

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 1**

MODUL 5



Disusun Oleh :

Rizki Agung Prasetya / 103112400263

IF-12-05

Asisten Praktikum :

Ayu Susilowati

Noviana Rizki Anisa Putri

Dosen Pengampu :

Yudha Islami Sulistya

PROGRAM STUDI S1 TEKNIK INFORMATIKA

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2024

TUGAS PENDAHULUAN

A. PRAKTIKUM Menghitung Deret Angka dari 1 Sampai n

Soal Studi Case

Buatlah sebuah program dalam bahasa Go untuk menjumlahkan deret angka yang meminta input berupa bilangan bulat positif n, kemudian menjumlahkan semua angka dari 1 hingga n. Petunjuk :

- Program akan meminta input n.
- Program harus menghitung jumlah dari deret angka 1 hingga n.
- Contoh : Jika input $n = 3$, maka program harus menjumlahkan angka 1 hingga 3

Sourcecode

```
package main

import (
    "fmt"
)

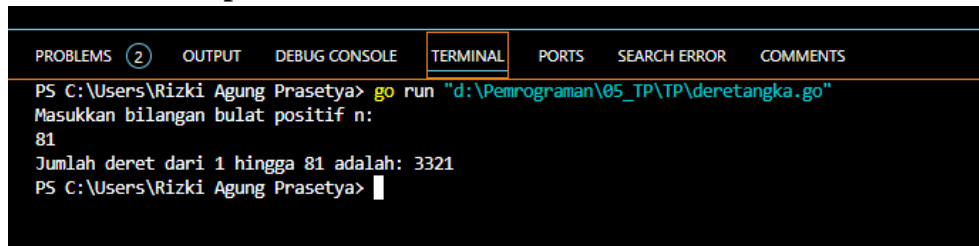
func main() {
    // Meminta pengguna memasukkan bilangan bulat positif
    var n int
    fmt.Println("Masukkan bilangan bulat positif n: ")
    fmt.Scan(&n) // Membaca input dari pengguna

    // Jika n lebih besar dari 0, lakukan penjumlahan
    if n > 0 {
        total := 0 // Variabel untuk menyimpan total penjumlahan

        // Perulangan untuk menjumlahkan angka dari 1 hingga n
        for i := 1; i <= n; i++ {
            total = total + i // Menambahkan angka i ke total
        }

        // Menampilkan hasil penjumlahan
        fmt.Println("Jumlah deret dari 1 hingga", n, "adalah:", total)
    } else {
        // Jika input tidak valid, tampilkan pesan kesalahan
        fmt.Println("Masukkan bilangan bulat positif yang valid!")
    }
}
```

Screenshoot Output



```
PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL PORTS SEARCH ERROR COMMENTS
PS C:\Users\Rizki Agung Prasetya> go run "d:\Pemrograman\05_TP\TP\deretangka.go"
Masukkan bilangan bulat positif n:
81
Jumlah deret dari 1 hingga 81 adalah: 3321
PS C:\Users\Rizki Agung Prasetya> |
```

Deskripsi Program

Program ini meminta pengguna memasukkan bilangan bulat positif n . Jika n valid (lebih besar dari 0), program menghitung jumlah deret angka dari 1 hingga n menggunakan perulangan `for`. Hasil penjumlahan ditampilkan setelah perulangan selesai. Jika pengguna memasukkan angka tidak valid (kurang dari atau sama dengan 0), program menampilkan pesan kesalahan.

Langkah-langkah utama:

- Minta input bilangan bulat positif.
- Periksa apakah n lebih dari 0.
- Jika ya, gunakan perulangan untuk menjumlahkan angka dari 1 hingga n .
- Tampilkan hasil penjumlahan.
- Jika tidak valid, tampilkan pesan kesalahan.

B. PRAKTIKUM Mencetak Segitiga Bintang

Soal Studi Case

Buatkan sebuah program yang dapat mencetak segitiga Bintang berdasarkan jumlah yang diinputkan Petunjuk :

- a. Program akan meminta sebuah input
- b. Program akan menampilkan output berupa perulangan Bintang sebanyak inputan yang dimasukkan

Sourcecode

```
package main

import (
    "fmt"
)

func main() {
    // Deklarasi variabel untuk menyimpan jumlah baris
    var n int
```

```

// Meminta pengguna memasukkan jumlah baris segitiga bintang
fmt.Print("Masukkan jumlah baris segitiga bintang: ")
fmt.Scan(&n)

// Memeriksa apakah jumlah baris yang dimasukkan lebih dari 0
if n > 0 {
    // Perulangan untuk mencetak segitiga bintang
    for i := 1; i <= n; i++ {
        // Perulangan untuk mencetak bintang di setiap baris
        for j := 1; j <= i; j++ {
            // Mencetak bintang tanpa pindah ke baris baru
            fmt.Print("*")
        }
        // Pindah ke baris baru setelah mencetak bintang pada satu baris
        fmt.Println()
    }
} else {
    // Jika input tidak valid, beri pesan kesalahan
    fmt.Println("Masukkan jumlah baris yang valid (lebih dari 0)!")
}
}

```

Screenshoot Output



Deskripsi Program

1. Program ini mencetak segitiga bintang berdasarkan input jumlah baris n . Jika $n > 0$, program menggunakan dua perulangan bersarang: perulangan luar untuk menentukan jumlah baris, dan perulangan dalam untuk mencetak bintang di setiap baris. Jika input tidak valid ($n \leq 0$), program menampilkan pesan kesalahan.

Langkah-langkah utama:

1. **Input:** Program meminta pengguna memasukkan jumlah baris n .
2. **Validasi:** Jika $n > 0$, program melanjutkan. Jika tidak, pesan kesalahan ditampilkan.
3. **Perulangan:**
 - o Perulangan luar (`for i := 1; i <= n; i++`) mengontrol jumlah baris.
 - o Perulangan dalam (`for j := 1; j <= i; j++`) mencetak bintang sebanyak nilai i di setiap baris.
4. Setelah mencetak bintang di setiap baris, program pindah ke baris baru dengan `fmt.Println()`.

c. Praktikum Menampilkan Bilangan Genap Dari 1 Sampai 50

Soal Studi Case

Buatlah sebuah program yang dapat menghitung dan menampilkan bilangan genap dari 1 hingga 50 Petunjuk :

1. Program akan menampilkan bilangan genap yang ada dari 1-50

Sourcecode

```
package main

import (
    "fmt"
)

func main() {
    // Menampilkan bilangan genap dari 1 hingga 50
    fmt.Println("Bilangan genap dari 1 hingga 50 adalah:")

    // Perulangan dari 1 hingga 50
    for i := 1; i <= 50; i++ {
        // Memeriksa apakah i adalah bilangan genap
        if i%2 == 0 {
            fmt.Println(i) // Jika genap, cetak angka
        }
    }
}
```

Screenshoot Output



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  SEARCH ERROR  COMMENTS
PS C:\Users\Rizki Agung Prasetya> go run "d:\Pemrograman\05_TP\TP\bilangangenap.go"
Bilangan genap dari 1 hingga 50 adalah:
2
4
6
8
10
12
14
16
18
20
22
```

Deskripsi Program

Program ini berfungsi untuk menampilkan semua bilangan genap dari 1 hingga 50 menggunakan perulangan dan kondisi sederhana.

Langkah-langkah:

1. **Perulangan:** Program melakukan perulangan dari angka 1 hingga 50 menggunakan `for i := 1; i <= 50; i++`.
2. **Cek bilangan genap:** Setiap angka diperiksa apakah genap menggunakan kondisi `if i%2 == 0`.
3. **Cetak:** Jika angka genap, program mencetaknya dengan `fmt.Println(i)`.

Hasilnya adalah semua bilangan genap dari 2 hingga 50 ditampilkan.

D. Praktikum Membuat Program Tebak angka

Soal Studi Case

Buatlah program yang dapat menjalankan permainan tebak angka. Program akan secara acak memilih sebuah angka antara 1 hingga 100. Pengguna diberikan kesempatan hingga 5 kali untuk menebak angka tersebut hingga benar. Program akan memberikan petunjuk apakah angka yang ditebak terlalu besar, terlalu kecil, ataupun sudah benar. Jika pengguna berhasil menebak sebelum pada percobaan ke 5, maka permainan berakhir!

Sourcecode

```
package main

import "fmt"

func main() {
    // Angka target ditentukan langsung sebagai 42 untuk
    kesederhanaan
    target := 81
    var guess int

    fmt.Println("Tebak angka antara 1 hingga 100. Anda punya 5 kali
    kesempatan.")

    // Loop untuk 5 kesempatan
    for i := 1; i <= 5; i++ {
        fmt.Printf("Percobaan %d: Masukkan tebakan Anda: ", i)
        fmt.Scan(&guess)
```

```

    // Cek apakah tebakan benar
    if guess == target {
        fmt.Println("Selamat! Tebakan Anda benar.")
        return
    } else if guess < target {
        fmt.Println("Terlalu kecil!")
    } else {
        fmt.Println("Terlalu besar!")
    }
}

// Jika tidak berhasil dalam 5 kali percobaan
fmt.Println("Maaf, Anda kehabisan kesempatan. Angka yang benar
adalah:", target)
}

```

Screenshoot Output

```

PS C:\Users\Rizki Agung Prasetya> go run "d:\Pemrograman\05_TP\TP\tebakangka.go"
Tebak angka antara 1 hingga 100. Anda punya 5 kali kesempatan.
Percobaan 1: Masukkan tebakan Anda: 87
Terlalu besar!
Percobaan 2: Masukkan tebakan Anda: 50
Terlalu kecil!
Percobaan 3: Masukkan tebakan Anda: 69
Terlalu kecil!
Percobaan 4: Masukkan tebakan Anda: 80
Terlalu kecil!
Percobaan 5: Masukkan tebakan Anda: 81
Selamat! Tebakan Anda benar.
PS C:\Users\Rizki Agung Prasetya>

```

Deskripsi Program

Program ini adalah permainan **tebak angka** sederhana, di mana pengguna diminta untuk menebak sebuah angka (yang dalam kode ini telah ditentukan sebagai 81). Pengguna diberikan 5 kali kesempatan untuk menebak angka tersebut dengan petunjuk apakah tebakan terlalu kecil, terlalu besar, atau benar.

1. Tentukan angka target: Angka target ditetapkan sebagai 81
2. Berikan instruksi: Program memberi tahu pengguna untuk menebak angka antara 1 hingga 100, dengan 5 kesempatan.
3. Perulangan 5 kali: Program meminta pengguna memasukkan tebakan hingga 5 kali.
4. Cek Tebakan:
 - Jika tebakan benar, program mencetak pesan kemenangan dan berhenti.
 - Jika tebakan terlalu kecil, program memberi tahu bahwa tebakan terlalu kecil.
 - Jika tebakan terlalu besar, program memberi tahu bahwa tebakan terlalu besar.
5. Kehabisan kesempatan : jika pengguna tidak berhasil menebak dalam 5 kali percobaan, program akan menampilkan angka yang benar.