

**LAPORAN PRAKTIKUM**  
**ALGORITMA DAN PEMROGRAMAN 1**  
**MODUL 13**  
**“TIPE DATA & VARIABEL”**



**DISUSUN OLEH:**  
**MUHAMMAD ZAKY MUBAROK**  
**103112400073 S1**  
**IF-12-01**  
**DOSEN:**  
**Yohani Setiya Rafika Nur, M. Kom.**

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

2024/2025

## DASAR TEORI

### Dasar Teori Komposisi

**1. Struktur Sekuensial: Menjalankan perintah secara berurutan, dari atas ke bawah.**

```
go
```

```
fmt.Println("Halo, dunia!")
```

```
fmt.Println("Belajar Komposisi")
```

**2. Struktur Kondisional: Memungkinkan program mengambil keputusan berdasarkan kondisi.**

```
go
```

```
if x > y {
```

```
    fmt.Println("x lebih besar")
```

```
} else {
```

```
    fmt.Println("x tidak lebih besar")
```

```
}
```

**3. Struktur Perulangan: Mengulang eksekusi perintah selama kondisi terpenuhi.**

```
go
```

```
for i := 0; i < 10; i++ {
```

```
    fmt.Println(i)
```

```
}
```

**Penggabungan: Menggunakan beberapa struktur kontrol bersama-sama untuk memecahkan masalah yang kompleks.**

- **Contoh: Mencetak bilangan ganjil dari 1 hingga 20.**

```
go
```

```
for i := 1; i <= 20; i++ {
```

```
    if i%2 != 0 {
```

```
        fmt.Println(i)
```

```
    }
```

```
}
```

**Komposisi membantu memecahkan masalah kompleks, meningkatkan keterbacaan, dan memudahkan pemeliharaan kode.**

## CONTOH SOAL

### 1. Latihan1

Source Code:

```
package main

import "fmt"

func main() {
    var bilangan, j int
    fmt.Scan(&bilangan)
    for j = 1 ; j <= bilangan; j+=1 {
        if j%2 != 0 {
            fmt.Print(j , " ")
        }
    }
}
```

Output:



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  SEARCH ERROR

PS D:\Belajar Html\Modul 14> go run "d:\Belajar Html\Modul 14\coso1\coso1.go"
3
1 3
PS D:\Belajar Html\Modul 14> go run "d:\Belajar Html\Modul 14\coso1\coso1.go"
2
1
PS D:\Belajar Html\Modul 14> go run "d:\Belajar Html\Modul 14\coso1\coso1.go"
7
1 3 5 7
PS D:\Belajar Html\Modul 14> go run "d:\Belajar Html\Modul 14\coso1\coso1.go"
10
1 3 5 7 9
PS D:\Belajar Html\Modul 14> █
```

## **Deskripsi Program:**

### **1. Inisialisasi variabel:**

**go**

**var bilangan, j int**

**Program mendeklarasikan dua variabel bilangan dan j yang bertipe integer.**

### **2. Membaca input:**

**go**

**fmt.Scan(&bilangan)**

**Program membaca nilai integer yang diinputkan dan menyimpannya dalam variabel bilangan.**

### **3. Perulangan dari 1 hingga nilai yang diinputkan:**

**go**

**for j = 1; j <= bilangan; j+=1 {**

**Program menjalankan loop dari 1 hingga nilai bilangan yang diinputkan. j+=1 berarti j akan bertambah 1 setiap kali loop dijalankan.**

### **4. Cek bilangan ganjil:**

**go**

**if j%2 != 0 {**

**Di dalam loop, program memeriksa apakah nilai j adalah bilangan ganjil dengan menggunakan operasi modulus (%). Jika j tidak habis dibagi 2 (j%2 != 0), maka j adalah bilangan ganjil.**

### **5. Mencetak bilangan ganjil:**

**go**

**fmt.Print(j, " ")**

**Jika kondisi if terpenuhi, program mencetak nilai j diikuti oleh spasi.**

**Secara keseluruhan, program ini efektif untuk mencetak semua bilangan ganjil mulai dari 1 hingga bilangan yang diinputkan**

## CONTOH SOAL

### 2. Latihan2

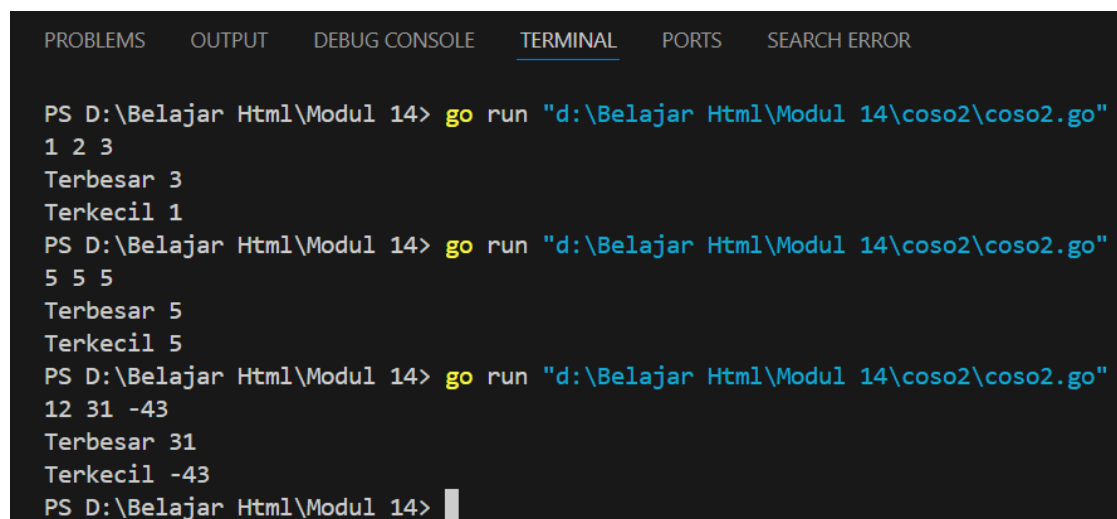
Source Code:

```
package main

import "fmt"

func main() {
    var b1, b2, b3, max, min int
    fmt.Scan(&b1, &b2, &b3)
    if b1 > b2 {
        max = b1
        min = b2
    } else {
        max = b2
        min = b1
    }
    if max < b3 {
        max = b3
    }
    if min > b3 {
        min = b3
    }
    fmt.Println("Terbesar", max)
    fmt.Println("Terkecil", min)
}
```

Output:



The screenshot shows a terminal window with a dark background and light-colored text. At the top, there are tabs for 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', 'TERMINAL' (which is active), 'PORTS', and 'SEARCH ERROR'. The terminal displays the execution of a Go program. The first command is 'go run "d:\Belajar Html\Modul 14\coso2\coso2.go"' followed by input '1 2 3'. The output is 'Terbesar 3' and 'Terkecil 1'. The second command is the same 'go run' command followed by input '5 5 5'. The output is 'Terbesar 5' and 'Terkecil 5'. The third command is the same 'go run' command followed by input '12 31 -43'. The output is 'Terbesar 31' and 'Terkecil -43'. The prompt 'PS D:\Belajar Html\Modul 14>' is visible at the end of each command line.

```
PS D:\Belajar Html\Modul 14> go run "d:\Belajar Html\Modul 14\coso2\coso2.go"
1 2 3
Terbesar 3
Terkecil 1
PS D:\Belajar Html\Modul 14> go run "d:\Belajar Html\Modul 14\coso2\coso2.go"
5 5 5
Terbesar 5
Terkecil 5
PS D:\Belajar Html\Modul 14> go run "d:\Belajar Html\Modul 14\coso2\coso2.go"
12 31 -43
Terbesar 31
Terkecil -43
PS D:\Belajar Html\Modul 14>
```

## **Deskripsi Program:**

### **1. Deklarasi variabel:**

```
var b1, b2, b3, max, min int
```

Program mendeklarasikan lima variabel integer: b1, b2, b3 untuk menyimpan tiga bilangan input, serta max dan min untuk menyimpan nilai maksimum dan minimum.

### **2. Membaca input:**

```
fmt.Scan(&b1, &b2, &b3)
```

Program membaca tiga nilai integer yang diinputkan dan menyimpannya dalam variabel b1, b2, dan b3.

### **3. Menentukan nilai maksimum dan minimum:**

```
if b1 > b2 {  
    max = b1  
    min = b2  
} else {  
    max = b2  
    min = b1  
}  
  
if max < b3 {  
    max = b3  
}  
  
if min > b3 {  
    min = b3  
}
```

Program membandingkan ketiga bilangan untuk menemukan nilai maksimum (max) dan minimum (min).

### **4. Mencetak hasil:**

```
go  
fmt.Println("Terbesar", max)  
fmt.Println("Terkecil", min)
```

Program mencetak nilai terbesar (max) dan terkecil (min) dari tiga bilangan yang diinputkan.

Program ini efektif untuk menemukan dan mencetak bilangan terbesar dan terkecil dari tiga bilangan yang diinputkan

## CONTOH SOAL

### 3. Latihan3

Source Code:

```
package main

import "fmt"

func main() {
    var bilangan, j int
    fmt.Scan(&bilangan)
    for j = 1 ; j <= bilangan; j+=1 {
        if bilangan%j == 0 {
            fmt.Print(j , " ")
        }
    }
}
```

Output:



The screenshot shows a terminal window with a dark background and light-colored text. At the top, there are tabs for 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', 'TERMINAL' (which is active), 'PORTS', and 'SEARCH ERROR'. Below the tabs, the terminal shows the execution of a Go program. The prompt is 'PS D:\Belajar Html\Modul 14>'. The user enters 'go run "d:\Belajar Html\Modul 14\coso3\coso3.go"'. The output is '5'. The user enters '1 5'. The prompt is 'PS D:\Belajar Html\Modul 14>'. The user enters 'go run "d:\Belajar Html\Modul 14\coso3\coso3.go"'. The output is '12'. The user enters '1 2 3 4 6 12'. The prompt is 'PS D:\Belajar Html\Modul 14>'. The user enters 'go run "d:\Belajar Html\Modul 14\coso3\coso3.go"'. The output is '20'. The user enters '1 2 4 5 10 20'. The prompt is 'PS D:\Belajar Html\Modul 14>'. The user enters 'go run "d:\Belajar Html\Modul 14\coso3\coso3.go"'. The output is '72'. The user enters '1 2 3 4 6 8 9 12 18 24 36 72'. The prompt is 'PS D:\Belajar Html\Modul 14>'. The cursor is at the end of the line.

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SEARCH ERROR

PS D:\Belajar Html\Modul 14> go run "d:\Belajar Html\Modul 14\coso3\coso3.go"
5
1 5
PS D:\Belajar Html\Modul 14> go run "d:\Belajar Html\Modul 14\coso3\coso3.go"
12
1 2 3 4 6 12
PS D:\Belajar Html\Modul 14> go run "d:\Belajar Html\Modul 14\coso3\coso3.go"
20
1 2 4 5 10 20
PS D:\Belajar Html\Modul 14> go run "d:\Belajar Html\Modul 14\coso3\coso3.go"
72
1 2 3 4 6 8 9 12 18 24 36 72
PS D:\Belajar Html\Modul 14> 
```



## **Deskripsi Program:**

### **1. Deklarasi variabel:**

```
var bilangan, j int
```

**Program mendeklarasikan dua variabel integer: bilangan untuk menyimpan nilai input dan j untuk perulangan.**

### **2. Membaca input:**

```
fmt.Scan(&bilangan)
```

**Program membaca nilai integer yang diinputkan dan menyimpannya dalam variabel bilangan.**

### **3. Perulangan untuk menemukan faktor:**

```
for j = 1; j <= bilangan; j+=1 {
```

**Program menjalankan loop dari 1 hingga nilai bilangan yang diinputkan. j+=1 berarti j akan bertambah 1 setiap kali loop dijalankan.**

### **4. Cek faktor:**

```
if bilangan%j == 0 {
```

```
    fmt.Print(j, " ")
```

```
}
```

**Di dalam loop, program memeriksa apakah j adalah faktor dari bilangan dengan menggunakan operasi modulus (%). Jika bilangan habis dibagi j (bilangan%j == 0), maka j adalah faktor dari bilangan, dan program mencetak nilai j diikuti oleh spasi.**

**Secara keseluruhan, program ini efektif untuk menemukan dan mencetak semua faktor dari sebuah bilangan yang diinputkan**

## SOAL LATIHAN

### Statement perulangan

1.

#### Source Code:

```
package main

import (
    "fmt"
)

func main() {
    var bilangan int
    fmt.Scan(&bilangan)

    if bilangan <= 0 {
        fmt.Println("Masukan harus bilangan bulat positif.")
    } else {
        jumlahGanjil := 0
        for i := 1; i <= bilangan; i++ {
            if i%2 != 0 {
                jumlahGanjil++
            }
        }

        fmt.Printf("Terdapat %d bilangan ganjil\n", jumlahGanjil)
    }
}
```

#### Output:



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  SEARCH ERROR

PS D:\Belajar Html\Modul 14> go run "d:\Belajar Html\Modul 14\latsol1\latsol1.go"
3
Terdapat 2 bilangan ganjil
PS D:\Belajar Html\Modul 14> go run "d:\Belajar Html\Modul 14\latsol1\latsol1.go"
2
Terdapat 1 bilangan ganjil
PS D:\Belajar Html\Modul 14> go run "d:\Belajar Html\Modul 14\latsol1\latsol1.go"
7
Terdapat 4 bilangan ganjil
PS D:\Belajar Html\Modul 14> go run "d:\Belajar Html\Modul 14\latsol1\latsol1.go"
10
Terdapat 5 bilangan ganjil
PS D:\Belajar Html\Modul 14> 
```

## **Deskripsikan Program :**

### **1. Deklarasi variabel:**

**var bilangan int**

**Program mendeklarasikan variabel integer bilangan untuk menyimpan nilai input.**

### **2. Membaca input:**

**fmt.Scan(&bilangan)**

**Program membaca nilai integer yang diinputkan dan menyimpannya dalam variabel bilangan.**

### **3. Memeriksa apakah bilangan positif:**

**if bilangan <= 0 {**

**fmt.Println("Masukan harus bilangan bulat positif.")**

**} else {**

**Program memeriksa apakah bilangan lebih besar dari 0. Jika tidak, program mencetak pesan bahwa input harus bilangan bulat positif.**

### **4. Perulangan untuk menghitung bilangan ganjil:**

**jumlahGanjil := 0**

**for i := 1; i <= bilangan; i++ {**

**if i%2 != 0 {**

**jumlahGanjil++**

**}**

**}**

**Jika bilangan lebih besar dari 0, program menginisialisasi variabel jumlahGanjil dengan nilai 0 dan menjalankan loop dari 1 hingga nilai bilangan. Di dalam loop, program memeriksa apakah i adalah bilangan ganjil dengan menggunakan operasi modulus (%). Jika i tidak habis dibagi 2 (i%2 != 0), maka i adalah bilangan ganjil, dan program menambahkan nilai jumlahGanjil sebanyak 1.**

### **5. Mencetak hasil:**

**fmt.Printf("Terdapat %d bilangan ganjil\n", jumlahGanjil)**

**Setelah loop selesai, program mencetak jumlah bilangan ganjil yang ditemukan.**

**Program ini efektif untuk menghitung dan mencetak jumlah bilangan ganjil dari 1 hingga bilangan yang diinputkan**

## SOAL LATIHAN

### 2. Latihan2 Source

Code:

```
package main

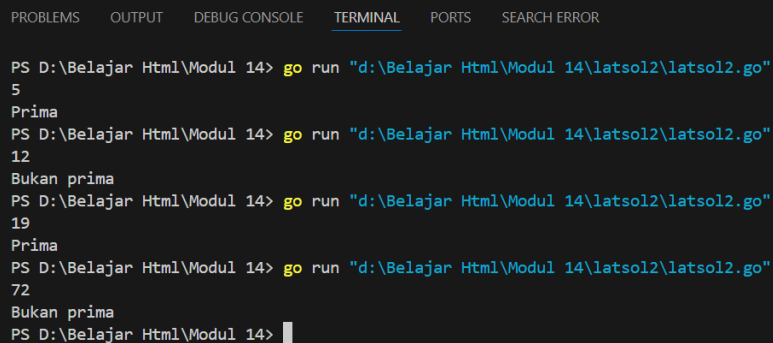
import (
    "fmt"
)

func main() {
    var bilangan int
    fmt.Scan(&bilangan)

    if bilangan <= 1 {
        fmt.Println("Bukan prima")
    } else {
        prima := true
        for i := 2; i*i <= bilangan; i++ {
            if bilangan%i == 0 {
                prima = false
                break
            }
        }

        if prima {
            fmt.Println("Prima")
        } else {
            fmt.Println("Bukan prima")
        }
    }
}
```

Output:



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  SEARCH ERROR

PS D:\Belajar Html\Modul 14> go run "d:\Belajar Html\Modul 14\latsol2\latsol2.go"
5
Prima
PS D:\Belajar Html\Modul 14> go run "d:\Belajar Html\Modul 14\latsol2\latsol2.go"
12
Bukan prima
PS D:\Belajar Html\Modul 14> go run "d:\Belajar Html\Modul 14\latsol2\latsol2.go"
19
Prima
PS D:\Belajar Html\Modul 14> go run "d:\Belajar Html\Modul 14\latsol2\latsol2.go"
72
Bukan prima
PS D:\Belajar Html\Modul 14> 
```

## **Deskripsi Program:**

### **1. Deklarasi variabel:**

```
var bilangan int
```

**Program mendeklarasikan variabel integer bilangan untuk menyimpan nilai input.**

### **2. Membaca input:**

```
fmt.Scan(&bilangan)
```

**Program membaca nilai integer yang diinputkan dan menyimpannya dalam variabel bilangan.**

### **3. Memeriksa apakah bilangan lebih kecil atau sama dengan 1:**

```
if bilangan <= 1 {
```

```
    fmt.Println("Bukan prima")
```

```
} else {
```

**Program memeriksa apakah bilangan lebih kecil atau sama dengan 1. Jika ya, maka bilangan tersebut bukan bilangan prima.**

### **4. Inisialisasi variabel prima dan perulangan untuk memeriksa bilangan prima:**

```
prima := true
```

```
for i := 2; i*i <= bilangan; i++ {
```

```
    if bilangan%i == 0 {
```

```
        prima = false
```

```
        break
```

```
    }
```

```
}
```

**Jika bilangan lebih besar dari 1, program menginisialisasi variabel prima dengan nilai true. Kemudian, program menjalankan loop dari 2 hingga akar kuadrat dari bilangan. Di dalam loop, program memeriksa apakah bilangan habis dibagi i. Jika ya, maka bilangan bukan bilangan prima dan variabel prima diatur ke false, kemudian loop dihentikan.**

### **5. Mencetak hasil:**

```
if prima {
```

```
    fmt.Println("Prima")
```

```
} else {
```

```
    fmt.Println("Bukan prima")
```

}

Setelah loop selesai, program memeriksa nilai variabel `prima`. Jika `prima` bernilai `true`, maka bilangan adalah bilangan prima dan program mencetak "Prima". Jika `prima` bernilai `false`, maka bilangan bukan bilangan prima dan program mencetak "Bukan prima".

Secara keseluruhan, program ini efektif untuk memeriksa dan mencetak apakah sebuah bilangan adalah bilangan prima atau bukan.

***DAFTAR PUSTAKA***

***MODUL PRAKTIKUM 14 - KOMPOSISI***