

**LAPORAN PRAKTIKUM
PEMROGRAMAN BERORIENTASI OBJEK**

**MODUL V
REKURSIF**



Oleh:

AULIA RADIX PUTRA WINARKO

2311102056

S1IF-11-02

**S1 TEKNIK INFORMATIKA
INSTITUT TEKNOLOGI TLKOM PURWOKERTO**

2024

I. DASAR TEORI

A. Pengantar Rekursif

modul-modul sebelumnya sudah dijelaskan bahwa suatu subprogram baik fungsi atau prosedur bisa memanggil subprogram lainnya. Hal ini tidak menutup kemungkinan bahwa subprogram yang dipanggil adalah dirinya sendiri. Dalam pemrograman teknik ini dikenal dengan istilah rekursif. Rekursif secara sederhana dapat diartikan sebagai cara menyelesaikan suatu masalah dengan cara menyelesaikan sub-masalah yang identik dari masalah utama.

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.

B. 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.

II. GUIDED

1. Source Code

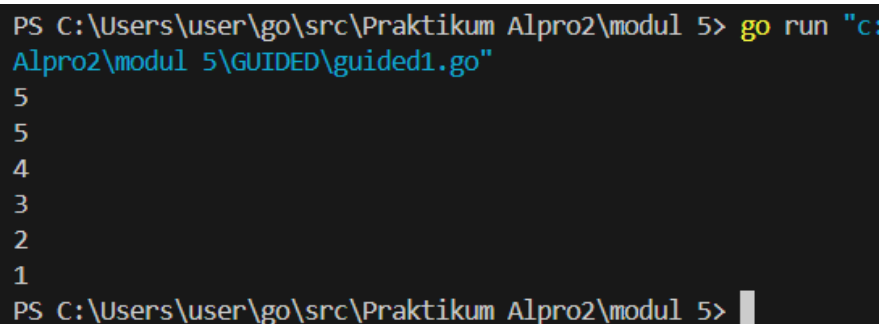
```
package main

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)
    }
}
```

Output



```
PS C:\Users\user\go\src\Praktikum Alpro2\modul 5> go run "c:\Users\user\go\src\Praktikum Alpro2\modul 5\GUIDED\guided1.go"
5
5
4
3
2
1
PS C:\Users\user\go\src\Praktikum Alpro2\modul 5> █
```

Keterangan

Program diatas dapat mencetak angka yang diinputkan oleh user hingga 1, Setelah menerima input melalui `fmt.Scan`, fungsi `baris (n)` mencetak angka dan memanggil dirinya sendiri dengan nilai dikurangi 1, sampai angka mencapai 1, di mana program akan berhenti.

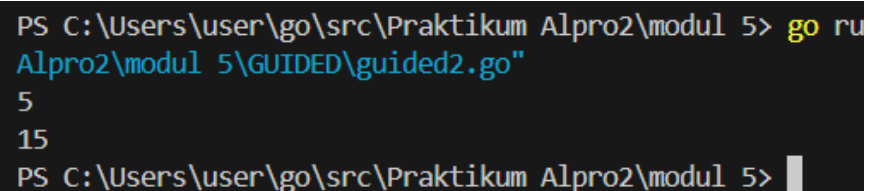
2. Source Code

```
package main

import "fmt"
```

```
func main() {  
    var n int  
    fmt.Scan(&n)  
    fmt.Println(penjualan(n))  
}  
  
func penjualan(n int) int {  
    if n == 1 {  
        return 1  
    } else {  
        return n + penjualan(n - 1)  
    }  
}
```

Output



```
PS C:\Users\user\go\src\Praktikum Alpro2\modul 5> go run Alpro2\modul 5\GUIDED\guided2.go  
5  
15  
PS C:\Users\user\go\src\Praktikum Alpro2\modul 5>
```

Keterangan

Program diatas dapat menghitung jumlah total dari bilangan 1 hingga bilangan yang diinputkan oleh user. Setelah menerima input angka melalui `fmt.Scan`, program memanggil fungsi `penjualan(n)` yang bekerja dengan memeriksa apakah `n` sama dengan 1. Jika `n` lebih besar dari 1, fungsi akan menambahkan nilai `n` dengan hasil dari `penjualan(n-1)`, menjumlahkan angka hingga mencapai 1. Program ini kemudian mencetak hasil penjumlahan tersebut.

III. UNGUIDED

1. Source Code

```
package main

import "fmt"

func fibo (n int) int {
    if n == 0{
        return 0
    } else if n == 1 {
        return 1
    } else {
        return fibo (n-1) + fibo (n-2)
    }
}

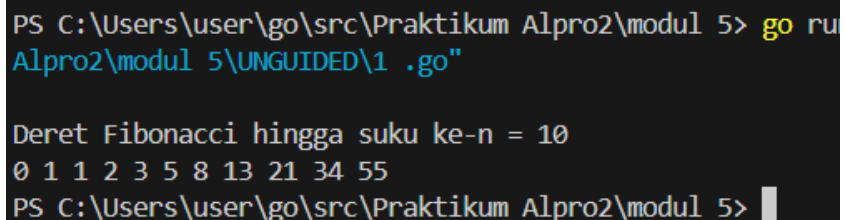
func main () {
    var bilangan int

    fmt.Print ("\nFungsi Rekrusif untuk mencari deret
Fibonacci\n")
    fmt.Print ("\nDeret Fibonacci hingga suku ke-n =")
    fmt.Scan (&bilangan)

    for n :=0; n <= bilangan; n++ {
        fmt.Printf ("%d", fibo(n) )
    }

    fmt.Println()
}
```

Output



```
PS C:\Users\user\go\src\Praktikum Alpro2\modul 5> go run
Alpro2\modul 5\UNGUIDED\1 .go

Deret Fibonacci hingga suku ke-n = 10
0 1 1 2 3 5 8 13 21 34 55
PS C:\Users\user\go\src\Praktikum Alpro2\modul 5>
```

Keterangan

Program diatas dapat mencetak deret Fibonacci hingga suku ke-n. Pada fungsi fibo(n) mengembalikan nilai Fibonacci ke-n dengan

memeriksa kondisi, jika n adalah 0, maka mengembalikan 0, dan jika n adalah 1, maka mengembalikan 1. Untuk nilai n yang lebih besar, fungsi memanggil dirinya sendiri untuk menghitung Fibonacci ke-n-1 dan ke-n-2, kemudian menjumlahkan keduanya.

2. Source Code

```
package main

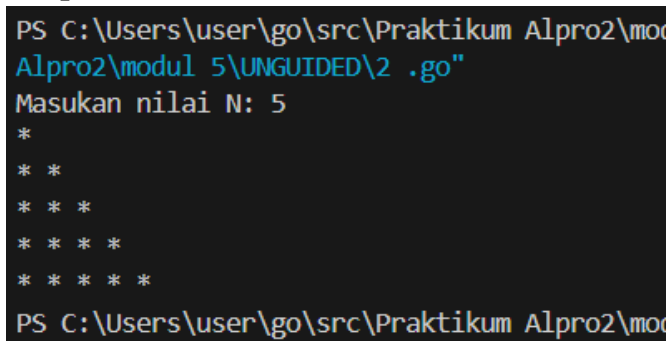
import "fmt"

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

func pola (n, c int) {
    if c > n {
        return
    }
    bintang (c)
    fmt.Println()
    pola (n, c+1)
}

func main () {
    var a int
    fmt.Print ("Masukan nilai N:")
    fmt.Scan(&a)
    pola(a, 1)
}
```

Output



```
PS C:\Users\user\go\src\Praktikum Alpro2\mod
Alpro2\modul 5\UNGUIDED\2 .go"
Masukan nilai N: 5
*
* *
* * *
* * * *
* * * * *
PS C:\Users\user\go\src\Praktikum Alpro2\mod
```

Keterangan

Kode diatas dapat mencetak deretan bintang sebanyak B kali dalam satu baris, dan berhenti ketika B mencapai 0. Pada fungsi pola(n, c) mencetak pola bintang dari baris pertama hingga baris ke-n, di mana setiap baris bertambah satu bintang dari baris sebelumnya.

3. Source Code

```
package main

import "fmt"

func faktorial(n, i int, faktors []int) []int {
    if i > n {
        return faktors
    }
    if n%i == 0 {
        faktors = append(faktors, i)
    }
    return faktorial(n, i+1, faktors)
}

func main() {
    var n int
    fmt.Print("Masukan Bilangan Bulat Positif N: ")
    fmt.Scan(&n)
    faktors := faktorial(n, 1, []int{})
    fmt.Println("Faktor dari", n, "Adalah", faktors)
}
```

Output

```
PS C:\Users\user\go\src\Praktikum Alpro2\modul 5> go r
Alpro2\modul 5\UNGUIDED\3 .go"
Masukan Bilangan Bulat Positif N: 5
Faktor dari 5 Adalah [1 5]
PS C:\Users\user\go\src\Praktikum Alpro2\modul 5> █
```

Keterangan

menghitung dan mencetak fak faktor-faktor dari bilangan bulat positif menggunakan rekursi. n sebagai bilangan yang akan dicari faktornya, i sebagai penghitung iterasi, dan faktors sebagai daftar faktor yang telah ditemukan. n sebagai bilangan yang akan dicari faktornya, i sebagai penghitung iterasi, dan faktors sebagai daftar faktor yang telah ditemukan.

4. Source Code

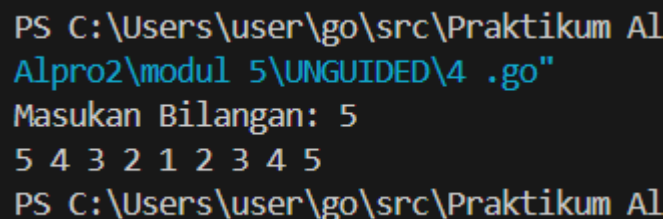
```
package main

import "fmt"

func main (){
    var n int
    fmt.Print("Masukan Bilangan: ")
    fmt.Scan (&n)
    baris (n)
}

func baris (bilangan int) {
    if bilangan < 1 {
        return
    }
    fmt.Print (bilangan," ")
    baris(bilangan - 1)
    if bilangan > 1 {
        fmt.Print (bilangan," ")
    }
}
```

Output



```
PS C:\Users\user\go\src\Praktikum Al
Alpro2\modul 5\UNGUIDED\4 .go"
Masukan Bilangan: 5
5 4 3 2 1 2 3 4 5
PS C:\Users\user\go\src\Praktikum Al
```

Keterangan

Program diatas meminta user untuk memasukkan sebuah bilangan bulat positif, kemudian mencetak bilangan tersebut secara berurutan dari bilangan yang dimasukkan hingga 1, dan kemudian mencetaknya kembali dari 2 hingga bilangan yang dimasukkan.

5. Source Code

```
package main

import "fmt"

func main (){
    var n int
    fmt.Print("Masukan Bilangan: ")
```



```

    fmt.Scan (&n)
    baris (n)
}

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

```

Output

```

PS C:\Users\user\go\src\Praktikum Al
Alpro2\modul 5\UNGUIDED\5 .go"
Masukan Bilangan: 10
1 3 5 7 9
PS C:\Users\user\go\src\Praktikum Al

```

Keterangan

Program diatas meminta user untuk memasukkan sebuah bilangan bulat positif dan kemudian mencetak bilangan ganjil dari 1 hingga bilangan yang dimasukkan secara rekursif. Fungsi baris akan memanggil dirinya sendiri dengan mengurangi nilai bilangan hingga mencapai 1.

6. Source Code

```

package main

import "fmt"

func pangkat (x int, y int) int {
    if y == 0 {
        return 1
    }
    if y < 0 {
        return 1 / pangkat(x, -y)
    }
    return x * pangkat(x, y-1)
}

func main(){
    var x, y int

```

```
fmt.Print("Masukan bilangan bulat: ")
fmt.Scan(&x)
fmt.Print("Masukan Pangkat: ")
fmt.Scan(&y)
hasil := pangkat(x, y)
fmt.Print("Hasil ", x, " Pangkat ", y, " Adalah: ", hasil )
}
```

Output

```
PS C:\Users\user\go\src\Praktikum Alpro2\modul 5> go run Alpro2\modul 5\UNGUIDED\tempCodeRunnerFile.go
Masukan bilangan bulat: 5
Masukan Pangkat: 3
Hasil 5 Pangkat 3 Adalah: 125
PS C:\Users\user\go\src\Praktikum Alpro2\modul 5> |
```

Keterangan

Program diatas dapat menghitung nilai pangkat dari sebuah bilangan bulat menggunakan fungsi rekursif.

Fungsi pangkat menerima dua parameter, yaitu x (bilangan yang akan dipangkatkan) dan y (pangkat yang diinginkan).