

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL 5
REKURSIF**



Oleh:

BERTHA ADELA

103112400041

IF-12-01

**S1 TEKNIK INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO**

2025

I. DASAR TEORI

Definisi Rekursif: Subprogram yang dipanggil adalah dirinya sendiri merupakan teknik yang dikenal dengan istilah rekursif. Rekursif secara sederhana dapat diartikan sebagai cara menyelesaikan suatu masalah dengan cara menyelesaikan sub-masalah yang identik dari masalah utama. Sebagai contoh:

```
1 package main
2 import "fmt"
3 func main(){
4     cetak(5)
5 }
6 func cetak(x int){
7     fmt.Println(x)
8     cetak(x+1)
9 }
```

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. 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*, yaitu bagian pemanggilan subprogramnya. Recursive-case adalah kondisi dimana proses pemanggilan dirinya sendiri dilakukan. Kondisi recursive-case adalah komplemen atau negasi dari base-case.

II. GUIDED

• GUIDED 1

Code:

```
SMT2 > Pertemuan4 > 103112400041_MODUL5 > -go 103112400041_Guided1.go > main

1  package main
2  import "fmt"
3  //Fungsi iteratif untuk menghitung pangkat (base^exp)
4  func pangkatIteratif(base, exp int) int {
5      hasil:= 1
6      for i := 0; i < exp; i++ {
7          hasil *= base
8      }
9      return hasil
10 }
11
12 //Fungsi iteratif untuk menghitung faktorial (n!)
13 func faktorialIteratif(n int) int {
14     hasil := 1
15     for i := 2; i <= n; i++ {
16         hasil *= i
17     }
18     return hasil
19 }
20

SMT2 > Pertemuan4 > 103112400041_MODUL5 > -go 103112400041_Guided1.go > main

20
21 func main() {
22     var base, exp, n int
23     // Input pangkat
24     fmt.Print("Masukkan bilangan: ")
25     fmt.Scanln(&base)
26     fmt.Print("Masukkan pangkat: ")
27     fmt.Scanln(&exp)
28
29     fmt.Printf("%d^%d = %d\n", base, exp, pangkatIteratif(base, exp))
30
31     // Input faktorial
32     fmt.Print("Masukkan angka untuk faktorial")
33     fmt.Scanln(&n)
34
35     fmt.Printf("%d! = %d\n", n, faktorialIteratif(n))
36 }
```

Output:

```
PS C:\Users\levina\OneDrive\Documents\golang> go run "c:\Users\levina\OneDrive\Documents\103112400041_Guided1.go"
Masukkan bilangan: 5
Masukkan pangkat: 2
5^2 = 25
Masukkan angka untuk faktorial: 12
12! = 479001600
PS C:\Users\levina\OneDrive\Documents\golang>
```

Penjelasan:

Program diatas berguna untuk menghitung pangkat dari suatu bilangan serta berguna untuk mencari faktorial dari suatu bilangan menggunakan fungsi iteratif.

• GUIDED 2


Code:

```
SMT2 > Pertemuan4 > 103112400041_MODUL5 > -go 103112400041_Guided2.go >  faktorial

1  package main
2  import "fmt"
3  //pangkat 0 adalah 1
4  func pangkatRekursif(base, exp int) int {
5      if exp == 0 {
6          return 1
7      }
8      return base * pangkatRekursif(base, exp-1)
9  }
10 func faktorialRekursif(n int) int {
11     if n == 0 || n == 1 {
12         return 1
13     }
14     return n * faktorialRekursif(n-1)
15 }

16 func main() {
17     var base, exp, n int
18     // Input pangkat
19     fmt.Print("Masukkan bilangan: ")
20     fmt.Scanln(&base)
21     fmt.Print("Masukkan pangkat: ")
22     fmt.Scanln(&exp)
23
24     fmt.Printf("%d^%d = %d\n", base, exp, pangkatRekursif(base, exp))
25
26     // Input faktorial
27     fmt.Print("Masukkan angka untuk faktorial: ")
28     fmt.Scanln(&n)
29
30     fmt.Printf("%d!=%d\n", n, faktorialRekursif(n))
31 }
```

Output:

```
PS C:\Users\levina\OneDrive\Documents\golang> go run "c:\Users\levina\OneDrive\Documents\103112400041_Guided1.go"
Masukkan bilangan: 5
Masukkan pangkat: 2
5^2 = 25
Masukkan angka untuk faktorial: 12
12! = 479001600
PS C:\Users\levina\OneDrive\Documents\golang> 
```

Penjelasan:

Program diatas berguna untuk menghitung pangkat dari suatu bilangan serta berguna untuk mencari faktorial dari suatu bilangan menggunakan fungsi rekursif.

III. UNGUIDED

- UNGUIDED 1

Code:

```
SMT2 > Pertemuan4 > 103112400041_MODUL5 > -go 103112400041_Unguided1.go > f
1 //BERTHA ADELA
2 //103112400041
3 package main
4 import "fmt"
5 func main() {
6     var n int
7     fmt.Scan(&n)
8     for i:= 1; i<= n; i++ {
9         fmt.Print(f(i), " ")
10    }
11 }
12 func f(n int) int {
13     if n == 0 {
14         return 0
15     } else if n == 1 {
16         return 1
17     }
18     return f(n-1)+f(n-2)
19 }
```

Output:

```
PS C:\Users\levina\OneDrive\Documents\golang> go run "c:\Users\levina\OneDrive\Documents\golang\103112400041_Unguided1.go"
10
1 1 2 3 5 8 13 21 34 55
PS C:\Users\levina\OneDrive\Documents\golang> 
```

Penjelasan:

Program ini berguna untuk mencari deret fibonacci dari suku ke-1 hingga suku ke n. Sebagai contoh untuk mencari deret fibonacci tersebut hanya perlu menginputkan suku ke-10, maka akan otomatis memberi output deret fibonacci dari suku ke-1 hingga suku ke-10 menggunakan fungsi rekursif.

• UNGUIDED 2

Code:

```
SMT2 > Pertemuan4 > 103112400041_MODUL5 > -go 103112400041_Unguided2.go
1  //BERTHA ADELA
2  //103112400041
3  package main
4  import "fmt"
5  func main() {
6      var n int
7      fmt.Scan(&n)
8      hitung(n, 1)
9  }
10 func hitung(n, skrg int) int {
11     if skrg > n {
12         return 0
13     }
14     cetakBintang(skrg)
15     fmt.Println()
16     hitung(n, skrg+1)
17     return 0
18 }
19 func cetakBintang(jumlah int) int {
20     if jumlah == 0 {
21         return 0
22     }
23     fmt.Print("*")
24     cetakBintang(jumlah-1)
25     return 0
26 }
```

Output:

```
PS C:\Users\levina\OneDrive\Documents\golang> go run "c:\Users\levina\OneDrive\Documents\golang\103112400041_Unguided2.go"
5
*
**
***
****
*****
PS C:\Users\levina\OneDrive\Documents\golang> go run "c:\Users\levina\OneDrive\Documents\golang\103112400041_Unguided2.go"
1
*
PS C:\Users\levina\OneDrive\Documents\golang> go run "c:\Users\levina\OneDrive\Documents\golang\103112400041_Unguided2.go"
3
*
**
***
```

Penjelasan:

Program ini digunakan untuk menampilkan pola bintang segitiga siku siku dengan menginputkan alas dari segitiga menggunakan fungsi rekursif.

• UNGUIDED 3

Code:

```
SMT2 > Pertemuan4 > 103112400041_MODUL5 > -go 103112400041_Unguided3.go
1 //BERTHA ADELA
2 //103112400041
3 package main
4 import "fmt"
5 func main() {
6     var n int
7     fmt.Scan(&n)
8     prima(n, 1)
9 }
10 func prima(p, x int) int {
11     if p<x {
12         return 0
13     }
14     if p%x == 0 {
15         fmt.Print(x, " ")
16     }
17     prima(p, x+1)
18     return 0
19 }
```

Output:

```
PS C:\Users\levina\OneDrive\Documents\golang> go run "c:\Users\levina\OneDrive\
DUL5\103112400041_Unguided3.go"
5
1 5
PS C:\Users\levina\OneDrive\Documents\golang> go run "c:\Users\levina\OneDrive\
DUL5\103112400041_Unguided3.go"
12
1 2 3 4 6 12
PS C:\Users\levina\OneDrive\Documents\golang> 
```

Penjelasan:

Program ini menampilkan faktor bilangan n menggunakan fungsi rekursif.

IV. KESIMPULAN

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.

REFERENSI
MODUL 5 REKURSIF