LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 1
MODUL X
“ELSE-IF”

DISUSUN OLEH:

FEROS PEDROSA VALENTINO

103112400055

S1 IF-12-01

DOSEN:

Yohani Setiya Rafika Nur, M. Kom.

PROGRAM STUDI S1 TEKNIK INFORMATIKA

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2024/2025

DASAR TEORI

Percabangan:

- Setiap baris instruksi dieksekusi satu persatu
- Adanya instruksi bersyarat (kondisi). Instruksi dieksekusi apabila memenuhi kondisi atau syarat tertentu
- Konstruksi dalam algoritma yang memungkinkan kita untuk melakukan pilihan instruksi yang berbeda-beda sesuai dengan kondisi yang dihadapi

Kondisi:

- Ekspresi yang bernilai TRUE atau FALSE (Boolean) Operasi Perbandingan atau Logika

Aksi:

- Kumpulan instruksi/ekspresi yang akan dieksekusi apabila kondisi bernilai TRUE
- Antara kondisi dan aksi terdapat suatu hubungan berpasangan

Sebelumnya telah dipelajari bahwa setiap baris kode program akan dieksekusi satu persatu secara sekuensial. Artinya kode program dari baris ke-1 hingga baris terakhis akan dieksekusi satu persatu. Bagaimana jika kita ingin baris kode program yang dieksekusi itu berdasarkan syarat atau suatu ketentuan tertentu? Sebagai analogi misalnya ketika kita berada dipersimpangan jalan ke kiri atau ke kanan. Maka di dalam pemrograman hal tersebut mungkin untuk dilakukan, struktur kontrol yang digunakan adalah else-if.

Pada dasarnya else-if tidak jauh berbeda dengan struktur kontrol percabangan menggunakan if-then. Perbedaannya terletak pada adanya aksi lain yang akan dieksekusi apabila kondisi tidak terjadi atau bernilai false. Penulisan struktur kontrol percabangan dengan menggunakan else-if terdiri dari dua bagian, yaitu:

1. Kondisi, yaitu sesuatu syarat atau ketentuan dari suatu percabangan. Kondisi ini harus bernilai boolean, baik itu variabel ataupun operasi tipe data.
2. Aksi, yaitu kumpulan instruksi yang akan dilakukan apabila kondisi terpenuhi atau bernilai true.
3. Aksi lain, yaitu kumpulan instruksi yang akan dilakukan apabila kondisi terpenuhi atau bernilai false. Artinya aksi dan aksi lain ini merupakan pilihan, yang mana hanya salah satu aksi saja yang akan dieksekusi sesuai dengan nilai dari kondisi.

Setiap aksi ke-i hanya akan dieksekusi apabila kondisi ke-i bernilai true atau benar, sedangkan aksi lain hanya akan dieksekusi apabila semua kondisi dari i = 1 s.d n bernilai false.

CONTOH SOAL

1. Coso1

Source code:

```
package main

import "fmt"

func main() {
    var usia int
    var kk bool
    fmt.Scan(&usia, &kk)
    if usia >= 17 && kk {
        fmt.Println("bisa membuat KTP")
    } else {
        fmt.Println("belum bisa membuat KTP")
    }
}
```

Output:

```
PS C:\semester1_alpro1\alpro1_week10> go run "c:\semester1_alpro1\alpro1_week10\coso1\coso1.go"
17
true
bisa membuat KTP
PS C:\semester1_alpro1\alpro1_week10> go run "c:\semester1_alpro1\alpro1_week10\coso1\coso1.go"
20
false
belum bisa membuat KTP
PS C:\semester1_alpro1\alpro1_week10> go run "c:\semester1_alpro1\alpro1_week10\coso1\coso1.go"
15
true
belum bisa membuat KTP
```

Deskripsi program:

Program di atas adalah sebuah program yang ditulis dalam bahasa Go, yang menentukan apakah seseorang memenuhi syarat untuk membuat Kartu Tanda Penduduk (KTP). Pertama deklarasi variabel usia sebagai tipe data integer, deklarasi variabel kk sebagai tipe data boolean. Program ini meminta pengguna untuk memasukkan dua data: usia (dalam bentuk bilangan bulat) dan status kepemilikan Kartu Keluarga (KK) dalam bentuk boolean (true/false). Setelah data dimasukkan, program memeriksa dua kondisi: apakah usia pengguna 17 tahun atau lebih, dan apakah pengguna memiliki KK. Jika kedua kondisi terpenuhi, program mencetak pesan "bisa membuat KTP". Sebaliknya, jika salah satu atau kedua kondisi tidak terpenuhi, program mencetak pesan "belum bisa membuat KTP".

2. Coso2

Source code:

```
package main

import "fmt"

func main() {
   var x rune
   var huruf, vKecil, vBesar bool
   fmt.Scanf("%c", &x)
   huruf = (x >= 'a' && x <= 'z') || (x >= 'A' && x <= 'Z')
   vKecil = x == 'a' || x == 'i' || x == 'u' || x == 'e' || x == 'o'
   vBesar = x == 'A' || x == 'I' || x == 'U' || x == 'E' || x == 'O'
   if huruf && (vKecil || vBesar) {
      fmt.Println("vokal")
   } else if huruf && !(vKecil || vBesar) {
      fmt.Println("konsonan")
   } else {
      fmt.Println("Bukan huruf")
   }
}
```

Output:

```
PS C:\semester1_alpro1\alpro1_week10> go run "c:\semester1_alpro1\alpro1_week10\coso2\coso2.go"
A
vokal
PS C:\semester1_alpro1\alpro1_week10> go run "c:\semester1_alpro1\alpro1_week10\coso2\coso2.go"
f
konsonan
PS C:\semester1_alpro1\alpro1_week10> go run "c:\semester1_alpro1\alpro1_week10\coso2\coso2.go"
1
Bukan huruf
PS C:\semester1_alpro1\alpro1_week10> go run "c:\semester1_alpro1\alpro1_week10\coso2\coso2.go"
$
Bukan huruf
```

Deskripsi program:

Program di atas adalah sebuah program yang ditulis dalam bahasa pemrograman Go. Program ini bertujuan untuk mengidentifikasi karakter yang dimasukkan oleh pengguna, apakah karakter tersebut merupakan huruf vokal, huruf konsonan, atau bukan huruf sama sekali. Pertama, deklarasikan variabel x sebagai tipe data rune karena rune dalam Go adalah tipe data yang menyimpan kode yang mewakili karakter Unicode kemudian deklarasikan variabel huruf, vKecil, vBesar sebagai tipe data boolean. Selanjutnya program meminta pengguna untuk memasukkan satu karakter. Setelah karakter diinput, program kemudian memeriksa apakah karakter tersebut adalah huruf dengan mengecek apakah karakter tersebut berada dalam rentang huruf kecil ('a' sampai 'z') atau huruf besar ('A' sampai 'Z'). Selanjutnya, program menentukan apakah karakter tersebut adalah huruf vokal dengan membandingkannya dengan huruf vokal baik dalam bentuk kecil maupun besar. Jika karakter yang dimasukkan adalah huruf vokal, program akan mencetak "vokal". Jika karakter tersebut adalah huruf konsonan, program akan mencetak "konsonan". Namun, jika karakter yang dimasukkan bukan merupakan huruf, program akan mencetak "Bukan huruf".

3. Coso3

Source code:

```
package main

import "fmt"

func main() {
   var bilangan, d1, d2, d3, d4 int
   var teks string
   fmt.Print("Bilangan: ")
   fmt.Scan(&bilangan)
   d4 = bilangan % 10
   d3 = (bilangan % 100) / 10
   d2 = (bilangan % 1000) / 100
   d1 = bilangan / 1000

   if d1 < d2 && d2 < d3 && d3 < d4 {
      teks = "terurut membesar"
   } else if d1 > d2 && d2 > d3 && d3 > d4 {
      teks = "terurut mengecil"
   } else {
      teks = "tidak terurut"
   }
   fmt.Println("Digit pada bilangan", bilangan, teks)
}
```

Output:

```
PS C:\semester1_alpro1\alpro1_week10> go run "c:\semester1_alpro1\alpro1_week10\coso3\coso3.go"
Bilangan: 2489
Digit pada bilangan 2489 terurut membesar
PS C:\semester1_alpro1\alpro1_week10> go run "c:\semester1_alpro1\alpro1_week10\coso3\coso3.go"
Bilangan: 3861
Digit pada bilangan 3861 tidak terurut
PS C:\semester1_alpro1\alpro1_week10> go run "c:\semester1_alpro1\alpro1_week10\coso3\coso3.go"
Bilangan: 9651
Digit pada bilangan 9651 terurut mengecil
```

Deskripsi program:

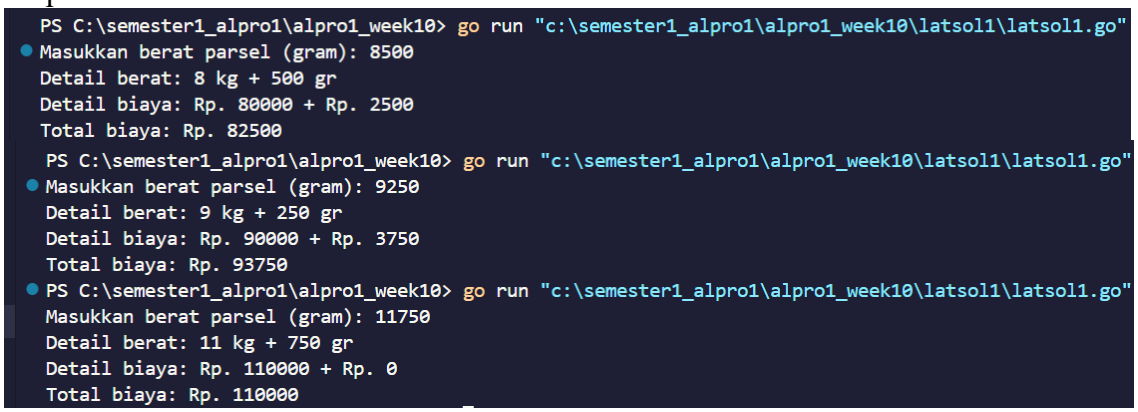
Program di atas adalah program yang ditulis dalam bahasa pemrograman Go, yang berfungsi untuk menganalisis urutan digit dari sebuah bilangan bulat empat digit yang dimasukkan oleh pengguna. Pertama, deklarasikan variabel bilangan, d1, d2, d3, d4 sebagai tipe data integer kemudian deklarasikan variabel sebagai tipe data string. Selanjutnya, program meminta pengguna untuk memasukkan sebuah bilangan 4 digit saja tentunya karena hanya ada empat variabel yang akan menyimpan nilai digit-digit tersebut. Setelah bilangan diinput, program memisahkan digit-digit dari bilangan tersebut menjadi empat variabel: d1, d2, d3, dan d4, yang masing-masing merepresentasikan digit ribuan, ratusan, puluhan, dan satuan. Selanjutnya, program melakukan pemeriksaan terhadap urutan digit tersebut. Jika digit-digit tersebut terurut membesar, maka program akan menyimpan string "terurut membesar" dalam variabel teks. Sebaliknya, jika digit-digit tersebut terurut mengecil, program akan menyimpan string "terurut mengecil". Jika digit-digit tersebut tidak memenuhi kedua kondisi tersebut, program akan menyimpan string "tidak terurut". Akhirnya, program mencetak hasil analisis yang menunjukkan digit-digit dari bilangan yang dimasukkan beserta status urutannya.

LATIHAN SOAL

1. PT POS membutuhkan aplikasi perhitungan biaya kirim berdasarkan berat parsel. Maka, **buatlah program BiayaPos untuk menghitung biaya pengiriman tersebut dengan ketentuan sebagai berikut!**
   Dari berat parsel (dalam gram), harus dihitung total berat dalam kg dan sisanya (dalam gram). Biaya jasa pengiriman adalah Rp. 10.000,- per kg. Jika sisa berat tidak kurang dari 500 gram, maka tambahan biaya kirim hanya Rp. 5,- per gram saja. Tetapi jika kurang dari 500 gram, maka tambahan biaya akan dibebankan sebesar Rp. 15,- per gram. Sisa berat (yang kurang dari 1kg) digratiskan biayanya apabila total berat ternyata lebih dari 10kg.
   Source code:

```
package main

import "fmt"

func main() {

   var berat, beratKg, sisaGram, biayaKg, biayaSisaGram, totalBiaya int

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

   beratKg = berat / 1000
   sisaGram = berat % 1000

   biayaKg = beratKg * 10000

   if sisaGram > 0 {
      if beratKg > 10 {
         biayaSisaGram = 0
      } else if sisaGram >= 500 {
         biayaSisaGram = sisaGram * 5
      } else {
         biayaSisaGram = sisaGram * 15
      }
   }

   totalBiaya = biayaKg + biayaSisaGram

   fmt.Printf("Detail berat: %d kg + %d gr\n", beratKg, sisaGram)
   fmt.Printf("Detail biaya: Rp. %d + Rp. %d\n", biayaKg, biayaSisaGram)
   fmt.Printf("Total biaya: Rp. %d\n", totalBiaya)
}
```

Output:



Deskripsi program:

Program ini adalah sebuah program sederhana yang ditulis dalam bahasa pemrograman Go, yang menghitung biaya pengiriman parsel berdasarkan beratnya dalam gram. Pertama, deklarasikan variabel berat, beratKg, sisaGram, biayaKg, biayaSisaGram, totalBiaya sebagai tipe data integer. Pengguna diminta untuk memasukkan berat parsel, yang kemudian diolah untuk menentukan berat dalam kilogram dan sisa gram. Biaya pengiriman dihitung berdasarkan tarif per kilogram dan sisa gram, dengan ketentuan khusus: jika berat parsel lebih dari 10 kilogram, biaya untuk sisa gram akan menjadi nol; jika sisa gram lebih dari atau sama dengan 500 gram, biaya per gram adalah Rp. 5; dan jika kurang dari 500 gram, biaya per gram adalah Rp. 15. Setelah menghitung biaya untuk kilogram dan sisa gram, program akan menjumlahkan kedua biaya tersebut untuk mendapatkan total biaya pengiriman. Hasil akhir ditampilkan dalam format yang jelas, menunjukkan detail berat, detail biaya, dan total biaya pengiriman.

2. Latihan soal 2

Source code:

```
package main

import "fmt"

func main() {
   var nam float64
   var nmk string
   fmt.Print("Nilai akhir mata kuliah: ")
   fmt.Scan(&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 if nam <= 40 {
      nmk = "E"
   }
   fmt.Println("Nilai mata kuliah: ", nmk)
}
```

Output:

```
PS C:\semester1_alpro1\alpro1_week10> go run "c:\semester1_alpro1\alpro1_week10\latsol2\latsol2.go"
Nilai akhir mata kuliah: 100
Nilai mata kuliah:  A
PS C:\semester1_alpro1\alpro1_week10> go run "c:\semester1_alpro1\alpro1_week10\latsol2\latsol2.go"
Nilai akhir mata kuliah: 75
Nilai mata kuliah:  AB
PS C:\semester1_alpro1\alpro1_week10> go run "c:\semester1_alpro1\alpro1_week10\latsol2\latsol2.go"
Nilai akhir mata kuliah: 70
Nilai mata kuliah:  B
PS C:\semester1_alpro1\alpro1_week10> go run "c:\semester1_alpro1\alpro1_week10\latsol2\latsol2.go"
Nilai akhir mata kuliah: 60
Nilai mata kuliah:  BC
PS C:\semester1_alpro1\alpro1_week10> go run "c:\semester1_alpro1\alpro1_week10\latsol2\latsol2.go"
Nilai akhir mata kuliah: 55
Nilai mata kuliah:  C
PS C:\semester1_alpro1\alpro1_week10> go run "c:\semester1_alpro1\alpro1_week10\latsol2\latsol2.go"
Nilai akhir mata kuliah: 45
Nilai mata kuliah:  D
PS C:\semester1_alpro1\alpro1_week10> go run "c:\semester1_alpro1\alpro1_week10\latsol2\latsol2.go"
Nilai akhir mata kuliah: 35
Nilai mata kuliah:  E
```

Deskripsi program:

Program ini adalah program yang digunakan untuk menentukan nilai huruf (grade) dari nilai akhir mata kuliah berdasarkan kriteria yang telah ditentukan. Program dimulai dengan mendeklarasikan dua variabel, yaitu nam yang bertipe float64 untuk menyimpan nilai akhir mata kuliah, dan nmk yang bertipe string untuk menyimpan hasil konversi nilai menjadi

huruf. Pengguna diminta untuk memasukkan nilai akhir dalam bentuk angka desimal, yang kemudian akan diproses untuk menentukan kategori nilai huruf yang sesuai. Program menggunakan serangkaian kondisi else if untuk mengevaluasi nilai yang dimasukkan: jika nilai lebih dari 80, pengguna akan mendapatkan grade "A"; untuk nilai antara 72.5 hingga 80, grade yang diberikan adalah "AB"; nilai antara 65 hingga 72.5 akan mendapatkan grade "B"; nilai antara 57.5 hingga 65 akan mendapatkan grade "BC"; nilai antara 50 hingga 57.5 akan mendapatkan grade "C"; nilai antara 40 hingga 50 akan mendapatkan grade "D"; dan hingga grade "E" untuk nilai 40 atau kurang. Setelah menentukan grade yang sesuai, program akan menampilkan hasilnya kepada pengguna dalam format yang jelas.

Jawaban soal dari program dibawah ini (program yang salah):

```
package main
import “fmt”
func main() {
   var nam float64
   var nmk string
   fmt.Print(“Nilai akhir mata kuliah: “)
   fmt.Scan(&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)
}
```

a. Jika nam diberikan adalah 80.1, apa keluaran dari program tersebut? Apakah eksekusi program tersebut sesuai spesifikasi soal?
   Keluaran: Program tidak akan berjalan dengan benar, karena terjadi kesalahan dalam penanganan nilai variabel nam dan nmk.
   - Variabel nam yang seharusnya bertipe numerik (float64) diubah ke tipe string tanpa casting yang benar.
   - Nilai nmk tidak pernah diubah, sehingga akan menghasilkan nilai default string kosong.

b. Apa saja kesalahan dari program tersebut? Mengapa demikian? Jelaskan alur program seharusnya!

1. Penanganan tipe data:
    - Variabel nam (bilangan riil) digunakan untuk menyimpan string nilai huruf seperti "A", "B", dll., yang menyebabkan kesalahan tipe data. Variabel ini seharusnya tetap menyimpan angka.
    - Variabel nmk (yang bertipe string) tidak pernah diisi, meskipun dimaksudkan untuk menyimpan hasil nilai huruf.
2. Struktur pengkondisian:
    - Semua kondisi menggunakan pernyataan if independen, sehingga setiap kondisi dievaluasi, meskipun kondisi sebelumnya sudah terpenuhi. Akibatnya, nilai nam akan terus diperiksa hingga kondisi terakhir, mengabaikan urutan prioritas.
3. Kesalahan pencetakan:
    - Variabel nmk tidak diisi, sehingga keluaran tidak sesuai.

Alur program yang benar:

1. Menerima input berupa nam (Nilai Akhir Mata Kuliah) bertipe float64.
2. Menggunakan logika kondisional berbasis urutan nilai nam untuk menentukan nmk (Nilai Mata Kuliah dalam huruf). Prioritas penilaian harus dari yang tertinggi ke yang terendah menggunakan if-else.
3. Menampilkan nilai NMK yang sesuai.

c. Perbaiki program tersebut! Ujilah dengan masukan: 93.5; 70.6; dan 49.5. Seharusnya keluaran yang diperoleh adalah ‘A’, ‘B’, dan ‘D’.
Program tersebut sudah di perbaiki ada di source code.
Output:

```
PS C:\semester1_alpro1\alpro1_week10> go run "c:\semester1_alpro1\alpro1_week10\latsol2\latsol2.go"
Nilai akhir mata kuliah: 93.5
Nilai mata kuliah:  A
PS C:\semester1_alpro1\alpro1_week10> go run "c:\semester1_alpro1\alpro1_week10\latsol2\latsol2.go"
Nilai akhir mata kuliah: 70.6
Nilai mata kuliah:  B
PS C:\semester1_alpro1\alpro1_week10> go run "c:\semester1_alpro1\alpro1_week10\latsol2\latsol2.go"
Nilai akhir mata kuliah: 49.5
Nilai mata kuliah:  D
```

3. Latihan soal 3

Source code:

```
package main

import "fmt"

func main() {
    var b int
    fmt.Print("Bilangan : ")
    fmt.Scan(&b)

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

    fmt.Println()

    if jumlahFaktor == 2 {
        fmt.Println("Prima : True")
    } else {
        fmt.Println("Prima : False")
    }
}
```

Output:

```
PS C:\semester1_alpro1\alpro1_week10> go run "c:\semester1_alpro1\alpro1_week10\latsol3\latsol3.go"
Bilangan : 12
Faktor :  1 2 3 4 6 12
Prima : False
PS C:\semester1_alpro1\alpro1_week10> go run "c:\semester1_alpro1\alpro1_week10\latsol3\latsol3.go"
Bilangan : 7
Faktor :  1 7
Prima : True
```

Deskripsi program:

Program ini adalah program yang ditulis dalam bahasa pemrograman Go, yang digunakan untuk menentukan faktor-faktor dari sebuah bilangan dan mengecek apakah bilangan tersebut merupakan bilangan prima. Program ini tidak hanya memberikan daftar faktor dari bilangan yang dimasukkan, tetapi juga memberikan informasi tambahan mengenai sifat prima dari bilangan tersebut. Pertama, deklarasikan variabel b sebagai tipe data integer. Pengguna diminta untuk memasukkan sebuah bilangan bulat, yang kemudian akan diproses untuk mencari semua faktor dari bilangan tersebut. Program ini menggunakan loop untuk memeriksa setiap angka dari 1 hingga bilangan yang dimasukkan, dan jika angka tersebut merupakan faktor (yaitu jika bilangan tersebut habis dibagi oleh angka tersebut), maka angka itu akan ditampilkan sebagai faktor dan jumlah faktor akan dihitung. Setelah semua faktor ditampilkan, program akan mengevaluasi jumlah faktor yang ditemukan. Jika jumlah faktor sama dengan 2, program akan mengkonfirmasi bahwa bilangan tersebut adalah

bilangan prima; sebaliknya, program akan menyatakan bahwa bilangan tersebut bukan bilangan prima.

DAFTAR PUSTAKA

A.13. Seleksi Kondisi. (n.d.). Retrieved from https://dasarpemrogramangolang.novalagung.com/A-seleksi-kondisi.html

B.6. Template: Actions & Variables. (n.d.). Retrieved from https://dasarpemrogramangolang.novalagung.com/B-template-actions-variables.html