

**LAPORAN PRAKTIKUM**  
**Algoritma Pemrograman**

**EVALUASI**



**Disusun oleh:**

**NAMA : PRIMATAMA SIGALINGGING**

**NIM : 109082500076**

**S1IF-13-07**

**PROGRAM STUDI S1 INFORMATIKA**  
**FAKULTAS INFORMATIKA**  
**TELKOM UNIVERSITY PURWOKERTO**  
**2025**

## SOAL

### 1. SOAL 1

#### Source Code

```
package main

import "fmt"

func main() {
    var n int

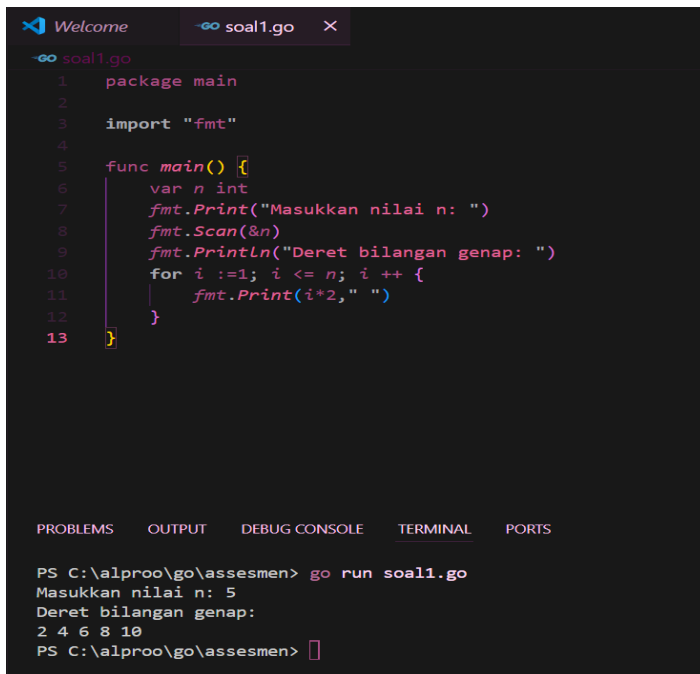
    fmt.Print("Masukkan nilai n: ")

    fmt.Scan(&n)

    fmt.Println("Deret bilangan genap: ")

    for i :=1; i <= n; i ++ {
        fmt.Print(i*2," ")
    }
}
```

#### Screenshoot program



```
Welcome  soal1.go x
-go soal1.go
1  package main
2
3  import "fmt"
4
5  func main() {
6      var n int
7      fmt.Print("Masukkan nilai n: ")
8      fmt.Scan(&n)
9      fmt.Println("Deret bilangan genap: ")
10     for i :=1; i <= n; i ++ {
11         fmt.Print(i*2," ")
12     }
13 }

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\alproo\go\assesmen> go run soal1.go
Masukkan nilai n: 5
Deret bilangan genap:
2 4 6 8 10
PS C:\alproo\go\assesmen> 
```

### **Deskripsi program**

#### ➤ **Tujuan**

Program ini dibuat untuk menampilkan deretan bilangan genap dari angka awal sampai angka akhir yang dimasukkan oleh pengguna. Tujuan utamanya adalah melatih pemahaman penggunaan perulangan (looping) dalam menghasilkan data berurutan.

#### ➤ **Proses**

Saat dijalankan, program meminta input dari pengguna, yaitu angka **n**(sebagai jumlah banyaknya deret).

Setelah itu, program menjalankan perulangan dengan variabel **i**.

Setiap kali nilai **i** berubah, program langsung menampilkan angka tersebut di layar, dipisahkan dengan spasi.

Misalnya pengguna memasukkan angka **5**, maka hasil yang muncul adalah:

**2 4 6 8 10**

Proses ini sangat sederhana tapi penting untuk memahami konsep dasar looping dengan kondisi batas.

#### ➤ **Kesimpulan**

Program ini menunjukkan bagaimana for loop bekerja untuk menghasilkan deretan angka secara otomatis. Dari sini, pengguna belajar bahwa komputer bisa menampilkan urutan nilai tanpa perlu menulis setiap angka satu per satu

## 2. SOAL 2

### Source Code

```
package main

import "fmt"

func main() {
    var x, y int
    fmt.Print("Masukkan nilai x: ")
    fmt.Scan(&x)
    fmt.Print("Masukkan nilai y: ")
    fmt.Scan(&y)

    hasil := 1
    for i := x; i <= y; i ++ {
        hasil = hasil * i
    }
    fmt.Println("Jumlah bakteri pada hari terakhir:",
        hasil)
}
```

## Screenshoot program

```
-go soal2.go
1  package main
2
3  import "fmt"
4
5  func main() {
6      var x, y int
7      fmt.Print("Masukkan nilai x: ")
8      fmt.Scan(&x)
9      fmt.Print("Masukkan nilai y: ")
10     fmt.Scan(&y)
11
12     hasil := 1
13     for i := x; i <= y; i ++ {
14         hasil = hasil * i
15     }
16     fmt.Println("Jumlah bakteri pada hari terakhir:", hasil)
17 }
```

PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS

```
Masukkan nilai x: 2
Masukkan nilai y: 4
Jumlah bakteri pada hari terakhir: 24
PS C:\alproo\go\assesmen> go run soal2.go
Masukkan nilai x: 3
Masukkan nilai y: 5
Jumlah bakteri pada hari terakhir: 60
```

### Deskripsi program

#### ➤ Tujuan

Tujuan utama program ini adalah untuk menghitung jumlah bakteri dari hari x sampai hari y dengan cara yang praktis. Selain itu, program ini juga melatih kemampuan pengguna dalam mengelola input yang lebih dari satu, serta memahami bagaimana data diproses berulang kali.

#### ➤ Proses

Pertama, pengguna akan diminta untuk memasukkan input nilai x dan juga nilai y. Program kemudian memproses setiap data satu per satu dan langsung menampilkan hasil perhitungannya di layar. Proses ini dilakukan secara berulang hingga semua data selesai dihitung.

#### ➤ Kesimpulan

Melalui program ini, kita bisa memahami bagaimana cara komputer mengolah banyak data secara efisien menggunakan perulangan. Walaupun hasil akhirnya hanya berupa angka, logika di balik program ini mencerminkan cara berpikir sistematis dan runtut dalam menyelesaikan masalah.

### 3. SOAL 3

#### Source Code

```
package main

import "fmt"

func main() {
    var keping, peti, karung, ikat, sisa int
    fmt.Print("Masukkan jumlah keping: ")
    fmt.Scan(&keping)

    peti = keping / 800
    sisa = keping % 800

    karung = sisa / 80
    sisa = sisa % 80

    ikat = sisa / 8
    keping = sisa % 8

    fmt.Print("Hasil: ")
    fmt.Print(peti, "peti, ")
    fmt.Print(karung, "karung, ")
    fmt.Print(ikat, "ikat, dan")
    fmt.Print(sisa, "keping\n")
}
```

## Screenshoot program

```
soal3.go
3  import "fmt"
4
5  func main() {
6      var keping, peti, karung, ikat, sisa int
7      fmt.Print("Masukkan jumlah keping: ")
8      fmt.Scan(&keping)
9
10     peti = keping / 800
11     sisa = keping % 800
12
13     karung = sisa / 80
14     sisa = sisa % 80
15
16     ikat = sisa / 8
17     keping = sisa % 8
18
19     fmt.Print("Hasil: ")
20     fmt.Print(peti, "peti, ")
21     fmt.Print(karung, "karung, ")
22     fmt.Print(ikat, "ikat, dan")
23     fmt.Print(sisa, "keping\n")
24 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\alproo\go\assesmen> go run soal3.go
Masukkan jumlah keping: 800
Hasil: 1peti, 0karung, 0ikat, dan0keping
```

### Deskripsi program

#### ➤ Tujuan

Program ini bertujuan untuk mengetahui berapa banyak peti, karung, ikat dan keping yang setara dengan barang yang pengguna miliki semisal si pengguna memiliki sejumlah keping.

#### ➤ Proses

Saat dijalankan, program meminta pengguna untuk memasukkan jumlah keping yang pengguna miliki. Misalnya pengguna memasukkan 800 keping, maka program akan menghitung hasil berapa banyak peti, karung, ikat, dan keping yang setara dengan 800 keping tersebut. Semua proses dilakukan secara otomatis, sehingga hasil akhir bisa langsung tampil di layar.

#### ➤ Kesimpulan

Program ini menggambarkan penerapan logika matematika yang terstruktur dan terkontrol.