LAPORAN PRAKTIKUM
PEMROGRAMAN BERORIENTASI OBJEK

MODUL II

REVIEW STRUKTUR KONTROL

Oleh:

NAMA : ARNANDA SETYA NOSA PUTRA

NIM : 2311102180

KELAS : IF 11 02

S1 TEKNIK INFORMATIKA

INSTITUT TEKNOLOGI TELKOM PURWOKERTO

2024

I. DASAR TEORI

Struktur Program Go
Dalam pemrograman menggunakan bahasa Go, program utama selalu memiliki dua komponen inti:
- **package main** menandakan bahwa file tersebut berisi program utama.
- **func main()** adalah tempat utama di mana kode program Go ditulis.

Komentar tidak dianggap sebagai bagian dari kode program dan bisa ditempatkan di mana saja dalam program. Ada dua jenis komentar:
- Komentar satu baris dimulai dengan dua garis miring ("//") hingga akhir baris.
- Komentar beberapa baris dimulai dengan "/" dan diakhiri dengan "/".

Tipe Data dan Instruksi Dasar

Data dan Variabel
Variabel adalah nama yang digunakan untuk merujuk pada lokasi di memori, di mana data dengan tipe tertentu dapat disimpan. Nama variabel harus diawali dengan huruf dan bisa diikuti oleh huruf, angka, atau garis bawah. Beberapa tipe data yang umum dalam Go termasuk integer, real, boolean, karakter, dan string. Nilai dari suatu variabel bisa diakses dengan menyebutkan nama variabel tersebut. Misalnya, jika variabel bernama "found", maka menyebut "found" akan mengambil nilai yang disimpan di dalamnya.

Alamat atau lokasi variabel di memori bisa diperoleh dengan menambahkan tanda "&" sebelum nama variabel. Sebagai contoh, "&found" akan mengembalikan alamat memori dari variabel "found". Jika variabel berisi alamat memori, tanda "*" sebelum nama variabel tersebut akan mengakses nilai yang tersimpan di memori yang ditunjuk.

Variabel harus dideklarasikan sebelum digunakan dan harus diinisialisasi dengan nilai awal agar program berjalan dengan benar. Dalam Go, variabel yang tidak diinisialisasi secara otomatis akan diisi dengan nilai default:
- 0 untuk bilangan integer,
- 0.0E+0 untuk bilangan real,
- false untuk boolean,
- karakter NUL (sesuai dengan tabel ASCII) untuk tipe karakter,
- string kosong ("") untuk tipe string,
- nil untuk alamat memori.

Struktur Kontrol Perulangan

Go hanya memiliki satu kata kunci **for** yang digunakan untuk semua jenis perulangan. Dua bentuk umum perulangan yang digunakan adalah while-loop dan repeat-until. Dalam pemrograman terstruktur, setiap algoritma harus memiliki satu titik masuk dan satu titik keluar. Oleh karena itu, penggunaan lebih dari satu pintu keluar dalam sebuah loop, seperti menggunakan break di dalam loop dan di instruksi **for,** tidak disarankan.

1. While-Loop
Pada while-loop, kondisi harus benar (true) agar loop bisa berjalan. Ketika kondisi menjadi salah (false), perulangan akan berhenti.

2. Repeat-Until
Perulangan ini terus dilakukan hingga kondisi keluar terpenuhi (benar/true). Jika kondisi belum terpenuhi (salah/false), perulangan akan tetap berjalan. Berbeda dengan while-loop, di repeat-until, tubuh loop akan dieksekusi setidaknya satu kali karena kondisi diperiksa setelah loop dijalankan.

Struktur Kontrol Percabangan

Go mendukung dua bentuk utama percabangan, yaitu if-else dan switch-case.

1. If-Else
Terdapat berbagai variasi struktur if-else dalam Go. Pada dasarnya, semua bentuk tersebut terdiri dari satu blok if-else-endif. Jika diperlukan, struktur bersarang dapat dibentuk dengan menggabungkan beberapa blok if-else-endif.

2. Switch-Case
Go memiliki dua variasi bentuk switch-case. Yang pertama, switch diikuti oleh ekspresi dan setiap case mengandung nilai untuk dibandingkan. Bentuk kedua tidak memerlukan ekspresi pada switch, namun setiap case bisa berisi ekspresi boolean. Bentuk kedua ini lebih fleksibel dan sering digunakan sebagai alternatif dari if-elseif-else.

II. GUIDED

Source Code + Screenshot hasil program beserta penjelasan

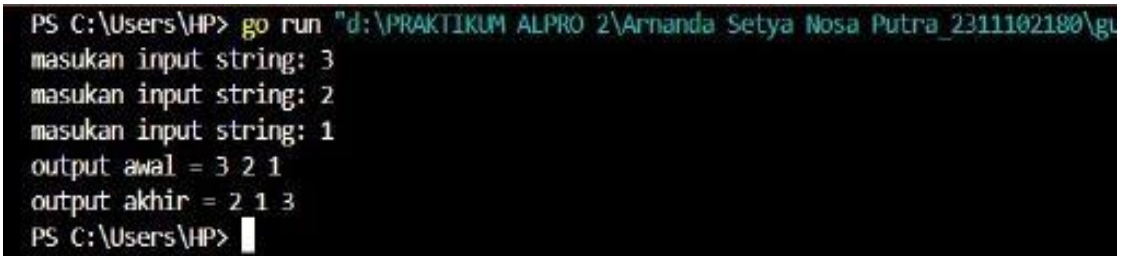
SOAL 2A

NO 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)

}
```

Output :



Penjelasan :

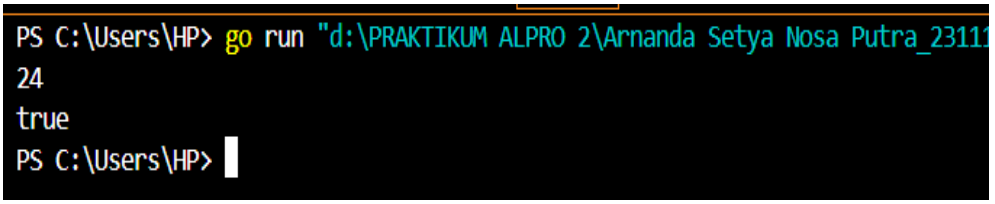
Program ini melakukan tugas sederhana yaitu membaca tiga input string dari pengguna, menampilkan urutan awalnya, lalu menukar urutannya secara melingkar dan menampilkannya kembali.

NO 2.

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")

    }
}
```

Output:



Penjelasan :

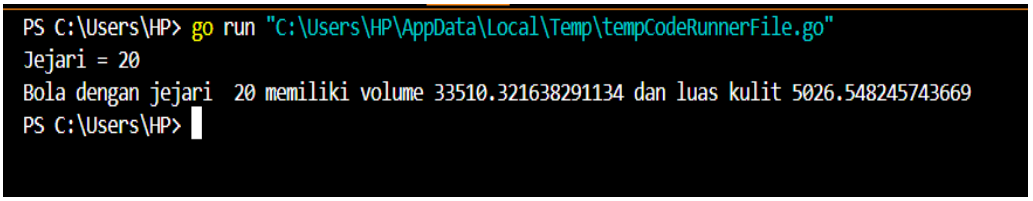
Program ini untuk menentukan apakah sebuah tahun yang dimasukkan oleh pengguna merupakan tahun kabisat atau tidak,

NO 3.

Soure 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)
}
```

Output :



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 yaitu Volume bola danLuas permukaan bola

III. UNGUIDED

Source Code + Screenshot hasil program beserta penjelasan

SOAL 2B

NO 1.

Source code :

```
D: > PRAKTIKUM ALPRO 2 > Arnanda Setya Nosa Putra_2311102180 > unguided modul 2 > 2B1.go > main
1   package main    No packages found for open file D:\PRAKTIKUM ALPRO 2\Arnanda Setya Nosa Putra_2311102180\unguided modul 2\2B1.go.
2
3   import (
4       "fmt"
5       "strings"
6   )
7
8   func main() {
9       const urutan = "merahkuninghijauungu"
10      var gelas [4]string
11      var hasil bool = true
12
13      for i := 0; i < 5; i++ {
14
15          fmt.Printf("Percobaan %d: ", i+1)
16          for j := 0; j < 4; j++ {
17              fmt.Scan(&gelas[j])
18          }
19
20          if urutan != strings.ToLower(strings.Join(gelas[:], "")) {
21              hasil = false
22          }
23      }
24
25      fmt.Printf("Berhasil: %t\n", hasil)
26  }
27
```

Output :

```
PS C:\Users\HP> go run "C:\Users\HP\AppData\Local\Temp\tempCodeRunnerFile.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\HP> go run "C:\Users\HP\AppData\Local\Temp\tempCodeRunnerFile.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
```

Penjelasan :

Program di atas melakukan pengecekan apakah urutan warna yang dimasukkan pengguna dalam 5 percobaan sesuai dengan urutan yang sudah ditentukan, yaitu "merahkuninghijauungu". Pada setiap percobaan, pengguna diminta memasukkan 4 warna yang disimpan dalam array gelas. Warna-warna yang dimasukkan kemudian digabung menjadi satu string dan diubah menjadi huruf kecil menggunakan strings.ToLower. Program kemudian membandingkan string ini dengan urutan warna yang sudah ditentukan. Jika ada satu kali percobaan yang tidak sesuai, variabel hasil diubah menjadi false. Setelah 5 percobaan selesai, program akan mencetak apakah percobaan berhasil atau tidak berdasarkan nilai variabel hasil.

NO 2.

Source Code :

```go
package main    No packages found for open file D:\PRAKTIKUM ALPRO 2\Arnanda Setya Nosa Putra_2311

import "fmt"

func inputBunga() []string {
    var bunga []string
    var namaBunga string

    for i := 1; ; i++ {
        fmt.Printf("Bunga %d: ", i)
        fmt.Scanln(&namaBunga)
        if namaBunga == "SELESAI" {
            break
        }
        bunga = append(bunga, namaBunga)
    }
    return bunga
}

func tampilkanPitaDanJumlah(bunga []string) {
    pita := ""
    for i, b := range bunga {
        if i > 0 {
            pita += " - "
        }
        pita += b
    }

    fmt.Printf("Pita: %s\n", pita)
    fmt.Printf("Jumlah bunga dalam pita: %d\n", len(bunga))
}

func main() {
    bunga := inputBunga()
    tampilkanPitaDanJumlah(bunga)
}
```

Output:

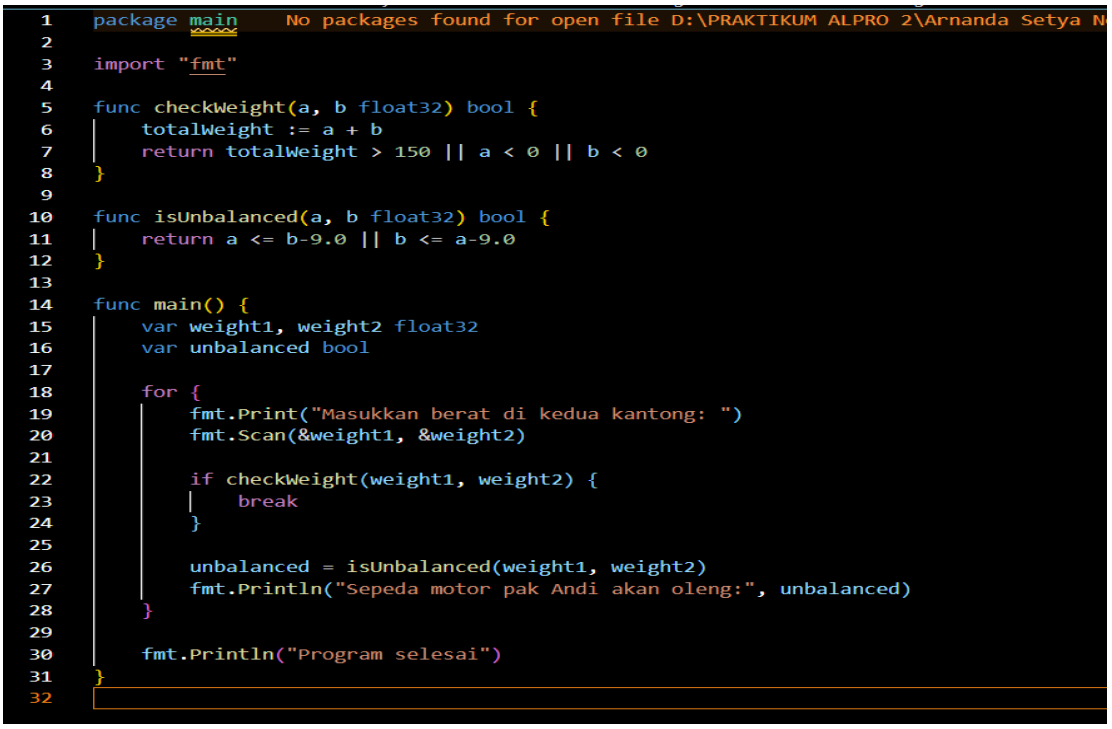
```
PS C:\Users\HP> go run "C:\Users\HP\AppData\Local\Temp\te
Bunga 1: kertas
Bunga 2: mawar
Bunga 3: tulip
Bunga 4: SELESAI
Pita: kertas - mawar - tulip
Jumlah bunga dalam pita: 3
PS C:\Users\HP>
```

Penjelasan :

Program di atas untuk menerima input nama bunga dari pengguna hingga pengguna mengetik "SELESAI." Fungsi inputBunga akan meminta pengguna untuk memasukkan nama bunga satu per satu dan menyimpannya dalam slice string. Setelah selesai, fungsi ini mengembalikan daftar bunga yang dimasukkan. Fungsi tampilkanPitaDanJumlah akan menerima daftar bunga tersebut dan menampilkannya dalam format yang dipisahkan oleh tanda " - " serta menghitung dan menampilkan jumlah bunga yang ada. Dalam fungsi main, program mengintegrasikan kedua fungsi ini dengan memanggil inputBunga untuk mengambil input dan tampilkanPitaDanJumlah untuk menampilkan hasilnya.

NO.3

Source code :



```go
package main    No packages found for open file D:\PRAKTIKUM ALPRO 2\Arnanda Setya N

import "fmt"

func checkWeight(a, b float32) bool {
    totalWeight := a + b
    return totalWeight > 150 || a < 0 || b < 0
}

func isUnbalanced(a, b float32) bool {
    return a <= b-9.0 || b <= a-9.0
}

func main() {
    var weight1, weight2 float32
    var unbalanced bool

    for {
        fmt.Print("Masukkan berat di kedua kantong: ")
        fmt.Scan(&weight1, &weight2)

        if checkWeight(weight1, weight2) {
            break
        }

        unbalanced = isUnbalanced(weight1, weight2)
        fmt.Println("Sepeda motor pak Andi akan oleng:", unbalanced)
    }

    fmt.Println("Program selesai")
}
```

Output :

```
PS C:\Users\HP> go run "C:\Users\HP\AppData\Local\Temp\tempCodeRunnerFile.go
Masukkan berat di kedua kantong: 5 10
Sepeda motor pak Andi akan oleng: false
Masukkan berat di kedua kantong: 55.6 70.2
Sepeda motor pak Andi akan oleng: true
Masukkan berat di kedua kantong: 72.3 66.9
Sepeda motor pak Andi akan oleng: false
Masukkan berat di kedua kantong: 59.5 98.7
Program selesai
PS C:\Users\HP>
```

Penjelasan :

Program di meminta pengguna untuk memasukkan berat dua kantong. Fungsi `checkWeight` memeriksa apakah total berat kedua kantong melebihi 150 atau jika salah satu berat kurang dari 0, yang akan menghentikan program. Sementara itu, fungsi `isUnbalanced` menentukan apakah perbedaan berat antara kedua kantong lebih dari 9, yang menunjukkan bahwa sepeda motor akan oleng. Program terus meminta input hingga kondisi berat terpenuhi, dan kemudian mencetak hasil keseimbangan sebelum menyelesaikan eksekusi.

NO.4

Source code :

D: > PRAKTIKUM ALPRO 2 > Arnanda Setya Nosa Putra_2311102180 > unguided modul 2 > 2B4.go > ...

```go
package main    No packages found for open file D:\PRAKTIKUM ALPRO 2\Arnanda Setya Nosa Putra

import (
    "fmt"
    "math"
)

func calculateResult(k int) float64 {
    result := 1.0
    for i := 0; i <= k; i++ {
        numerator := math.Pow(4*float64(i)+2, 2)
        denominator := (4*float64(i) + 1) * (4*float64(i) + 3)
        result *= numerator / denominator
    }
    return result
}

func main() {
    var k int

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

    hasil := calculateResult(k)
    fmt.Printf("Nilai Akar 2 = %.10f \n", hasil)
}

```

Output :

```
PS C:\Users\HP> go run "C:\Users\HP\AppData\Local\Temp\tempCodeRunnerFile.go"
Nilai K = 10
Nilai Akar 2 = 1.4062058441
PS C:\Users\HP> go run "C:\Users\HP\AppData\Local\Temp\tempCodeRunnerFile.go"
Nilai K = 100
Nilai Akar 2 = 1.4133387072
PS C:\Users\HP> go run "C:\Users\HP\AppData\Local\Temp\tempCodeRunnerFile.go"
Nilai K = 1000
Nilai Akar 2 = 1.4141252651
PS C:\Users\HP>
```

Penjelasan :

Program di atas adalah aplikasi dalam bahasa Go yang menghitung nilai akar dua menggunakan metode matematis. Pengguna diminta untuk memasukkan nilai integer $$ k $$, yang kemudian digunakan dalam fungsi `calculateResult`. Fungsi ini menghitung hasil dengan mengalikan serangkaian pecahan yang terdiri dari kuadrat dari $$ 4i + 2 $$ sebagai pembilang dan hasil kali dari $$ (4i + 1)(4i + 3) $$ sebagai penyebut, untuk setiap $$ i $$ dari 0 hingga $$ k $$. Hasil akhir ditampilkan dengan presisi sepuluh desimal, memberikan estimasi nilai akar dua berdasarkan input yang diberikan.

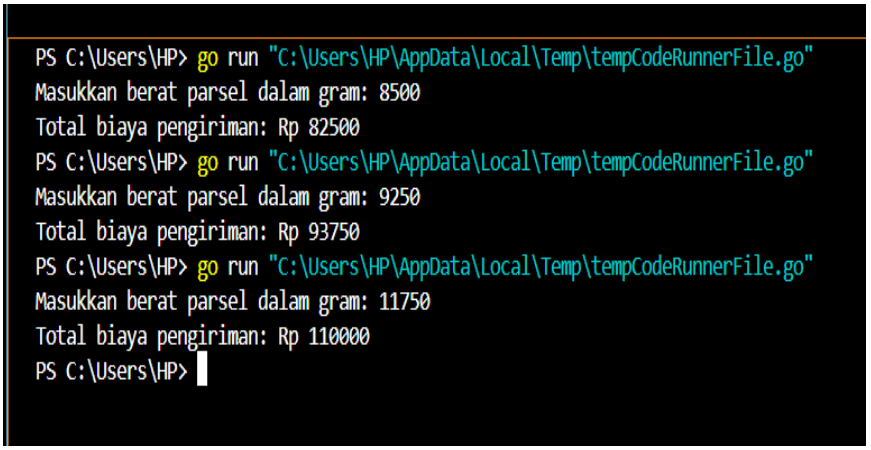
SOAL 2C

NO.1

Source code :

```go
package main    No packages found for open file D:\PRAKTIKUM ALPRO 2\Arnanda Setya Nosa F

import "fmt"

func main() {
    var berat, biaya, val, tambahan int

    fmt.Print("Masukkan berat parsel dalam gram: ")
    fmt.Scan(&berat)

    kg := berat / 1000
    sisaBerat := berat % 1000
    biaya = kg * 10000

    if sisaBerat >= 500 {
        tambahan = sisaBerat * 5
    } else if sisaBerat > 0 {
        tambahan = sisaBerat * 15
    }

    val = biaya + tambahan

    if kg > 10 {
        fmt.Printf("Total biaya pengiriman: Rp %d\n", biaya)
    } else {
        fmt.Printf("Total biaya pengiriman: Rp %d\n", val)
    }
}

```

Output :



Penjelasan :

Program di atas adalah aplikasi dalam bahasa Go yang menghitung biaya pengiriman parsel berdasarkan beratnya dalam gram. Pengguna diminta untuk memasukkan berat parsel, yang kemudian dikonversi menjadi kilogram dan sisa berat dalam gram. Biaya dasar dihitung dengan mengalikan jumlah kilogram dengan tarif Rp 10.000. Jika sisa berat mencapai 500 gram atau lebih, biaya tambahan dihitung dengan tarif Rp 5 per gram; jika sisa berat kurang dari 500 gram tetapi lebih dari 0, tarifnya menjadi Rp 15 per gram. Total biaya pengiriman ditentukan berdasarkan berat: jika berat lebih dari 10 kg, hanya biaya dasar yang ditampilkan; jika tidak, total biaya termasuk tambahan juga ditampilkan.

NO.2

Source code :

```go
package main    No packages found for open file D:\PRAKTIKUM ALPRO 2\Arnanda

import "fmt"

func main() {
    var nilaiAkhir float64
    var nilaiHuruf string

    fmt.Print("Nilai akhir mata kuliah: ")
    fmt.Scanln(&nilaiAkhir)

    switch {
    case nilaiAkhir > 80:
        nilaiHuruf = "A"
    case nilaiAkhir > 72.5:
        nilaiHuruf = "AB"
    case nilaiAkhir > 65:
        nilaiHuruf = "B"
    case nilaiAkhir > 57.5:
        nilaiHuruf = "BC"
    case nilaiAkhir > 50:
        nilaiHuruf = "C"
    case nilaiAkhir > 40:
        nilaiHuruf = "D"
    default:
        nilaiHuruf = "E"
    }

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

```

Output :

```
PS C:\Users\HP> go run "C:\Users\HP\AppData\Local\Temp\tempCodeRunnerFile.go"
Nilai akhir mata kuliah: 93.5
Nilai mata kuliah: A
PS C:\Users\HP> go run "C:\Users\HP\AppData\Local\Temp\tempCodeRunnerFile.go"
Nilai akhir mata kuliah: 70.6
Nilai mata kuliah: B
PS C:\Users\HP> go run "C:\Users\HP\AppData\Local\Temp\tempCodeRunnerFile.go"
Nilai akhir mata kuliah: 49.5
Nilai mata kuliah: D
PS C:\Users\HP>
```

Penjelasan :

Program di atas adalah aplikasi dalam bahasa Go yang mengonversi nilai akhir mata kuliah menjadi nilai huruf. Pengguna diminta untuk memasukkan nilai akhir dalam bentuk angka desimal. Berdasarkan nilai yang dimasukkan, program menggunakan pernyataan `switch` untuk menentukan nilai huruf yang sesuai, dengan kriteria sebagai berikut: nilai di atas 80 mendapatkan "A", di atas 72.5 mendapatkan "AB", dan seterusnya hingga nilai di bawah atau sama dengan 40 yang mendapatkan "D" atau "E". Setelah menentukan nilai huruf, program mencetak hasilnya ke layar.

NO.3

Source code :

```go
package main    No packages found for open file D:\PRAKTIKUM AL

import "fmt"

func main() {
    var x int

    fmt.Print("Masukkan x: ")
    fmt.Scan(&x)

    fmt.Print("Faktor-faktor dari ", x, ": ")
    for i := 1; i <= x; i++ {
        if x%i == 0 {
            fmt.Print(i, " ")
        }
    }
    fmt.Println()
}
```

Output :

```
PS C:\Users\HP> go run "C:\Users\HP\AppData\Local\Temp\tempCodeRunnerFile.go"
Masukkan x: 12
Faktor-faktor dari 12: 1 2 3 4 6 12
PS C:\Users\HP> go run "C:\Users\HP\AppData\Local\Temp\tempCodeRunnerFile.go"
Masukkan x: 7
Faktor-faktor dari 7: 1 7
PS C:\Users\HP>
```

Penjelasan :

Program di atas adalah aplikasi dalam bahasa Go yang menghitung dan menampilkan faktor-faktor dari sebuah bilangan bulat yang dimasukkan oleh pengguna. Setelah meminta pengguna untuk memasukkan nilai $$ x $$, program menggunakan loop `for` untuk memeriksa setiap angka dari 1 hingga $$ x $$ apakah merupakan faktor dari $$ x $$ dengan memeriksa apakah sisa pembagian $$ x $$ oleh angka tersebut sama dengan 0. Jika ya, angka tersebut dicetak sebagai faktor. Hasilnya ditampilkan dalam format yang jelas di konsol.