

**LAPORAN PRAKTIKUM  
ALGORITMA DAN PEMROGRAMAN**

**TUGAS PENDAHULUAN MODUL 14**



**Disusun Oleh :**

**Chilya Fadhilatin Nisa / 103112430010**

**IF-05**

**Asisten Praktikum :**

**Ayu Susilowati**

**Noviana Rizki Anisa Putri**

**Dosen Pengampu :**

**Yuda Islami Sulistia**

**PROGRAM STUDI S1 TEKNIK INFORMATIKA**

**FAKULTAS INFORMATIKA**

**TELKOM UNIVERSITY PURWOKERTO**

**2024**

**TUGAS PENDAHULUAN**

## A. PRAKTIKUM

### 1. Soal Studi Case

Buatlah program untuk menampilkan bilangan sejumlah bilangan prima dengan inputan berupa bilangan bulat dari user Output berupa barisan bilangan prima dari 1 hingga bilangan yang diinputkan user.

#### Sourcecode

```
package main

import (
    "fmt"
)

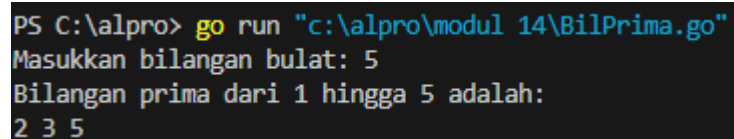
func isPrime(num int) bool {
    if num < 2 {
        return false
    }
    for i := 2; i*i <= num; i++ {
        if num%i == 0 {
            return false
        }
    }
    return true
}

func main() {
    var n int

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

    fmt.Printf("Bilangan prima dari 1 hingga %d adalah:\n", n)
    for i := 2; i <= n; i++ {
        if isPrime(i) {
            fmt.Printf("%d ", i)
        }
    }
    fmt.Println()
}
```

#### Screenshoot Output



```
PS C:\alpro> go run "c:\alpro\modul 14\BilPrima.go"
Masukkan bilangan bulat: 5
Bilangan prima dari 1 hingga 5 adalah:
2 3 5
```

**a. Deskripsi Program**

Program di atas adalah program sederhana dalam bahasa Go (Golang) yang digunakan untuk menampilkan bilangan prima dari angka 1 hingga angka yang diinputkan oleh pengguna. Bilangan prima adalah bilangan bulat positif yang hanya memiliki dua faktor, yaitu 1 dan bilangan itu sendiri.

**b. Algoritma Program**

- Mulai
- Input bilangan bulat  $n$  dari pengguna
- Buat fungsi `isPrime(num)` untuk memeriksa apakah `num` adalah bilangan prima:
  1. Jika  $num < 2$ , kembalikan `false` (karena bilangan kurang dari 2 bukan bilangan prima)
  2. Lakukan perulangan dari  $i = 2$  hingga  $i * i \leq num$ :
    - a. Jika  $num \% i == 0$ , kembalikan `false` (bukan bilangan prima)
  3. Jika perulangan selesai tanpa menemukan pembagi, kembalikan `true` (bilangan adalah bilangan prima)
- Lakukan perulangan untuk setiap bilangan  $i$  dari 2 hingga  $n$ :
  1. Panggil fungsi `isPrime(i)`:
    1. Jika hasilnya `true`, cetak  $i$
- Akhiri program
- Selesai

**c. Cara Kerja**

- Input Data:
  1. Pengguna diminta memasukkan angka bulat  $n$
- Logika Utama:
  1. Periksa setiap angka mulai dari 2 hingga  $n$ :
    - a. Gunakan fungsi `isPrime` untuk menentukan apakah angka tersebut prima
    - b. Jika angka prima, cetak angka tersebut.
- Output:
  1. Program mencetak bilangan prima dari 1 hingga angka  $n$

## 2. Soal Studi Case

Bilangan sempurna adalah bilangan yang sama dengan jumlah dari factor-faktornya. Buatlah program untuk menentukan apakah sebuah bilangan merupakan bilangan sempurna.

### Sourcecode

```
package main

import (
    "fmt"
)

func isPerfectNumber(num int) bool {
    if num <= 0 {
        return false
    }

    sum := 0
    for i := 1; i < num; i++ {
        if num%i == 0 {
            sum += i
        }
    }

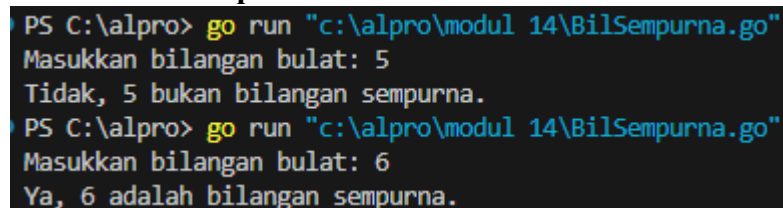
    return sum == num
}

func main() {
    var n int

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

    if isPerfectNumber(n) {
        fmt.Printf("Ya, %d adalah bilangan sempurna.\n", n)
    } else {
        fmt.Printf("Tidak, %d bukan bilangan sempurna.\n", n)
    }
}
```

### Screenshot Output



```
PS C:\alpro> go run "c:\alpro\modul 14\BilSempurna.go"
Masukkan bilangan bulat: 5
Tidak, 5 bukan bilangan sempurna.
PS C:\alpro> go run "c:\alpro\modul 14\BilSempurna.go"
Masukkan bilangan bulat: 6
Ya, 6 adalah bilangan sempurna.
```

#### a. Deskripsi Program

Program di atas digunakan untuk menentukan apakah sebuah bilangan adalah bilangan sempurna. Bilangan sempurna adalah bilangan yang sama dengan jumlah semua faktor pembaginya (tidak termasuk bilangan itu sendiri).

**b. Algoritma Program**

- Mulai
- Buat fungsi isPerfectNumber(num):
  1. Jika  $\text{num} \leq 0$ , kembalikan false
  2. Inisialisasi  $\text{sum} = 0$
  3. Lakukan perulangan dari  $i = 1$  hingga  $i < \text{num}$ :
    - a. Jika  $\text{num} \% i == 0$ , tambahkan  $i$  ke  $\text{sum}$
  4. Setelah selesai, kembalikan true jika  $\text{sum} == \text{num}$ ; jika tidak, kembalikan false
- Di fungsi main:
  1. Minta input bilangan  $n$  dari pengguna
  2. Panggil fungsi isPerfectNumber( $n$ ):
    - a. Jika true, cetak "Ya,  $n$  adalah bilangan sempurna"
    - b. Jika false, cetak "Tidak,  $n$  bukan bilangan sempurna"
- Selesai

**c. Cara Kerja**

- Input: Pengguna memasukkan bilangan bulat
- Proses: Program memeriksa apakah jumlah faktor (tidak termasuk bilangan itu sendiri) sama dengan bilangan itu
- Output: Menampilkan apakah bilangan tersebut adalah bilangan sempurna atau bukan.