

**LAPORAN PRAKTIKUM  
ALGORITMA PEMROGRAMAN 2**

**MODUL 5  
REKURSIF**



Oleh:

Ben Waiz Pintus Widyosaputro

2311102169

IF-11-02

**S1 TEKNIK INFORMATIKA  
INSTITUT TEKNOLOGI TELKOM PURWOKERTO**

**2024**

## **I. DASAR TEORI**

### **1. Pengertian :**

Rekursif adalah suatu fungsi yang memanggil dirinya sendiri, baik secara langsung maupun tidak langsung, dengan input yang semakin menyempit hingga mencapai kondisi dasar. Tujuannya adalah untuk memecahkan masalah yang kompleks menjadi sub-masalah yang lebih sederhana dan solvable.

### **2. Komponen Utama Fungsi Rekursif :**

Base Case (Kasus Dasar):

- Merupakan kondisi di mana fungsi tidak akan memanggil dirinya sendiri lagi dan langsung memberikan hasil. Base case diperlukan untuk menghentikan rekursi dan mencegah terjadinya loop tak berujung.

Recursive Case (Kasus Rekursif):

- Bagian ini adalah di mana fungsi memanggil dirinya sendiri dengan versi yang lebih sederhana atau lebih kecil dari masalah aslinya. Hal ini memungkinkan fungsi untuk terus memecahkan masalah menjadi ukuran yang lebih kecil hingga mencapai base case.

### **3. Struktur Logika Rekursif :**

Proses rekursif dapat dijelaskan sebagai berikut:

1. Panggilan Awal: Fungsi dimulai dengan memanggil dirinya sendiri dengan input yang sesuai.
2. Iterasi: Fungsi kemudian memanggil dirinya sendiri dengan input yang telah diminimalkan.
3. Termination: Ketika mencapai base case, fungsi tidak memanggil dirinya sendiri lagi dan mengembalikan hasil.

## II. GUIDED

### Guided 1

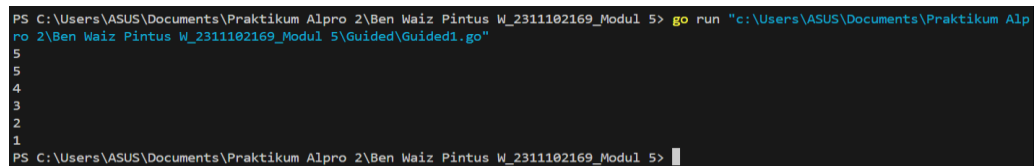
```
package main

import "fmt"

func main() {
    var n int
    fmt.Scan(&n)
    baris(n)
}

func baris(bilangan int) {
    if bilangan == 1 {
        fmt.Println(1)
    } else {
        fmt.Println(bilangan)
        baris(bilangan - 1)
    }
}
```

Screenshot output



```
PS C:\Users\ASUS\Documents\Praktikum Alpro 2\Ben Waiz Pintus W_2311102169_Modul 5> go run "c:\Users\ASUS\Documents\Praktikum Alpro 2\Ben Waiz Pintus W_2311102169_Modul 5\Guided\Guided1.go"
5
5
4
3
2
1
PS C:\Users\ASUS\Documents\Praktikum Alpro 2\Ben Waiz Pintus W_2311102169_Modul 5> |
```

Penjelasan :

Program meminta pengguna untuk memasukkan sebuah angka  $n$  dan mencetak deretan angka mulai dari angka  $n$  hingga 1 secara menurun. Pertama, nilai  $n$  dibaca menggunakan fungsi `fmt.Scan()`. Kemudian, fungsi `baris()` dipanggil dengan parameter  $n$ . Fungsi `baris()` adalah fungsi rekursif yang mencetak angka yang diterimanya dan kemudian memanggil dirinya sendiri dengan angka yang lebih kecil (yaitu  $\text{bilangan} - 1$ ) sampai mencapai angka 1. Ketika nilai yang diterima adalah 1, fungsi akan mencetak 1 dan menghentikan rekursi.

## Guided 2

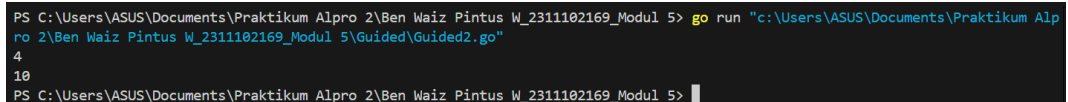
```
package main

import "fmt"

func main() {
    var n int
    fmt.Scan(&n)
    fmt.Println(penjumlahan(n))
}

func penjumlahan(n int) int {
    if n == 1 {
        return 1
    } else {
        return n + penjumlahan(n-1)
    }
}
```

Output :



```
PS C:\Users\ASUS\Documents\Praktikum Alpro 2\Ben Waiz Pintos W_2311102169_Modul 5> go run "c:\Users\ASUS\Documents\Praktikum Alpro 2\Ben Waiz Pintos W_2311102169_Modul 5\Guided\Guided2.go"
4
10
PS C:\Users\ASUS\Documents\Praktikum Alpro 2\Ben Waiz Pintos W_2311102169_Modul 5>
```

Penjelasan :

Program meminta pengguna untuk memasukkan sebuah angka  $n$  dan menghitung jumlah total dari semua bilangan bulat dari 1 hingga  $n$ . Pertama, program membaca input menggunakan `fmt.Scan()` dan kemudian memanggil fungsi rekursif `penjumlahan(n)`. Fungsi `penjumlahan()` akan terus menambahkan nilai  $n$  dengan hasil dari `penjumlahan(n-1)` sampai mencapai kondisi dasar di mana  $n == 1$ , yang akan mengembalikan nilai 1. Hasil akhir, yaitu penjumlahan dari semua angka dari 1 hingga  $n$ , akan ditampilkan menggunakan `fmt.Println()`.

### III. UNGUIDED

#### Unguided 1

```
//Ben Waiz Pintus W
//2311102169

package main

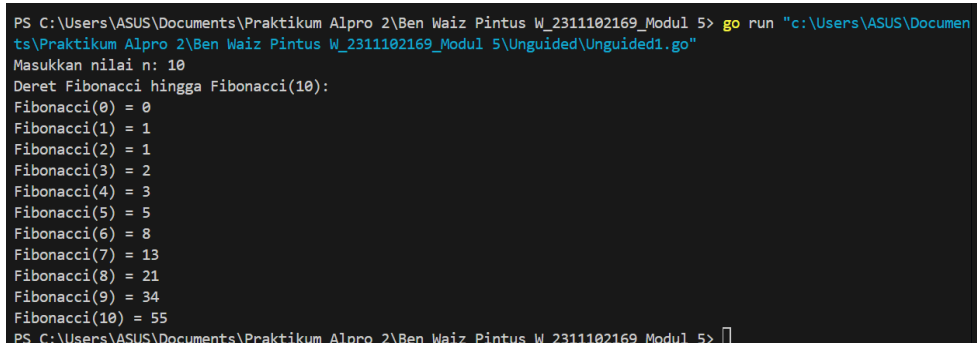
import "fmt"

func fibonacci(n int) int {
    if n <= 1 {
        return n
    }
    return fibonacci(n-1) + fibonacci(n-2)
}

func main() {
    var n int
    fmt.Print("Masukkan nilai n: ")
    fmt.Scan(&n)

    fmt.Printf("Deret Fibonacci hingga Fibonacci(%d):\n", n)
    for i := 0; i <= n; i++ {
        fmt.Printf("Fibonacci(%d) = %d\n", i, fibonacci(i))
    }
}
```

Screenshot output



```
PS C:\Users\ASUS\Documents\Praktikum Alpro 2\Ben Waiz Pintus W_2311102169_Modul 5> go run "c:\Users\ASUS\Documents\Praktikum Alpro 2\Ben Waiz Pintus W_2311102169_Modul 5\Unguided\Unguided1.go"
Masukkan nilai n: 10
Deret Fibonacci hingga Fibonacci(10):
Fibonacci(0) = 0
Fibonacci(1) = 1
Fibonacci(2) = 1
Fibonacci(3) = 2
Fibonacci(4) = 3
Fibonacci(5) = 5
Fibonacci(6) = 8
Fibonacci(7) = 13
Fibonacci(8) = 21
Fibonacci(9) = 34
Fibonacci(10) = 55
PS C:\Users\ASUS\Documents\Praktikum Alpro 2\Ben Waiz Pintus W_2311102169_Modul 5> █
```

#### Penjelasan

Untuk menghitung deret Fibonacci, fungsi rekursif `fibonacci(n)` digunakan untuk menjumlahkan dua nilai sebelumnya, kecuali nilai dasar 0 dan 1, dan kemudian mencetak deret Fibonacci dari 0 hingga `n`. Setelah mencapai nilai dasar, fungsi rekursif memanggil dirinya sendiri, dan proses iteratif di

bagian main() mencetak hasil Fibonacci untuk setiap angka dari 0 hingga nilai yang dimasukkan.

## Unguided 2

```
//Ben Waiz Pintus W
//2311102169

package main

import "fmt"

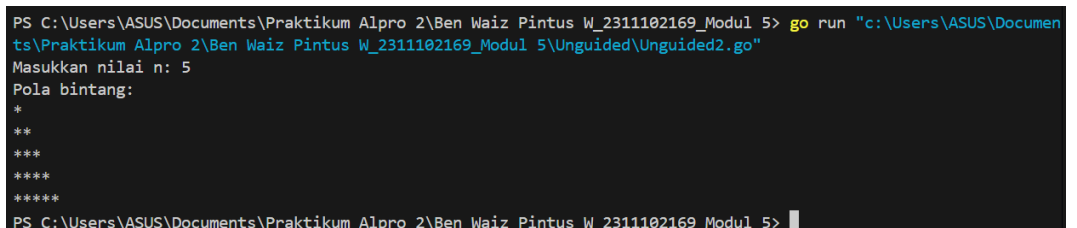
func cetakBintang(n int) {
    if n == 0 {
        return
    }
    fmt.Print("*")
    cetakBintang(n - 1)
}

func cetakPola(n, current int) {
    if current > n {
        return
    }
    cetakBintang(current)
    fmt.Println()
    cetakPola(n, current+1)
}

func main() {
    var n int
    fmt.Print("Masukkan nilai n: ")
    fmt.Scan(&n)

    fmt.Println("Pola bintang:")
    cetakPola(n, 1)
}
```

Screenshot output



```
PS C:\Users\ASUS\Documents\Praktikum Alpro 2\Ben Waiz Pintus W_2311102169_Modul 5> go run "c:\Users\ASUS\Documents\Praktikum Alpro 2\Ben Waiz Pintus W_2311102169_Modul 5\Unguided\Unguided2.go"
Masukkan nilai n: 5
Pola bintang:
*
**
***
****
*****
PS C:\Users\ASUS\Documents\Praktikum Alpro 2\Ben Waiz Pintus W_2311102169_Modul 5>
```

Penjelasan :

Program tersebut berfungsi untuk mencetak pola bintang secara bertahap menggunakan dua fungsi rekursif, cetakBintang(n) dan cetakPola(n, current). Fungsi cetakBintang(n) mencetak sejumlah n bintang pada satu baris, sementara cetakPola(n, current) mencetak pola bertahap mulai dari 1 hingga n baris. Program menerima input dari pengguna berupa nilai n, kemudian mencetak pola bintang bertingkat mulai dari 1 bintang hingga n bintang pada baris terakhir.

### Unguided 3

```
//Ben Waiz Pintus W
//2311102169

package main

import "fmt"

func cetakFaktor(n, i int) {
    if i > n {
        return
    }
    if n%i == 0 {
        fmt.Printf("%d ", i)
    }
    cetakFaktor(n, i+1)
}

func main() {
    var n int
    fmt.Print("Masukkan nilai N: ")
    fmt.Scan(&n)

    fmt.Printf("Nilai faktor dari %d adalah: ", n)
    cetakFaktor(n, 1)
    fmt.Println()
}
```

Screenshot output

```
PS C:\Users\ASUS\Documents\Praktikum Alpro 2\Ben Waiz Pintus W_2311102169_Modul 5> go run "c:\Users\ASUS\Documents\Praktikum Alpro 2\Ben Waiz Pintus W_2311102169_Modul 5\Unguided\Unguided3.go"
Masukkan nilai N: 12
Nilai faktor dari 12 adalah: 1 2 3 4 6 12
PS C:\Users\ASUS\Documents\Praktikum Alpro 2\Ben Waiz Pintus W_2311102169_Modul 5> |
```

Penjelasan :

Program menggunakan fungsi rekursif untuk mencetak semua faktor dari sebuah bilangan bulat positif N. Fungsi cetakFaktor(n, i) memeriksa apakah setiap angka i dari 1 hingga N habis membagi N, dan jika iya, angka tersebut dicetak sebagai faktor. Fungsi ini dipanggil secara rekursif dengan menaikkan nilai i hingga mencapai N. Setelah program menerima input N dari pengguna, ia kemudian mencetak faktor-faktor dari bilangan tersebut dalam urutan dari 1 hingga N.

#### Unguided 4

```
//Ben Waiz Pintus W
//2311102169

package main

import "fmt"

func cetakUrutan(jumlah int) {
    if jumlah == 1 {
        fmt.Print(jumlah, " ")
        return
    }
    fmt.Print(jumlah, " ")
    cetakUrutan(jumlah - 1)
    fmt.Print(jumlah, " ")
}

func main() {
    var jumlah int
    fmt.Print("Masukkan jumlah bilangan: ")
    fmt.Scanln(&jumlah)

    fmt.Print("Keluaran : ")
    cetakUrutan(jumlah)
    fmt.Println()
}
```

Screenshot output

```
PS C:\Users\ASUS\Documents\Praktikum Alpro 2\Ben Waiz Pintus W_2311102169_Modul 5> go run "c:\Users\ASUS\Documents\Praktikum Alpro 2\Ben Waiz Pintus W_2311102169_Modul 5\Unguided\Unguided4.go"
Masukkan jumlah bilangan: 9
Keluaran : 9 8 7 6 5 4 3 2 1 2 3 4 5 6 7 8 9
PS C:\Users\ASUS\Documents\Praktikum Alpro 2\Ben Waiz Pintus W_2311102169_Modul 5> █
```



Penjelasan :

Program mencetak urutan angka dari n hingga 1, kemudian kembali mencetak urutan angka dari 1 hingga n tanpa mencetak angka 1 dua kali. Pada kondisi dasar, rekursi berhenti ketika jumlah sama dengan 1, dan fungsi rekursif cetak urutan(jumlah) menampilkan angka jumlah dan memanggil dirinya sendiri. Setelah itu, saat kembali ke tahap awal, program mencetak angka dari 2 hingga n. Ini menghasilkan pola simetris yang tidak duplikasi angka 1.

### Unguided 5

```
//Ben Waiz Pintus W
//2311102169

package main

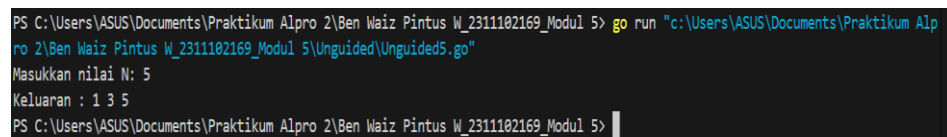
import "fmt"

func cetakBilanganGanjil(n int) {
    if n < 1 {
        return
    }
    cetakBilanganGanjil(n - 2)
    if n%2 != 0 {
        fmt.Print(n, " ")
    }
}

func main() {
    var N int
    fmt.Print("Masukkan nilai N: ")
    fmt.Scanln(&N)

    fmt.Print("Keluaran : ")
    cetakBilanganGanjil(N)
    fmt.Println()
}
```

Screenshot output



```
PS C:\Users\ASUS\Documents\Praktikum Alpro 2\Ben Waiz Pintus W_2311102169_Modul 5> go run "c:\Users\ASUS\Documents\Praktikum Alpro 2\Ben Waiz Pintus W_2311102169_Modul 5\Unguided\Unguided5.go"
Masukkan nilai N: 5
Keluaran : 1 3 5
PS C:\Users\ASUS\Documents\Praktikum Alpro 2\Ben Waiz Pintus W_2311102169_Modul 5>
```

Penjelasan :

Ketika program dijalankan, pengguna akan diminta memasukkan nilai N. Fungsi cetakBilanganGanjil kemudian akan dipanggil secara rekursif untuk mencetak bilangan ganjil. Fungsi ini bekerja dengan cara membagi masalah menjadi submasalah yang lebih kecil, yaitu dengan mengurangi nilai N sebesar 2 pada setiap pemanggilan rekursif. Proses ini berulang hingga nilai N kurang dari 1. Pada setiap pemanggilan fungsi, program akan memeriksa apakah nilai N saat itu adalah bilangan ganjil. Jika ya, maka nilai N tersebut akan dicetak.

### Unguided 6

```
//Ben Waiz Pintus W
//2311102169

package main

import (
    "fmt"
)

func hitungPangkat(x int, y int) int {
    if y < 0 {
        return 0
    }

    hasil := 1
    for i := 0; i < y; i++ {
        hasil *= x
    }
    return hasil
}

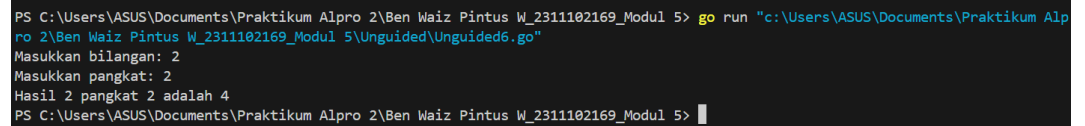
func main() {
    var bilangan int
    var pangkat int

    fmt.Print("Masukkan bilangan: ")
    fmt.Scanln(&bilangan)
    fmt.Print("Masukkan pangkat: ")
    fmt.Scanln(&pangkat)

    hasil := hitungPangkat(bilangan, pangkat)
```

```
    fmt.Printf("Hasil %d pangkat %d adalah %d\n", bilangan, pangkat,
hasil)
}
```

Screenshot output



```
PS C:\Users\ASUS\Documents\Praktikum Alpro 2\Ben Waiz Pintos W_2311102169_Modul 5> go run "c:\Users\ASUS\Documents\Praktikum Alpro 2\Ben Waiz Pintos W_2311102169_Modul 5\Unguided\Unguided6.go"
Masukkan bilangan: 2
Masukkan pangkat: 2
Hasil 2 pangkat 2 adalah 4
PS C:\Users\ASUS\Documents\Praktikum Alpro 2\Ben Waiz Pintos W_2311102169_Modul 5>
```

Penjelasan :

Pengguna diminta untuk memasukkan bilangan dan pangkat yang diinginkan, kemudian hasil perhitungan ditampilkan ke layar dalam format yang mudah dibaca. Program ini memberikan gambaran yang jelas tentang penggunaan fungsi, input, dan output.