

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

**EVALUASI**



**Disusun oleh:**

**Alvin setya wardana**

**109082500107**

**S1IF-13-02**

**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 bilangan: ") // Perbaikan:
    fmt.Print, bukan fmt.print

    fmt.Scan(&n)

    if n <= 0 {
        fmt.Println("input harus bilangan positif")
        return
    }

    fmt.Printf("deret %d bilangan ganjil:\n", n) //
    Perbaikan: gunakan fmt.Printf untuk format string

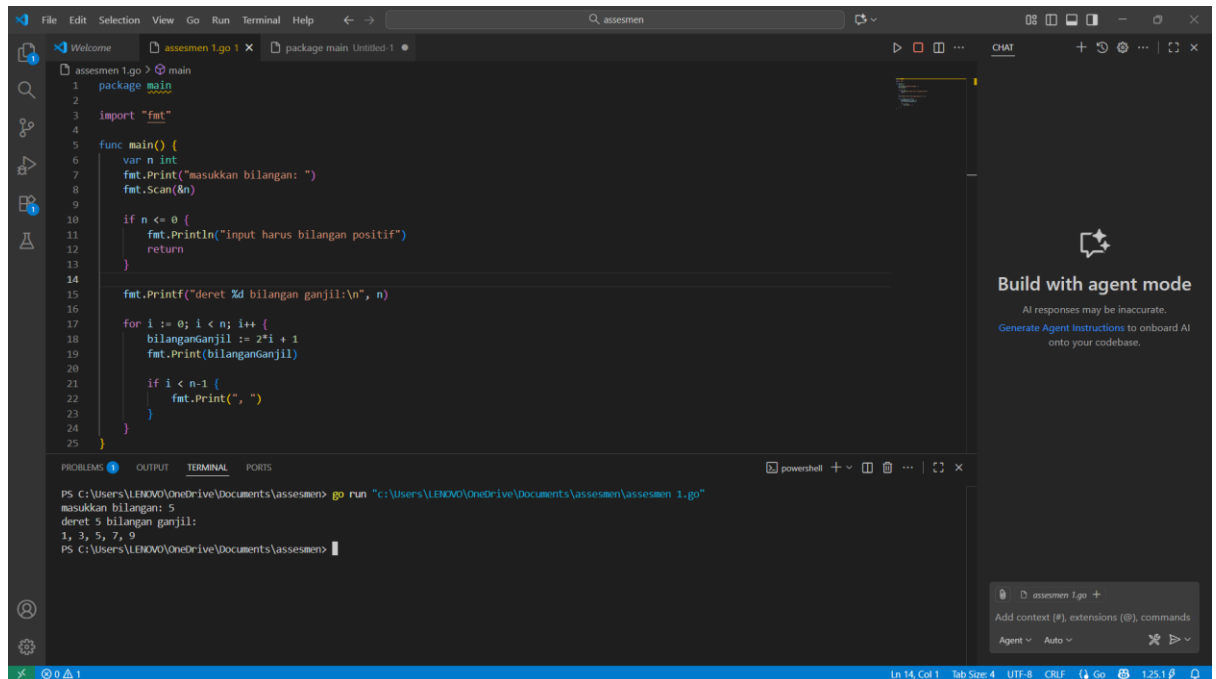
    for i := 0; i < n; i++ {
        bilanganGanjil := 2*i + 1
        fmt.Print(bilanganGanjil)

        if i < n-1 {
            fmt.Print(", ")
        }
    }
}
```

#### Screenshoot program

//tambahkan tangkapan layar dari program (boleh lebih dari 1 jika diperlukan)

## CONTOH TANGKAPAN LAYAR: (GUNAKAN NOTEPAD)



The screenshot shows a Visual Studio Code editor with a Go file named `assesmen 1.go`. The code defines a `main` package and a `main` function. It prompts the user to enter a positive integer `n`. If `n` is non-positive, it prints an error message. Otherwise, it prints the first `n` odd numbers. The terminal output shows the program being run, the input `5`, and the output `deret 5 bilangan ganjil: 1, 3, 5, 7, 9`.

```
1 package main
2
3 import "fmt"
4
5 func main() {
6     var n int
7     fmt.Print("masukkan bilangan: ")
8     fmt.Scan(&n)
9
10    if n <= 0 {
11        fmt.Println("input harus bilangan positif")
12        return
13    }
14
15    fmt.Printf("deret %d bilangan ganjil:\n", n)
16
17    for i := 0; i < n; i++ {
18        bilanganGanjil := 2*i + 1
19        fmt.Print(bilanganGanjil)
20
21        if i < n-1 {
22            fmt.Print(", ")
23        }
24    }
25 }
```

Terminal Output:

```
PS C:\Users\LENOVO\OneDrive\Documents\assesmen> go run "c:\Users\LENOVO\OneDrive\Documents\assesmen\assesmen 1.go"
masukkan bilangan: 5
deret 5 bilangan ganjil:
1, 3, 5, 7, 9
PS C:\Users\LENOVO\OneDrive\Documents\assesmen>
```

### Deskripsi program

Program ini meminta input sebuah bilangan bulat positif `n`, lalu mencetak deret sebanyak `n` bilangan ganjil pertama secara berurutan

## 2. SOAL 2

### Source Code

```
package main

import "fmt"

func main() {

    var x, y int

    fmt.Print("Masukkan hari awal: ")

    fmt.Scan(&x)

    fmt.Print("Masukkan hari akhir: ")

    fmt.Scan(&y)

    if x > y || x <= 0 {
```

```

        fmt.Println("Input tidak valid. Pastikan  $x \leq y$ 
        dan  $x > 0$ .")
        return
    }

    jumlah := 1
    for i := x; i <= y; i++ {
        jumlah *= i
    }

    fmt.Printf("Jumlah bakteri pada hari ke-%d sampai
    ke-%d adalah: %d\n", x, y, jumlah)
}

```

## Screenshoot program

The screenshot shows a Go program in VS Code. The code defines a `main` function that takes two integers `x` and `y` as input. It validates that `x` is less than or equal to `y` and `x` is greater than 0. If the input is invalid, it prints an error message. If valid, it calculates the total number of bacteria by multiplying all integers from `x` to `y` (inclusive) and prints the result.

The terminal output shows the program being run with `x=2` and `y=4`, resulting in a total of 24 bacteria.

```

1 package main
2
3 import "fmt"
4
5 func main() {
6     var x, y int
7     fmt.Print("Masukkan hari awal: ")
8     fmt.Scan(&x)
9     fmt.Print("Masukkan hari akhir: ")
10    fmt.Scan(&y)
11
12    if x > y || x <= 0 {
13        fmt.Println("Input tidak valid. Pastikan  $x \leq y$  dan  $x > 0$ .")
14        return
15    }
16
17    jumlah := 1
18    for i := x; i <= y; i++ {
19        jumlah *= i
20    }
21
22    fmt.Printf("Jumlah bakteri pada hari ke-%d sampai ke-%d adalah: %d\n", x, y, jumlah)
23 }
24

```

```

PS C:\Users\LENOVO\OneDrive\Documents\assessmen> go run "c:\Users\LENOVO\OneDrive\Documents\assessmen\assessmen_2.go"
Masukkan hari awal: 2
Masukkan hari akhir: 4
Jumlah bakteri pada hari ke-2 sampai ke-4 adalah: 24
PS C:\Users\LENOVO\OneDrive\Documents\assessmen>

```

## Deskripsi program

Program ini menghitung total jumlah bakteri dari hari ke- $x$  sampai hari ke- $y$ , di mana setiap hari jumlah bakteri dikalikan dengan nomor hari tersebut.

- Input dua bilangan bulat  $x$  dan  $y$  sebagai batas hari.
- Validasi agar  $x \leq y$  dan  $x > 0$ .

-Mengalikan semua bilangan dari x hingga y untuk mendapatkan total pertumbuhan bakteri.

### 3. SOAL 3

#### Source Code

```
package main

import "fmt"

func main() {
    var keping int
    fmt.Print("Masukkan jumlah dalam satuan keping: ")
    fmt.Scan(&keping)

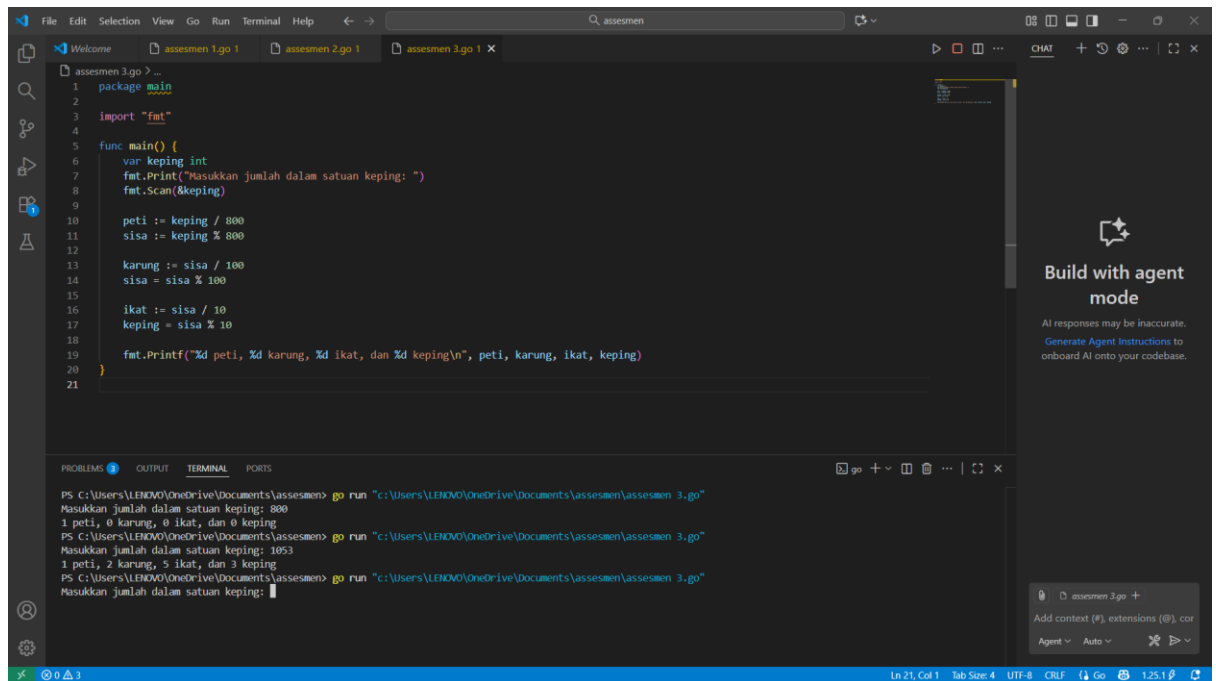
    peti := keping / 800
    sisa := keping % 800

    karung := sisa / 100
    sisa = sisa % 100

    ikat := sisa / 10
    keping = sisa % 10

    fmt.Printf("%d peti, %d karung, %d ikat, dan %d keping\n", peti, karung, ikat, keping)
}
```

#### Screenshoot program



```
1 package main
2
3 import "fmt"
4
5 func main() {
6     var keping int
7     fmt.Print("Masukkan jumlah dalam satuan keping: ")
8     fmt.Scan(&keping)
9
10    peti := keping / 800
11    sisa := keping % 800
12
13    karung := sisa / 100
14    sisa = sisa % 100
15
16    ikat := sisa / 10
17    keping = sisa % 10
18
19    fmt.Printf("%d peti, %d karung, %d ikat, dan %d keping\n", peti, karung, ikat, keping)
20 }
21
```

PROBLEMS OUTPUT TERMINAL PORTS

```
PS C:\Users\LENOVO\OneDrive\Documents\assamen> go run "c:\Users\LENOVO\OneDrive\Documents\assamen\assamen 3.go"
Masukkan jumlah dalam satuan keping: 800
1 peti, 0 karung, 0 ikat, dan 0 keping
PS C:\Users\LENOVO\OneDrive\Documents\assamen> go run "c:\Users\LENOVO\OneDrive\Documents\assamen\assamen 3.go"
Masukkan jumlah dalam satuan keping: 1053
1 peti, 2 karung, 5 ikat, dan 3 keping
PS C:\Users\LENOVO\OneDrive\Documents\assamen> go run "c:\Users\LENOVO\OneDrive\Documents\assamen\assamen 3.go"
Masukkan jumlah dalam satuan keping: 
```

## Deskripsi program

Program ini digunakan untuk mengonversi jumlah barang dalam satuan keping menjadi satuan yang lebih besar secara bertingkat: peti, karung, ikat, dan sisa keping.

### Logika Konversi:

- 1 peti = 800 keping
- 1 karung = 100 keping
- 1 ikat = 10 keping