LAPORAN PRAKTIKUM ALGORITMA DAN PEMROGRAMAN 1

MODUL V FOR-LOOP



Disusun oleh:

BENING PUTRI NARESWARI SUKARNO 109082500211

S1IF-13-02

Asisten Praktikum

Adithana dharma putra Alfin Ilham Berlianto

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

   fmt.Println("Masukkan dua bilangan bulat a dan b: ")
   fmt.Scan(&a, &b)

   for i := a; i <= b; i++ {
      fmt.Print(i, "")
   }
   fmt.Println()
}</pre>
```

```
🚥 perulangan1.go 🗦 ...
      package main
      import "fmt"
      func main() {
          var a, b int
          fmt.Println("Masukkan dua bilangan bulat a dan b: ")
          fmt.Scan(&a, &b)
          for i := a; i <= b; i++ {
              fmt.Print(i, "")
          fmt.Println()
16
                                                     powershell + v III III ····
                    DEBUG CONSOLE TERMINAL PORTS
PROBLEMS 5 OUTPUT
PS C:\Users\vero0\go\pkg\mod\golang.org\x\text@v0.27.0\internal\language\perulang
an> go run nyoba.go
Masukkan dua bilangan bulat a dan b:
                                                                            PS C:\Users\vero0\go\pkg\mod\golang.org\x
                                         File
                                                Edit
                                                       View
                                          NAMA : BENING PUTRI NARESWARI SUKARNO
                                          NIM: 109082500211
                                          KELAS: IF-13-02
```

Deskripsi program

Program berfungsi untuk menampilkan baris bilangan dari a sampai b. Cara Kerja:

- 1. Program mendeklarasikan dua variabel integer, a, dan b, untuk menyimpan dua bilangan bulat masukan.
- 2. Program meminta pengguna memasukkan dua bilangan bulat a dan b
- 3. Program menggunakan perulangan for. Memulai penghitung (i) dari nilai a (i :=
- a). Terus berjalan selama nilai i kurang dari atau sama dengan nilai b (i <= b).

- 1. Program meminta pengguna memasukkan dua bilangan bulat a dan b
- 2. Pengguna memasukkan dua bilangan 4 dan 7
- 3. Program mencetak deret angka tersebut: 47 (Angka 4, 5, 6, dan 7 dicetak, dan pada output terminal terlihat digabungkan menjadi 4567).
- 4. Program menampilkan hasil: 4567.

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)
    }
}</pre>
```

```
perulangan2.go 🗦
      package main
      import "fmt"
      func main() {
          var j, alas, ting
                              type float64 float64
          var luas float64
           fmt.Scan(&n)
                              float64 is the set of all IEEE 754 64-bit floating-point numbers.
           for j = 1; j <= n
               fmt.Scan(&ala float64 on pkg.go.dev
               luas = 0.5 * float64(alas*tinggi)
               fmt.Println(luas)
 15
PROBLEMS 4
                                               ∑ powershell + ∨ □ · □ · □ · □ · □ ×
                      TERMINAL
                                                                                Σ
x\text@v0.27.0\internal\language\perulangan> go run perulangan2.go
                                                                                Σ
11 2
                                                                                兦
11
32 14
224
                                                                               6 2
15 15
                                                                                     33
                                           File
                                                  Edit
112.5
20 35
                                           NAMA : BENING PUTRI NARESWARI SUKARNO
350
                                           NIM: 109082500211
PS C:\Users\vero0\go\pkg\mod\golang.org\>
                                           KELAS : IF-13-02
```

Deskripsi program

Program tersebut sedang menghitung luas segitiga secara berulang berdasarkan sejumlah pasangan nilai alas dan tinggi yang dimasukkan pengguna.

Cara Kerja:

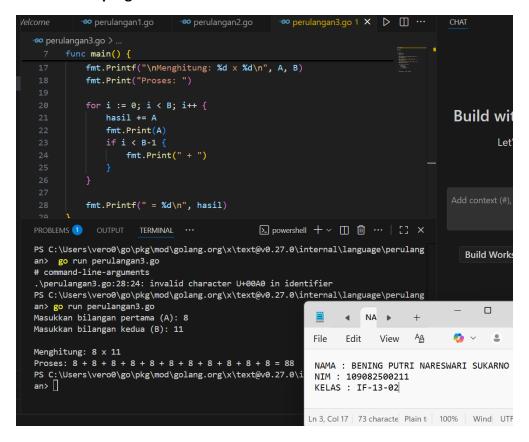
- 1. Program pertama-tama meminta satu masukan integer (n), yang menentukan berapa kali perhitungan luas segitiga akan dilakukan.
- 2. Program memulai perulangan for dari j=1 hingga j mencapai nilai n.
- 3. Di setiap putaran perulangan:
- 4. Program meminta dua masukan integer alas dan tinggi dari pengguna.
- 5. Program menghitung luas segitiga menggunakan rumus Luas = 0.5 x (alas x tinggi).
- 6. Setelah perhitungan, program segera mencetak nilai luas dari segitiga tersebut pada baris baru. Proses ini diulang hingga n kali.

3. Guided 3 Source Code

```
package main
import (
    "fmt"
)
func main() {
   var A, B int
   hasil := 0
    fmt.Print("Masukkan bilangan pertama (A): ")
    fmt.Scanln(&A)
    fmt.Print("Masukkan bilangan kedua (B): ")
    fmt.Scanln(&B)
    fmt.Printf("\nMenghitung: %d x %d\n", A, B)
    fmt.Print("Proses: ")
    for i := 0; i < B; i++ {
       hasil += A
        fmt.Print(A)
        if i < B-1 {
            fmt.Print(" + ")
```

```
}

fmt.Printf(" = %d\n", hasil)
}
```



Deskripsi program

Program tersebut sedang melakukan operasi perkalian (A x B) menggunakan metode penjumlahan berulang, sambil menampilkan langkah-langkah proses perhitungannya.

Cara Kerja:

- 1. Program mendeklarasikan dua variabel integer, A (bilangan yang akan dijumlahkan) dan B(jumlah pengulangan), serta variabel hasil yang diinisialisasi dengan 0.
- 2. Program meminta dua masukan dari pengguna: Bilangan pertama (A) bilangan yang akan dijumlahkan berulang, dan bilangan kedua (B) jumlah kali pengulangan.
- 3. Program menggunakan perulangan for yang berjalan sebanyak B kali. Program mencetak tanda "+" setelah setiap bilangan A, kecuali pada iterasi terakhir (if i < B-1).

4. Setelah perulangan selesai, program mencetak total hasil dari penjumlahan berulang tersebut, yang merupakan hasil dari A x B.

- 1. Program meminta Bilangan pertama A: Pengguna memasukkan 8.
- 2. Program meminta Bilangan kedua B: Pengguna memasukkan 11.
- 3. Program menampilkan perhitungan yang akan dilakukan: Menghitung: 8 x 11
- 4. Program menampilkan proses penjumlahan berulang:
- 5. Angka 8 dijumlahkan berulang sebanyak 11 kali.
- 7. Program menampilkan hasil akhir: = 88 (karena $8 \times 11 = 88$).

TUGAS

1. Tugas 1

```
package main
import "fmt"
func main() {
   var n int
   total := 0
    fmt.Print("Masukkan bilangan bulat positif (n): ")
    fmt.Scanln(&n)
    for i := 1; i <= n; i++ {
       total += i
    }
    fmt.Printf("Hasil penjumlahan dari 1 sampai %d adalah:
d\n", n, total)
}
```

```
co tugas1mod5.go 1 X
Welcome
                                                                       ▷ □ ··
🕶 tugas1mod5.go > ...
      package main
      import "fmt"
      func main() {
          var n int
           total := 0
           fmt.Print("Masukkan bilangan bulat positif (n): ")
           fmt.Scanln(&n)
           for i := 1; i <= n; i++ {
               total += i
PROBLEMS (1)
                     DEBUG CONSOLE TERMINAL
                                                      ∑ powershell + ∨ □ ····
PS C:\Users\vero0\go\pkg\mod\golang.org\x\text@v0.27.0\internal\language\tugas pe
rulangan> go run tugas1mod5.go
Masukkan bilangan bulat positif (n): 8
Hasil penjumlahan dari 1 sampai 8 adalah: 36
PS C:\Users\vero0\go\pkg\mod\golang.org\x\text@v0.27.0\internal\language\tugas pe
rulangan>
                                                         X
                         NA ▶
                                    AД
                                                         £
                File
                      Edit
                             View
                NAMA : BENING PUTRI NARESWARI SUKARNO
                NIM: 109082500211
                KELAS : IF-13-02
```

Deskripsi program

Tujuan program adalah untuk menghitung jumlah (penjumlahan) semua bilangan bulat positif mulai dari 1 hingga sebuah bilangan bulat positif n yang diinputkan oleh pengguna.

Cara Kerja:

- 1. Program meminta dan menerima angka 8 dari pengguna, menyimpannya sebagai nilai n
- 2. Program memulai hitungan dari angka 1 (variabel i).
- 3. Dalam setiap putaran, nilai hitungan saat ini (i) ditambahkan ke dalam total.
- 4. Proses ini diulang terus-menerus, dengan hitungan (i) bertambah satu, sampai i mencapai 8
- 5. Setelah perulangan selesai, program mencetak nilai akhir dari total (36) ke terminal.

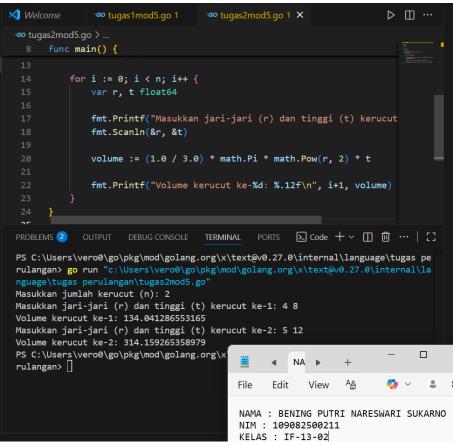
- 1. Program meminta input dari pengguna.
- 2. Pengguna memasukkan angka 8 sebagai nilai untuk n.
- 3. Program menampilkan hasil output setelah perhitungan. Ini berarti program menghitung 1 + 2 + 3 + 4 + 5 + 6 + 7 + 8, dan hasilnya adalah 36.

2. Tugas 2

```
package main
import (
    "fmt"
    "math"
)
func main() {
   var n int
    fmt.Print("Masukkan jumlah kerucut (n): ")
    fmt.Scanln(&n)
    for i := 0; i < n; i++ {
       var r, t float64
        fmt.Printf("Masukkan jari-jari (r) dan tinggi (t)
kerucut ke-%d: ", i+1)
        fmt.Scanln(&r, &t)
        volume := (1.0 / 3.0) * math.Pi * math.Pow(r, 2) * t
        fmt.Printf("Volume kerucut ke-%d: %.12f\n", i+1,
volume)
    }
```

}

Screenshoot program



Deskripsi program

Program adalah untuk menghitung volume kerucut berdasarkan rumus $\frac{1}{3}$ x π x r² x t , di mana r adalah jari-jari dan t adalah tinggi kerucut. Jumlah kerucut yang akan dihitung ditentukan oleh pengguna.

Cara Kerja:

- 1. Program meminta pengguna memasukkan berapa jumlah kerucut (n) yang ingin dihitung volumenya.
- 2. Program kemudian mengulangi proses perhitungan sebanyak n kali.
- 3. Dalam setiap pengulangan, program meminta pengguna untuk memasukkan jari-jari (r) dan tinggi (t) kerucut. Kemudian menghitungnya dengan rumus : $\frac{1}{3}$ x π x r² x t.
- 4. Program menampilkan hasil volume untuk kerucut yang sedang dihitung.

- 1. Program meminta pengguna memasukkan jumlah kerucut (n) yang akan dihitung. Pengguna memasukkan n= 2.
- 2. Perulangan ke-1:
 - Program meminta input untuk kerucut ke-1 (jari-jari r dan tinggi t). Pengguna memasukkan r=4 dan t=8. Program langsung menghitung volume dan menampilkan hasilnya (134.041...).
- 3. Perulangan ke-2:
 - Program mengulang langkah tadi untuk kerucut ke-2. Pengguna memasukkan r=5 dan t=12. Program kembali menghitung volume dan menampilkan hasilnya (314.159...).
- 4. Setelah menyelesaikan hitungan untuk kerucut kedua, program berhenti karena telah memenuhi jumlah perulangan yang diminta (n=2).

3. Tugas 3

```
package main

import "fmt"

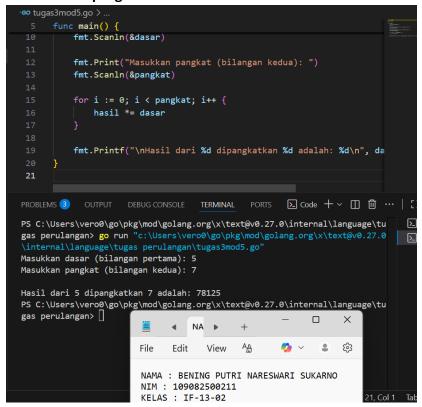
func main() {
   var dasar, pangkat int
   hasil := 1

   fmt.Print("Masukkan dasar (bilangan pertama): ")
   fmt.Scanln(&dasar)

   fmt.Print("Masukkan pangkat (bilangan kedua): ")
   fmt.Scanln(&pangkat)

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

   fmt.Printf("\nHasil dari %d dipangkatkan %d adalah:
%d\n", dasar, pangkat, hasil)
}</pre>
```



Deskripsi program

Program tersebut sedang menghitung hasil perpangkatan dari dua bilangan bulat yang dimasukkan oleh pengguna.

Cara Kerja:

- 1. Program mendeklarasikan tiga variabel:
 - dasar (bilangan yang akan dipangkatkan).
 - pangkat (jumlah kali bilangan dasar dikalikan).
 - hasil (untuk menyimpan hasil akhir), diinisialisasi dengan nilai 1.
- 2. Program meminta pengguna memasukkan bilangan dasar dan menyimpannya di variabel dasar.
- 3. Program meminta pengguna memasukkan bilangan pangkat dan menyimpannya di variabel pangkat.
- 4. Program menggunakan perulangan for yang akan berjalan sebanyak nilai dari pangkat.
- 5. Setelah perulangan selesai, program mencetak hasil akhir perhitungan dengan format: "Hasil dari [dasar] dipangkatkan [pangkat] adalah: [hasil]".

- 1. Program meminta masukan bilangan dasar:
- 2. Pengguna memasukkan nilai: 5
- 3. Program meminta masukan bilangan pangkat:
- 4. Pengguna memasukkan nilai: 7
- 6. Program menampilkan hasil: Hasil dari 5 dipangkatkan 7 adalah: 78125.

4. Tugas 4

```
package main

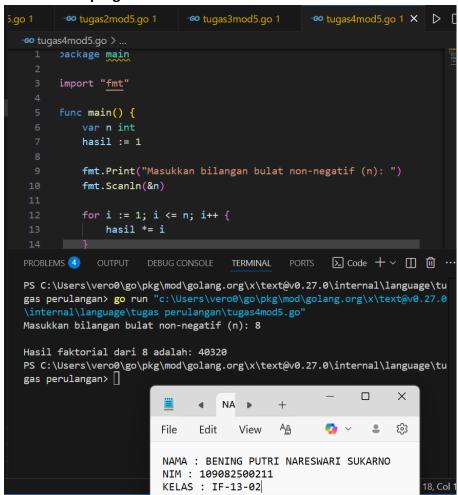
import "fmt"

func main() {
    var n int
    hasil := 1

    fmt.Print("Masukkan bilangan bulat non-negatif (n): ")
    fmt.Scanln(&n)

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

    fmt.Printf("\nHasil faktorial dari %d adalah: %d\n", n,
hasil)
}</pre>
```



Deskripsi program

Program tersebut sedang menghitung nilai faktorial (n!) dari bilangan bulat non-negatif (n) yang dimasukkan pengguna.

Cara Kerja:

- 1. Program mendeklarasikan variabel n (untuk menyimpan bilangan masukan) dan variabel hasil yang diinisialisasi dengan nilai 1 (karena 0! = 1 dan ini adalah nilai awal perkalian)
- 2. Program menampilkan pesan "Masukkan bilangan bulat non-negatif (n): " di terminal untuk meminta pengguna memasukkan nilai n.
- 3. Program membaca dan menyimpan bilangan bulat yang dimasukkan pengguna ke dalam variabel n.
- 4. Program menggunakan perulangan for yang dimulai dari i=1 hingga i mencapai nilai n.
- 5. Setelah perulangan selesai, program menampilkan hasil akhir perhitungan faktorial dengan format: "Hasil faktorial dari [nilai n] adalah: [nilai hasil]".

- 1. Program meminta pengguna memasukkan bilangan bulat non-negatif (n):
- 2. Pengguna memasukkan nilai: 8

- 3. Program menjalankan perhitungan faktorial 8! = 8 x 7 x 6 x 5 x 4 x 3 x 2 x 1 = 40320.
- 4. Program menampilkan hasil faktorial dari 8 adalah: 40320