

**LAPORAN PRAKTIKUM
PEMROGRAMAN BERORIENTASI OBJEK**

MODUL X

MATERI



Oleh:

NAMA : HAIKAL SATRIATAMA

NIM : 2311102066

KELAS : IF 11 02

**S1 TEKNIK INFORMATIKA
INSTITUT TEKNOLOGI TELKOM PURWOKERTO**

2024

I. DASAR TEORI

Dasar Teori

Struktur program Go terdiri dari beberapa komponen utama, termasuk deklarasi paket, impor, fungsi utama, variabel, struktur kontrol, dan fungsi tambahan. Program Go sangat terstruktur dan mendukung penulisan kode yang bersih dan terorganisir, membuatnya mudah untuk dibaca dan dikelola.

Dalam program utama di golang, selalu terdapat dua komponen

- package main merupakan bahwa ini merupakan program utama
- func main merupakan fungsi yang akan dieksekusi Ketika program berjalan.

- Satu baris teks yang diawali dengan garis miring ganda ('//') s.d akhir baris atau
- Beberapa baris teks yang dimulai dengan pasangan karakter '/*' dan diakhiri dengan '*/'

1). Koding, kompilasi dan eksekusi go

- Koding
Tidak berbeda dengan penulisan program sumber dalam bahasa lain, program go harus dibuat menggunakan penyunting teks dan disimpan dalam format teks, bukan dalam format dokumen (doc, docx, atau lainnya)
- Setiap program go harus disimpan dalam file teks dengan ekstensi *.go, dengan nama bebas sebaiknya nama file adalah nama untuk program tersebut.
- Setiap satu program lengkap go harus 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.

2). Kompilasi

Beberapa bahasa pemrograman dirancang untuk diimplementasikan sebagai interpreter dan lainnya sebagai kompilator. Interpreter akan membaca setiap baris intruksi 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 eksekusi. Selain itu karena program dibuat menjadi eksekutabel lebih dahulu, proses optimasi dapat dilakukan sehingga program menjadi sangat efisien.

Catatan :

Semua proses terkait bahasa go dilakukan melalui utilitas go. Beberapa opsi dengan utilitas go :

- Go build : mengkompilasi program sumber yang ada dalam folder menjadi sebuah program
- Go build file.go : mengkompilasi program sumber file.go saja.
- Go fmt : membaca semua program sumber dalam folder dan mereformat penulisannya agar sesuai dengan standar penulisan program sumber go.
- Go clean : membersihkan file-file dalam folder sehingga tersisa program sumber nya saja.

2). Latihan

- Selidiki bahasa-bahasa pemrograman berikut, apakah termasuk diinterpretasi, dikompilasi, dikompilasi (ke intruksi perantara) kemudian diinterpretasi :
 - Pasca
 - C dan C++
 - Java
 - Python
- Unduh kompilator go di komputer yang anda gunakan. Kemudian salin contoh program atas ke dalam folder C:\Users\userid\go\hello\hello.go, yaitu buat folder go dalam direktori home anda, kemudian buat subfolder hello dan taruh file hello.go di dalamnya. Hidupkan terminal (cmd.exe) dan panggil go build di dalam folder hello tersebut. Periksa apakah hello.exe muncul di folder tersebut? Jika yaa, coba eksekusi program tersebut, juga melalui terminal cmd.exe tersebut.(jangan di klik melalui browser folder).
-

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.
Contoh: *ketemu, found, rerata, mhs1, data_2...*
- Tipe data yang umum tersedia adalah integer, real, boolean, karakter, dan string.
- Nilai data yang tersimpan dalam variabel dapat diperoleh dengan menyebutkan langsung nama variabelnya.
- Informasi alamat atau lokasi dari variabel dapat diperoleh dengan menambahkan prefiks & di depan nama variabel tersebut.
- Jika variabel berisi alamat memori, prefiks * pada variabel tersebut akan memberikan nilai yang tersimpan dalam memori yang lokasinya disimpan dalam variabel tersebut.
- 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 scan dari paket fmt.
 - Memanfaatkan fungsi-fungsi dalam paket strconv, seperti Atol, Itoa, dan ParseBool.
 - 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) Instruksi Dasar

3) Konstanta Simbolik

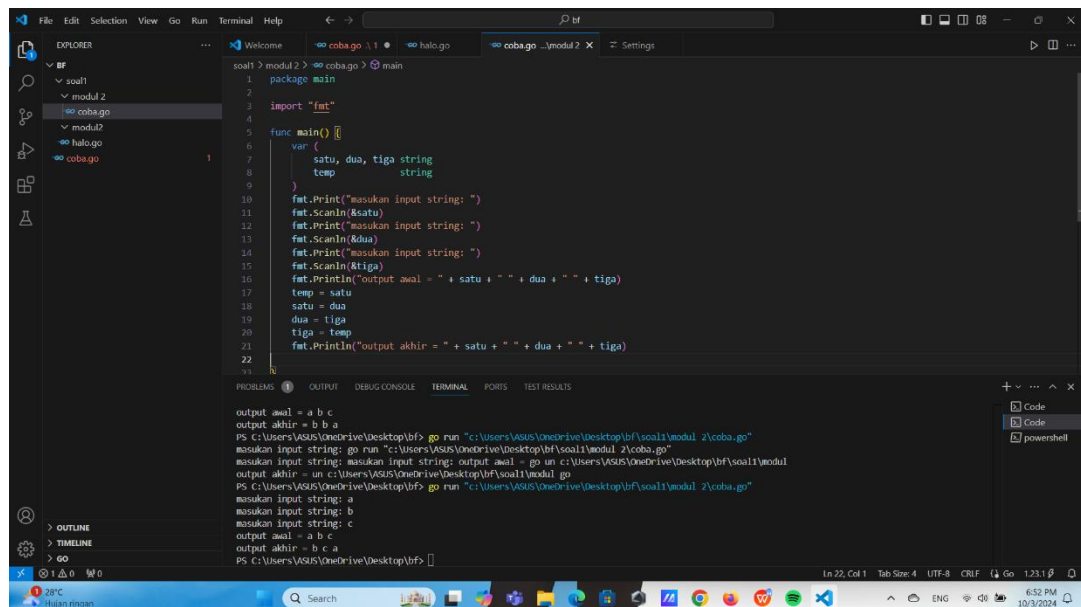
Konstanta dapat diberi nama untuk memudahkan mengingat maksud dan manfaat dari nilai yang diberi nama tersebut, Seperti PI untuk mempresentasikan konstanta.

II. GUIDED

Source Code + Screenshot hasil program beserta penjelasan

MODUL 2A

NO 1.



The screenshot shows a Go IDE with the following source code in `soal1/modul2/coba.go`:

```
1 package main
2
3 import "fmt"
4
5 func main() {
6     var (
7         satu, dua, tiga string
8         temp string
9     )
10    fmt.Println("masukan input string: ")
11    fmt.Scanln(&satu)
12    fmt.Println("masukan input string: ")
13    fmt.Scanln(&dua)
14    fmt.Println("masukan input string: ")
15    fmt.Scanln(&tiga)
16    fmt.Println("output awal = " + satu + " " + dua + " " + tiga)
17    temp = satu
18    satu = dua
19    dua = tiga
20    tiga = temp
21    fmt.Println("output akhir = " + satu + " " + dua + " " + tiga)
22 }
```

The terminal output shows the execution of the program with the following input and output:

```
output awal = a b c
output akhir = b b a
PS C:\Users\VASUS\OneDrive\Desktop\bf> go run "c:\Users\VASUS\OneDrive\Desktop\bf\soal1\modul 2\coba.go"
masukan input string: go run "c:\Users\VASUS\OneDrive\Desktop\bf\soal1\modul 2\coba.go"
masukan input string: masukan input string: output awal = go un c:\Users\VASUS\OneDrive\Desktop\bf\soal1\modul
output akhir = un c:\Users\VASUS\OneDrive\Desktop\bf\soal1\modul go
PS C:\Users\VASUS\OneDrive\Desktop\bf> go run "c:\Users\VASUS\OneDrive\Desktop\bf\soal1\modul 2\coba.go"
masukan input string: a
masukan input string: b
masukan input string: c
output awal = a b c
output akhir = b c a
PS C:\Users\VASUS\OneDrive\Desktop\bf> []
```

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)

}

```

Penjelasan :

Program ini melakukan tugas sederhana yaitu membaca tiga input string dari pengguna, menampilkan urutan awalnya, lalu menukar urutannya secara melingkar dan menampilkannya kembali.

NO 2.

```
1 package main
2
3 import "fmt"
4
5 func main() {
6     var tahun int
7     fmt.Print("Tahun : ")
8     fmt.Scan(&tahun)
9     if tahun%4 == 0 && tahun%100 != 0 {
10         fmt.Print("Kabisat : True")
11     } else {
12         fmt.Print("Kabisat : False")
13     }
14 }
15
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS TEST RESULTS

```
PS C:\Users\ASUS\OneDrive\Desktop\bf> go run "c:\Users\ASUS\OneDrive\Desktop\bf\soal1\modul 2\coba.go"
masukan input string: a
masukan input string: b
masukan input string: c
output awal = a b c
output akhir = b c a
PS C:\Users\ASUS\OneDrive\Desktop\bf> go run "c:\Users\ASUS\OneDrive\Desktop\bf\soal2\soal2.go"
Tahun : 2014
Kabisat : False
PS C:\Users\ASUS\OneDrive\Desktop\bf> go run "c:\Users\ASUS\OneDrive\Desktop\bf\soal2\soal2.go"
Tahun : 2016
Kabisat : True
PS C:\Users\ASUS\OneDrive\Desktop\bf>
```

Source Code :

package main

import "fmt"

func main() {

var tahun int

fmt.Print("Tahun : ")

fmt.Scan(&tahun)

if tahun%4 == 0 && tahun%100 != 0 {

fmt.Print("Kabisat : True")

} else {

fmt.Print("Kabisat : False")

}

```
}
```

Penjelasan :

++`+source code ini bertujuan untuk menentukan apakah sebuah tahun yang dimasukkan oleh pengguna merupakan tahun kabisat atau tidak, berdasarkan aturan sederhana yang memeriksa apakah tahun tersebut habis dibagi 4 dan tidak habis dibagi 100. Jika kedua kondisi terpenuhi, tahun tersebut dianggap sebagai tahun kabisat.

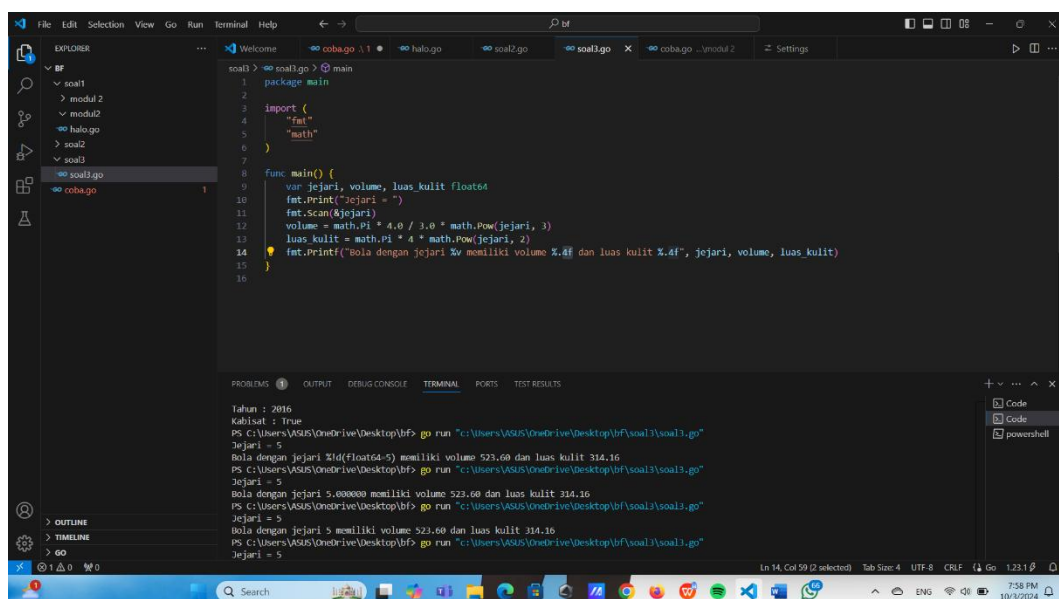
Namun, kode ini tidak memperhitungkan kasus khusus untuk tahun yang habis dibagi 400 (yang seharusnya tetap dianggap kabisat, meskipun habis dibagi 100). Untuk membuat logika lebih lengkap, perlu ditambahkan pengecekan tambahan apakah tahun habis dibagi 400.

Aturan Tahun Kabisat Lengkap:

1. Tahun yang habis dibagi 4 adalah tahun kabisat kecuali habis dibagi 100.
2. Tahun yang habis dibagi 100 bukan tahun kabisat kecuali habis dibagi 400.

Dengan logika lengkap tersebut, hasil program akan lebih akurat dalam menentukan apakah sebuah tahun kabisat atau bukan.

NO 3.



```
soal3.go | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
package main
import (
    "fmt"
    "math"
)
func main() {
    var jejari, volume, luas_kulit float64
    fmt.Print("Jejari = ")
    fmt.Scan(&jejari)
    volume = math.Pi * 4.0 / 3.0 * math.Pow(jejari, 3)
    luas_kulit = math.Pi * 4 * math.Pow(jejari, 2)
    fmt.Printf("Bola dengan jejari %v memiliki volume %.2f dan luas kulit %.2f", jejari, volume, luas_kulit)
}
```

Terminal Output:

```
Tahun : 2016
Kabisat : True
PS C:\Users\ASUS\OneDrive\Desktop> go run "c:\Users\ASUS\OneDrive\Desktop\brf\soal3\soal3.go"
Jejari = 5
Bola dengan jejari 5.000000 memiliki volume 523.60 dan luas kulit 314.16
PS C:\Users\ASUS\OneDrive\Desktop> go run "c:\Users\ASUS\OneDrive\Desktop\brf\soal3\soal3.go"
Jejari = 5
Bola dengan jejari 5.000000 memiliki volume 523.60 dan luas kulit 314.16
PS C:\Users\ASUS\OneDrive\Desktop> go run "c:\Users\ASUS\OneDrive\Desktop\brf\soal3\soal3.go"
Jejari = 5
Bola dengan jejari 5 memiliki volume 523.60 dan luas kulit 314.16
PS C:\Users\ASUS\OneDrive\Desktop> go run "c:\Users\ASUS\OneDrive\Desktop\brf\soal3\soal3.go"
Jejari = 5
```

Source Code :


```
package main
```

```
import (  
    "fmt"  
    "math"  
)
```

```
func main() {  
    var jejari, volume, luas_kulit float64  
  
    fmt.Print("Jejari = ")  
  
    fmt.Scan(&jejari)  
  
    volume = math.Pi * 4.0 / 3.0 * math.Pow(jejari, 3)  
  
    luas_kulit = math.Pi * 4 * math.Pow(jejari, 2)  
  
    fmt.Printf("Bola dengan jejari %v memiliki volume %.4f dan luas kulit %.4f",  
        jejari, volume, luas_kulit)  
}
```

Penjelasan :

Kesimpulannya, source code ini adalah sebuah program Go yang menerima input berupa jejari (radius) bola dari pengguna, kemudian menghitung dua hal utama menggunakan rumus matematika:

Volume bola menggunakan rumus : yang sudah di tentukan di modul

Luas permukaan bola menggunakan rumus : yang sudah di tentukan di modul

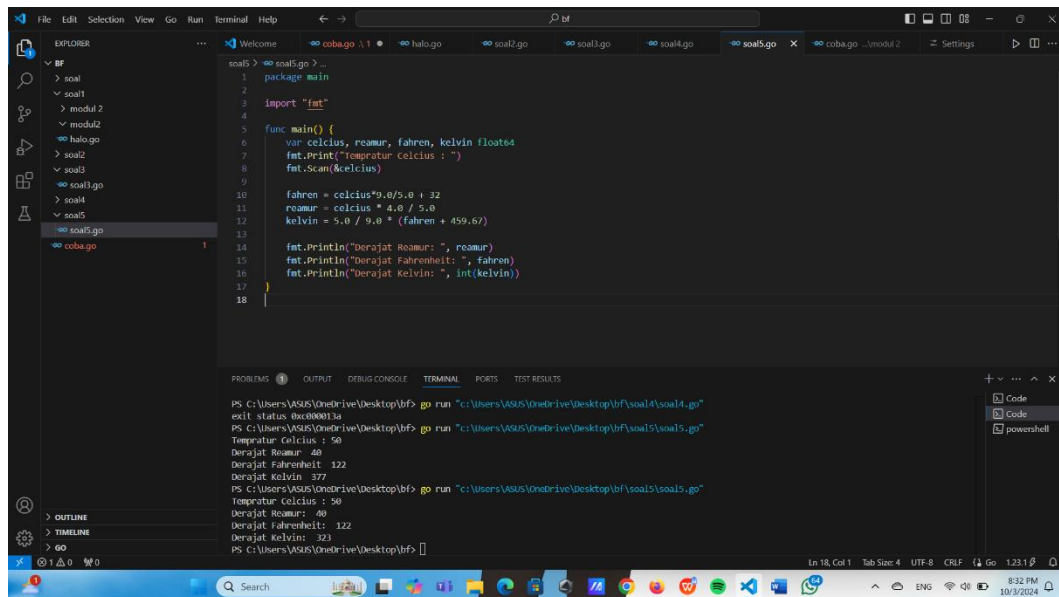
Program ini efektif untuk menghitung dan menampilkan volume serta luas permukaan sebuah bola berdasarkan nilai jejari yang dimasukkan oleh pengguna.

III. UNGUIDED

Source Code + Screenshot hasil program beserta penjelasan

Soal 2A

NO 4.



Source code :

```
package main
```

```
import "fmt"
```

```
func main() {
```

```
    var celcius, reamur, fahrenheit, kelvin float64
```

```
    fmt.Print("Tempratur Celcius : ")
```

```
    fmt.Scan(&celcius)
```

```
    fahrenheit = celcius*9.0/5.0 + 32
```

```
reamur = celcius * 4.0 / 5.0
```

```
kelvin = 5.0 / 9.0 * (fahren + 459.67)
```

```
fmt.Println("Derajat Reamur: ", reamur)
```

```
fmt.Println("Derajat Fahrenheit: ", fahren)
```

```
fmt.Println("Derajat Kelvin: ", int(kelvin))
```

```
}
```

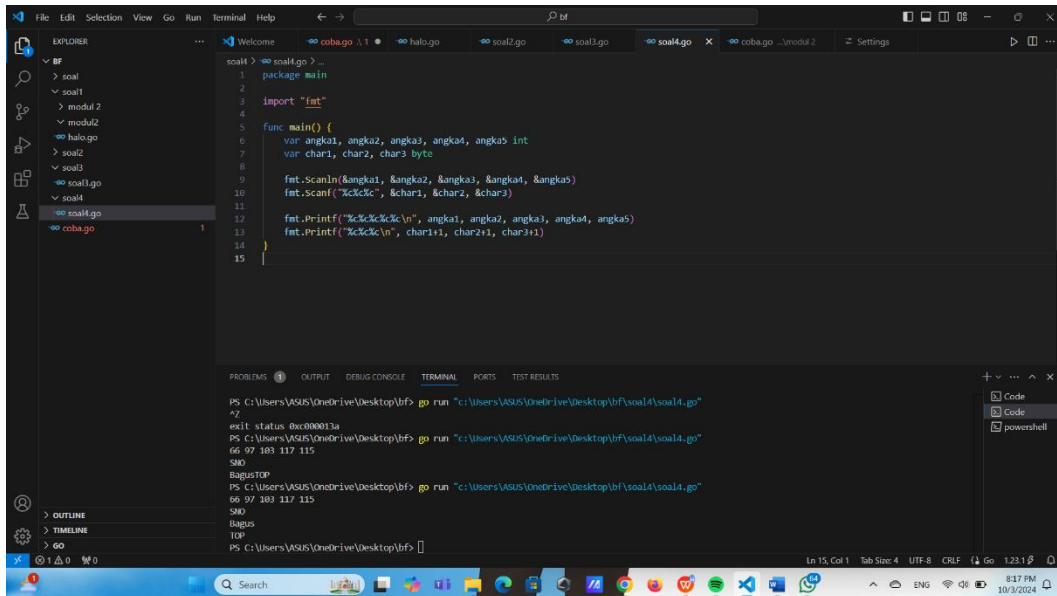
Penjelasan :

Kesimpulan:

- Program ini menerima input temperatur dalam skala Celsius dan mengkonversinya ke tiga skala temperatur lainnya: Reaumur, Fahrenheit, dan Kelvin.
- Perhitungan konversi dilakukan dengan menggunakan rumus matematika yang sesuai untuk masing-masing skala.
- Hasil konversi ditampilkan dalam format yang mudah dibaca, dengan Kelvin ditampilkan dalam bentuk integer.

Program ini membantu pengguna untuk dengan cepat mengonversi temperatur dari Celsius ke beberapa skala yang berbeda, yaitu Reaumur, Fahrenheit, dan Kelvin.

NO 5.



```
1 package main
2
3 import "fmt"
4
5 func main() {
6     var angka1, angka2, angka3, angka4, angka5 int
7     var char1, char2, char3 byte
8
9     fmt.Scanln(&angka1, &angka2, &angka3, &angka4, &angka5)
10    fmt.Scanf("%c%c%c", &char1, &char2, &char3)
11
12    fmt.Printf("%c%c%c%c%c\n", angka1, angka2, angka3, angka4, angka5)
13    fmt.Printf("%c%c%c\n", char1, char2, char3)
14 }
15
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS TEST RESULTS

```
PS C:\Users\ASUS\OneDrive\Desktop\hrf> go run "c:\Users\ASUS\OneDrive\Desktop\hrf\soal4\soal4.go"
^Z
exit status 0xc000013a
PS C:\Users\ASUS\OneDrive\Desktop\hrf> go run "c:\Users\ASUS\OneDrive\Desktop\hrf\soal4\soal4.go"
66 97 103 117 115
SHO
BagustOP
PS C:\Users\ASUS\OneDrive\Desktop\hrf> go run "c:\Users\ASUS\OneDrive\Desktop\hrf\soal4\soal4.go"
66 97 103 117 115
SHO
Bagus
TOP
PS C:\Users\ASUS\OneDrive\Desktop\hrf>
```

Source Code :

package main

import "fmt"

func main() {

var angka1, angka2, angka3, angka4, angka5 int

var char1, char2, char3 byte

fmt.Scanln(&angka1, &angka2, &angka3, &angka4, &angka5)

fmt.Scanf("%c%c%c", &char1, &char2, &char3)

fmt.Printf("%c%c%c%c%c\n", angka1, angka2, angka3, angka4, angka5)

```

        fmt.Printf("%c%c%c%c\n", char1+1, char2+1, char3+1)
    }
}

```

Penjelasan :

- Program ini meminta pengguna untuk memasukkan lima angka dan tiga karakter.
- Angka yang dimasukkan ditampilkan sebagai karakter sesuai dengan nilai ASCII mereka.
- Karakter yang dimasukkan digeser satu posisi dalam tabel ASCII dan kemudian ditampilkan.

Program ini menggunakan konversi nilai integer menjadi karakter ASCII dan pergeseran karakter berdasarkan nilai ASCII untuk menampilkan hasil yang menarik dari input pengguna.

B. MODUL 2B

NO 1.

```

1 package main
2
3 import "fmt"
4
5 func main() {
6     var bunga1, bunga2, bunga3, bunga4 string
7     var beda bool = false
8
9     for i := 1; i <= 5; i++ {
10        fmt.Printf("percobaan : %i, ", i)
11        fmt.Scan(&bunga1, &bunga2, &bunga3, &bunga4)
12        if (bunga1 != "merah" || bunga2 != "kuning" || bunga3 != "hijau" || bunga4 != "ungu") && !beda {
13            beda = true
14        }
15    }
16    fmt.Println("BERHASIL : ", !beda)
17 }
18

```

```

Prjase : true
PS C:\Users\ASUS\OneDrive\Desktop\lbf> run "c:\Users\ASUS\OneDrive\Desktop\lbf\modul2c\soal3 lanjutan.go"
Bilangan: 12
Faktor : 1 2 3 4 6 12
Prjase : false
PS C:\Users\ASUS\OneDrive\Desktop\lbf> run "c:\Users\ASUS\OneDrive\Desktop\lbf\modul2b\soal1.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 C:\Users\ASUS\OneDrive\Desktop\lbf>

```

Source code :

package main

```

import "fmt"

func main() {
    var bunga1, bunga2, bunga3, bunga4 string
    var beda bool = false

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

        fmt.Scan(&bunga1, &bunga2, &bunga3, &bunga4)

        if (bunga1 != "merah" || bunga2 != "kuning" || bunga3 != "hijau" ||
            bunga4 != "ungu") && !beda {
            beda = true
        }
    }

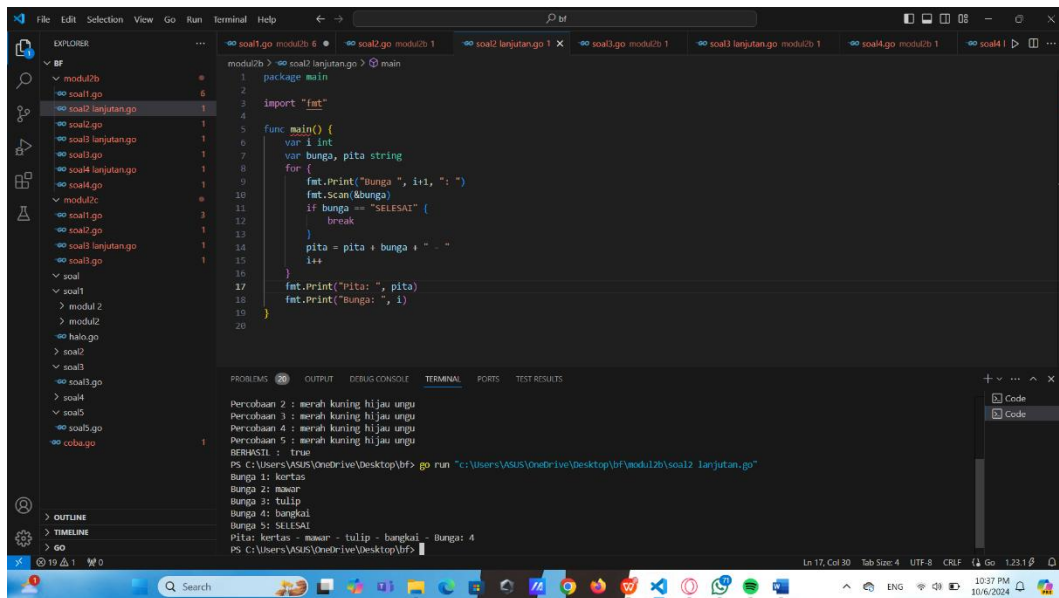
    fmt.Println("BERHASIL : ", !beda)
}

```

Penjelasan :

Program ini melakukan validasi apakah pengguna memasukkan kombinasi warna yang tepat (merah, kuning, hijau, ungu) dalam lima percobaan. Jika ada satu kali saja input yang tidak sesuai dengan kombinasi yang diharapkan, program akan menandai bahwa percobaan tidak berhasil, dan hasil akhirnya akan mencetak "BERHASIL: false". Namun, jika semua input dalam lima percobaan sesuai dengan kombinasi yang diharapkan, hasil akhirnya akan mencetak "BERHASIL: true"

NO 2.



The screenshot shows a Go IDE with a file explorer on the left, a source code editor in the center, and a terminal at the bottom. The source code is a Go program that prints a sequence of flowers and a final status. The terminal output shows the program's execution, including the sequence of flowers and the final status.

```
1 package main
2
3 import "fmt"
4
5 func main() {
6     var i int
7     var bunga, pita string
8     for {
9         fmt.Print("Bunga ", i+1, ": ")
10        fmt.Scan(&bunga)
11        if bunga == "SELESAI" {
12            break
13        }
14        pita = pita + bunga + " - "
15        i++
16    }
17    fmt.Print("Pita: ", pita)
18    fmt.Print("Bunga: ", i)
19 }
20
```

Terminal Output:

```
Percoobaan 2 : merah kuning hijau ungu
Percoobaan 3 : merah kuning hijau ungu
Percoobaan 4 : merah kuning hijau ungu
Percoobaan 5 : merah kuning hijau ungu
SELESAI : true
Bunga 1: kertas
Bunga 2: mawar
Bunga 3: tulip
Bunga 4: bangkai
Bunga 5: SELESAI
Pita: kertas - mawar - tulip - bangkai - Bunga: 4
PS C:\Users\VASUS\OneDrive\Desktop\bf> go run "c:\Users\VASUS\OneDrive\Desktop\bf\modul2b\soal2_lanjutan.go"
```

Source code :

package main

import "fmt"

func main() {

var i int

var bunga, pita string

for {

fmt.Print("Bunga ", i+1, ": ")

fmt.Scan(&bunga)

if bunga == "SELESAI" {

break

}

```

        pita = pita + bunga + " - "

        i++

    }

    fmt.Print("Pita: ", pita)

    fmt.Print("Bunga: ", i)

}

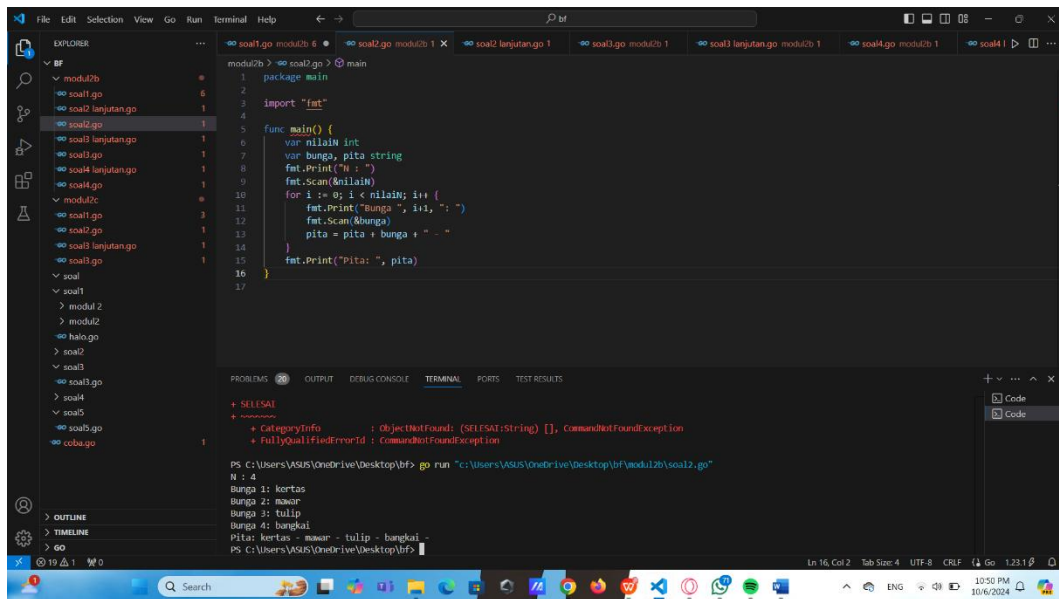
```

Penjelasan :

Program ini berfungsi sebagai alat untuk mengumpulkan daftar nama bunga yang dimasukkan oleh pengguna, dengan input yang dihentikan ketika pengguna mengetik "SELESAI". Pada akhir program, nama-nama bunga tersebut dicetak sebagai sebuah string pita, dan program juga menampilkan jumlah total bunga yang dimasukkan.

Program ini menggunakan perulangan tak terbatas dengan `for {}`, yang merupakan cara efektif untuk membuat loop yang terus berlanjut sampai syarat tertentu terpenuhi, seperti mendeteksi input "SELESAI" untuk berhenti.

Modifikasi program sebelumnya, proses input akan berhenti apabila user mengetikkan “SELESAI” . Kemudian tampilkan isi pita beserta banyaknya bunga yang ada di dalam pita.



Source code :

```
package main
```

```
import "fmt"
```

```
func main() {
```

```
    var nilaiN int
```

```
    var bunga, pita string
```

```
    fmt.Print("N : ")
```

```
    fmt.Scan(&nilaiN)
```

```
    for i := 0; i < nilaiN; i++ {
```

```
        fmt.Print("Bunga ", i+1, ": ")
```

```
        fmt.Scan(&bunga)
```

```
        pita = pita + bunga + " - "
```

```
    }
```

```

    fmt.Print("Pita: ", pita)
}

```

Penjelasan :

Program ini memungkinkan pengguna untuk menentukan jumlah bunga yang akan dimasukkan terlebih dahulu, kemudian mengambil input nama bunga sesuai dengan jumlah yang ditentukan. Nama-nama bunga yang dimasukkan oleh pengguna akan digabungkan ke dalam satu string yang dipisahkan oleh tanda strip " - " dan kemudian dicetak setelah semua input selesai.

Program ini bersifat sederhana dan memungkinkan pengguna untuk mengontrol berapa banyak input yang harus diberikan, lalu menghasilkan sebuah pita yang berisi semua nama bunga yang dimasukkan dengan format yang konsisten.

NO 3.

The screenshot shows a Go IDE with the following source code in `modul2b.go`:

```

package main

import "fmt"

func main() {
    var kantongA, kantongB float64
    for {
        fmt.Print("Masukkan berat belanjaan kedua kantong: ")
        fmt.Scan(&kantongA, &kantongB)
        if kantongA >= 9 || kantongB >= 9 {
            fmt.Print("Proses selesai")
            break
        }
    }
}

```

The terminal output shows the execution of the program:

```

N : 4
Bunga 1: kertas
Bunga 2: mawar
Bunga 3: tulip
Bunga 4: bangkai
Pita: kertas - mawar - tulip - bangkai -
PS C:\Users\ASUS\OneDrive\Desktop\bfs> go run "C:\Users\ASUS\OneDrive\Desktop\bfs\modul2b\soal3.go"
Masukkan berat belanjaan kedua kantong: 5.5 1.0
Masukkan berat belanjaan kedua kantong: 7.0 0.5
Masukkan berat belanjaan kedua kantong: 2.0
Masukkan berat belanjaan kedua kantong: 9.5.8
Proses selesai
PS C:\Users\ASUS\OneDrive\Desktop\bfs>

```

Source code :

```
package main
```

```

import "fmt"

func main() {
    var kantonga, kantongb float64

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

        fmt.Scan(&kantonga, &kantongb)

        if kantonga >= 9 || kantongb >= 9 {
            fmt.Print("Proses selesai")

            break
        }
    }
}

```

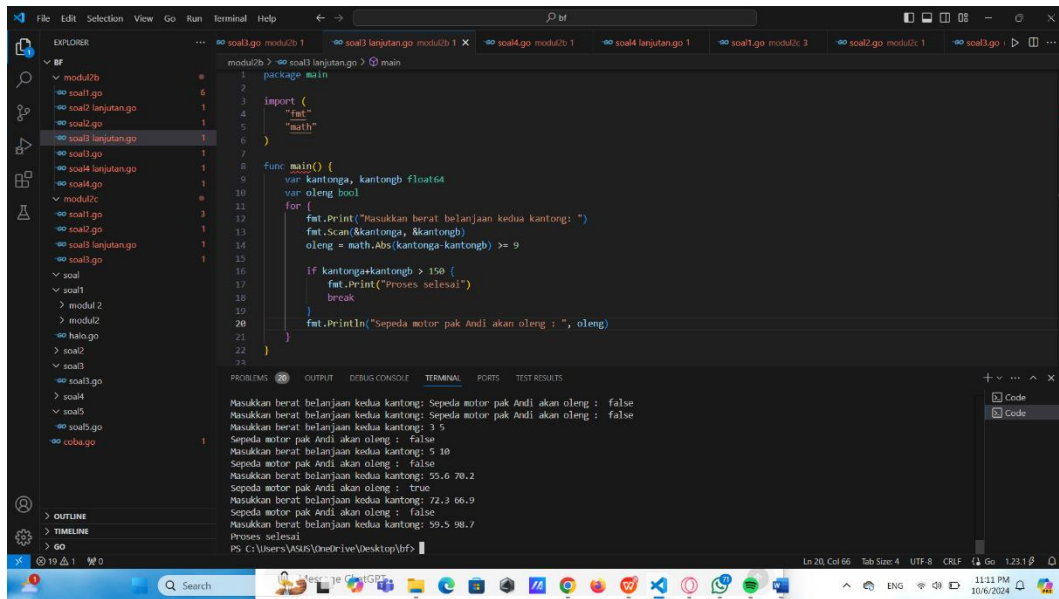
Penjelasan :

Program ini digunakan untuk memeriksa berat belanjaan dua kantong secara berulang. Program akan terus meminta pengguna untuk memasukkan berat sampai salah satu kantong memiliki berat 9 kg atau lebih, kemudian program akan menghentikan proses. Kondisi ini menunjukkan adanya batas maksimum berat kantong yang tidak boleh dilampaui, sehingga jika salah satu kantong terlalu berat, program akan menghentikan pengisian lebih lanjut.

Program ini dapat digunakan dalam situasi di mana ada batasan berat maksimum untuk kantong belanja, dan membantu mencegah pengguna memasukkan lebih banyak barang ke dalam kantong yang sudah mendekati batas tersebut.

Pada modifikasi program tersebut, program akan menampilkan true jika selisih kedua isi kantong lebih dari atau sama dengan

9 kg. Program berhenti memproses apabila total berat isi kedua kantong melebihi 150 kg atau salah satu kantong beratnya negatif.



Source code :

```
package main
```

```
import (
```

```
    "fmt"
```

```
    "math"
```

```
)
```

```
func main() {
```

```
    var kantonga, kantongb float64
```

```
    var oleng bool
```

```
    for {
```

```

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

    fmt.Scan(&kantonga, &kantongb)

    oleng = math.Abs(kantonga-kantongb) >= 9

    if kantonga+kantongb > 150 {

        fmt.Print("Proses selesai")

        break

    }

    fmt.Println("Sepeda motor pak Andi akan oleng : ", oleng)

}

}

```

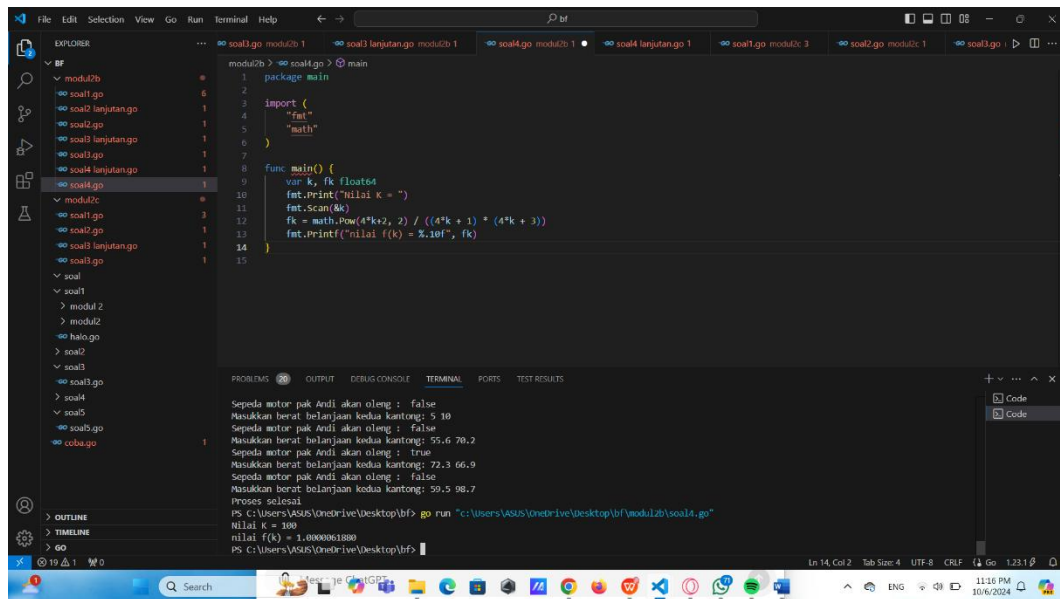
Penjelasan :

Program ini berfungsi untuk memantau dua hal penting terkait berat belanjaan yang dimasukkan ke dua kantong:

1. Apakah sepeda motor Pak Andi akan oleng jika selisih berat antara dua kantong terlalu besar (9 kg atau lebih).
2. Apakah total berat belanjaan di kedua kantong melebihi 150 kg, di mana jika melebihi, program akan menghentikan proses lebih lanjut.

Program ini membantu mengatur distribusi berat belanjaan secara seimbang antara dua kantong dan memastikan agar total berat tidak melebihi batas tertentu untuk mencegah sepeda motor oleng atau terlalu berat.

NO 4.



Source Code :

```
package main
```

```
import (
```

```
    "fmt"
```

```
    "math"
```

```
)
```

```
func main() {
```

```
    var k, fk float64
```

```
    fmt.Print("Nilai K = ")
```

```
    fmt.Scan(&k)
```

```
    fk = math.Pow(4*k+2, 2) / ((4*k + 1) * (4*k + 3))
```

```
    fmt.Printf("nilai f(k) = %.10f", fk)
```

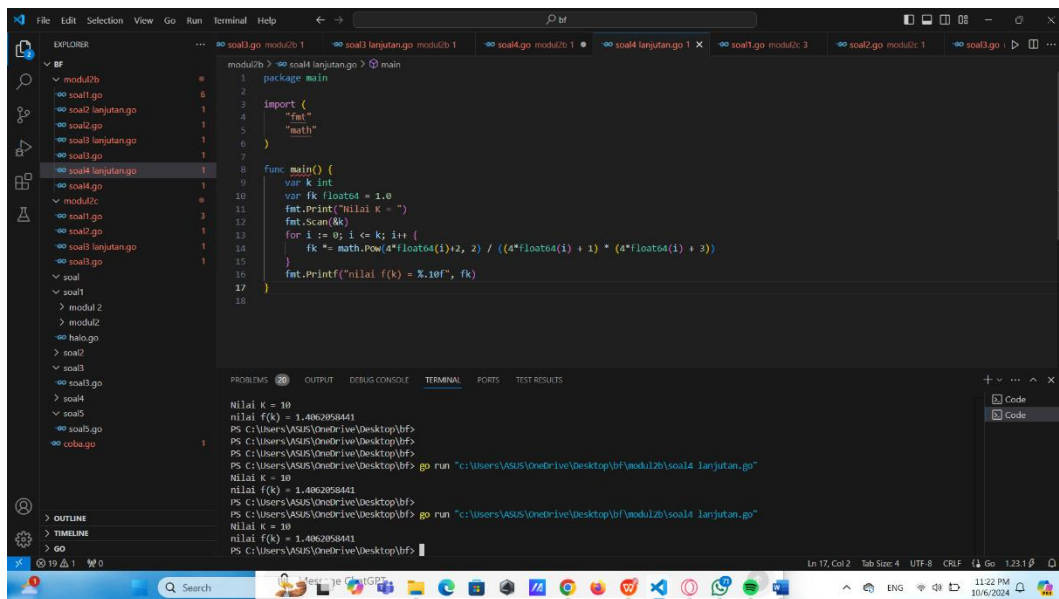
```
}
```

Penjelasan :

Program ini berfungsi untuk menghitung nilai fungsi matematika $f(k)$ yang berbentuk pecahan dengan numerator berupa kuadrat dari ekspresi linear dan denominator berupa hasil perkalian dua ekspresi linear. Pengguna memasukkan nilai k , dan program menghitung hasil perhitungan tersebut dengan presisi tinggi menggunakan fungsi matematika `math.Pow`. Hasil akhir ditampilkan dengan 10 angka desimal untuk memastikan akurasi.

Program ini sangat cocok digunakan untuk menghitung fungsi-fungsi matematis dengan presisi tinggi, terutama jika melibatkan operasi pangkat dan pecahan.

Modifikasi program sebelumnya yang menerima input integer ? dan menghitung $\sqrt{2}$ untuk ? tersebut. Hampiran $\sqrt{2}$ dituliskan dalam ketelitian 10 angka di belakang koma.



```
1 package main
2
3 import (
4     "fmt"
5     "math"
6 )
7
8 func main() {
9     var k int
10    var fk float64 = 1.0
11    fmt.Println("Nilai K = ")
12    fmt.Scan(&k)
13    for i := 0; i <= k; i++ {
14        fk = math.Pow(4*float64(i)+2, 2) / ((4*float64(i) + 1) * (4*float64(i) + 3))
15    }
16    fmt.Printf("Nilai f(k) = %.10f", fk)
17 }
18
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS TEST RESULTS

```
Nilai K = 10
Nilai f(k) = 1.4862958441
PS C:\Users\VASUS\OneDrive\Desktop\bf>
PS C:\Users\VASUS\OneDrive\Desktop\bf>
PS C:\Users\VASUS\OneDrive\Desktop\bf> go run "c:\Users\VASUS\OneDrive\Desktop\bf\modul2b\soal4_lanjutan.go"
Nilai K = 10
Nilai f(k) = 1.4862958441
PS C:\Users\VASUS\OneDrive\Desktop\bf>
PS C:\Users\VASUS\OneDrive\Desktop\bf> go run "c:\Users\VASUS\OneDrive\Desktop\bf\modul2b\soal4_lanjutan.go"
Nilai K = 10
Nilai f(k) = 1.4862958441
PS C:\Users\VASUS\OneDrive\Desktop\bf>
```

Source code :

```
package main
```

```
import (
```

```
    "fmt"
```

```

    "math"
)

func main() {
    var k int

    var fk float64 = 1.0

    fmt.Print("Nilai K = ")

    fmt.Scan(&k)

    for i := 0; i <= k; i++ {
        fk *= math.Pow(4*float64(i)+2, 2) / ((4*float64(i) + 1) * (4*float64(i)
+ 3))
    }

    fmt.Printf("nilai f(k) = %.10f", fk)
}

```

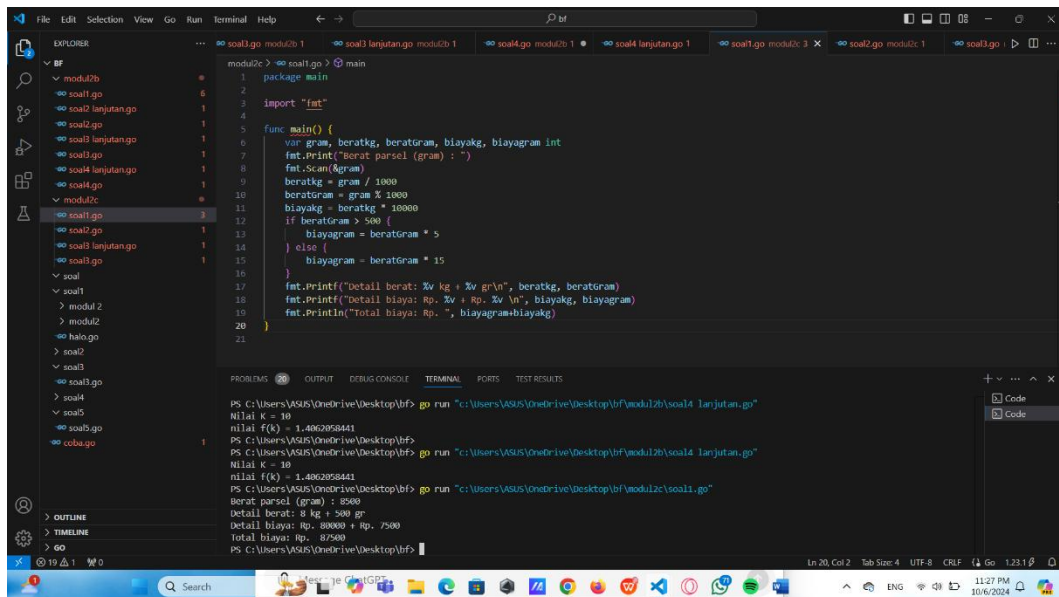
Penjelasan :

Program ini menghitung nilai fungsi $f(k)$ melalui iterasi. Pada setiap iterasi, nilai fk diperbarui dengan hasil dari ekspresi matematis yang melibatkan kuadrat dan perkalian beberapa bilangan. Fungsi ini dapat dianggap sebagai bentuk rekursif atau iteratif dari sebuah seri matematis. Program ini berguna untuk menghitung produk berkelanjutan berdasarkan formula tertentu yang melibatkan k sebagai batas atas perhitungan.

Dengan pendekatan perulangan ini, program menghitung hasil akhir yang akurat dan menampilkannya dengan presisi hingga 10 digit desimal.

C. MODUL 2C.

NO 1.



Source code :

```
package main
```

```
import "fmt"
```

```
func main() {
```

```
    var gram, beratkg, beratGram, biayakg, biayagram int
```

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

```
    fmt.Scan(&gram)
```

```
    beratkg = gram / 1000
```

```
    beratGram = gram % 1000
```

```
    biayakg = beratkg * 10000
```

```
    if beratGram > 500 {
```

```
        biayagram = beratGram * 5
```

```
    } else {
```

```

        biayagram = beratGram * 15
    }

    fmt.Printf("Detail berat: %v kg + %v gr\n", beratkg, beratGram)

    fmt.Printf("Detail biaya: Rp. %v + Rp. %v \n", biayakg, biayagram)

    fmt.Println("Total biaya: Rp. ", biayagram+biayakg)

}

```

Penjelasan :

Program ini menghitung total biaya pengiriman parcel berdasarkan berat yang dimasukkan oleh pengguna dalam gram. Berat dipecah menjadi kilogram dan gram, di mana setiap kilogram dikenakan tarif tetap **Rp. 10.000**, sementara gram dikenakan biaya yang bervariasi tergantung apakah beratnya lebih dari atau kurang dari 500 gram. Program ini dirancang untuk memberikan informasi terperinci tentang berat dan biaya, serta menghitung total biaya secara otomatis.

Ini sangat berguna dalam konteks logistik atau layanan pengiriman di mana biaya dihitung berdasarkan berat dan tarif yang berbeda untuk berat dalam satuan kecil seperti gram.

NO 2.

```

modul2c > go run modul2c.go
5 func main() {
6     rnmk = "A"
7 } else if nam > 72.5 {
8     rnmk = "B"
9 } else if nam > 65 {
10    rnmk = "C"
11 } else if nam > 57.5 {
12    rnmk = "D"
13 } else if nam > 50 {
14    rnmk = "E"
15 } else if nam > 40 {
16    rnmk = "F"
17 } else {
18    rnmk = "G"
19 }
20 fmt.Println("Nilai mata kuliah: ", rnmk)
21 }
22
23
24
25
26
27

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS TEST RESULTS

```

Nilai mata kuliah: B
PS C:\Users\ASUS\OneDrive\Desktop\bf> go run "c:\Users\ASUS\OneDrive\Desktop\bf\modul2c\soal2.go"
Nilai akhir mata kuliah: 93.5
Nilai mata kuliah: A
PS C:\Users\ASUS\OneDrive\Desktop\bf> go run "c:\Users\ASUS\OneDrive\Desktop\bf\modul2c\soal2.go"
Nilai akhir mata kuliah: 70.6
Nilai mata kuliah: B
PS C:\Users\ASUS\OneDrive\Desktop\bf> go run "c:\Users\ASUS\OneDrive\Desktop\bf\modul2c\soal2.go"
Nilai akhir mata kuliah: 49.5
Nilai mata kuliah: D
PS C:\Users\ASUS\OneDrive\Desktop\bf>

```

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)
    }
}
```

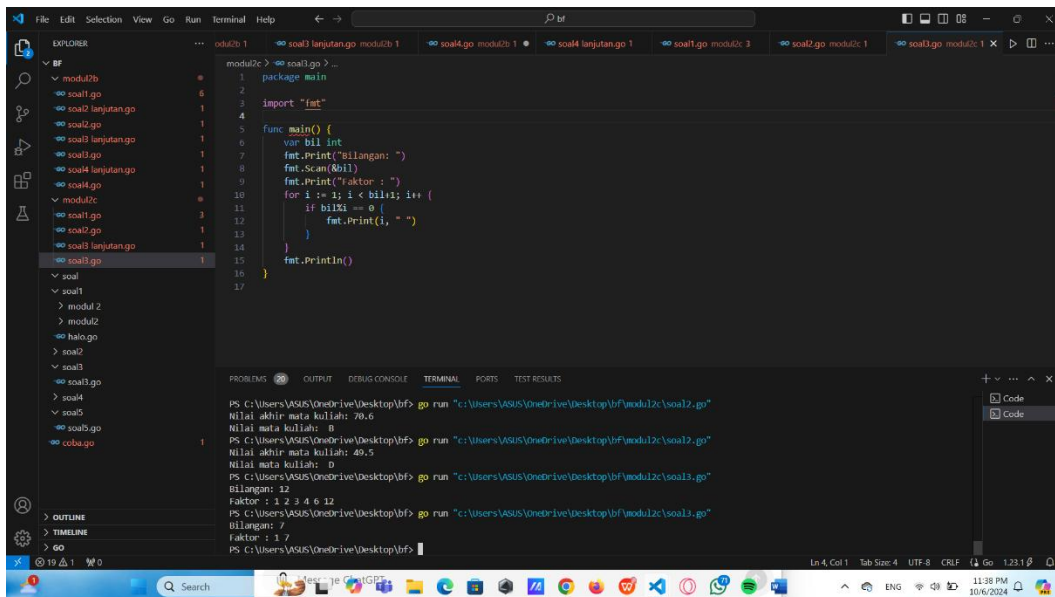
Penjelasan :

Program ini berfungsi untuk mengonversi nilai akhir mata kuliah yang berupa angka menjadi huruf nilai (grade) sesuai dengan rentang yang telah ditentukan. Pengguna memasukkan nilai numerik, dan program secara otomatis memberikan hasil dalam bentuk huruf berdasarkan klasifikasi berikut:

- A (nilai > 80)
- AB (nilai > 72.5)
- B (nilai > 65)
- BC (nilai > 57.5)
- C (nilai > 50)
- D (nilai > 40)
- E (nilai <= 40)

Program ini berguna dalam konteks pendidikan, terutama untuk mengonversi nilai ujian atau nilai mata kuliah ke dalam sistem penilaian yang umum digunakan di perguruan tinggi atau sekolah.

NO 03.



Source code :

package main

import "fmt"

func main() {

var bil int

fmt.Println("Bilangan: ")

fmt.Scan(&bil)

fmt.Println("Faktor : ")

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

if bil%i == 0 {

fmt.Print(i, " ")

}

}

```

    fmt.Println()
}

```

Penjelasan :

Program ini dirancang untuk menemukan dan mencetak semua faktor dari bilangan bulat positif yang dimasukkan oleh pengguna. Program ini mengambil input satu bilangan dan kemudian menghitung faktor-faktornya dengan cara membagi bilangan tersebut dengan setiap bilangan dari 1 hingga bilangan itu sendiri. Semua faktor yang ditemukan dicetak ke layar.

Faktor-faktor dari sebuah bilangan adalah bilangan yang dapat membagi bilangan tersebut tanpa menghasilkan sisa. Program ini berguna dalam konteks matematika, terutama untuk analisis numerik dan pembelajaran dasar tentang bilangan.

Lanjutkan program sebelumnya. Setelah menerima masukan sebuah bilangan bulat $b > 0$. Program tersebut mencari dan menampilkan semua faktor bilangan tersebut. Kemudian, program menentukan apakah b merupakan bilangan prima.

The screenshot shows a Go IDE with the following source code in `module2c.go`:

```

1 package main
2
3 import "fmt"
4
5 func main() {
6     var bil, count int
7     fmt.Print("Bilangan: ")
8     fmt.Scan(&bil)
9     fmt.Print("Faktor: ")
10    for i := 1; i <= bil; i++ {
11        if bil%i == 0 {
12            fmt.Print(i, " ")
13            count++
14        }
15    }
16    fmt.Println("\nPrima : ", count == 2)
17 }
18
19

```

The terminal output shows the results of running the program for different inputs:

```

PS C:\Users\ASUS\OneDrive\Desktop\bf> go run "c:\Users\ASUS\OneDrive\Desktop\bf\module2c\soal1.go"
Faktor : 1 2 3 4 6 12
PS C:\Users\ASUS\OneDrive\Desktop\bf> go run "c:\Users\ASUS\OneDrive\Desktop\bf\module2c\soal1.go"
Bilangan: 7
Faktor : 1 7
PS C:\Users\ASUS\OneDrive\Desktop\bf> go run "c:\Users\ASUS\OneDrive\Desktop\bf\module2c\soal1_lanjutan.go"
Bilangan: 12
Faktor : 1 2 3 4 6 12
Prima : false
PS C:\Users\ASUS\OneDrive\Desktop\bf> go run "c:\Users\ASUS\OneDrive\Desktop\bf\module2c\soal1_lanjutan.go"
Bilangan: 7
Faktor : 1 7
Prima : true
PS C:\Users\ASUS\OneDrive\Desktop\bf>

```

Source code :

```
package main
```

```

import "fmt"

func main() {
    var bil, count int

    fmt.Print("Bilangan: ")

    fmt.Scan(&bil)

    fmt.Print("Faktor : ")

    for i := 1; i < bil+1; i++ {
        if bil%i == 0 {
            fmt.Print(i, " ")

            count++
        }
    }

    fmt.Println("\nPrima : ", count == 2)
}

```

Penjelasan :

Program ini memiliki dua fungsi utama:

1. Menghitung dan mencetak semua faktor dari bilangan bulat positif yang dimasukkan pengguna.
2. Menentukan dan mencetak apakah bilangan tersebut adalah bilangan prima dengan memeriksa jumlah faktor yang ditemukan.

Dengan menggunakan program ini, pengguna dapat dengan mudah mengetahui faktor-faktor dari sebuah bilangan serta status primanya. Hal ini bermanfaat dalam pembelajaran matematika, khususnya dalam memahami konsep bilangan prima dan faktor. Program ini juga

mengajarkan penggunaan perulangan, kondisi, dan penghitungan dalam pemrograman.