

LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 1
MODUL 2
“TIPE DATA & VARIABEL”



DISUSUN OLEH:

M. HANIF AL FAIZ

103112400042

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:

- Tipe data adalah kategori yang menentukan jenis nilai yang bisa kita simpan dan manipulasi. Contohnya, bilangan bulat (integer), angka desimal (float), dan teks (string).
- Variabel adalah wadah yang digunakan untuk menyimpan data. Setiap variabel memiliki nama dan tipe data, seperti bilangan bulat (int), desimal (float), atau teks (string).
- Input/Output (I/O) adalah proses menerima data dari pengguna (input) dan menampilkan hasilnya ke layar (output).
- Dalam pemrograman, kita sering melakukan berbagai operasi pada variabel, seperti operasi matematika, logika, atau perbandingan. Sebelum digunakan, variabel harus dideklarasikan (diberi nama dan tipe data) serta diinisialisasi (diberi nilai awal).
- Konversi tipe data diperlukan jika kita ingin menggunakan dua jenis data yang berbeda dalam satu operasi, misalnya menggabungkan angka dengan teks.

Poin Penting:

1. Tipe Data: Seperti integer (bilangan bulat), float (bilangan desimal), boolean (benar atau salah), karakter, dan string (teks).
2. Variabel: Wadah untuk menyimpan data, harus diberi nama, jenis, dan nilai awal.
3. Input: Cara mengambil data dari pengguna.
4. Output: Cara menampilkan hasil ke layar.
5. Operasi Aritmatika: Penjumlahan, pengurangan, perkalian, pembagian, sisa bagi (modulo).
6. Operasi Logika: Seperti AND, OR, NOT untuk keputusan logis.
7. Konversi Tipe Data: Ketika kita perlu menggunakan tipe data berbeda bersama-sama.

SOAL LATIHAN

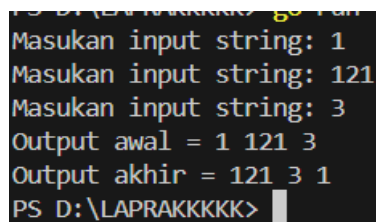
1. Latihan1

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)
}
```



```
PS D:\LAPRAK\GO> go run
Masukan input string: 1
Masukan input string: 121
Masukan input string: 3
Output awal = 1 121 3
Output akhir = 121 3 1
PS D:\LAPRAK\GO>
```

Output:

Deskripsi Program:

Program ini berfungsi untuk membaca tiga input string dari pengguna, menampilkan input awal, menukar posisi string secara berurutan, dan kemudian menampilkan output akhir setelah penukaran. Berikut adalah penjelasan bagaimana program ini berjalan:

1. ****Deklarasi Variabel****:

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

- `satu`, `dua`, dan `tiga`: Variabel untuk menyimpan tiga input string dari pengguna.
- `temp`: Variabel sementara untuk menyimpan nilai saat proses penukaran string.

2. Input dari Pengguna:

```
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 satu per satu dan menyimpannya dalam variabel `satu`, `dua`, dan `tiga` dengan menggunakan `fmt.Scanln()` untuk membaca input dari keyboard.

3. Menampilkan Output Awal:

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

- Setelah input, program menampilkan nilai awal dari variabel `satu`, `dua`, dan `tiga` dalam format string yang disatukan.

4. Penukaran Nilai Variabel:

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

- Langkah 1: Nilai variabel `satu` disimpan dalam variabel `temp` (agar tidak hilang).
- Langkah 2: Nilai `satu` diisi dengan nilai `dua`.
- Langkah 3: Nilai `dua` diisi dengan nilai `tiga`.
- Langkah 4: Nilai `tiga` diisi dengan nilai dari `temp` (yang menyimpan nilai awal `satu`).

5. Menampilkan Output Akhir:

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

```

- Setelah proses penukaran, program menampilkan nilai baru dari `satu`, `dua`, dan `tiga` setelah penukaran urutan.

## Contoh Jalannya Program:

Jika pengguna memasukkan string berikut:

- Input pertama: `A`
- Input kedua: `B`
- Input ketiga: `C`

## Output yang dihasilkan:

- Output awal: `A B C`
- Output akhir: `B C A` (setelah penukaran posisi).

Program ini pada dasarnya menukar posisi string sehingga string kedua menjadi yang pertama, string ketiga menjadi yang kedua, dan string pertama menjadi yang terakhir.

## 2.Latihan2

### Source Code:

```
package main

import "fmt"

func main() {
 var nama, nim, kelas string
 fmt.Scan(&kelas, &nama, &nim)
 fmt.Print("Perkenalkan saya adalah ", nama, ", salah satu mahasiswa Prodi S1-IF
dari kelas ", kelas, " dengan NIM ", nim)
}
```

### Output:

```
PS D:\LAPRAK\> go run "d:\LAPRAK\laprak\cosol2.go"
HANIF IF 01 12345678
Perkenalkan saya adalah IF, salah satu mahasiswa Prodi S1-IF dari kelas HANIF dengan NIM 01
PS D:\LAPRAK\> █
```

### Deskripsi Program:

Program ini berfungsi untuk menerima input dari pengguna, yaitu kelas, nama, dan NIM (Nomor Induk Mahasiswa), kemudian menampilkan pesan perkenalan yang menggabungkan ketiga input tersebut dalam format kalimat yang terstruktur. Berikut adalah deskripsi lebih rinci tentang bagaimana program ini bekerja:

#### 1. Deklarasi Variabel:

```
var nama, nim, kelas string
```

- nama: Variabel untuk menyimpan nama mahasiswa.
- nim: Variabel untuk menyimpan NIM (Nomor Induk Mahasiswa).
- kelas: Variabel untuk menyimpan kelas mahasiswa.

#### 2. Menerima Input dari Pengguna:

```
fmt.Scan(&kelas, &nama, &nim)
```

- Program menunggu pengguna untuk memasukkan tiga data (kelas, nama, dan NIM) dalam satu baris input. Data tersebut akan disimpan ke variabel `kelas`, `nama`, dan `nim` secara berurutan.

- fmt.Scan digunakan untuk membaca input dari pengguna. Dalam hal ini, input dipisahkan oleh spasi saat dimasukkan oleh pengguna.

### 3. Menampilkan Output Perkenalan:

```
fmt.Print("Perkenalkan saya adalah ", nama, ", salah satu mahasiswa Prodi S1-IF dari kelas ", kelas, " dengan NIM ", nim)
```

- Program akan menampilkan pesan perkenalan yang berisi:
  - Nama mahasiswa (`nama`).
  - Kelas mahasiswa (`kelas`).
  - NIM mahasiswa (`nim`).
- fmt.Print digunakan untuk menampilkan teks tanpa memindahkan kursor ke baris baru setelah selesai, sehingga semuanya ditampilkan dalam satu baris.

### 4. Alur Program:

- Pertama, program meminta pengguna untuk memasukkan kelas, nama, dan NIM dalam satu baris input.
- Setelah pengguna memasukkan data tersebut, program akan menampilkan kalimat perkenalan yang menggabungkan input tersebut dalam format yang sesuai.

Contoh Jalannya Program:

Jika pengguna memasukkan input berikut:

IF-44-01 JohnDoe 12345678

Program akan menampilkan output:

Perkenalkan saya adalah JohnDoe, salah satu mahasiswa Prodi S1-IF dari kelas IF-44-01 dengan NIM 12345678

### Fungsi Program:

- Program ini bisa digunakan untuk menerima informasi dari mahasiswa dan menampilkannya dalam format perkenalan yang otomatis berdasarkan input yang dimasukkan.

## 2. Latihan3

### Source Code:

```
package main

import "fmt"

func main() {
 var r int
 var phi, L float64
 fmt.Scan(&r)
 phi = 3.14
 L = phi * (float64(r)) * (float64(r))
}
```

```
 fmt.Print(L)
}
```

Output:

```
PS D:\LAPRAKKKKK> go run "d:\LAPRAKKKKK\laprak\laprak2\tempCodeRunnerFile.go"
7
153.86
PS D:\LAPRAKKKKK> go run "d:\LAPRAKKKKK\laprak\laprak2\tempCodeRunnerFile.go"
14
615.44
PS D:\LAPRAKKKKK> go run "d:\LAPRAKKKKK\laprak\laprak2\tempCodeRunnerFile.go"
20
1256
PS D:\LAPRAKKKKK> █
```

Deskripsi Program:

Program ini digunakan untuk menghitung **luas lingkaran** berdasarkan **jari-jari** yang dimasukkan oleh pengguna. Hasil perhitungan luas lingkaran akan ditampilkan di layar. Berikut adalah penjelasan lebih rinci mengenai cara kerja program ini:

1. Deklarasi Variabel:

```
var r int
var phi, L float64
```

- `r`: Variabel bertipe `int` untuk menyimpan jari-jari lingkaran yang akan dimasukkan oleh pengguna. Nilai ini berupa angka bulat (integer).
- `phi`: Variabel bertipe `float64` yang digunakan untuk menyimpan nilai konstanta  $\pi$  (pi), yang dalam program ini didefinisikan sebagai 3.14.
- `L`: Variabel bertipe `float64` yang akan menyimpan hasil perhitungan luas lingkaran.

2. Input dari Pengguna:

```
fmt.Scan(&r)
```

- Program menunggu pengguna untuk memasukkan nilai jari-jari lingkaran (`r`), yang merupakan input berupa bilangan bulat.

3. Inisialisasi Nilai `phi`:

```
phi = 3.14
```

- Konstanta  $\pi$  (pi) diinisialisasi dengan nilai 3.14, yang merupakan nilai perkiraan dari konstanta matematika pi.

4. Perhitungan Luas Lingkaran:

```
L = phi * (float64(r)) * (float64(r))
```

- Rumus untuk menghitung luas lingkaran adalah:

$$\text{Luas} = \pi * r^2$$

Di dalam program:

- `phi` mewakili nilai  $\pi$  (3.14).
- `r` adalah jari-jari lingkaran yang dimasukkan oleh pengguna.

- Karena variabel `r` dideklarasikan sebagai `int` (bilangan bulat), tetapi perhitungan dilakukan dalam tipe `float64` (bilangan desimal), kita perlu melakukan casting atau konversi dari `int` ke `float64` dengan menggunakan `(float64(r))`. Hal ini untuk memastikan perhitungan dapat dilakukan dengan presisi yang benar.

5. Menampilkan Hasil (Luas Lingkaran):

`fmt.Print(L)`

- Setelah perhitungan selesai, nilai dari luas lingkaran (`L`) akan dicetak ke layar menggunakan `fmt.Print()`.

Contoh Jalannya Program:

Misalnya, jika pengguna memasukkan nilai:

5

Maka program akan melakukan perhitungan luas lingkaran:

$$L = 3.14 * 5 * 5 = 78.5$$

Hasil yang akan ditampilkan oleh program adalah:

78.5

Ringkasan Alur Program:

1. Program meminta pengguna memasukkan jari-jari lingkaran.
2. Program menggunakan nilai  $\pi$  (`phi = 3.14`) dan menghitung luas lingkaran menggunakan rumus  $L = \pi * r^2$ .
3. Hasil perhitungan luas lingkaran ditampilkan di layar.

Fungsi Program:

- Program ini menghitung luas lingkaran berdasarkan input jari-jari yang diberikan oleh pengguna, menggunakan rumus matematika dasar untuk luas lingkaran.

### 3. Latihan4

```
package main

import "fmt"

func main() {
 var F, C float64
 fmt.Scan(&F)
 C = (F - 32) * 5 / 9
 fmt.Print(C)
}
```



Output:

```
PS D:\LAPRAK\> go run "d:\LAPRAK\laprak\laprak2\tempCodeRunnerFile.go"
32
0
PS D:\LAPRAK\> go run "d:\LAPRAK\laprak\laprak2\tempCodeRunnerFile.go"
77
25
PS D:\LAPRAK\> go run "d:\LAPRAK\laprak\laprak2\tempCodeRunnerFile.go"
212
100
PS D:\LAPRAK\>
```

Deskripsi Program:

Program ini berfungsi untuk mengonversi suhu dari Fahrenheit ke Celsius. Berikut adalah penjelasan bagaimana program ini berjalan:

1. Deklarasi Variabel:

var F, C float64

- `F`: Variabel bertipe `float64` yang akan menyimpan nilai suhu dalam Fahrenheit yang diinput oleh pengguna.

- `C`: Variabel bertipe `float64` yang akan menyimpan hasil perhitungan konversi suhu dalam Celsius.

2. Input dari Pengguna:

fmt.Scan(&F)

- Program meminta pengguna untuk memasukkan suhu dalam Fahrenheit. Nilai tersebut akan disimpan dalam variabel `F`.

3. Konversi Suhu dari Fahrenheit ke Celsius:

$C = (F - 32) * 5 / 9$

- Rumus untuk mengonversi suhu dari Fahrenheit ke Celsius adalah:

$[C = \frac{5}{9} \times (F - 32)]$

- F adalah suhu dalam Fahrenheit yang dimasukkan oleh pengguna.

- C adalah hasil konversi suhu dalam Celsius.

Program ini menggunakan rumus tersebut untuk menghitung nilai suhu dalam Celsius dan menyimpannya di variabel `C`.

4. Menampilkan Hasil:

fmt.Print(C)

- Setelah proses konversi selesai, program menampilkan hasil konversi (nilai suhu dalam Celsius) ke layar.

Contoh Jalannya Program:

Misalnya, jika pengguna memasukkan suhu:

100

Program akan melakukan konversi suhu dari Fahrenheit ke Celsius:

$$[C = \frac{5}{9} \times (100 - 32) = 37.7778]$$

Output yang ditampilkan oleh program adalah:

37.7778

Alur Program:

1. Program meminta pengguna memasukkan suhu dalam Fahrenheit.
2. Program menghitung suhu yang setara dalam Celsius menggunakan rumus konversi.
3. Program menampilkan hasil suhu dalam Celsius.

Fungsi Program:

Program ini digunakan untuk mengonversi suhu dari skala Fahrenheit ke skala Celsius berdasarkan input yang diberikan oleh pengguna.

## CONTOH SOAL

1. Cosol1

```
2. package main
3.
4. import "fmt"
5.
6. func main() {
7. var a, b, c, d, e int
8. var hasil int
9. fmt.Scanln(&a, &b, &c, &d, &e)
10. hasil = a + b + c + d + e
11. fmt.Println("Hasil penjumlahan", a, b, c, d, e, "adalah", hasil)
12.}
```

```
PS D:\LAPRAKKKK> go run "d:\LAPRAKKKK\cosol\cosol1.go"
3 2 7 10 2
Hasil penjumlahan 3 2 7 10 2 adalah 24
PS D:\LAPRAKKKK> go run "d:\LAPRAKKKK\cosol\cosol1.go"
11 22 33 44 55
Hasil penjumlahan 11 22 33 44 55 adalah 165
PS D:\LAPRAKKKK> █
```

Output:

Deskripsi Program:

Program ini digunakan untuk menjumlahkan lima bilangan bulat yang diinput oleh pengguna, kemudian menampilkan hasil penjumlahannya. Berikut adalah penjelasan bagaimana program ini berjalan:

#### 1. Deklarasi Variabel:

```
var a, b, c, d, e int
```

```
var hasil int
```

- `a, b, c, d, e`: Lima variabel bertipe `int` yang digunakan untuk menyimpan lima bilangan bulat yang akan dimasukkan oleh pengguna.

- `hasil`: Variabel bertipe `int` yang akan menyimpan hasil penjumlahan dari lima bilangan bulat yang dimasukkan.

#### 2. Input dari Pengguna:

```
fmt.Scanln(&a, &b, &c, &d, &e)
```

- Program meminta pengguna untuk memasukkan lima bilangan bulat sekaligus dalam satu baris input, dipisahkan oleh spasi. Bilangan tersebut akan disimpan ke dalam variabel `a`, `b`, `c`, `d`, dan `e`.

- `fmt.Scanln()` digunakan untuk membaca input dari pengguna, dengan memisahkan setiap nilai berdasarkan spasi.

#### 3. Penjumlahan Bilangan:

```
hasil = a + b + c + d + e
```

- Setelah input diperoleh, program menghitung hasil penjumlahan dari kelima bilangan yang dimasukkan pengguna dan menyimpannya dalam variabel `hasil`.

#### 4. Menampilkan Hasil:

```
fmt.Println("Hasil penjumlahan", a, b, c, d, e, "adalah", hasil)
```

- Program kemudian menampilkan hasil penjumlahan dengan format: "Hasil penjumlahan [a] [b] [c] [d] [e] adalah [hasil]".

- `fmt.Println()` digunakan untuk mencetak nilai dengan menambahkan baris baru setelah mencetak output.

#### Contoh Jalannya Program:

Misalnya, jika pengguna memasukkan input berikut:

```
1 2 3 4 5
```

Program akan melakukan penjumlahan:

\[hasil = 1 + 2 + 3 + 4 + 5 = 15\]

Output yang ditampilkan oleh program:

Hasil penjumlahan 1 2 3 4 5 adalah 15

Alur Program:

1. Program meminta pengguna memasukkan lima bilangan bulat.
2. Program menjumlahkan kelima bilangan tersebut.
3. Program menampilkan hasil penjumlahan.

Fungsi Program:

Program ini berfungsi untuk menghitung penjumlahan dari lima bilangan bulat yang diinput oleh pengguna dan menampilkan hasilnya dalam format yang terstruktur.

2.cosol2

```
package main

import "fmt"

func main() {

 var x, fx float64
 fmt.Scan(&x)
 fx = 2/(x+5) + 5
 fmt.Println(fx)
}
```

Output:

```
PS D:\LAPRAK\cosol2> go run "d:\LAPRAK\cosol2\cosol2.go"
5
5.2
PS D:\LAPRAK\cosol2>
PS D:\LAPRAK\cosol2> go run "d:\LAPRAK\cosol2\cosol2.go"
-23
4.888888888888889
PS D:\LAPRAK\cosol2>
```

### Deskripsi Program:

Program ini digunakan untuk menghitung nilai fungsi matematika yang dirumuskan sebagai:

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

Program akan meminta pengguna memasukkan nilai x, lalu menghitung hasil dari fungsi tersebut, dan menampilkan hasilnya ke layar. Berikut adalah penjelasan lebih rinci mengenai bagaimana program ini berjalan:

#### 1. Deklarasi Variabel:

```
var x, fx float64
```

- `x`: Variabel bertipe `float64` yang akan menyimpan nilai yang dimasukkan oleh pengguna (nilai dari x).

- `fx`: Variabel bertipe `float64` yang akan menyimpan hasil perhitungan fungsi f(x).

#### 2. Input dari Pengguna:

```
fmt.Scan(&x)
```

- Program menunggu pengguna untuk memasukkan nilai x, yang merupakan input berupa bilangan desimal (float). Nilai ini akan disimpan dalam variabel `x`.

#### 3. Perhitungan Fungsi Matematika:

$$fx = 2/(x+5) + 5$$

- Program menghitung nilai f(x) berdasarkan rumus:

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

- Pertama, nilai x yang dimasukkan oleh pengguna ditambahkan dengan 5.

- Kemudian, 2 dibagi dengan hasil penjumlahan x + 5.

- Hasil dari pembagian tersebut ditambahkan dengan 5.

#### 4. Menampilkan Hasil:

```
fmt.Println(fx)
```

- Setelah perhitungan selesai, program menampilkan hasil dari fungsi f(x) ke layar dengan menggunakan `fmt.Println()`, yang juga menambahkan baris baru setelah output.

#### Contoh Jalannya Program:

Misalnya, jika pengguna memasukkan nilai:

5

Program akan menghitung:

$$f(5) = \frac{2}{5+5} + 5 = \frac{2}{10} + 5 = 0.2 + 5 = 5.2$$

Output yang ditampilkan oleh program:

## 5.2

### Alur Program:

1. Program meminta pengguna untuk memasukkan nilai x
2. Program menghitung nilai fungsi  $f(x)$  berdasarkan rumus  $\left(\frac{2}{x+5} + 5\right)$ .
3. Program menampilkan hasil perhitungan fungsi tersebut ke layar.

### Fungsi Program:

Program ini digunakan untuk menghitung dan menampilkan nilai fungsi matematika berdasarkan input yang diberikan oleh pengguna sesuai dengan rumus yang telah ditentukan.

### 3.cosol3

```
package main

import "fmt"

func main() {
 var c1, c2, c3, c4, c5 byte
 var b1, b2, b3 int
 fmt.Scan(&c1, &c2, &c3, &c4, &c5)
 fmt.Scanf("%c", &b1)
 fmt.Scanf("%c", &b2)
 fmt.Scanf("%cc", &b3)
 fmt.Printf("%c%c%c%c%c", c1, c2, c3, c4, c5)
 fmt.Printf("%c%c%c", b1+1, b2+1, b3+1)
}
```

```
PS D:\LAPRAK\cosol3> go run "d:\LAPRAK\cosol3\cosol3.go"
66 97 103 117 115
```

Output: Bagus

Deskripsi Program:

Program ini menerima input berupa lima data karakter dalam bentuk byte dan tiga data karakter yang diproses secara spesifik dengan format ASCII. Program kemudian mencetak karakter-karakter tersebut, dan untuk tiga karakter terakhir, ditampilkan dengan karakter yang berada satu posisi setelahnya dalam tabel ASCII.

Berikut adalah penjelasan rinci tentang bagaimana program ini berjalan:

#### 1. Deklarasi Variabel:

```
var c1, c2, c3, c4, c5 byte
```

```
var b1, b2, b3 int
```

- `c1, c2, c3, c4, c5`: Variabel bertipe `byte` yang digunakan untuk menyimpan lima input pertama (karakter). Tipe `byte` di sini adalah alias untuk `uint8` (8-bit integer), yang umumnya digunakan untuk menyimpan data karakter (sesuai dengan tabel ASCII).

- `b1, b2, b3`: Variabel bertipe `int` yang digunakan untuk menyimpan tiga karakter tambahan, namun diproses dalam bentuk integer untuk dilakukan operasi aritmatika.

#### 2. Menerima Input dari Pengguna (Lima Karakter):

```
fmt.Scan(&c1, &c2, &c3, &c4, &c5)
```

- Program menunggu pengguna untuk memasukkan lima karakter, dan setiap karakter disimpan dalam variabel `c1` hingga `c5`. Karena menggunakan tipe `byte`, karakter yang dimasukkan akan diterjemahkan ke nilai numerik berdasarkan kode ASCII.

#### 3. Menerima Input dari Pengguna (Tiga Karakter Lainnya):

```
fmt.Scanf("%c", &b1)
```

```
fmt.Scanf("%c", &b2)
```

```
fmt.Scanf("%cc", &b3)
```

- `fmt.Scanf("%c", &b1)`: Program menunggu pengguna untuk memasukkan satu karakter yang disimpan dalam variabel `b1`. Format `%c` menunjukkan bahwa input tersebut adalah karakter.

- `fmt.Scanf("%c", &b2)`: Sama seperti sebelumnya, namun untuk karakter kedua, yang disimpan dalam `b2`.

- `fmt.Scanf("%cc", &b3)`: Ini untuk karakter ketiga, disimpan dalam `b3`. Format `%cc` berfungsi untuk menerima satu karakter.

#### 4. Menampilkan Lima Karakter Pertama:

```
fmt.Printf("%c%c%c%c%c", c1, c2, c3, c4, c5)
```

- `fmt.Printf` digunakan untuk menampilkan lima karakter pertama (`c1` hingga `c5`) dalam format karakter. Setiap variabel karakter yang disimpan sebagai byte diubah kembali ke bentuk karakter menggunakan format `%c`.

#### 5. Menampilkan Tiga Karakter dengan Perubahan (Berdasarkan Tabel ASCII):

```
fmt.Printf("%c%c%c", b1+1, b2+1, b3+1)
```

- Karakter `b1`, `b2`, dan `b3` diproses dengan menambahkan `1` pada nilai ASCII-nya. Dengan kata lain, karakter yang akan dicetak adalah karakter berikutnya dalam tabel ASCII setelah karakter yang dimasukkan oleh pengguna.

- Sebagai contoh, jika pengguna memasukkan karakter `A`, nilai ASCII dari `A` adalah 65, maka yang akan ditampilkan adalah karakter dengan nilai ASCII 66, yaitu `B`.

Contoh Jalannya Program:

Jika pengguna memasukkan input berikut:

A B C D E

X Y Z

Maka alurnya adalah:

1. Lima karakter pertama `A`, `B`, `C`, `D`, `E` disimpan dalam `c1`, `c2`, `c3`, `c4`, `c5`.

2. Tiga karakter berikutnya, yaitu `X`, `Y`, `Z`, disimpan dalam `b1`, `b2`, `b3`.

3. Output pertama:

ABCDE

4. Output kedua: Setelah menambahkan `1` ke masing-masing nilai ASCII dari `X`, `Y`, `Z`, output yang ditampilkan adalah:

YZA

Alur Program:

1. Program menerima lima karakter dari pengguna, lalu menyimpannya dalam variabel `c1` hingga `c5`.

2. Program menerima tiga karakter tambahan, menyimpannya dalam `b1`, `b2`, dan `b3`.

3. Program menampilkan lima karakter pertama.

4. Program menampilkan tiga karakter tambahan yang telah diubah ke karakter berikutnya dalam tabel ASCII.

Fungsi Program:

- Program ini digunakan untuk memanipulasi data karakter, terutama dalam pengolahan nilai-nilai ASCII. Karakter pertama hingga kelima ditampilkan langsung, sementara tiga karakter berikutnya ditampilkan setelah diubah menjadi karakter yang ada satu posisi setelahnya dalam tabel ASCII.



## DAFTAR PUSTAKA

*Laporan Dasar Pemrograman Tipe Data.* (2015-05-17). Retrieved from <http://Academia.edu>:  
<http://Academia.edu>

*Laporan Hasil Praktikum Modul II Tipe Data.* (2014-11-05). Retrieved from <http://Slideshare.net>:  
<http://Slideshare.net>

*Laporan Praktikum 3 Tipe Data dan Operasi I/O.* (2020-01-01). Diambil kembali dari  
<http://Academia.edu>: <http://Academia.edu>

*Polinema Programming.* (2020-09-21). Retrieved from Jobsheet 03.

*Praktikum 4 & 5 – Tipe Data, Variabel, Konstanta dan Operator.* (2016-05-26). Retrieved from  
<http://Matheusrumetna.com>: <http://Matheusrumetna.com>