

**LAPORAN PRAKTIKUM ALGORITMA
DAN PEMROGRAMAN 1**

MODUL 5-6

MODUL 5-6 FOR-LOOP



Disusun oleh:

Yedija Johanan Siregar

109082500075

S1IF-13-07

Asisten Praktikum

Adithana dharma putra

Apri pandu wicaksono

**PROGRAM STUDI S1 INFORMATIKA
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO**

2025

LATIHAN KELAS - GUIDED

1. Guided 1

Source Code

```
package main

import "fmt"

func main() {

    var a, b int

    var j int

    fmt.Scan(&a, &b)

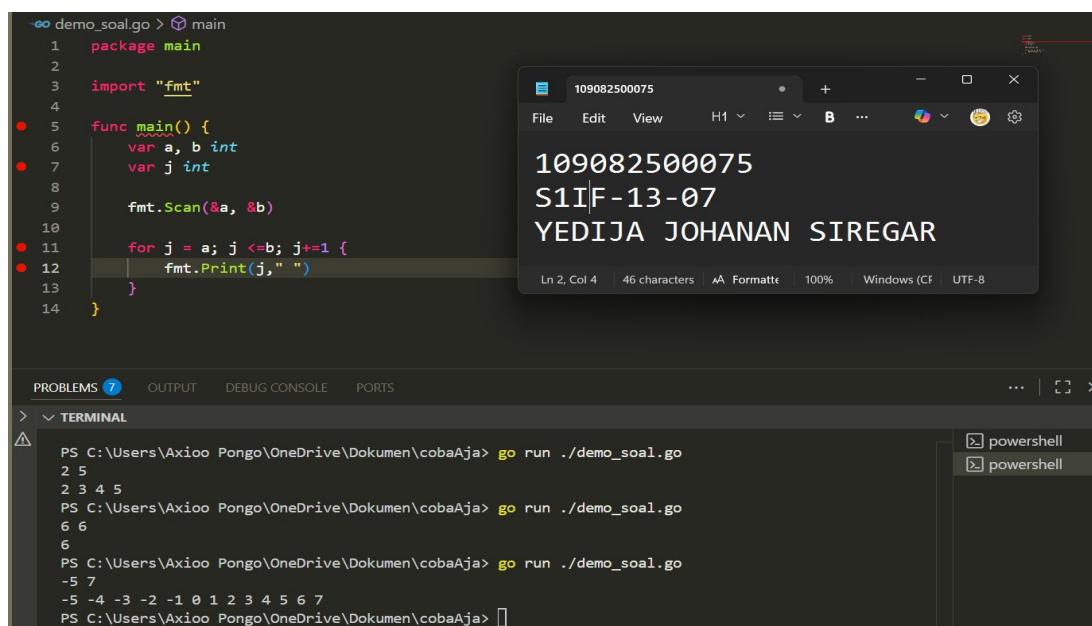
    for j = a; j <=b; j+=1 {

        fmt.Print(j, " ")

    }

}
```

Screenshoot program



Deskripsi program

Program ini digunakan untuk menampilkan deretan bilangan bulat dari nilai awal hingga nilai akhir yang dimasukkan oleh pengguna. Saat dijalankan, program meminta dua input berupa bilangan bulat, yaitu a sebagai batas awal dan b sebagai batas akhir. Setelah itu, program menggunakan perulangan for yang dimulai dari a hingga b, di mana setiap nilai j dalam rentang tersebut akan dicetak ke layar menggunakan `fmt.Print`. Hasilnya adalah deretan angka dari a sampai b yang dipisahkan dengan spasi.

2. Guided 2

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)

    }

}
```

Screenshoot program

The screenshot shows a Go IDE with a source code editor and a terminal. The source code is as follows:

```
demo_soal.go > main
1 package main
2 import "fmt"
3
4 func main() {
5     var j, alas, tinggi, n int
6     var luas float64
7     fmt.Scan(&n)
8
9     for j = 1; j <= n; j++ {
10         fmt.Scan(&alas, &tinggi)
11         luas = 0.5 * float64(alas * tinggi)
12         fmt.Println(luas)
13     }
14 }
```

The terminal shows the execution of the program with the following input and output:

```
PS C:\Users\Axiio Pongo\OneDrive\Dokumen\cobaAja> go run ./demo_soal.go
5
11 2
11
32 14
224
6 2
6
15 15
112.5
20 35
```

A floating window displays the output of the program:

```
109082500075
S1IF-13-07
YEDIJA JOHANAN SIREGAR
```

Deskripsi program

Program ini digunakan untuk menghitung luas beberapa segitiga berdasarkan data alas dan tinggi yang dimasukkan oleh pengguna. Ketika dijalankan, program meminta pengguna memasukkan sebuah bilangan n yang menunjukkan banyaknya segitiga yang akan dihitung. Setelah itu, program menjalankan perulangan `for` dari 1 hingga n , dan di setiap iterasi pengguna diminta memasukkan nilai alas dan tinggi segitiga. Nilai tersebut kemudian dikonversi ke tipe `float64` dan dihitung luasnya menggunakan rumus $luas = 0.5 \times alas \times tinggi$. Setelah perhitungan selesai, hasil luas dari setiap segitiga langsung ditampilkan di layar.

3. Guided 3

Source Code

```
package main

import "fmt"

func main() {

    var j, v1, v2 int

    var hasil int
```

```

        fmt.Scan(&v1, &v2)

        hasil = 0

        for j = 1; j <= v2; j+=1 {

            hasil = hasil + v1

        }

        fmt.Println(hasil)

    }

```

Screenshoot program

```

demo_soal.go > main
1  package main
2  import "fmt"
3
4  func main() {
5      var j, v1, v2 int
6      var hasil int
7      fmt.Scan(&v1, &v2)
8
9      hasil = 0
10
11     for j = 1; j <= v2; j+=1 {
12         hasil = hasil + v1
13     }
14     fmt.Println(hasil)
15 }

```

109082500075
S1IF-13-07
YEDIJA JOHANAN SIREGAR

```

PS C:\Users\Axioo Pongo\OneDrive\Dokumen\cobaAja> go run ./demo_soal.go
2 100
200
PS C:\Users\Axioo Pongo\OneDrive\Dokumen\cobaAja> go run ./demo_soal.go
7 6
42
PS C:\Users\Axioo Pongo\OneDrive\Dokumen\cobaAja>

```

Deskripsi program

Program ini digunakan untuk menghitung hasil perkalian antara dua bilangan bulat positif menggunakan operasi penjumlahan berulang. Ketika dijalankan, program meminta pengguna memasukkan dua bilangan, yaitu v1 dan v2. Variabel hasil diinisialisasi dengan nilai 0 sebagai tempat menyimpan hasil perhitungan. Selanjutnya,

program menggunakan perulangan for dari 1 sampai v2, dan pada setiap iterasi, nilai v1 akan ditambahkan ke variabel hasil. Setelah perulangan selesai, program menampilkan nilai akhir dari hasil, yang merupakan hasil perkalian antara v1 dan v2.

TUGAS

1. Tugas 1

Source code

```
package main

import "fmt"

func main() {
    var n int

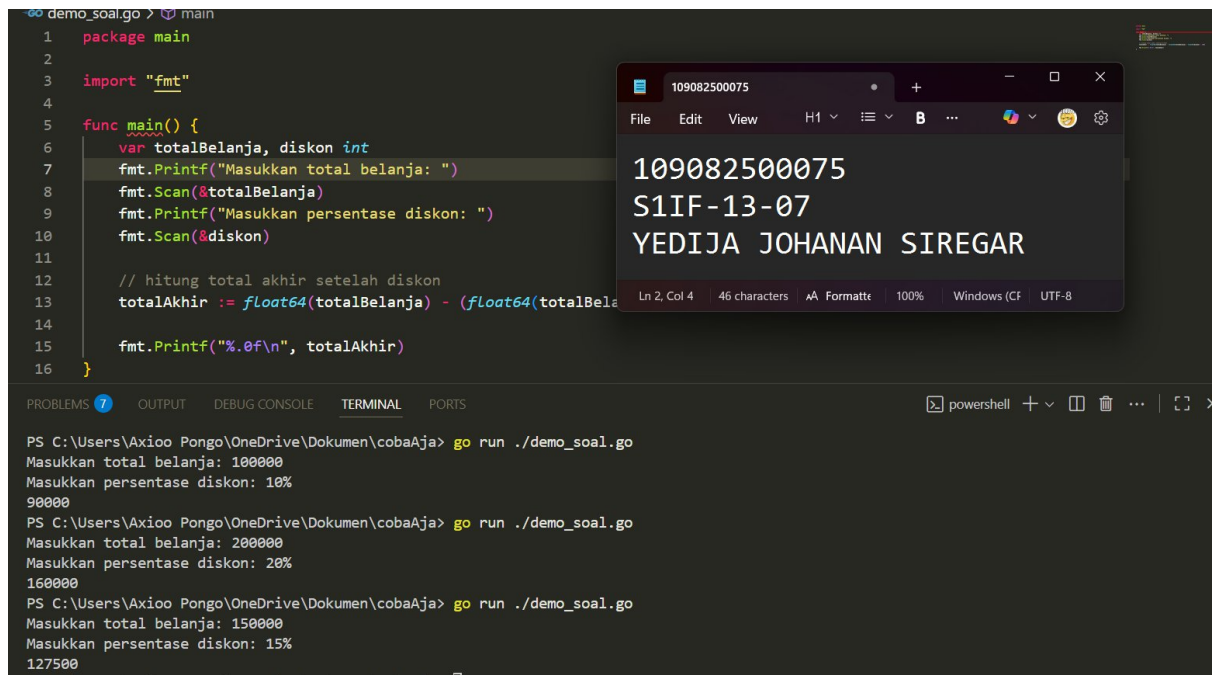
    fmt.Print("Masukkan bilangan bulat positif n: ")
    fmt.Scan(&n)

    jumlah := 0

    for i := 1; i <= n; i++ {
        jumlah += i
    }

    fmt.Println("Hasil penjumlahan dari 1 sampai", n, "adalah",
jumlah)
}
```

Screenshoot program



The screenshot shows a Go IDE with a source code editor on the left and a terminal on the right. The source code is a Go program named `demo_soal.go` that calculates the total price after a discount. It prompts the user for the total price and the discount percentage, then calculates the final amount. The terminal shows three runs of the program with different inputs.

```
demo_soal.go > main
1 package main
2
3 import "fmt"
4
5 func main() {
6     var totalBelanja, diskon int
7     fmt.Printf("Masukkan total belanja: ")
8     fmt.Scan(&totalBelanja)
9     fmt.Printf("Masukkan persentase diskon: ")
10    fmt.Scan(&diskon)
11
12    // hitung total akhir setelah diskon
13    totalAkhir := float64(totalBelanja) - (float64(totalBelanja) * float64(diskon) / 100)
14
15    fmt.Printf("%.0f\n", totalAkhir)
16 }
```

Terminal Output:

```
PS C:\Users\Axioo Pongo\OneDrive\Dokumen\cobaAja> go run ./demo_soal.go
Masukkan total belanja: 100000
Masukkan persentase diskon: 10%
90000
PS C:\Users\Axioo Pongo\OneDrive\Dokumen\cobaAja> go run ./demo_soal.go
Masukkan total belanja: 200000
Masukkan persentase diskon: 20%
160000
PS C:\Users\Axioo Pongo\OneDrive\Dokumen\cobaAja> go run ./demo_soal.go
Masukkan total belanja: 150000
Masukkan persentase diskon: 15%
127500
```

Deskripsi program

Program ini digunakan untuk menghitung hasil penjumlahan dari bilangan 1 sampai dengan n. Ketika dijalankan, program akan meminta pengguna memasukkan sebuah bilangan bulat positif dan menyimpannya ke dalam variabel n. Selanjutnya, program menginisialisasi variabel jumlah dengan nilai 0, kemudian menggunakan perulangan for untuk menambahkan setiap bilangan dari 1 hingga n ke dalam variabel tersebut. Setelah proses perulangan selesai, program menampilkan hasil akhir penjumlahan di layar sebagai keluaran. Dengan cara ini, pengguna dapat mengetahui total penjumlahan dari deret bilangan 1 hingga n sesuai nilai yang dimasukkan.

2. Tugas 2

Source code

```
package main

import (
    "fmt"
    "math"
```

```
)

func main() {

    var n int

    fmt.Print("Masukkan jumlah kerucut: ")

    fmt.Scan(&n)

    fmt.Println("Masukkan jari-jari dan tinggi setiap kerucut:")

    for i := 1; i <= n; i++ {

        var r, t float64

        fmt.Printf("Kerucut ke-%d (r t): ", i)

        fmt.Scan(&r, &t)

        volume := (1.0 / 3.0) * math.Pi * r * r * t

        fmt.Printf("Volume kerucut ke-%d = %.14f\n", i, volume)

    }

}
```

Screenshoot program

The screenshot shows a Go program in a text editor and its execution in a terminal. The program, named `demo_soal.go`, prompts the user to enter the number of cones (`n`), then for each cone, it prompts for radius (`r`) and height (`t`). It calculates the volume of each cone using the formula $V = \frac{1}{3} \pi r^2 t$ and prints the result for each cone.

```
1 package main
2
3 import (
4     "fmt"
5     "math"
6 )
7
8 func main() {
9     var n int
10    fmt.Print("Masukkan jumlah kerucut: ")
11    fmt.Scan(&n)
12
13    fmt.Println("Masukkan jari-jari dan tinggi setiap kerucut:")
14
15    for i := 1; i <= n; i++ {
16        var r, t float64
17        fmt.Printf("Kerucut ke-%d (r t): ", i)
18        fmt.Scan(&r, &t)
19
20        volume := (1.0 / 3.0) * math.Pi * r * r * t
21        fmt.Printf("Volume kerucut ke-%d = %.14f\n", i, volume)
22    }
23 }
```

The terminal output shows the program being run from a PowerShell prompt. The user enters 1 for the number of cones, and then 4 for the radius and 3 for the height of the first cone. The program outputs the volume of the first cone as 37.69911184307752.

```
PS C:\Users\Axioo Pongo\OneDrive\Dokumen\cobaAja> go run ./demo_soal.go
Masukkan jumlah kerucut: 1
Masukkan jari-jari dan tinggi setiap kerucut:
Kerucut ke-1 (r t): 3
4
Volume kerucut ke-1 = 37.69911184307752
PS C:\Users\Axioo Pongo\OneDrive\Dokumen\cobaAja>
```

Deskripsi program

Program ini digunakan untuk menghitung volume beberapa kerucut berdasarkan jari-jari dan tinggi yang dimasukkan pengguna. Pertama, pengguna diminta memasukkan jumlah kerucut (n), kemudian memasukkan pasangan nilai jari-jari (r) dan tinggi (t) sebanyak n kali. Setiap kali nilai dimasukkan, program menghitung volume menggunakan rumus, $V = \frac{1}{3}\pi r^2 t$, lalu menampilkan hasilnya di layar untuk setiap kerucut.

3. Tugas 3

Source code

```
package main

import "fmt"

func main() {
    var a, b int
    fmt.Print("Masukkan dua bilangan bulat positif (a b): ")
    fmt.Scan(&a, &b)

    hasil := 1
    for i := 1; i <= b; i++ {
```

```

        hasil *= a
    }

    fmt.Println("Hasil pemangkatan:", hasil)
}

```

Screenshoot program

```

demo_soal.go > main
1  package main
2
3  import "fmt"
4
5  func main() {
6      var a, b int
7      fmt.Print("Masukkan dua bilangan bulat positif (a b): ")
8      fmt.Scan(&a, &b)
9
10     hasil := 1
11     for i := 1; i <= b; i++ {
12         hasil *= a
13     }
14
15     fmt.Println("Hasil pemangkatan:", hasil)
16 }

```

109082500075
S1IF-13-07
YEDIJA JOHANAN SIREGAR

```

PS C:\Users\Axioo Pongo\OneDrive\Dokumen\cobaAja> go run ./demo_soal.go
Masukkan dua bilangan bulat positif (a b): 4 2
Hasil pemangkatan: 16
PS C:\Users\Axioo Pongo\OneDrive\Dokumen\cobaAja> go run ./demo_soal.go
Masukkan dua bilangan bulat positif (a b): 2 10
Hasil pemangkatan: 1024
PS C:\Users\Axioo Pongo\OneDrive\Dokumen\cobaAja> go run ./demo_soal.go
Masukkan dua bilangan bulat positif (a b): 10 3
Hasil pemangkatan: 1000
PS C:\Users\Axioo Pongo\OneDrive\Dokumen\cobaAja>

```

Deskripsi program

Program ini digunakan untuk menghitung hasil pemangkatan dari dua bilangan bulat positif dengan menggunakan operasi perkalian dan perulangan. Ketika dijalankan, pengguna diminta memasukkan dua bilangan, di mana bilangan pertama (a) merupakan angka yang akan dipangkatkan, dan bilangan kedua (b) adalah pangkatnya. Program kemudian melakukan perkalian berulang sebanyak b kali menggunakan struktur perulangan for. Setiap iterasi mengalikan nilai hasil dengan a hingga perulangan selesai. Setelah itu, program menampilkan hasil akhir pemangkatan ke layar.

4. Tugas 3

Source code

```
package main
```

```

import "fmt"

func main() {
    var n int

    fmt.Print("Masukkan bilangan bulat: ")
    fmt.Scan(&n)

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

    fmt.Println("Hasil faktorial dari", n, "adalah",
faktorial)
}

```

Screenshoot program

The screenshot shows a Go IDE with the following code in `demo_soal.go`:

```

1 package main
2
3 import "fmt"
4
5 func main() {
6     var n int
7     fmt.Print("Masukkan bilangan bulat: ")
8     fmt.Scan(&n)
9
10    faktorial := 1
11    for i := 1; i <= n; i++ {
12        faktorial *= i
13    }
14
15    fmt.Println("Hasil faktorial dari", n, "adalah", faktorial)
16 }

```

The terminal output shows the program being run three times with different inputs:

```

PS C:\Users\Axioo Pongo\OneDrive\Dokumen\cobaAja> go run ./demo_soal.go
Masukkan bilangan bulat: 0
Hasil faktorial dari 0 adalah 1
PS C:\Users\Axioo Pongo\OneDrive\Dokumen\cobaAja> go run ./demo_soal.go
Masukkan bilangan bulat: 1
Hasil faktorial dari 1 adalah 1
PS C:\Users\Axioo Pongo\OneDrive\Dokumen\cobaAja> go run ./demo_soal.go
Masukkan bilangan bulat: 10
Hasil faktorial dari 10 adalah 3628800
PS C:\Users\Axioo Pongo\OneDrive\Dokumen\cobaAja>

```

Deskripsi program

Program ini digunakan untuk menghitung hasil faktorial dari suatu bilangan bulat non-negatif. Ketika dijalankan, program akan meminta pengguna memasukkan

sebuah bilangan n . Kemudian program menginisialisasi variabel faktorial dengan nilai 1, karena nilai awal faktorial selalu dimulai dari 1. Setelah itu, program menggunakan perulangan `for` dari 1 sampai n untuk mengalikan setiap bilangan secara berurutan dengan variabel faktorial. Setelah perulangan selesai, hasil akhir perhitungan ditampilkan di layar sebagai nilai faktorial dari n .