

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



DISUSUN OLEH:
MUHAMMAD ZAKY MUBAROK
103112400073
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

For & Loop

For-

loop adalah salah satu struktur kontrol perulangan yang paling umum digunakan dalam pemrograman. Berikut adalah dasar teori for-loop:

Definisi: For-

loop adalah pernyataan perulangan yang digunakan untuk mengulang blok kode tertentu sejumlah kali yang telah ditentukan. Ini sangat berguna untuk iterasi melalui elemen-elemen dalam array, daftar, atau rentang angka.

Struktur Umum: Struktur dasar for-

loop terdiri dari tiga bagian utama: inisialisasi, kondisi, dan iterasi

```
for (inisialisasi; kondisi; iterasi) {
```

```
    // blok kode yang akan diulang
```

```
}
```

- **Inisialisasi:** Bagian ini menginisialisasi variabel loop dan dieksekusi hanya sekali sebelum loop dimulai.
- **Kondisi:** Bagian ini dievaluasi sebelum setiap iterasi. Jika kondisi benar (true), blok kode dieksekusi; jika salah (false), loop berakhir.
- **Iterasi:** Bagian ini dieksekusi setelah setiap iterasi, biasanya digunakan untuk memperbarui variabel loop.

```
for i := 1; i <= 5; i++ {
```

```
    fmt.Println(i)
```

```
}
```

Dalam contoh ini:

- **Inisialisasi:** `i := 1`
- **Kondisi:** `i <= 5`
- **Iterasi:** `i++` (increment nilai `i` sebesar 1 setelah setiap iterasi)

Keuntungan:

1. **Sederhana dan Mudah Dibaca:** For-loop memiliki struktur yang jelas dan mudah dipahami.
2. **Fleksibilitas:** Bisa digunakan untuk berbagai jenis perulangan, termasuk iterasi pada array dan daftar.
3. **Kontrol Penuh:** Memberikan kontrol penuh atas inisialisasi, kondisi, dan iterasi.

CONTOH SOAL

1. Latihan1

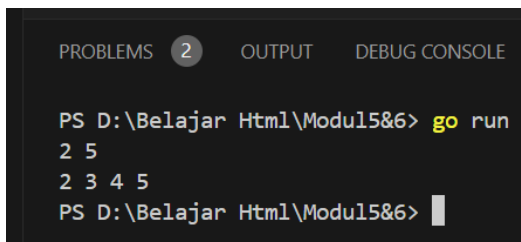
Source Code:

```
package main

import "fmt"

func main() {
    var a, b int
    var j int
    fmt.Scan(&a, &b)
    for j = a ; j <= b; j = j + 1 {
        fmt.Print(j, " ")
    }
}
```

Output:



```
PROBLEMS 2 OUTPUT DEBUG CONSOLE

PS D:\Belajar Html\Modul5&6> go run
2 5
2 3 4 5
PS D:\Belajar Html\Modul5&6>
```

Deskripsi Program:

Program: Menampilkan deretan bilangan dari a sampai b.

Langkah-langkah Program:

1. **Import Package:** Mengimpor package `fmt` untuk menggunakan fungsi-fungsi input dan output.
2. **Deklarasi Variabel:** Mendeklarasikan dua variabel integer `a` dan `b` untuk input bilangan bulat serta `j` untuk perulangan.
3. **Input Pengguna:** Meminta pengguna untuk memasukkan dua bilangan bulat `a` dan `b` menggunakan `fmt.Scan(&a, &b)`.
4. **Perulangan For:**
 - Memulai perulangan dari nilai `j` yang sama dengan `a`.
 - Melanjutkan perulangan selama `j` kurang dari atau sama dengan `b`.
 - Meningkatkan nilai `j` dengan 1 setiap iterasi (`j = j + 1`).

5. Output: Mencetak setiap nilai j dengan spasi menggunakan `fmt.Print(j, " ")`.

Contoh Penggunaan:

- **Jika memasukkan `a = 3` dan `b = 7`, program akan mencetak: 3 4 5 6**

CONTOH SOAL

2. Latihan2

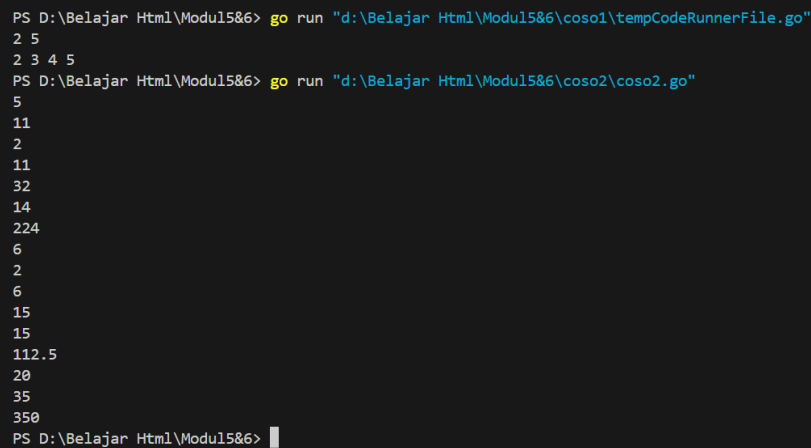
Source Code:

```
package main

import "fmt"

func main() {
    var j, alas, tinggi, n int
    var luas float64
    fmt.Scan(&n)
    for j = 1; j <= n; j += 1 {
        fmt.Scan(&alas, &tinggi)
        luas = 0.5 * float64(alas * tinggi)
        fmt.Println(luas)
    }
}
```

Output:



```
PS D:\Belajar Html\Modul5&6> go run "d:\Belajar Html\Modul5&6\coso1\tempCodeRunnerFile.go"
2 5
2 3 4 5
PS D:\Belajar Html\Modul5&6> go run "d:\Belajar Html\Modul5&6\coso2\coso2.go"
5
11
2
11
32
14
224
6
2
6
15
15
112.5
20
35
350
PS D:\Belajar Html\Modul5&6>
```

Deskripsi Program:

Program: Menghitung luas sejumlah n segitiga berdasarkan input panjang alas dan tinggi.

Langkah-langkah Program:

1. **Import Package:** Mengimpor package fmt untuk menggunakan fungsi-fungsi input dan output.
2. **Deklarasi Variabel:** Mendeklarasikan variabel-variabel integer j, alas, tinggi, dan n, serta variabel luas bertipe float64.

3. **Input Pengguna:** Meminta pengguna untuk memasukkan bilangan bulat n , yang menyatakan jumlah segitiga.
4. **Perulangan For:**
 - Memulai perulangan dari $j = 1$ hingga $j \leq n$.
 - Setiap iterasi, meminta pengguna untuk memasukkan panjang alas dan tinggi untuk segitiga ke- j .
 - Menghitung luas segitiga menggunakan rumus: $0.5 * \text{float64}(\text{alas} * \text{tinggi})$, lalu menyimpannya dalam variabel luas.
 - Mencetak hasil perhitungan luas untuk setiap segitiga.

Contoh Penggunaan:

- Jika memasukkan $n = 2$ dan panjang alas dan tinggi untuk dua segitiga berturut-turut, misalnya alas = 3, tinggi = 4 dan alas = 5, tinggi = 6, program akan menghitung luas masing-masing segitiga dan mencetak hasilnya.

CONTOH SOAL

3. Latihan3

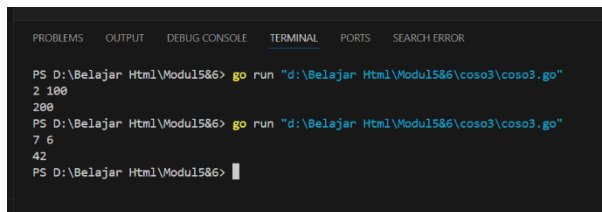
Source Code:

```
package main

import "fmt"

func main() {
    var j, hasil, v1, v2 int
    fmt.Scan(&v1, &v2)
    for j = 1; j <= v2 ; j+=1 {
        hasil = hasil + v1
    }
    fmt.Print(hasil)
}
```

Output :



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SEARCH ERROR
PS D:\Belajar Html\Modul5&6> go run "d:\Belajar Html\Modul5&6\coso3\coso3.go"
2 100
200
PS D:\Belajar Html\Modul5&6> go run "d:\Belajar Html\Modul5&6\coso3\coso3.go"
7 6
42
PS D:\Belajar Html\Modul5&6> █
```

Deskripsi Program :

Program: Menghitung hasil perkalian dua bilangan dengan menggunakan operator penjumlahan dan perulangan.

Langkah-langkah Program:

1. **Import Package:** Mengimpor package fmt untuk menggunakan fungsi-fungsi input dan output.
2. **Deklarasi Variabel:** Mendeklarasikan variabel-variabel integer j, hasil, v1, dan v2.
3. **Input Pengguna:** Meminta pengguna untuk memasukkan dua bilangan bulat positif v1 dan v2 menggunakan fmt.Scan(&v1, &v2).
4. **Perulangan For:**
 - Memulai perulangan dari j = 1 hingga j <= v2.

- Setiap iterasi, menambahkan nilai $v1$ ke variabel hasil.
5. **Output:** Mencetak hasil penjumlahan (yang merupakan hasil perkalian $v1$ dengan $v2$) menggunakan `fmt.Print(hasil)`.

Contoh Penggunaan:

- Jika memasukkan $v1 = 3$ dan $v2 = 4$, program akan menghitung $3 + 3 + 3 + 3$ yang sama dengan 12 dan mencetak hasil tersebut.

SOAL LATIHAN

Statement perulangan

1.

Source Code:

```
package main

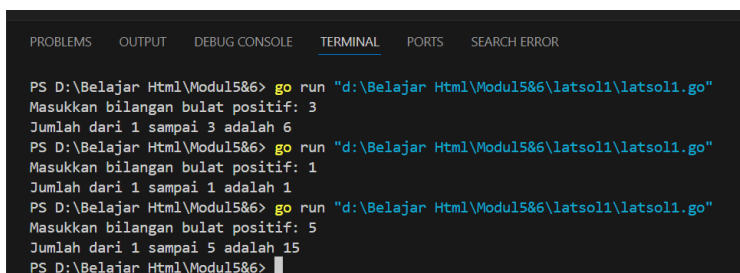
import (
    "fmt"
)

func main() {
    var n int
    fmt.Print("Masukkan bilangan bulat positif: ")
    fmt.Scan(&n)

    bil := 0
    for i := 1; i <= n; i++ {
        bil += i
    }

    fmt.Printf("Jumlah dari 1 sampai %d adalah %d\n", n, bil)
}
```

Output:



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  SEARCH ERROR

PS D:\Belajar Html\Modul5&6> go run "d:\Belajar Html\Modul5&6\latsol1\latsol1.go"
Masukkan bilangan bulat positif: 3
Jumlah dari 1 sampai 3 adalah 6
PS D:\Belajar Html\Modul5&6> go run "d:\Belajar Html\Modul5&6\latsol1\latsol1.go"
Masukkan bilangan bulat positif: 1
Jumlah dari 1 sampai 1 adalah 1
PS D:\Belajar Html\Modul5&6> go run "d:\Belajar Html\Modul5&6\latsol1\latsol1.go"
Masukkan bilangan bulat positif: 5
Jumlah dari 1 sampai 5 adalah 15
PS D:\Belajar Html\Modul5&6> 
```

Deskripsi Program:

Program: Menghitung jumlah dari 1 sampai dengan bilangan bulat positif n.

Langkah-langkah Program:

1. **Import Package:** Mengimpor package `fmt` untuk menggunakan fungsi-fungsi input dan output.

2. **Deklarasi Variabel:** Mendeklarasikan variabel integer `n` untuk menyimpan input bilangan bulat positif, dan `bil` untuk menyimpan hasil penjumlahan.
3. **Input Pengguna:** Meminta pengguna untuk memasukkan bilangan bulat positif `n` menggunakan `fmt.Scan(&n)`.
4. **Inisialisasi Variabel `bil`:** Menginisialisasi variabel `bil` dengan nilai 0.
5. **Perulangan For:**
 - Memulai perulangan dari `i = 1` hingga `i <= n`.
 - Setiap iterasi, menambahkan nilai `i` ke variabel `bil`.
6. **Output:** Mencetak hasil penjumlahan bilangan dari 1 sampai `n` menggunakan `fmt.Printf`.

Contoh Penggunaan:

- Jika memasukkan `n = 5`, program akan menghitung $1 + 2 + 3 + 4 + 5$ dan mencetak hasilnya sebagai 15

SOAL LATIHAN

Statement perulangan

2.

Source Code:

```
package main

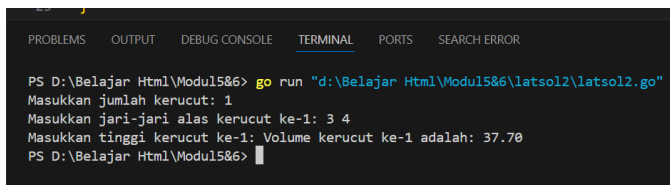
import (
    "fmt"
    "math"
)

func main() {
    var n int
    fmt.Print("Masukkan jumlah kerucut: ")
    fmt.Scan(&n)

    // Loop untuk memasukkan jari-jari dan tinggi, lalu menghitung volume
    for i := 1; i <= n; i++ {
        var jariJari, tinggi float64
        fmt.Printf("Masukkan jari-jari alas kerucut ke-%d: ", i)
        fmt.Scan(&jariJari)
        fmt.Printf("Masukkan tinggi kerucut ke-%d: ", i)
        fmt.Scan(&tinggi)

        // Menghitung volume kerucut
        volume := (1.0 / 3.0) * math.Pi * math.Pow(jariJari, 2) * tinggi
        fmt.Printf("Volume kerucut ke-%d adalah: %.2f\n", i, volume)
    }
}
```

Output:



```
PS D:\Belajar Html\Modul5&6> go run "d:\Belajar Html\Modul5&6\latsol2\latsol2.go"
Masukkan jumlah kerucut: 1
Masukkan jari-jari alas kerucut ke-1: 3 4
Masukkan tinggi kerucut ke-1: Volume kerucut ke-1 adalah: 37.70
PS D:\Belajar Html\Modul5&6>
```

Deskripsi Program:

Program: Menghitung volume sejumlah n kerucut berdasarkan input jari-jari alas dan tinggi masing-masing kerucut.

Langkah-langkah Program:

1. **Import Package:** Mengimpor package `fmt` untuk fungsi input/output dan `math` untuk fungsi matematika.
2. **Deklarasi Variabel:** Mendeklarasikan variabel integer `n` untuk menyimpan jumlah kerucut yang akan dihitung.
3. **Input Pengguna:** Meminta pengguna untuk memasukkan jumlah kerucut (`n`) menggunakan `fmt.Scan(&n)`.
4. **Perulangan For:**
 - Menginisialisasi perulangan dari `i = 1` hingga `i <= n`.
 - Setiap iterasi, meminta pengguna untuk memasukkan panjang `jariJari` dan `tinggi` dari kerucut ke-`i`.
 - Menghitung volume kerucut menggunakan rumus: $\text{volume} = (1.0 / 3.0) * \text{math.Pi} * \text{math.Pow}(\text{jariJari}, 2) * \text{tinggi}$.
 - Mencetak volume kerucut dengan format desimal dua angka menggunakan `fmt.Printf`.

Contoh Penggunaan:

- Jika pengguna `n = 2` dan untuk kerucut pertama `jariJari = 3` dan `tinggi = 4`, dan untuk kerucut kedua `jariJari = 5` dan `tinggi = 6`, program akan menghitung dan mencetak volume masing-masing kerucut.

SOAL LATIHAN

Statement perulangan

3.

Source Code:

```
package main

import (
    "fmt"
)

func main() {
    var dasar, pangkat int
    fmt.Print("Masukkan bilangan dasar: ")
    fmt.Scan(&dasar)
    fmt.Print("Masukkan bilangan pangkat: ")
    fmt.Scan(&pangkat)

    Hasil := 1
    for i := 0; i < pangkat; i++ {
        Hasil *= dasar
    }

    fmt.Printf("Hasil dari %d dipangkatkan dengan %d adalah %d\n", dasar, pangkat, Hasil)
}
```

Output:

```
PS D:\Belajar Html\Modul5&6> go run "d:\Belajar Html\Modul5&6\latsol3\latsol3.go"
Masukkan bilangan dasar: 4 2
Masukkan bilangan pangkat: Hasil dari 4 dipangkatkan dengan 2 adalah 16
PS D:\Belajar Html\Modul5&6> go run "d:\Belajar Html\Modul5&6\latsol3\latsol3.go"
Masukkan bilangan dasar: 2 10
Masukkan bilangan pangkat: Hasil dari 2 dipangkatkan dengan 10 adalah 1024
PS D:\Belajar Html\Modul5&6> go run "d:\Belajar Html\Modul5&6\latsol3\latsol3.go"
Masukkan bilangan dasar: 10 3
Masukkan bilangan pangkat: Hasil dari 10 dipangkatkan dengan 3 adalah 1000
PS D:\Belajar Html\Modul5&6>
```

Deskripsi Program:

Program: Menghitung hasil pemangkatan dua bilangan menggunakan operasi perkalian dan struktur kontrol perulangan.

Langkah-langkah Program:

1. **Import Package:** Mengimpor package `fmt` untuk menggunakan fungsi-fungsi input dan output.
2. **Deklarasi Variabel:** Mendeklarasikan variabel integer dasar, pangkat, dan Hasil.
3. **Input Pengguna:** Meminta pengguna untuk memasukkan bilangan dasar dan bilangan pangkat menggunakan `fmt.Scan(&dasar)` dan `fmt.Scan(&pangkat)`.
4. **Inisialisasi Variabel Hasil:** Menginisialisasi variabel Hasil dengan nilai 1.
5. **Perulangan For:**
 - Memulai perulangan dari $i = 0$ hingga $i < \text{pangkat}$.
 - Setiap iterasi, mengalikan nilai dasar dengan Hasil dan menyimpannya kembali ke Hasil.
6. **Output:** Mencetak hasil pemangkatan bilangan dasar dengan bilangan pangkat menggunakan `fmt.Printf`.

Contoh Penggunaan:

- Jika memasukkan dasar = 2 dan pangkat = 3, program akan menghitung $2 * 2 * 2$ dan mencetak hasilnya sebagai 8.

SOAL LATIHAN

Statement perulangan

4.

Source Code:

```
package main

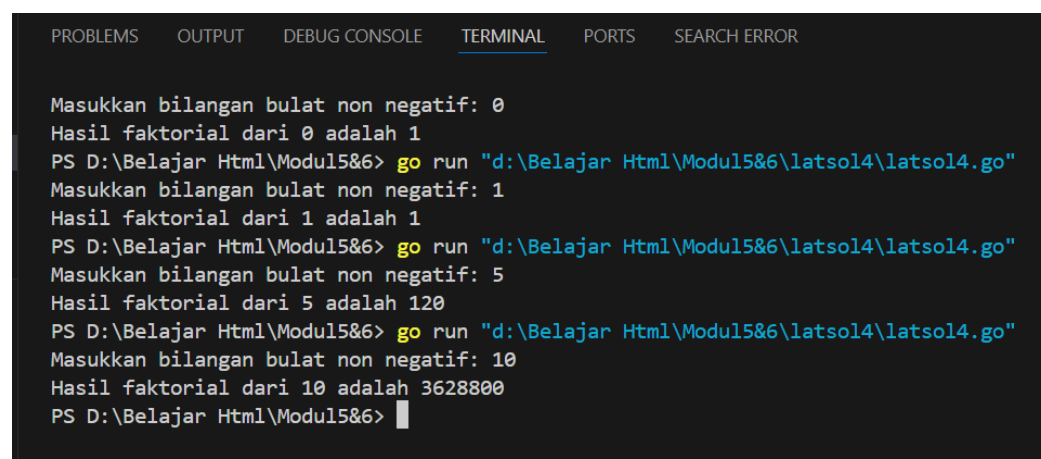
import (
    "fmt"
)

func main() {
    var n int
    fmt.Print("Masukkan bilangan bulat non negatif: ")
    fmt.Scan(&n)

    faktorial := 1
    for i := 1; i <= n; i++ {
        faktorial *= i
    }

    fmt.Printf("Hasil faktorial dari %d adalah %d\n", n, faktorial)
}
```

Output :



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  SEARCH ERROR

Masukkan bilangan bulat non negatif: 0
Hasil faktorial dari 0 adalah 1
PS D:\Belajar Html\Modul5&6> go run "d:\Belajar Html\Modul5&6\latsol4\latsol4.go"
Masukkan bilangan bulat non negatif: 1
Hasil faktorial dari 1 adalah 1
PS D:\Belajar Html\Modul5&6> go run "d:\Belajar Html\Modul5&6\latsol4\latsol4.go"
Masukkan bilangan bulat non negatif: 5
Hasil faktorial dari 5 adalah 120
PS D:\Belajar Html\Modul5&6> go run "d:\Belajar Html\Modul5&6\latsol4\latsol4.go"
Masukkan bilangan bulat non negatif: 10
Hasil faktorial dari 10 adalah 3628800
PS D:\Belajar Html\Modul5&6> 
```

Deskripsi Program :

Program: Menghitung hasil faktorial dari suatu bilangan bulat non negatif.

Langkah-langkah Program:

1. **Import Package:** Mengimpor package fmt untuk menggunakan fungsi-fungsi input dan output.
2. **Deklarasi Variabel:** Mendeklarasikan variabel integer n untuk menyimpan input bilangan bulat non negatif, dan faktorial untuk menyimpan hasil faktorial.
3. **Input Pengguna:** Meminta pengguna untuk memasukkan bilangan bulat non negatif n menggunakan `fmt.Scan(&n)`.
4. **Inisialisasi Variabel faktorial:** Menginisialisasi variabel faktorial dengan nilai 1.
5. **Perulangan For:**
 - Memulai perulangan dari $i = 1$ hingga $i \leq n$.
 - Setiap iterasi, mengalikan nilai i dengan faktorial dan menyimpannya kembali ke variabel faktorial.
6. **Output:** Mencetak hasil faktorial dari n menggunakan `fmt.Printf`.

Contoh Penggunaan:

- Jika memasukkan $n = 5$, program akan menghitung $1 * 2 * 3 * 4 * 5$ dan mencetak hasilnya sebagai 120

DAFTAR PUSTAKA

<https://www.detik.com/jabar/berita/d-6185689/rumus-volume-kerucut-cara-menghitung-contoh-soal-dan-luas-permukaannya>

https://www.ni.com/en/support/documentation/supplemental/08/labview-for-loops-and-while-loops-explained.html?srsId=AfmBOopjF3sYdStYtiBFyWeXaXK59BP7U03reY_bebeDF1NTXzLPvtPK

[Documentation - The Go Programming Language](#)