

LAPORAN PRAKTIKUM
ALGORITMA PEMROGRAMAN 2
MODUL 5
REKURSIF



Oleh :

Dimas Bagus Firmansyah

2311102002

S1 IF 11 02

S1 TEKNIK INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2024

I. DASAR TEORI

6.1 Pengantar Rekursif

Rekursif secara sederhana dapat diartikan sebagai cara menyelesaikan suatu masalah dengan cara menyelesaikan sub-masalah yang identic dari masalah utama. Sebagai contoh perhatikan prosedur berikut ini!

	Notasi Algoritma	Notasi dalam bahasa GO
1	procedure cetak(in x:integer)	func cetak(x int){
2	algoritma	fmt.Println(x)
3	output(x)	cetak(x+1)
4	cetak(x+1)	}
5	endprocedure	

Catatan :

- Teknik rekursif ini merupakan salah satu alternatif untuk mengganti struktur kontrol perulangan dengan memanfaatkan subprogram (bisa fungsi ataupun prosedur).
- Untuk menghentikan proses rekursif digunakan percabangan (if-then).
- Base-case adalah kondisi proses rekursif berhenti. Base-case merupakan hal terpenting dan pertama yang harus diketahui ketika akan membuat program rekursif. Mustahil membuat program rekursif tanpa mengetahui base-case terlebih dahulu.
- Recursive-case adalah kondisi dimana proses pemanggilan dirinya sendiri dilakukan. Kondisi recursive-case adalah komplemen atau negasi dari base-case.
- Setiap algoritma rekursif selalu memiliki padanan dalam bentuk algoritma iteratif.

6.2 Komponen Rekursif

Algoritma rekursif terdiri dari dua komponen utama :

- **Base-case (Basis)**, yaitu bagian untuk menghentikan proses rekursif dan menjadi komponen terpenting di dalam sebuah rekursif.
- **Recursive-case**, yaitu bagian pemanggilan subprogramnya.

6.3 Contoh Program dengan menggunakan Rekursif

a. Membuat baris bilangan dari n hingga 1

Base-case: bilangan == 1

```
1 package main
2 import "fmt"
3 func main(){
4     var n int
5     fmt.Scan(&n)
6     baris(n)
7 }
8
9 func baris(bilangan int){
10     if bilangan == 1 {
11         fmt.Println(1)
12     }else{
13         fmt.Println(bilangan)
14         baris(bilangan - 1)
15     }
16 }
```

II. GUIDED

GUIDED 1 SOURCE CODE

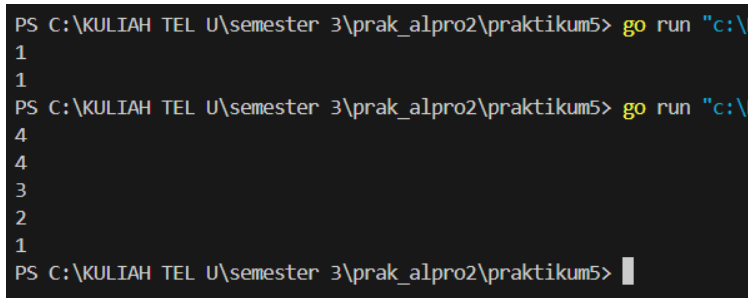
```
package main

import "fmt"

func main() {
    var n int
    fmt.Scan(&n) // Membaca input pengguna
    baris(n)     // memanggil fungsi rekursif 'baris'
}

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

SCREENSHOOT PROGRAM



```
PS C:\KULIAH TEL U\semester 3\prak_alpro2\praktikum5> go run "c:\V
1
1
PS C:\KULIAH TEL U\semester 3\prak_alpro2\praktikum5> go run "c:\V
4
4
3
2
1
PS C:\KULIAH TEL U\semester 3\prak_alpro2\praktikum5> |
```

DESKRIPSI PROGRAM

Fungsi baris melakukan cetak bilangan, kemudian memanggil dirinya sendiri dengan nilai bilangan yang dikurangi satu. Proses akan terus berlanjut hingga bilangan mencapai satu, di mana ia mencetak satu dan berhenti. Hasilnya yaitu deretan bilangan yang dicetak dari n hingga secara menurun 1.

GUIDED 2 SOURCE CODE

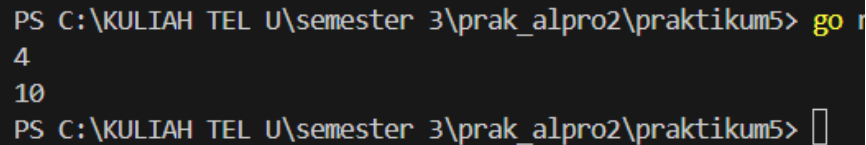
```
package main

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)  
        }  
    }  
}
```

SCREENSHOOT PROGRAM



```
PS C:\KULIAH TEL U\semester 3\prak_alpro2\praktikum5> go r  
4  
10  
PS C:\KULIAH TEL U\semester 3\prak_alpro2\praktikum5> 
```

DESKRIPSI PROGRAM

Program tersebut membaca bilangan n yang diinput pengguna dan akan mencetak hasil penjumlahan dari n hingga 1 menggunakan fungsi rekursif penjumlahan. Fungsi itu akan menambahkan n dengan hasil pemanggilan rekursif pada n-1 hingga mencapai 1.

III. UNGUIDED

UNGUIDED 1

```
package main

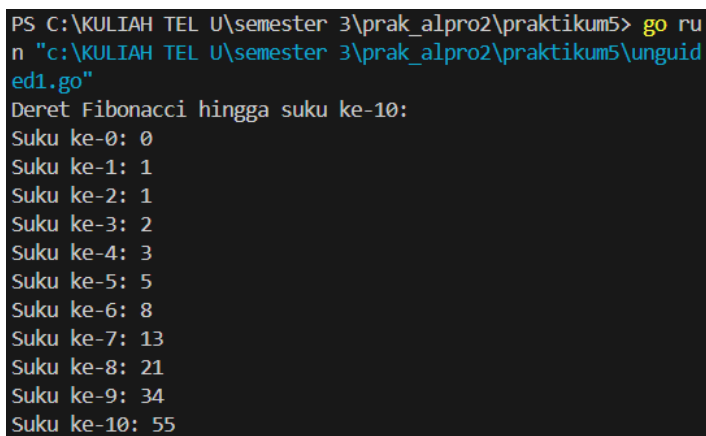
// 2311102002

import "fmt"

// Fungsi rekursif untuk menghitung deret Fibonacci
func fibonacci(n int) int {
    if n <= 1 {
        return n // Jika nilai n adalah 0 atau 1, maka langsung kembalikan nilainya
    }
    return fibonacci(n-1) + fibonacci(n-2) // Rekursi untuk menghitung nilai suku
    sebelumnya
}

func main() {
    fmt.Println("Deret Fibonacci hingga suku ke-10:")
    for i := 0; i <= 10; i++ { // Langsung gunakan angka 10 di sini
        fmt.Printf("Suku ke-%d: %d\n", i, fibonacci(i))
    }
}
```

SCRENSHOOT PROGRAM



```
PS C:\KULIAH TEL U\semester 3\prak_alpro2\praktikum5> go run "c:\KULIAH TEL U\semester 3\prak_alpro2\praktikum5\unguided1.go"
Deret Fibonacci hingga suku ke-10:
Suku ke-0: 0
Suku ke-1: 1
Suku ke-2: 1
Suku ke-3: 2
Suku ke-4: 3
Suku ke-5: 5
Suku ke-6: 8
Suku ke-7: 13
Suku ke-8: 21
Suku ke-9: 34
Suku ke-10: 55
```

DESKRIPSI PROGRAM

Fibonacci menerima parameter n dan akan menghitung nilai dari suku ke- n secara rekursif. Jika n adalah 0 atau 1, fungsi akan mengembalikan nilai n , jika tidak, fungsi akan memanggil $n-1$ dan $n-2$ dan menambahkan hasilnya.

UNGUIDED 2

```
package main

import "fmt"

// Fungsi rekursif untuk mencetak bintang
func printStars(n int) {
    if n == 0 {
```

```

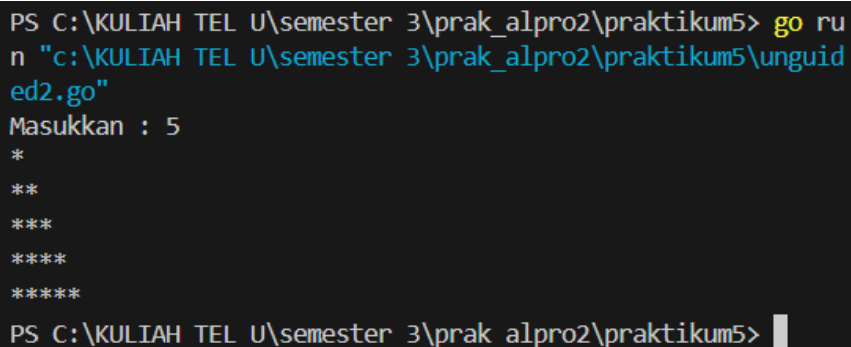
        return
    }
    printStars(n - 1)
    fmt.Println(string(make([]rune, n, n))) // Mencetak n bintang
}

// Fungsi rekursif untuk mencetak pola hingga N baris
func printPattern(n, i int) {
    if i > n {
        return
    }
    for j := 0; j < i; j++ {
        fmt.Print("*")
    }
    fmt.Println()
    printPattern(n, i+1)
}

func main() {
    var n int
    fmt.Print("Masukkan : ")
    fmt.Scan(&n)
    printPattern(n, 1) // Memulai dari baris pertama
}

```

SCREENSHOOT PROGRAM



```

PS C:\KULIAH TEL U\semester 3\prak_alpro2\praktikum5> go run
n "c:\KULIAH TEL U\semester 3\prak_alpro2\praktikum5\unguid
ed2.go"
Masukkan : 5
*
**
***
****
*****
PS C:\KULIAH TEL U\semester 3\prak_alpro2\praktikum5>

```

DESKRIPSI PROGRAM

Fungsi `printPattern` mengambil dua parameter, `n` (jumlah baris) dan `i` (baris saat ini), dan mencetak baris dengan jumlah bintang sama dengan nilai `i`. Ia mencetak bintang menggunakan loop untuk setiap baris, memanggil dirinya sendiri pada `i+1` hingga jumlah baris mencapai `n`.

UNGUIDED 3

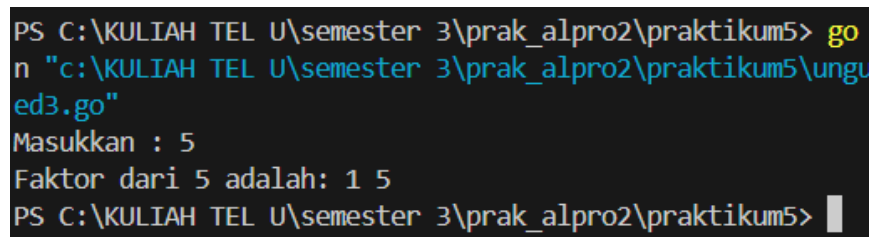
```
package main

import "fmt"

// Fungsi untuk mencetak faktor n
func printFactors(n, i int) {
    if i > n {
        return
    }
    if n%i == 0 {
        fmt.Print(i, " ")
    }
    printFactors(n, i+1)
}

func main() {
    var n int
    fmt.Print("Masukkan : ")
    fmt.Scan(&n)
    fmt.Print("Faktor dari ", n, " adalah: ")
    printFactors(n, 1)
    fmt.Println()
}
```

SCREENSHOOT PROGRAM



```
PS C:\KULIAH TEL U\semester 3\prak_alpro2\praktikum5> go
n "c:\KULIAH TEL U\semester 3\prak_alpro2\praktikum5\unguid3.go"
Masukkan : 5
Faktor dari 5 adalah: 1 5
PS C:\KULIAH TEL U\semester 3\prak_alpro2\praktikum5>
```

DESKRIPSI PROGRAM

Fungsi `printFactors` menerima dua parameter: `n` (angka yang ingin Anda cari faktornya) dan `i` (angka yang ingin Anda periksa untuk melihat apakah faktornya). Jika `i` adalah faktor dari `n` (`n % i == 0`), maka `i` dicetak. Fungsi ini dijalankan secara rekursif dari nilai `i = 1` hingga `i > n`. Pada fungsi utama, pengguna diminta memasukkan nilai `n`. Program kemudian mencetak faktor `n` menggunakan fungsi `printFactors`.

UNGUIDED 4

```
package main

import "fmt"

// Fungsi menampilkan bilangan n hingga 1 dan kembali
func printSequence(n, current int) {
    if current == 0 {
        return
    }
    fmt.Print(current, " ")
    printSequence(n, current-1)
    if current != n {
```

```

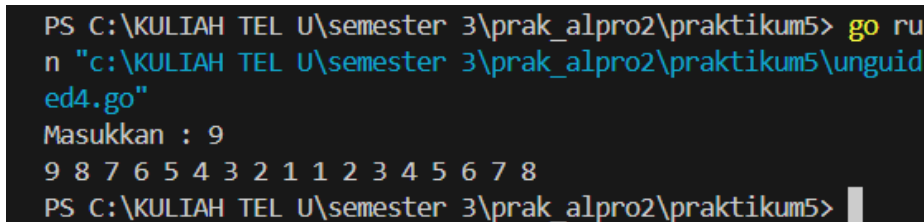
        fmt.Print(current, " ")
    }
}

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

    printSequence(n, n)
    fmt.Println() // 2311102002
}

```

SCREENSHOOT PROGRAM



```

PS C:\KULIAH TEL U\semester 3\prak_alpro2\praktikum5> go run
n "c:\KULIAH TEL U\semester 3\prak_alpro2\praktikum5\unguid
ed4.go"
Masukkan : 9
9 8 7 6 5 4 3 2 1 1 2 3 4 5 6 7 8
PS C:\KULIAH TEL U\semester 3\prak_alpro2\praktikum5>

```

DESKRIPSI PROGRAM

n (nilai awal yang dimasukkan oleh pengguna) dan saat ini (nilai yang diproses). Fungsi ini mencetak nilai saat ini dan memanggil dirinya sendiri dengan current-1 hingga mencapai 0. Kemudian, jika nilai saat ini tidak sama dengan n, program akan mencetak ulang nilai saat ini untuk membuat urutan menaik.

UNGUIDED 5

```

package main

import "fmt"

// Fungsi rekursif untuk menampilkan bilangan ganjil dari 1 hingga N
func bilGanjil(n, current int) {
    if current > n {
        return
    }
    fmt.Print(current, " ")
    bilGanjil(n, current+2)
}

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

    bilGanjil(n, 1)
    fmt.Println() // 2311102002
}

```


SCREENSHOOT PROGRAM

```
PS C:\KULIAH TEL U\semester 3\prak_alpro2\praktikum5> go
n "c:\KULIAH TEL U\semester 3\prak_alpro2\praktikum5\ung
ed5.go"
Masukkan: 20
1 3 5 7 9 11 13 15 17 19
PS C:\KULIAH TEL U\semester 3\prak_alpro2\praktikum5> |
```

DESKRIPSI PROGRAM

Fungsi bilangan ganjil menerima dua parameter n (batas atas yang dimasukkan oleh pengguna) dan saat ini (angka ganjil yang akan diproses). Fungsi ini mencetak nilai saat ini dan memanggil dirinya sendiri, yang menambahkan 2 ke nilai saat ini hingga nilainya melebihi n. Pada fungsi utama, program meminta pengguna untuk memasukkan nilai n dan memanggil bilangan ganjil untuk mencetak semua bilangan ganjil dari 1 hingga n.

UNGUIDED 6

```
package main

import "fmt"

// Fungsi rekursif untuk menghitung x pangkat y
func pangkat(x, y int) int {
    if y == 0 {
        return 1
    }
    return x * pangkat(x, y-1)
}

func main() {
    var x, y int
    fmt.Print("Masukkan x dan y: ")
    fmt.Scan(&x, &y) // 2311102002

    fmt.Println(pangkat(x, y))
}
```

SCREENSHOOT PROGRAM

```
PS C:\KULIAH TEL U\semester 3\prak_alpro2\praktikum
n "c:\KULIAH TEL U\semester 3\prak_alpro2\praktikum
ed6.go"
Masukkan x dan y: 2 2
4
PS C:\KULIAH TEL U\semester 3\prak_alpro2\praktikum
```

DESKRIPSI PROGRAM

Jika y adalah 0, fungsi tersebut mengembalikan 1 karena bilangan yang dipangkatkan 0 adalah 1. Jika tidak, fungsi tersebut akan memanggil dirinya sendiri dengan y-1 dan mengalikan hasilnya dengan x. Pada fungsi utama, pengguna diminta memasukkan nilai x dan y. Kemudian, dengan menggunakan fungsi pangkat, hasil x pangkat y dihitung dan ditampilkan di layar.