

LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 1
MODUL II
“TIPE DATA & VARIABEL”



Disusun oleh :
NAMA : Felix Pedrosa Valentino
NIM : 103112400056
S1 IF-12-01
Dosen Pengampu :
Yohani Setiya Rafika Nur, M. Kom.

PROGRAM STUDI S1 TEKNIK INFORMATIKA
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2024/2025

BAB I

TUJUAN PRAKTIKUM

1. Mahasiswa dapat memahami beberapa tipe data dasar (jenis dan jangkauannya)
2. Mahasiswa dapat memahami variabel
3. Mahasiswa dapat memahami konstanta
4. Mahasiswa dapat memahami berbagai jenis operator dan pemakaiannya
5. Mahasiswa dapat memahami instruksi Input dan Output

BAB II DASAR TEORI

1.) Input dan Output (I/O)

Salah satu Instruksi dasar di dalam pemrograman adalah input/scan/read dan output/print/write.

a.) Input

Merupakan instruksi dasar untuk membaca data yang diberikan dari pengguna. Data yang diberikan oleh pengguna akan disimpan ke dalam suatu wadah yang disebut variabel. Data ini untuk selanjutnya akan diproses oleh program komputer. Pembacaan data ini tentunya memerlukan perangkat/device tambahan yang terhubung oleh komputer. Misalnya: keyboard, mouse, layar sentuh, dan lainnya. Informasi yang diketik oleh pengguna akan disimpan sebagai data komputer di dalam variabel. Penulisan instruksi input beragam menyesuaikan bahasa pemrograman yang digunakan (misalnya scan atau read).

b.) Output

Berbeda dengan Input, maka Output merupakan perintah untuk menampilkan data ke layar monitor* (bisa dari perangkat komputer ataupun mobile). Data yang sudah diproses atau diolah oleh program komputer perlu ditampilkan ke layar sehingga pengguna bisa memperoleh informasi dari hasil pengolahan data yang dilakukan oleh program.

Kata kunci instruksi yang digunakan juga beragam, menyesuaikan dengan bahasa pemrograman yang digunakan. Contohnya write dan print.

Catatan*: Pada konteks ini, perangkat untuk menampilkan data dipersempit menjadi layar komputer, tujuannya untuk membuat mahasiswa lebih mudah memahami instruksi output

2.) Data, Variabel, dan Instruksi Dasar

Variabel adalah nama dari suatu lokasi di memori, yang data dengan tipe tertentu dapat disimpan.

a.) Nama variabel dimulai dengan huruf dan dapat diikuti dengan sejumlah huruf, angka, atau garisbawah.

Contoh: ketemu, found, rerata, mhs1, data_2, ...

b.) Tipe data yang umum tersedia adalah integer, real, boolean, karakter, dan string.

c.) **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.

d.) **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**.

- e.) 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=).
- f.) 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 **Sprintf** dan **Scan** dari paket **fmt**.
 - ✓ Memanfaatkan fungsi-fungsi dalam paket **strconv**, seperti **Atoi**, **Itoa**, dan **ParseBool**. Lihat lampiran untuk contoh penggunaan.
- g.) 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 dahulu otomatis diisi dengan nilai default yang ekuivalen dengan bit 0.
 - ✓ Nilai 0 untuk bilangan integer
 - ✓ 0.0E+0 untuk bilangan real
 - ✓ **false** untuk boolean
 - ✓ Karakter NUL (lihat tabel ASCII) untuk karakter
 - ✓ "" (string kosong, string dengan panjang 0) untuk string
 - ✓ **nil** untuk alamat memori
- h.) Variabel bisa dianalogikan sebagai wadah yang akan digunakan untuk menyimpan data.

3.) Konstanta Simbolik

Konstanta dapat diberi nama untuk memudahkan mengingat maksud dan manfaat dari nilai yang diberi nama tersebut. Seperti **PI** untuk merepresentasikan konstanta π .

4.) Tipe Data Character

Tipe data character pada dasarnya merupakan tipe data integer, tetapi digunakan untuk merepresentasikan suatu simbol tertentu berdasarkan tabel acuan ASCII/UTF-8 atau UTF-16. Simbol pada tipe data character ditulis dengan menambahkan petik satu, Contohnya bilangan

64 dapat ditulis dengan simbol '@' (dengan petik) dan bilangan 97 dapat ditulis dengan simbol

'a'.

BAB III

CONTOH SOAL

1.) Contoh Soal 1

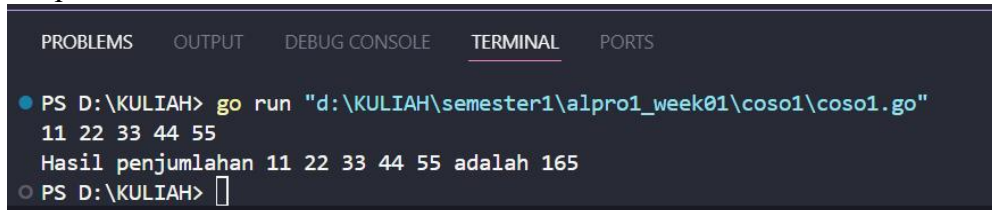
Source Code :

```
package main

import "fmt"

func main() {
    var a, b, c, d, e int
    var hasil int
    fmt.Scanln(&a, &b, &c, &d, &e)
    hasil = a + b + c + d + e
    fmt.Println("Hasil penjumlahan", a, b, c, d, e, "adalah", hasil)
}
```

Output :



```
PS D:\KULIAH> go run "d:\KULIAH\semester1\alpro1_week01\coso1\coso1.go"
11 22 33 44 55
Hasil penjumlahan 11 22 33 44 55 adalah 165
PS D:\KULIAH>
```

Deskripsi Program :

Program di atas adalah sebuah program dalam bahasa Go yang menerima lima input angka dari pengguna, kemudian menjumlahkan kelima angka tersebut dan menampilkan hasilnya.

Penjelasan :

1.) Deklarasi Variabel :

Program ini mendeklarasikan enam variabel integer: `a`, `b`, `c`, `d`, `e` untuk menyimpan input dari pengguna, dan `hasil` untuk menyimpan hasil penjumlahan.

2.) Input dari Pengguna :

Fungsi `fmt.Scanln` digunakan untuk menerima lima input integer secara bersamaan dari pengguna. Input ini disimpan dalam variabel `a`, `b`, `c`, `d`, dan `e`.

3.) Penjumlahan :

Nilai dari variabel `a`, `b`, `c`, `d`, dan `e` dijumlahkan, dan hasilnya disimpan dalam variabel `hasil`.

4.) Output :

Fungsi `fmt.Println` digunakan untuk mencetak kalimat yang menampilkan hasil penjumlahan beserta kelima angka yang dijumlahkan.

Contoh Input dan Output :

- Input: 1 2 3 4 5

- Output: "Hasil penjumlahan 1 2 3 4 5 adalah 15"

Kesimpulan :

Program ini berfungsi untuk menjumlahkan lima angka yang dimasukkan oleh pengguna dan memberikan hasil penjumlahan dalam bentuk yang mudah dipahami.

2.) Contoh Soal 2

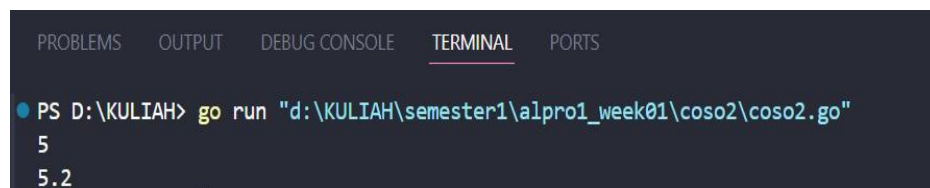
Source Code :

```
package main

import "fmt"

func main() {
    // fx = 2 / (x + 5) + 5
    // masukkan input x
    var x, fx float32
    fmt.Scan(&x)
    fx = 2/(x+5) + 5
    fmt.Print(fx)
}
```

Output :



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS D:\KULIAH> go run "d:\KULIAH\semester1\alpro1_week01\coso2\coso2.go"
5
5.2
```

Deskripsi Program :

Program di atas adalah sebuah program dalam bahasa Go yang menghitung nilai dari sebuah fungsi matematika berdasarkan input pengguna. Fungsi matematika yang digunakan adalah

$$f(x) = \frac{2}{x+5} + 5$$

Penjelasan :

1. Deklarasi Variabel :

Program mendeklarasikan dua variabel bertipe float32 : x untuk menerima input dari pengguna dan fx untuk menyimpan hasil perhitungan fungsi $f(x)$.

2. Input dari Pengguna :

Fungsi `fmt.Scan` digunakan untuk menerima satu input berupa bilangan desimal dari pengguna yang disimpan dalam variabel `x`.

3. Perhitungan Fungsi :

Program kemudian menghitung nilai dari fungsi $f(x) = \frac{2}{x+5} + 5$

berdasarkan nilai input `'x'` yang dimasukkan pengguna, dan hasilnya disimpan dalam variabel `'fx'`.

4. Output :

Hasil perhitungan fungsi $f(x)$ dicetak ke layar menggunakan fungsi `'fmt.Print'`.

Contoh Input dan Output:

- Input: 3

- Perhitungan : $f(3) = \frac{2}{3+5} + 5 = 5.25$

- Output: 5.25

Kesimpulan :

Program ini digunakan untuk menghitung nilai fungsi matematika $f(x)$ berdasarkan input pengguna dan menampilkan hasilnya.

3.) Contoh Soal 3

Source code :

```
package main

import "fmt"

func main() {
    // 1. input 5 buah byte, output 5 buah karakter
    var c1, c2, c3, c4, c5 byte
    fmt.Scan(&c1, &c2, &c3, &c4, &c5)
    fmt.Printf("%c%c%c%c%c", c1, c2, c3, c4, c5)
    fmt.Println()
    fmt.Scanln()
    // 2. input 3 buah int, output karakter setelahnya
    var b1, b2, b3 int
    fmt.Scanf("%c%c%c", &b1, &b2, &b3)
    fmt.Printf("%c%c%c", b1+1, b2+1, b3+1)
}
```

Output :

```
PS D:\KULIAH> go run "d:\KULIAH\semester1\alpro1_week01\coso3\coso3.go"
66 97 103 117 115
Bagus
SNO
TOP
```

Deskripsi Program :

Program ini merupakan program dalam bahasa Go yang menerima input berupa beberapa byte dan integer dari pengguna, lalu mengubahnya menjadi karakter dan menampilkannya ke layar. Program ini memiliki dua bagian utama yang melakukan operasi pada input byte dan integer.

Penjelasan:

1. Bagian 1 - Input 5 Byte, Output 5 Karakter :

- Lima variabel bertipe byte yaitu `c1`, `c2`, `c3`, `c4`, dan `c5` dideklarasikan.
- Program menggunakan fungsi `fmt.Scan` untuk menerima lima input byte dari pengguna.
- Input byte tersebut kemudian ditampilkan sebagai karakter dengan menggunakan fungsi `fmt.Printf` dengan format `%c`, yang mengonversi byte menjadi karakter berdasarkan kode ASCII.
- Misalnya, jika pengguna memasukkan angka-angka yang sesuai dengan kode ASCII, karakter yang sesuai akan ditampilkan.

2. Bagian 2 - Input 3 Integer, Output Karakter Setelahnya :

- Tiga variabel bertipe int yaitu `b1`, `b2`, dan `b3` dideklarasikan untuk menyimpan input integer dari pengguna.
- Fungsi `fmt.Scanf` digunakan untuk membaca tiga input integer yang dianggap sebagai karakter ASCII, di mana input dimasukkan dalam format karakter (`%c`).
- Setelah input diterima, program menambah nilai 1 pada setiap input integer dan menampilkannya sebagai karakter menggunakan `fmt.Printf` dengan format `%c`. Hal ini menggeser karakter ke kode ASCII berikutnya.
- Sebagai contoh, jika pengguna memasukkan 'A', maka hasilnya akan menjadi 'B' karena nilai ASCII untuk 'A' adalah 65, dan setelah ditambah 1 akan menjadi 66, yang merupakan 'B'.

Kesimpulan :

Program ini mengambil input berupa byte dan integer yang dikonversi menjadi karakter berdasarkan kode ASCII. Pada bagian pertama, lima input byte ditampilkan sebagai karakter. Pada bagian kedua, tiga input karakter diubah ke karakter ASCII berikutnya dengan menambah nilai integer sebesar 1.

BAB IV

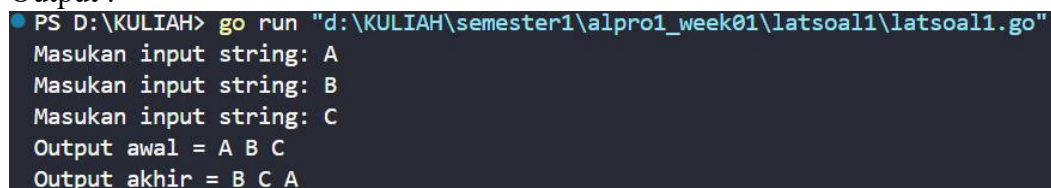
SOAL LATIHAN

1.) Latihan Soal 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 :



```
PS D:\KULIAH> go run "d:\KULIAH\semester1\alpro1_week01\latsoal1\latsoal1.go"
Masukan input string: A
Masukan input string: B
Masukan input string: C
Output awal = A B C
Output akhir = B C A
```

Deskripsi program :

Program Go di atas bertujuan untuk meminta tiga input string dari pengguna sekaligus, lalu menukar posisi string tersebut, dan akhirnya menampilkan hasil sebelum dan sesudah pertukaran.

Penjelasan Program:

1. Deklarasi Variabel :

```
var (
    satu, dua, tiga string
    temp      string
)
```

Program diatas mendeklarasikan empat variabel bertipe string, yaitu satu, dua, tiga, dan temp. Variabel satu, dua, dan tiga digunakan untuk menyimpan input dari pengguna, sementara temp digunakan sebagai variabel sementara untuk membantu pertukaran nilai.

2.Membaca Input :

```
fmt.Print("Masukan input string: ")
fmt.Scanln(&satu)
fmt.Print("Masukan input string: ")
fmt.Scanln(&dua)
fmt.Print("Masukan input string: ")
fmt.Scanln(&tiga)
```

Program meminta pengguna memasukkan tiga string secara berturut-turut. String pertama akan disimpan di variabel satu, string kedua di variabel dua, dan string ketiga di variabel tiga

3.Mencetak Output Awal :

```
fmt.Println("Output awal = " + satu + " " + dua + " " + tiga)
```

Program mencetak ketiga string yang dimasukkan dalam urutan yang sama seperti saat dimasukkan oleh pengguna.

4.Pertukaran Nilai :

```
temp = satu
satu = dua
dua = tiga
tiga = temp
```

Pada bagian ini, program menukar nilai dari ketiga variabel:

- Nilai variabel satu disimpan sementara di temp.
- Nilai variabel dua dipindahkan ke satu.
- Nilai variabel tiga dipindahkan ke dua.
- Nilai dari temp (yang berisi nilai awal dari satu) dipindahkan ke tiga.

5.Mencetak Output Akhir :

```
fmt.Println("Output akhir = " + satu + " " + dua + " " + tiga)
```

Setelah pertukaran, program mencetak kembali ketiga string dengan urutan yang baru.

Kesimpulan :

Program ini berfungsi untuk menerima tiga input string dari pengguna sekaligus, menampilkan urutan awal string tersebut, kemudian menukar urutannya dengan memindahkan string pertama ke posisi terakhir, string kedua ke posisi pertama, dan string ketiga ke posisi kedua, lalu menampilkan hasil akhirnya.

2.) Latihan Soal 2

Source code :

```
package main

import "fmt"

func main() {

var (

nama, nim, kelas, prodi string

)

// Meminta input dari pengguna

fmt.Print("Masukan Nama: ")

fmt.Scanln(&nama)

fmt.Print("Masukan NIM: ")

fmt.Scanln(&nim)

fmt.Print("Masukan Kelas: ")

fmt.Scanln(&kelas)

fmt.Print("Masukan Prodi: ")

fmt.Scanln(&prodi)

// Menampilkan biodata dalam format resume

fmt.Println("\n=== Resume Mahasiswa ===")

fmt.Println("Nama : " + nama)

fmt.Println("NIM : " + nim)

fmt.Println("Kelas : " + kelas)

fmt.Println("Prodi : " + prodi)

fmt.Println("=====")

fmt.Printf("Perkenalkan saya adalah %s, salah satu mahasiswa Prodi %s dari
```

```

    kelas %s dengan NIM %s.\n", nama, prodi, kelas, nim)

    fmt.Println("Selamat belajar, semoga sukses dalam studi Anda!")

}

```

Output :

```

PS D:\KULIAH> go run "d:\KULIAH\semester1\alpro1_week01\soal2\soal2.go"
Masukan Nama: Burhan
Masukan NIM: 1029384750
Masukan Kelas: IF-69-GAB
Masukan Prodi: S1-IF

=== Resume Mahasiswa ===
Nama : Burhan
NIM : 1029384750
Kelas : IF-69-GAB
Prodi : S1-IF
=====
Perkenalkan saya adalah Burhan, salah satu mahasiswa Prodi S1-IF dari kelas IF-69-GAB dengan NIM 1029384750.
Selamat belajar, semoga sukses dalam studi Anda!

```

Deskripsi Program :

Program di atas merupakan program dalam bahasa Go yang digunakan untuk menerima input biodata mahasiswa dari pengguna, kemudian menampilkan informasi tersebut dalam bentuk resume yang terformat dengan baik.

Penjelasan Program:

1. Deklarasi Variabel:
 - Program mendeklarasikan empat variabel bertipe string: nama, nim, kelas, dan prodi, yang akan digunakan untuk menyimpan input dari pengguna.
2. Menerima Input dari Pengguna:
 - Program meminta pengguna untuk memasukkan data berupa nama, NIM, kelas, dan program studi (prodi) menggunakan fungsi `fmt.Scanln`.
 - Setiap input akan disimpan dalam variabel yang sesuai, dan program menampilkan prompt berupa teks seperti "Masukan Nama:", "Masukan NIM:", dll., untuk memandu pengguna memasukkan data.
3. Menampilkan Biodata dalam Format Resume:
 - Setelah menerima semua input, program menampilkan informasi biodata mahasiswa dalam dua bagian:
 - Resume Terformat: Informasi yang dimasukkan pengguna dicetak dalam format yang menyerupai resume, dengan label yang terorganisir seperti "Nama", "NIM", "Kelas", dan "Prodi".
 - Kalimat Perkenalan: Program juga menampilkan kalimat perkenalan yang menggunakan variabel-variabel yang telah diisi oleh pengguna. Ini bertujuan untuk membuat output yang lebih interaktif, dengan menggabungkan informasi yang diinputkan pengguna ke dalam kalimat yang jelas.
4. Pesan Motivasi:
 - Program ditutup dengan pesan motivasi untuk mahasiswa, "Selamat belajar, semoga sukses dalam studi Anda!", yang memberikan kesan akhir yang positif.

Kesimpulan :

Program ini digunakan untuk mengumpulkan data biodata mahasiswa melalui input dari pengguna, dan kemudian menampilkan informasi tersebut dalam format resume serta sebuah kalimat perkenalan. Program ini berguna dalam berbagai aplikasi yang memerlukan pengolahan data pengguna secara interaktif.

3.) Latihan Soal 3

Source code :

```
package main

import "fmt"

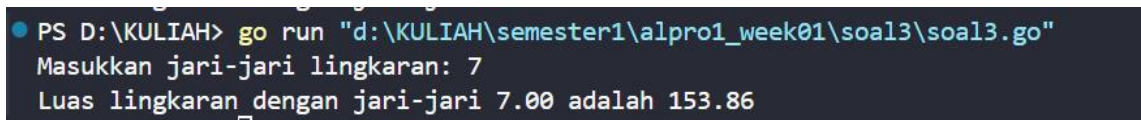
func main() {
    var r, luas float64
    const pi = 3.14

    // Meminta input jari-jari lingkaran dari pengguna
    fmt.Print("Masukkan jari-jari lingkaran: ")
    fmt.Scanln(&r)

    // Menghitung luas lingkaran menggunakan rumus  $\pi * r^2$ 
    luas = pi * r * r

    // Menampilkan hasil perhitungan luas lingkaran
    fmt.Printf("Luas lingkaran dengan jari-jari %.2f adalah %.2f\n", r,
luas)
}
```

Output :



```
PS D:\KULIAH> go run "d:\KULIAH\semester1\alpro1_week01\soal3\soal3.go"
Masukkan jari-jari lingkaran: 7
Luas lingkaran dengan jari-jari 7.00 adalah 153.86
```

Deskripsi Program :

Program ini merupakan program dalam bahasa Go yang digunakan untuk menghitung luas lingkaran berdasarkan jari-jari yang diberikan oleh pengguna. Program menggunakan rumus dasar luas lingkaran, yaitu:

$$\text{Luas} = \pi * r * r$$

Dimana:

- r adalah jari-jari lingkaran.
- π (pi) di dalam program ini memiliki nilai konstan 3.14.

Penjelasan Program:

1. Deklarasi Variabel dan Konstanta:
 - Variabel r (jari-jari) dan luas bertipe `float64` digunakan untuk menyimpan input dan hasil perhitungan luas lingkaran.
 - Konstanta π disetel dengan nilai 3.14 untuk menggantikan nilai π (pi).
2. Input dari Pengguna:
 - Program meminta pengguna untuk memasukkan nilai jari-jari lingkaran menggunakan `fmt.Scanln`, dan nilai yang diinputkan disimpan dalam variabel r .
3. Perhitungan Luas Lingkaran:
 - Program menghitung luas lingkaran dengan menggunakan rumus $\pi * r * r$, di mana nilai π digantikan dengan konstanta pi dan r adalah jari-jari yang diberikan oleh pengguna. Hasil perhitungan disimpan dalam variabel `luas`.
4. Menampilkan Hasil:
 - Program menampilkan hasil perhitungan luas lingkaran dalam format desimal dua angka di belakang koma menggunakan `fmt.Printf`. Hasil ditampilkan dalam bentuk: "Luas lingkaran dengan jari-jari [nilai] adalah [hasil]".

Kesimpulan :

Program ini menghitung luas lingkaran berdasarkan jari-jari yang dimasukkan oleh pengguna menggunakan rumus sederhana $\pi \times r^2$. Program ini menampilkan hasil perhitungan dalam format yang mudah dipahami, dengan dua angka desimal untuk memastikan keakuratan hasil.

4.) Latihan Soal 4

Source code :

```
package main

import "fmt"

func main() {
    var fahrenheit, celsius float64

    // Meminta input suhu dalam Fahrenheit dari pengguna
    fmt.Print("Masukkan suhu dalam Fahrenheit: ")
```

```

fmt.Scanln(&fahrenheit)

// Mengonversi suhu dari Fahrenheit ke Celcius

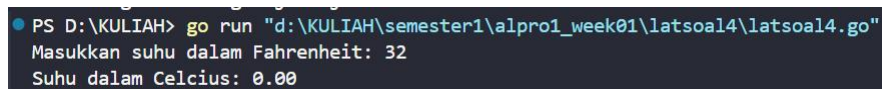
celsius = (fahrenheit - 32) * 5 / 9

// Menampilkan hasil konversi suhu dalam Celcius

fmt.Printf("Suhu dalam Celcius: %.2f\n", celsius)
}

```

Output :



```

PS D:\KULIAH> go run "d:\KULIAH\semester1\alpro1_week01\latsoal4\latsoal4.go"
Masukkan suhu dalam Fahrenheit: 32
Suhu dalam Celcius: 0.00

```

Deskripsi Program :

Program di atas adalah sebuah program dalam bahasa Go yang digunakan untuk melakukan konversi suhu dari Fahrenheit ke Celcius. Program menerima input dari pengguna berupa suhu dalam satuan Fahrenheit, kemudian menghitung dan menampilkan hasil konversi dalam satuan Celcius menggunakan rumus konversi yang telah dikenal.

Penjelasan Program:

1. Deklarasi Variabel:
 - Program mendeklarasikan dua variabel bertipe float64, yaitu fahrenheit untuk menyimpan input suhu dalam Fahrenheit dan celsius untuk menyimpan hasil konversi suhu ke Celcius.
2. Input dari Pengguna:
 - Program menggunakan fmt.Print untuk menampilkan pesan kepada pengguna agar memasukkan suhu dalam Fahrenheit.
 - Input suhu dimasukkan oleh pengguna melalui konsol menggunakan fmt.Scanln, yang menyimpan nilai input tersebut ke dalam variabel fahrenheit.
3. Konversi Suhu:
 - Program menghitung konversi suhu dari Fahrenheit ke Celcius menggunakan rumus : $C = \frac{5}{9} \times (F - 32)$

Di mana :

 - F adalah suhu dalam Fahrenheit (input dari pengguna),
 - C adalah suhu dalam Celcius (hasil konversi).

- Rumus ini dikodekan dalam perhitungan: $\text{celcius} = (\text{fahrenheit} - 32) * 5 / 9$.

4. Menampilkan Hasil:

- Setelah perhitungan selesai, program menampilkan hasil konversi dalam format dua angka di belakang koma menggunakan `fmt.Printf`. Hasil ini menunjukkan suhu yang setara dalam satuan Celcius.

Kesimpulan :

Program ini digunakan untuk mengonversi suhu dari Fahrenheit ke Celcius. Dengan memasukkan suhu dalam Fahrenheit, program menghitung suhu dalam Celcius dan menampilkannya dengan format yang rapi dan akurat. Program ini sangat berguna untuk kebutuhan konversi suhu secara cepat dan interaktif.

DAFTAR PUSTAKA

Alfandi, A. S. (2018). PRAKTIKUM 4 (1/2) – Tipe Data Dan Operasi I/O. Retrieved from <https://alfanbro99.wordpress.com/2018/11/08/praktikum-4-1-2-tipe-data-dan-operasi-i-o/>

LAPORAN PRAKTIKUM Variabel Dan Tipe Data. (n.d.). Retrieved from <https://id.scribd.com/document/671779921/LAPORAN-PRAKTIKUM-variabel-dan-tipe-data-1-Copy>