk nilai boolean (true atau false).

Unguided 2b_4

```

package main

import (
    "fmt"
    "math"
)

func main() {
    var k, hasil float64 = 0, 1
    fmt.Print("Nilai K = ")
}

```

```

    fmt.Scan(&k)

    for i := 0; i < int(k); i++ {

        hasil *= math.Pow(4*float64(i)+2, 2) / ((4*float64(i) + 1) *
(4*float64(i) + 3))

    }

    fmt.Printf("Nilai akar 2 = %.10f", hasil)

}

```

Screenshoot Output

```

PS D:\Bedor\KULIAH\Semester 3\Alpro 2\Alpro2> go run "d:\Bedor\KULIAH\Semester 3\Alpro 2\Alpro2\Modul 2\Unguided2b_4.go"
Nilai K = 10
012345678910Nilai Akar 2 = 1.4062058441
PS D:\Bedor\KULIAH\Semester 3\Alpro 2\Alpro2>

```

```

PS D:\Bedor\KULIAH\Semester 3\Alpro 2\Alpro2> go run "d:\Bedor\KULIAH\Semester 3\Alpro 2\Alpro2\Modul 2\Unguided2b_4.go"
Nilai K = 100
Nilai akar 2 = 1.4133299615
PS D:\Bedor\KULIAH\Semester 3\Alpro 2\Alpro2>

```

```

PS D:\Bedor\KULIAH\Semester 3\Alpro 2\Alpro2> go run "d:\Bedor\KULIAH\Semester 3\Alpro 2\Alpro2\Modul 2\Unguided2b_4.go"
Nilai K = 1000
Nilai akar 2 = 1.4141251768
PS D:\Bedor\KULIAH\Semester 3\Alpro 2\Alpro2>

```

Deskripsi Program

Program tersebut menghitung nilai perkiraan akar kuadrat dari 2 menggunakan sebuah deret matematika. Pengguna diminta memasukkan nilai k, yang menentukan jumlah iterasi dalam perhitungan deret tersebut. Di dalam loop, pada setiap iterasi, program menghitung elemen baru dari deret dengan rumus tertentu yang melibatkan operasi pangkat dan perkalian. Hasil akhirnya adalah nilai perkiraan akar 2 yang ditampilkan dengan presisi hingga 10 angka desimal.

Unguided 2c_1

```
package main

import (
    "fmt"
)

func main() {
    var berat int

    fmt.Print("Masukkan berat parsel (gram): ")

    fmt.Scanf("%d", &berat)

    kg := berat / 1000

    grams := berat % 1000

    original := kg * 10000

    tambahan := 0

    if kg >= 10 {
        tambahan = 0
    } else {
        if grams < 500 {
            tambahan = grams * 15
        } else {
            tambahan = grams * 5
        }
    }

    total := original + tambahan

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

    fmt.Printf("Detail biaya: Rp. %d + Rp. %d\n", original, tambahan)
```

```
        fmt.Printf("Total biaya: Rp. %d\n", total)
    }
}
```

Screenshoot Output

```
PS D:\Bedor\KULIAH\Semester 3\Alpro 2\Alpro2> go run "d:\Bedor\KULIAH\Semester 3\Alpro 2\Alpro2\Modul 2\Unguided2c_1.go"
Masukkan berat parcel (gram): 8500
Detail berat: 8 kg + 500 gr
Detail biaya: Rp. 80000 + Rp. 2500
Total biaya: Rp. 82500
PS D:\Bedor\KULIAH\Semester 3\Alpro 2\Alpro2> █
```

```
PS D:\Bedor\KULIAH\Semester 3\Alpro 2\Alpro2> go run "d:\Bedor\KULIAH\Semester 3\Alpro 2\Alpro2\Modul 2\Unguided2c_1.go"
Masukkan berat parcel (gram): 9250
Detail berat: 9 kg + 250 gr
Detail biaya: Rp. 90000 + Rp. 3750
Total biaya: Rp. 93750
PS D:\Bedor\KULIAH\Semester 3\Alpro 2\Alpro2> █
```

```
PS D:\Bedor\KULIAH\Semester 3\Alpro 2\Alpro2> go run "d:\Bedor\KULIAH\Semester 3\Alpro 2\Alpro2\Modul 2\Unguided2c_1.go"
Masukkan berat parcel (gram): 11750
Detail berat: 11 kg + 750 gr
Detail biaya: Rp. 110000 + Rp. 0
Total biaya: Rp. 110000
PS D:\Bedor\KULIAH\Semester 3\Alpro 2\Alpro2> █
```

Deskripsi Program

Program tersebut menghitung total biaya pengiriman berdasarkan berat parcel yang dimasukkan pengguna dalam gram. Pertama, berat parcel dipecah menjadi kilogram (kg) dan gram (gr). Biaya dasar dihitung sebagai Rp. 10.000 per kilogram. Jika total berat kurang dari 10 kg, biaya tambahan dihitung berdasarkan sisa gramnya: jika sisa gram kurang dari 500, biaya tambahan adalah Rp. 15 per gram, sedangkan jika lebih dari atau sama dengan 500 gram, biaya tambahan adalah Rp. 5 per gram. Program kemudian menampilkan rincian berat parcel, rincian biaya, dan total biaya pengiriman.

Unguided 2c_2

```
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 Output

```

PS D:\Bedor\KULIAH\Semester 3\Alpro 2\Alpro2> go run "d:\Bedor\KULIAH\Semester 3\Alpro 2\Alpro2\Modul 2\Unguided2c_2.go"
Nilai akhir mata kuliah: 93.5
Nilai mata kuliah: A
PS D:\Bedor\KULIAH\Semester 3\Alpro 2\Alpro2>

```

```

PS D:\Bedor\KULIAH\Semester 3\Alpro 2\Alpro2> go run "d:\Bedor\KULIAH\Semester 3\Alpro 2\Alpro2\Modul 2\Unguided2c_2.go"
Nilai akhir mata kuliah: 70.6
Nilai mata kuliah: B
PS D:\Bedor\KULIAH\Semester 3\Alpro 2\Alpro2>

```

```

PS D:\Bedor\KULIAH\Semester 3\Alpro 2\Alpro2> go run "d:\Bedor\KULIAH\Semester 3\Alpro 2\Alpro2\Modul 2\Unguided2c_2.go"
Nilai akhir mata kuliah: 49.5
Nilai mata kuliah: D
PS D:\Bedor\KULIAH\Semester 3\Alpro 2\Alpro2>

```


Deskripsi Program

Program menentukan nilai mata kuliah berdasarkan nilai akhir yang dimasukkan pengguna. Pengguna diminta untuk memasukkan nilai akhir (dalam bentuk angka desimal). Program kemudian menggunakan struktur kondisi if-else untuk memetakan nilai akhir tersebut ke nilai huruf yang sesuai, dengan rentang nilai mulai dari A hingga E. Setelah penentuan nilai huruf, program menampilkan nilai mata kuliah yang sesuai berdasarkan nilai akhir tersebut.

Unguided 2c_3

```
package main

import "fmt"

func main() {

    var bil int

    var prima bool = true

    fmt.Print("Bilangan: ")

    fmt.Scan(&bil)

    fmt.Print("Faktor: ")

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

        if bil%i == 0 {

            fmt.Print(i, " ")

            if !(i == 1 || i == bil) {

                prima = false

            }

        }

    }

    fmt.Println()

    fmt.Print("Prima: ", prima)

}
```

Screenshoot Output

```
PS D:\Bedor\KULIAH\Semester 3\Alpro 2\Alpro2> go run "d:\Bedor\KULIAH\Semester 3\Alpro 2\Alpro2\Modul 2\Unguided2c_3.go"
Bilangan: 12
Faktor: 1 2 3 4 6 12
Prima: false
PS D:\Bedor\KULIAH\Semester 3\Alpro 2\Alpro2> |
```

```
PS D:\Bedor\KULIAH\Semester 3\Alpro 2\Alpro2> go run "d:\Bedor\KULIAH\Semester 3\Alpro 2\Alpro2\Modul 2\Unguided2c_3.go"
Bilangan: 7
Faktor: 1 7
Prima: true
PS D:\Bedor\KULIAH\Semester 3\Alpro 2\Alpro2> |
```

Deskripsi Program

Program tersebut berfungsi untuk memeriksa apakah sebuah bilangan yang dimasukkan oleh pengguna adalah bilangan prima dan menampilkan faktor-faktornya. Pengguna diminta untuk memasukkan sebuah bilangan bulat, lalu program mencetak semua faktor dari bilangan tersebut. Jika bilangan hanya memiliki dua faktor, yaitu 1 dan bilangan itu sendiri, maka program menganggapnya sebagai bilangan prima dan mencetak true. Jika ada faktor lain di luar 1 dan bilangan itu sendiri, program mencetak false, menandakan bahwa bilangan tersebut bukan prima.