

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN**

**MODUL V
TUGAS PENDAHULUAN MODUL 5**



**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 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 .

Sourcecode

```
package main

import (
    "fmt"
)

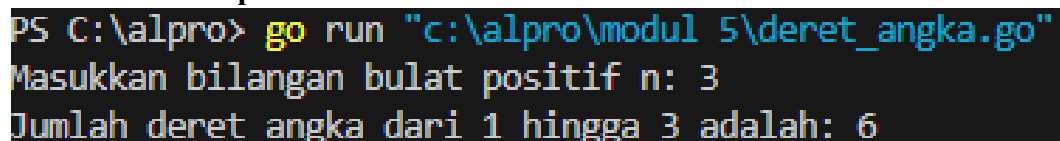
// Fungsi rekursif untuk menjumlahkan angka dari 1 hingga n
func sum(n int) int {
    if n == 1 {
        return 1
    }
    return n + sum(n-1)
}

func main() {
    var n int

    // Meminta input dari user
    fmt.Print("Masukkan bilangan bulat positif n: ")
    fmt.Scan(&n)

    // Memanggil fungsi rekursif dan menampilkan hasilnya
    fmt.Printf("Jumlah deret angka dari 1 hingga %d adalah: %d\n", n, sum(n))
}
```

Screenshoot Output



```
PS C:\alpro> go run "c:\alpro\modul 5\deret_angka.go"
Masukkan bilangan bulat positif n: 3
Jumlah deret angka dari 1 hingga 3 adalah: 6
```

a. Deskripsi Program

Program diatas meminta untuk memasukan angka bulat positif (n) . Setelah itu, dia bakal pakai fungsi yang namanya `sum` buat ngitung semua angka dari 1 sampai (n) . Fungsi ini terus manggil dirinya sendiri sambil ngurangin nilai (n) sampai dia sampai ke 1. Pas semua angka udah dijumlahin, hasilnya bakal ditampilkan.

b. Algoritma Program

- Mulai
- Minta Input: Tanyakan kepada pengguna untuk memasukkan bilangan bulat positif n
- Definisikan Fungsi Rekursif: Buat fungsi $\text{sum}(n)$:
- Base Case: Jika n sama dengan 1, kembalikan 1
- Rekurens: Kembalikan $n + \text{sum}(n-1)$
- Panggil Fungsi: Panggil $\text{sum}(n)$ dan simpan hasilnya
- Tampilkan Hasil: Tampilkan hasil penjumlahan kepada pengguna
- Selesai

c. Cara Kerja

- Program meminta pengguna untuk memasukkan angka bulat positif n
- Kemudian, program memanggil fungsi $\text{sum}(n)$, yang akan menjumlahkan angka dari 1 hingga n secara rekursif
- Fungsi sum memeriksa apakah n sudah mencapai 1; jika iya, ia mengembalikan 1. Jika belum, ia menambahkan nilai n dengan hasil dari $\text{sum}(n-1)$
- Proses ini berlanjut sampai n mencapai 1, dan hasil akhirnya ditampilkan kepada pengguna

2. Soal Studi Case

Buatkan sebuah program yang dapat mencetak segitiga Bintang berdasarkan jumlah yang diinputkan.

Sourcecode

```
package main

import (
    "fmt"
)

// Fungsi rekursif untuk mencetak bintang pada setiap baris
func printStars(n, i int) {
    if i > n {
        return
    }
    fmt.Println(starString(i))
    printStars(n, i+1)
}

// Fungsi rekursif untuk membuat string bintang
func starString(k int) string {
    if k == 0 {
        return ""
    }
    return "*" + starString(k-1)
}

func main() {
    var n int
    fmt.Print("Masukkan jumlah baris: ")
}
```

```

        fmt.Scan(&n)

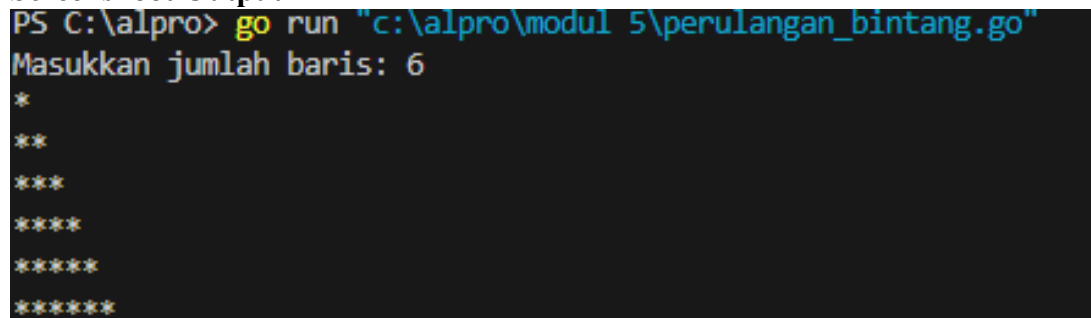
        printStars(n, 1)
    }

    // Hitung total gaji bulanan
    totalGaji = (jamNormal * upahPerJam * 4) + (lembur * 1.5 *
    upahPerJam * 4)

    // Tampilkan cara menghitung dan hasil
    fmt.Printf("Perhitungan:\n")
    fmt.Printf("Jam Normal: %.2f jam\n", jamNormal)
    fmt.Printf("Lembur: %.2f jam\n", lembur)
    fmt.Printf("Gaji dari jam normal: %.2f x %.2f x 4 = %.2f\n",
    jamNormal, upahPerJam, jamNormal*upahPerJam*4)
    fmt.Printf("Gaji dari lembur: %.2f x 1.5 x %.2f x 4 =
    %.2f\n", lembur, upahPerJam, lembur*1.5*upahPerJam*4)
    fmt.Printf("Total gaji bulanan: %.2f\n", totalGaji)
}

```

Screenshoot Output



```

PS C:\alpro> go run "c:\alpro\modul 5\perulangan_bintang.go"
Masukkan jumlah baris: 6
*
**
***
****
*****
*****

```

a. Deskripsi Program

Program ini untuk mencetak bintang dalam bentuk segitiga berdasarkan jumlah baris yang dimasukkan oleh pengguna. Setelah pengguna memasukkan jumlah baris n , program memanggil fungsi rekursif `printStars`, yang akan mencetak bintang. Fungsi ini memanggil fungsi `starString` untuk menghasilkan string bintang, di mana jumlah bintang yang dicetak bertambah satu setiap barisnya.

b. Algoritma Program

- Mulai
- Minta Input: Tanyakan kepada pengguna untuk memasukkan jumlah baris n
- Panggil Fungsi `printStars(n, 1)`: Memulai proses pencetakan bintang dengan baris pertama
- Definisikan Fungsi `printStars(n, i)`:
- Base Case: Jika i lebih besar dari n , hentikan eksekusi
- Cetak Bintang: Panggil `starString(i)` untuk mencetak bintang sebanyak i pada baris tersebut
- Panggil Diri Sendiri: Panggil `printStars(n, i+1)` untuk mencetak baris berikutnya
- Definisikan Fungsi `starString(k)`:
- Base Case: Jika k sama dengan 0, kembalikan string kosong

- Konstruksi String Bintang: Kembalikan string bintang yang terdiri dari satu bintang ditambah hasil dari `starString(k-1)`
- **Selesai**

e. **Cara Kerja**

- Program meminta pengguna untuk memasukkan jumlah baris `nnn` untuk segitiga bintang
- Fungsi `printStars` dipanggil dengan parameter `nnn` dan indeks baris `iii` yang dimulai dari 1
- Fungsi ini mencetak bintang dengan memanggil fungsi `starString`, yang membentuk string bintang sesuai dengan baris yang diinginkan
- Setelah mencetak bintang untuk baris saat ini, `printStars` dipanggil lagi dengan `iii` yang ditingkatkan, hingga semua baris selesai dicetak

3. Soal Studi Case

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

Sourcecode

```
package main

import (
    "fmt"
)

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

    printEven(2) // Memulai dari angka genap pertama
}

// Fungsi rekursif untuk mencetak bilangan genap
func printEven(n int) {
    if n > 50 {
        return
    }
    fmt.Println(n)
    printEven(n + 2) // Melanjutkan ke bilangan genap berikutnya
}
```

Screenshoot Output

```
PS C:\alpro> go run "c:\alpro\modul 5\bil_genap.go"
Bilangan genap dari 1 hingga 50:
2
4
6
8
10
12
14
16
18
20
22
24
26
28
30
32
34
36
38
40
42
44
46
48
50
```

a. Deskripsi Program

Program ini untuk mencetak bilangan genap dari 1 sampai 50 dengan cara yang simpel. Pertama nyetak judul "Bilangan genap dari 1 hingga 50:". Lalu, panggil fungsi `printEven` mulai dari angka genap pertama, yaitu 2. Di dalam fungsi `printEven`, program akan terus mencetak angka genap sampai angka tersebut lebih dari 50. Setiap kali dia nyetak angka, dia panggil lagi fungsi itu dengan menambah 2, jadi semua angka genap tercetak dengan rapi.

b. Algoritma Program

- Mulai
- Cetak Judul: Tampilkan "Bilangan genap dari 1 hingga 50:"
- Panggil Fungsi printEven(2): Mulai dari angka genap pertama yaitu 2
- Definisikan Fungsi printEven(n):
- Base Case: Jika nnn lebih besar dari 50, hentikan eksekusi.
- Cetak nnn: Tampilkan angka genap saat ini
- Panggil Diri Sendiri: Panggil printEven($n + 2$) untuk melanjutkan ke bilangan genap berikutnya
- Selesai

c. Cara Kerja

- Program mulai dengan mencetak judul "Bilangan genap dari 1 hingga 50:"
- Fungsi `printEven` dipanggil dengan parameter 2, yang merupakan bilangan genap pertama
- Di dalam fungsi `printEven`, program memeriksa apakah `nnn` lebih besar dari 50. Jika tidak, program mencetak angka tersebut
- Kemudian, fungsi memanggil dirinya sendiri dengan menambahkan 2 ke nilai `nnn`, sehingga angka genap berikutnya akan dicetak
- Proses ini berulang hingga semua bilangan genap sampai 50 dicetak

4. 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"
    "math/rand"
    "time"
)

func main() {
    rand.Seed(time.Now().UnixNano()) // Menginisialisasi
    generator angka acak
    target := rand.Intn(100) + 1      // Memilih angka acak antara
    1 hingga 100

    fmt.Println("Tebak angka antara 1 hingga 100. Anda punya 5
    kali kesempatan!")
    playGame(target, 1) // Memulai permainan dengan percobaan
    pertama
}

func playGame(target, attempt int) {
    var guess int

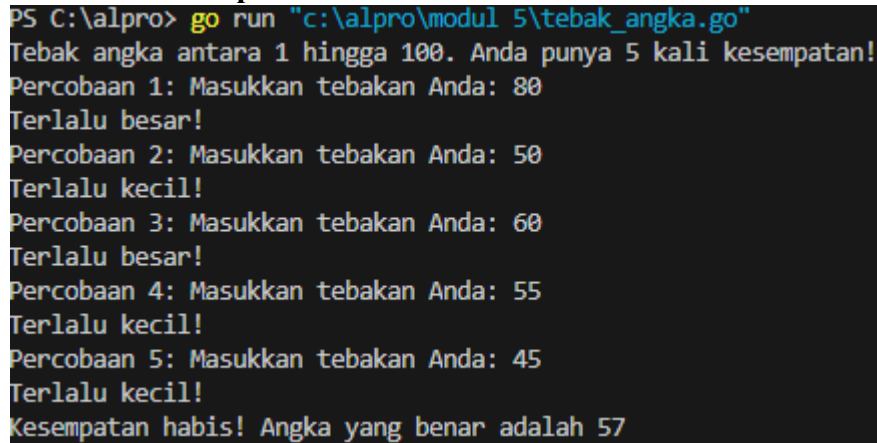
    // Jika percobaan melebihi 5, permainan berakhir
    if attempt > 5 {
        fmt.Printf("Kesempatan habis! Angka yang benar adalah
        %d\n", target)
        return
    }

    fmt.Printf("Percobaan %d: Masukkan tebakan Anda: ", attempt)
    fmt.Scan(&guess)

    // Cek apakah tebakan benar, terlalu besar, atau kecil
    if guess == target {
```

```
        fmt.Println("Selamat! Anda berhasil menebak angka dengan benar.")
    } else if guess < target {
        fmt.Println("Terlalu kecil!")
        playGame(target, attempt+1)
    } else {
        fmt.Println("Terlalu besar!")
        playGame(target, attempt+1)
    }
}
```

Screenshoot Output



```
PS C:\alpro> go run "c:\alpro\modul 5\tebak_angka.go"
Tebak angka antara 1 hingga 100. Anda punya 5 kali kesempatan!
Percobaan 1: Masukkan tebakan Anda: 80
Terlalu besar!
Percobaan 2: Masukkan tebakan Anda: 50
Terlalu kecil!
Percobaan 3: Masukkan tebakan Anda: 60
Terlalu besar!
Percobaan 4: Masukkan tebakan Anda: 55
Terlalu kecil!
Percobaan 5: Masukkan tebakan Anda: 45
Terlalu kecil!
Kesempatan habis! Angka yang benar adalah 57
```

a. Deskripsi Program

Untuk menebak angka dengan memilih angka acak antara 1 hingga 100. Setelah itu, pemain diberikan 5 kesempatan untuk menebak angka tersebut. Setiap kali pemain memasukkan tebakan, program akan memberi tahu apakah tebakan itu terlalu kecil, terlalu besar, atau benar. Jika pemain tidak berhasil menebak dalam 5 kali percobaan, program akan memberi tahu angka yang benar.

b. Algoritma Program

- Mulai
- Inisialisasi: Pilih angka acak antara 1-100
- Tampilkan Pesan: Informasikan pemain tentang 5 kesempatan
- Panggil `playGame(target, 1)`
- Definisikan `playGame(target, attempt)`:
- Jika `attempt > 5`, cetak angka benar dan berhenti
- Minta tebakan pemain
- Cek tebakan:
- Jika benar, cetak pesan sukses
- Jika terlalu kecil/besar, panggil `playGame` dengan `attempt + 1`
- Selesai

c. **Cara Kerja**

Cara kerja program ini dimulai dengan memilih angka acak antara 1 hingga 100. Kemudian, pemain diberikan 5 kesempatan untuk menebak angka tersebut. Setiap tebakan yang dimasukkan akan diperiksa, dan program memberikan clue apakah tebakan terlalu kecil, terlalu besar, atau benar. Jika pemain menebak dengan benar, permainan berakhir, tetapi jika tidak, mereka bisa terus mencoba hingga mencapai batas 5 kali percobaan. Jika kesempatan habis, program akan menampilkan angka yang benar.