

LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 1
MODUL 10
“ELSE-IF”



DISUSUN OLEH:
ANASTASIA ADINDA NARENDRA INDRIANTO
103112400085
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

1. Konsep Input dan Output

Input merupakan intruksi dasar untuk membaca data yang diberikan dari pengguna. Data yang diberikan oleh pengguna akan disimpan ke dalam suatu wadah yang disebut *variable*. Penulisan intruksi input beragam menyesuaikan Bahasa pemrograman yang digunakan. Output merupakan perintah untuk menampilkan data ke layar monitor. Data yang sudah diproses atau diolah oleh program komputer perlu ditampilkan ke layar sehingga pengguna bisa memperoleh informasi dari hasil pengolahan data yang dilakukan program.

2. Konsep Data, Variabel, dan Intruksi Dasar

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 adalah integer, real, Boolean, karakter dan string. Nilai data yang tersimpan dalam variabel dapat diperoleh dengan menyebutkan langsung nama variabelnya. Informasi Alamat atau Lokasi variabel dapat diperoleh dengan menambahkan prefix dan di depan nama variabel tersebut.

3. Konsep Bahasa Pemrograman Go

Bahasa Go menganut kesesuaian tipe data yang ketat. Tipe data yang berbeda tidak boleh dicampur dalam satu ekspresi, bahkan tipe data masih yang sejenis. Menyesuaikan tipe data ada beberapa cara yaitu *casting* tipe (data) mengubah tipe dari data yang diberikan ke tipe data yang diinginkan, memanfaatkan fungsi *Sprint* dan *Sscan* dari paket *fmt*, dan memanfaatkan fungsi-fungsi dalam paket *strconv*, seperti *Atoi*, *Itoa* dan *ParseBool*. Variabel harus dideklarasikan terlebih dahulu sebelum digunakan. Variabel juga harus diinisialisasi dulu agar nilai yang tersimpan diketahui dengan jelas dan eksekusi algoritma menjadi terprediksi. Dalam Bahasa Go, variabel yang tidak diinisialisasi lebih dahulu otomatis diisi dengan nilai default ekuivalen dengan bit 0.

4. Konsep Tipe Data

Melalui ilmu komputer tentunya kita sering mendengar istilah data. Data terbagi menjadi bermacam-macam tipe data yang terklasifikasi dan memiliki fungsi sendiri. Tentunya, berbagai macam tipe data tersebut sangat bermanfaat bagi kinerja komputer melalui kode-kode dalam bahasa pemrograman. Data types atau tipe data adalah sebuah pengklasifikasian data berdasarkan jenis data tersebut. Tipe data dibutuhkan agar kompiler dapat mengetahui bagaimana sebuah data akan digunakan. Untuk mengembangkan sebuah program ada beberapa tipe data yang dibutuhkan. Tipe data memiliki 4 data type yang sering digunakan yaitu,

- 1) Bilangan Bulat (Integer) Tipe bilangan bulat (Integer) adalah tipe data numerik yang biasa digunakan apabila bertemu dengan bilangan bulat, seperti 1, 27, 100, dll. Bilangan ini juga mengenal nilai positif dan negatif. Tipe data numerik yang termasuk ke dalam bilangan bulat adalah *byte*, *short*, *int*, dan *long*.
- 2) Bilangan Pecahan (Floating Point) Tipe bilangan pecahan atau *floating point* adalah bilangan yang menangani bilangan desimal atau perhitungan secara detail. Karena kemampuannya, float point berbanding terbalik dengan integer. Terdapat dua tipe pada bilangan pecahan ini yaitu *float* dan *double*.
- 3) Karakter (Char)
Tipe data karakter tunggal yang biasa didefinisikan dengan tanda petik (') di awal dan di akhir karakternya. Tipe ini mengikuti aturan “unicode” sehingga bilangan harus

diawali kode “/u”. Tetapi juga biasa menggunakan bilangan heksadesimal dari 0000 sampai FFFF.

4) Boolean

Tipe data *boolean* merupakan tipe yang memiliki dua nilai yaitu benar (*true*) atau salah (*false*). Nilai yang digunakan pada tipe ini sangat penting dalam mengambil keputusan suatu kejadian tertentu.

5. **Deklarasi dan Inisialisasi**

Deklarasi variabel dalam bahasa pemrograman komputer adalah pernyataan yang digunakan untuk menentukan nama variabel dan tipe datanya. Deklarasi memberi tahu kompiler tentang keberadaan entitas dalam program dan lokasinya. Saat Anda mendeklarasikan variabel, Anda juga harus menginisialisasinya. Inisialisasi adalah proses pemberian nilai pada Variabel. Setiap bahasa pemrograman memiliki metode tersendiri untuk menginisialisasi variabel. Jika nilai tidak diberikan pada Variabel, maka proses tersebut hanya disebut Deklarasi.

6. **Konstanta dalam Bahasa Pemrograman**

Konstanta adalah entitas yang merujuk pada nilai data yang tetap dan tidak dapat diubah. Selama eksekusi/perhitungan pemrograman, nilai suatu konstanta tidak dapat diubah, tetap konstan. Jenis Konstanta ada konstanta Bilangan Bulat. Konstanta bilangan bulat adalah barisan bilangan bulat yang nilainya tetap. Mereka tidak boleh mengandung koma desimal atau angka pecahan. Konstanta bilangan bulat dapat berupa bilangan positif atau negatif. Mereka termasuk bilangan bulat sistem desimal, bilangan bulat sistem oktal, bilangan bulat sistem heksadesimal.

7. **Integer Division dan Modulo**

Pembagian pada tipe data integer (**integer division** atau **div**) sedikit berbeda dengan pembagian yang sudah sering kita pelajari dari sekolah dasar. Hasil pembagian akan bertipe data integer, artinya kita akan mengabaikan bilangan yang muncul setelah tanda koma atau floating point. Hasil dari operasi div ini dikenal dengan istilah **quotient**. Modulo atau modulus "mod" adalah operasi untuk mencari nilai integer dari sisa pembagian pada integer division, biasanya dikenal juga dengan istilah remainder. Misalnya untuk operasi 10 dibagi dengan 3, maka sisa pembagian adalah 1. Operator modulo menggunakan "%" di dalam bahasa pemrograman Go.

8. **Casting atau Konversi Tipe Data**

Pada Bahasa pemrograman Go, tipe data bersifat statis, artinya tipe data yang sudah didefinisikan tidak dapat diganti selama program berjalan. Casting merupakan salah satu Teknik konversi antar tipe data di dalam Bahasa pemrograman.

9. **Paradigma Perulangan**

Perulangan merupakan salah satu struktur kontrol yang memungkinkan suatu instruksi yang sama dilakukan berulang kali dalam waktu atau jumlah yang lama. Tanpa instruksi perulangan, maka suatu instruksi akan ditulis dalam jumlah yang sangat banyak. Salah satu instruksi perulangan yang paling mudah adalah **for-loop**, yang mana dengan instruksi ini dapat digunakan untuk mengulangi instruksi sebanyak **n** kali (iterasi). Batasan besar nilai dari **n** menyesuaikan dengan batasan dari tipe data integer yang digunakan.

10. **Paradigma Percabangan if-else**

Percabangan if-else adalah cara untuk menjalankan kode yang berbeda tergantung pada apakah kondisi yang diberikan bernilai benar (*true*) atau salah (*false*). Bentuk paling sederhana dari percabangan ini adalah dengan menggunakan if untuk memeriksa kondisi, artinya kode program dari baris ke-1 hingga baris terakhir akan dieksekusi satu persatu.

CONTOH SOAL

1. Contoh Latihan Soal 1

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 D:\Semester1\Semester 1 APro\Go\WEEK10LAPRAK> go run "d:\Semester1\Semester 1 APro\Go\WEEK10LAPRAK\lcoso.go"
17
true
bisa membuat KTP
PS D:\Semester1\Semester 1 APro\Go\WEEK10LAPRAK> go run "d:\Semester1\Semester 1 APro\Go\WEEK10LAPRAK\lcoso.go"
20
false
belum bisa membuat KTP
PS D:\Semester1\Semester 1 APro\Go\WEEK10LAPRAK> go run "d:\Semester1\Semester 1 APro\Go\WEEK10LAPRAK\lcoso.go"
15
true
belum bisa membuat KTP
```

Deskripsi Program:

Program lcoso.go dibuat untuk menentukan apakah seorang penduduk bisa membuat KTP atau tidak. Syarat utama membuat KTP adalah berusia minimal 17 tahun dan memiliki kartu keluarga. Input terdiri dari dua baris. Baris pertama adalah bilangan bulat yang menyatakan usia, sedangkan baris kedua adalah boolean yang menyatakan memiliki KTP atau tidak. Output berupa teks "bisa membuat KTP" atau "belum bisa membuat KTP" sesuai dengan masukan yang diberikan. User dapat menginputkan data yang diinginkan didalam bagian terminal setelah program dijalankan dengan runner program. Berikut input dan output;

No	Masukan	Keluaran
1	17 true	bisa membuat KTP
2	20 false	belum bisa membuat KTP
3	15 true	belum bisa membuat KTP

2. Contoh Latihan Soal 2

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 D:\Semester1\Semester 1 APro\Go\WEEK10LAPRAK> go run "d:\Semester1\Semester 1 APro\Go\WEEK10LAPRAK\2coso.go"
A
vokal
PS D:\Semester1\Semester 1 APro\Go\WEEK10LAPRAK> go run "d:\Semester1\Semester 1 APro\Go\WEEK10LAPRAK\2coso.go"
f
konsonan
PS D:\Semester1\Semester 1 APro\Go\WEEK10LAPRAK> go run "d:\Semester1\Semester 1 APro\Go\WEEK10LAPRAK\2coso.go"
1
bukan huruf
PS D:\Semester1\Semester 1 APro\Go\WEEK10LAPRAK> go run "d:\Semester1\Semester 1 APro\Go\WEEK10LAPRAK\2coso.go"
$
bukan huruf
```

Deskripsi Program:

Program 2coso.go dibuat dengan menentukan suatu alfabet yang diberikan adalah vokal atau konsonan. Input terdiri dari suatu huruf. Output berupa teks yang menyatakan huruf adalah "vokal" atau "konsonan" atau "bukan huruf". User dapat menginputkan data yang diinginkan didalam bagian terminal setelah program dijalankan dengan runner program. Berikut input dan ouput;

No	Masukan	Keluaran
1	A	vokal
2	f	konsonan
3	1	bukan huruf
4	\$	bukan huruf

3. Contoh Latihan Soal 3

Source Code:

```
package main

import "fmt"

func main() {
    var bilangan, d1, d2, d3, d4 int
    var teks string
    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 D:\Semester1\Semester 1 APro\Go\WEEK10LAPRAK> go run "d:\Semester1\Semester 1 APro\Go\WEEK10LAPRAK\3coso.go"
2489
Digit pada bilangan 2489 terurut membesar
PS D:\Semester1\Semester 1 APro\Go\WEEK10LAPRAK> go run "d:\Semester1\Semester 1 APro\Go\WEEK10LAPRAK\3coso.go"
3861
Digit pada bilangan 3861 tidak terurut
PS D:\Semester1\Semester 1 APro\Go\WEEK10LAPRAK> go run "d:\Semester1\Semester 1 APro\Go\WEEK10LAPRAK\3coso.go"
9651
Digit pada bilangan 9651 terurut mengecil
```

Deskripsi Program:

Program 3coso.go dibuat untuk Sapakah digit dalam suatu bilangan terurut membesar, mengecil, atau tidak terurut. Bilangan hanya terdiri dari empat digit saja, atau lebih besar atau sama dengan 1000 dan kecil sama atau sama dengan 9999. User dapat menginputkan data yang diinginkan didalam bagian terminal setelah program dijalankan dengan runner program. Berikut input dan ouput;

No.	Contoh masukan dan keluaran
1	Bilangan: <u>2489</u> Digit pada bilangan 2489 terurut membesar
2	Bilangan: <u>3861</u> Digit pada bilangan 3861 tidak terurut
3	Bilangan: <u>9651</u> Digit pada bilangan 9651 terurut mengecil

SOAL LATIHAN

1. Latihan Soal 1

Source Code:

```
package main

import "fmt"

func main() {

    //untuk input
    var berat int
    fmt.Print("Berat parsel (gram): ")
    fmt.Scan(&berat)

    //perhitungan berat dalam kg dan sisa gram
    kg := berat / 1000
    sisa := berat % 1000

    //biaya dalam kg
    biayaPerKg := 10000
    biayaTambahan := 0

    //kondisi untuk sisa berat
    if kg > 10 {
        biayaTambahan = 0
    } else if sisa >= 500 {
        biayaTambahan = sisa * 5
    } else {
        biayaTambahan = sisa * 15
    }

    //hitung totl biaya
    total := (kg * biayaPerKg) + biayaTambahan

    //menampilkan hasil
    fmt.Printf("Detail berat: %d kg + %d gr\n", kg, sisa)
    fmt.Printf("Detail biaya: Rp. %d + Rp. %d\n", kg*biayaPerKg, biayaTambahan)
    fmt.Printf("Total biaya: Rp. %d\n", total)
}
```

Output:

```
PS D:\Semester1\Semester 1 AlPro\Go\WEEK10LAPRAK> go run "d:\Semester1\Semester 1 AlPro\Go\WEEK10LAPRAK\Soal1.go"
Berat parcel (gram): 8500
Detail berat: 8 kg + 500 gr
Detail biaya: Rp. 80000 + Rp. 2500
Total biaya: Rp. 82500
PS D:\Semester1\Semester 1 AlPro\Go\WEEK10LAPRAK> go run "d:\Semester1\Semester 1 AlPro\Go\WEEK10LAPRAK\Soal1.go"
Berat parcel (gram): 9250
Detail berat: 9 kg + 250 gr
Detail biaya: Rp. 90000 + Rp. 3750
Total biaya: Rp. 93750
PS D:\Semester1\Semester 1 AlPro\Go\WEEK10LAPRAK> go run "d:\Semester1\Semester 1 AlPro\Go\WEEK10LAPRAK\Soal1.go"
Berat parcel (gram): 11750
Detail berat: 11 kg + 750 gr
Detail biaya: Rp. 110000 + Rp. 0
Total biaya: Rp. 110000
```

Deskripsi Program:

Program Soal1.go dibuat dengan tujuan untuk menghitung biaya pengiriman dengan ketentuan Dari berat parcel (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. Berikut input dan ouput;

No.	Contoh masukan dan keluaran
1	Berat parcel (gram): 8500 Detail berat: 8 kg + 500 gr Detail biaya: Rp. 80000 + Rp. 2500 Total biaya: Rp. 82500
2	Berat parcel (gram): 9250 Detail berat: 9 kg + 250 gr Detail biaya: Rp. 90000 + Rp. 3750 Total biaya: Rp. 93750
3	Berat parcel (gram): 11750 Detail berat: 11 kg + 750 gr Detail biaya: Rp. 110000 + Rp. 3750 Total biaya: Rp. 110000

2. Latihan Soal 2

Source Code:

```
package main

import "fmt"

func main() {
    var nam float64

    fmt.Print("Nilai Akhir Mata Kuliah: ")
    fmt.Scan(&nam)

    if nam <= 40 {
        fmt.Println("E")
    } else if nam > 40 && nam <= 50 {
        fmt.Println("D")
    } else if nam > 50 && nam <= 57.5 {
        fmt.Println("C")
    } else if nam > 57.5 && nam <= 65 {
        fmt.Println("BC")
    } else if nam > 65 && nam <= 72.5 {
        fmt.Println("B")
    } else if nam > 72.5 && nam <= 80 {
        fmt.Println("AB")
    } else if nam > 80 {
        fmt.Println("A")
    }
}
```

Output:

```
PS D:\Semester1\Semester 1 AlPro\Go\WEEK10LAPRAK> go run "d:\Semester1\Semester 1 AlPro\Go\WEEK10LAPRAK\Soal2.go"
Nilai Akhir Mata Kuliah: 80.1
A
PS D:\Semester1\Semester 1 AlPro\Go\WEEK10LAPRAK> go run "d:\Semester1\Semester 1 AlPro\Go\WEEK10LAPRAK\Soal2.go"
Nilai Akhir Mata Kuliah: 93.5
A
PS D:\Semester1\Semester 1 AlPro\Go\WEEK10LAPRAK> go run "d:\Semester1\Semester 1 AlPro\Go\WEEK10LAPRAK\Soal2.go"
Nilai Akhir Mata Kuliah: 70.6
B
PS D:\Semester1\Semester 1 AlPro\Go\WEEK10LAPRAK> go run "d:\Semester1\Semester 1 AlPro\Go\WEEK10LAPRAK\Soal2.go"
Nilai Akhir Mata Kuliah: 49.5
D
```

Deskripsi Program:

Program 2Soal.go dibuat dengan tujuan mengetahui nilai akhir mata kuliah (NAM) [0..100] dan standar penilaian nilai mata kuliah (NMK). Jika nilai lebih dari 80 maka akan mendapatkan 'A', jika nilai lebih dari 72.5 maka akan mendapatkan 'AB', jika nilai lebih dari 65 maka akan mendapatkan 'B', jika nilai lebih dari 57.5 maka akan mendapatkan 'BC', jika nilai lebih dari 50 maka akan mendapatkan 'C', jika nilai lebih dari 40 maka akan mendapatkan 'D', dan jika nilai kurang dari sama dengan 40 maka akan mendapatkan 'E'.

Pada soal 2 diberikan pertanyaan karena hasil awal codingan awal didalam soal salah.
Jawaban :

- a. Input nam dimasukan angka 80.1 menghasilkan output A yang sudah sesuai spesifikasi soal, dimana A hanya bisa didapatkan jika nilai yang dimasukan lebih dari 80.
- b. Kesalahan yang ada didalam program awal sebelum dibetulkan adalah tidak perlu memasukan variabel nmk dan tipe data string karena bisa menggunakan format print. Kesalahan lainnya adalah terlalu banyak if yang tidak sesuai dengan standar aturan if-else, seharusnya jika ingin memasukan banyak if maka setelah membuat if maka dilanjutkan dengan else if dan ditutup dengan else.
- c. Input nam dengan angka 93.5, 70.6, 49.5 mengeluarkan hasil Output 'A' untuk 93.5, 'B' untuk 70.6, dan 'C' untuk 49.5.

3. Latihan Soal 3

Source Code:

```
package main

import "fmt"

func main() {

    var bil int

    //bikin input
    fmt.Print("Bilangan:")
    fmt.Scan(&bil)

    //rumus mencari faktor bilangan
    fmt.Print("Faktor:")
    jumlahFaktor := 0
    for i := 1; i <= bil; i++ {
        if bil%i == 0 {
            fmt.Print(i, " ")
            jumlahFaktor++
        }
    }
    fmt.Println()

    isPrima := jumlahFaktor == 2
    //output hasil

    fmt.Println("prima", isPrima)
}
```

Output:

```
PS D:\Semester1\Semester 1 A1Pro\Go\WEEK10LAPRAK> go run "d:\Semester1\Semester 1 A1Pro\Go\WEEK10LAPRAK\Soal3.go"
Bilangan:12
Faktor:1234612
prima false
PS D:\Semester1\Semester 1 A1Pro\Go\WEEK10LAPRAK> go run "d:\Semester1\Semester 1 A1Pro\Go\WEEK10LAPRAK\Soal3.go"
Bilangan:7
Faktor:17
prima true
```

Deskripsi Program:

Program 3Soal.go dibuat dengan Program harus dapat mencari dan menampilkan semua faktor dari bilangan 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. User dapat menginputkan data yang diinginkan didalam bagian terminal setelah program dijalankan dengan runner program. Berikut input dan ouput;

Bilangan: 12	Bilangan: 7
Faktor: 1 2 3 4 6 12	Faktor: 1 7
Prima: false	Prima: true

DAFTAR PUSTAKA

<file:///D:/Semester%201%20AIPro/Modul/MODUL%202.pdf>

<https://www.toppr.com/guides/computer-science/introduction-to-c/data-types-variables-and-constants/declaration-of-variables/>

<https://www.toppr.com/guides/computer-science/introduction-to-c/data-types-variables-and-constants/constants-in-programming-language/>

<https://www.dicoding.com/blog/macam-macam-tipe-data/>

<file:///D:/Semester%201%20AIPro/Modul/MODUL%203.pdf>

<file:///D:/Semester1/Semester%201%20AIPro/Modul/MODUL%205%20dan%206.pdf>

<file:///D:/Semester1/Semester%201%20AIPro/WEEKMATERI/MODUL%209.pdf>