

**LAPORAN PRAKTIKUM
ALGORITMA PEMROGRAMAN 2**

**MODUL 5
REKURSIF**



Oleh:

WILDAN DAFFA' HAKIM PUTRA ANTARA

2311102055

S1IF-11-02

**S1 TEKNIK INFORMATIKA
INSTITUT TEKNOLOGI TELKOM PURWOKERTO
2024**

I. DASAR TEORI

rekursif merupakan kondisi dimana suatu sub program memanggil dirinya sendiri. Atau bisa disebut pemecahan masalah dengan menyelesaikan sub masalah yang identik dengan masalah utama.

Berikut merupakan cara deklarasi re

Notasi Algoritma

```
procedure cetak(in x:integer)
  algoritma
    output(x)
    cetak(x+1)
endprocedure
```

Notasi dalam Bahasa GO

```
func cetak(x int){
    fmt.Println(x)
    cetak(x+1)
}
```

pada contoh diatas, apabila dijalankan maka rekursif tidak akan pernah berhenti. Maka dari itu, kita perlu menambahkan base case sehingga ketika memiliki nilai seperti pada base-case maka rekursif akan berhenti. Yang terjadi ketika menemui kondisi base-case adalah program melakukan backward ke subprogram sebelumnya sampai pada program utama. Sedangkan langkah memanggil subprogram lain sebelum kondisi base-case disebut forward.

Notasi Algoritma

```
procedure cetak(in x:integer)
  algoritma
    if x == 6 then
      output(x)
    else
      output(x)
      cetak(x+1)
    endif
endprocedure
```

Notasi dalam Bahasa GO

```
package main
import "fmt"
func main(){
    cetak(5)
}
func cetak(x int){
    if x == 10 {
        fmt.Println(x)
    } else {
        fmt.Println(x)
        cetak(x+1)
    }
}
```

II. GUIDED

1. Membuat baris bilangan dari n hingga 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)
    }
}
```

```

λ MacBook-Air-Daffa Modul5 → go run "/Users/daffahakim/Documents/
7
7
6
5
4
3
2
1

```

Program ini digunakan untuk mencetak nomor secara menurun dari 7 sampai ke 1. Base-case dari program ini adalah ketika kondisi sama dengan satu. Jika tidak, maka bilangan akan dicetak dan fungsi baris akan dipanggil dengan parameter aktual bilangan – 1.

2. Menghitung hasil penjumlahan 1 hingga n

```

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)
    }
}

```

```

λ MacBook-Air-Daffa Modul5 → go run "/Users/daffahakim/Documents/Kuliah/S
5
15
λ MacBook-Air-Daffa Modul5 →

```

Program ini digunakan untuk menghitung jumlah dari deret bilangan. misalkan 5, maka hasilnya $5+4+3+2+1 = 15$, fungsi penjumlahan n. Base – case nya adalah jika n sama dengan 1 jika tidak, maka fungsi akan mengembalikan nilai n saat fungsi dipanggil dan memanggil fungsi penjumlahan dengan parameter aktual n-1. Sehingga ketika backward nilai yang ditambahkan akan dikumpulkan menjadi 1 (misalkan : 5, maka 5,4,3,2,1 yang berasal dari pemanggilan fungsi sebelumnya akan ditambahkan)

III. UNGUIDED

1. Deret fibonacci adalah sebuah deret dengan nilai suku ke-0 dan ke-1 adalah 0 dan 1, dan nilai suku ke-n selanjutnya adalah hasil penjumlahan dua suku sebelumnya.

```
package main

import (
    "fmt"
    "strconv"
)

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

func main() {
    var inp int
    var bilke, fibo string
    fmt.Scan(&inp)
    for i := 1; i <= inp; i++ {
        bilke += "\t" + strconv.Itoa(i)
        fibo += "\t" +
        strconv.Itoa(fibonacci(i))
    }
    fmt.Printf("n: %s \nSn: %s \n", bilke, fibo)
}
```

Program ini digunakan untuk menampilkan deret fibonacci. Pada fungsi rekursif yang bernama fibonnaci memiliki base-case ketika n kurang dari sama dengan 1 maka dia akan mengembalikan nilai n saat itu. Jika tidak, maka fungsi akan mengembalikan fibonacci dengan parameter aktual n-1 ditambah dengan fibonacci dengan parameter aktual n-2. Di program main, pengguna akan diminta inputan yang akan disimpan pada variable inp, lalu dilakukann perulangan dari 1 sampai nilai yang diinputkan user. Selama perulangan akan dilakukan pertambahan string

ke variable bilke dan fibo. Variable bilke digunakan untuk menyimpan nilai bilangan ke dan fibo untuk menyimpan nilai fibonnaci pada posisi tersebut. Di akhir program ketika perulangan selesai semua angka yang ditambahkan ke variable bilke dan fibo ditampilkan dengan fmt.Printf

2. Buatlah sebuah program yang digunakan untuk menampilkan pola bintang berikut ini dengan menggunakan fungsi rekursif. N adalah masukan dari user.

```
package main

import "fmt"

func columnPrint(n, max int) {
    if n > max {
        return
    } else {
        rowPrint(1, n)
        fmt.Println()
        ColumnPrint(n+1, max)
    }
}

func rowPrint(current, limit int) {
    if current > limit {
        return
    } else {
        fmt.Print("*")
        rowPrint(current+1, limit)
    }
}

func main() {
    var n int
    fmt.Scan(&n)
    columnPrint(1, n)
}
```

Program ini digunakan untuk mencetak segitiga, pada fungsi ini digunakan rekursif ColumnPrint dan rowPrint. rowPrint digunakan untuk mencetak bintang dengan parameter aktual current, limit per baris base-case nya adalah ketika current > limit maka fungsi rekursif akan berhenti dan melakukan backward. Sedangkan untuk columnPrint,

digunakan untuk mencetak baris baru setelah fungsi row dieksekusi. Base-case dari fungsi ini ketika $n > \text{max}$.

3. Buatlah program yang mengimplementasikan rekursif untuk menampilkan faktor bilangan dari suatu N, atau bilangan yang apa saja yang habis membagi N.

```
package main

import "fmt"

func faktor(n, i int) {
    if n%i == 0 {
        fmt.Print(i, " ")
    }
    if n == i {
        return
    } else {
        faktor(n, i+1)
    }
}

func main() {
    var faktorN int
    fmt.Scan(&faktorN)
    faktor(faktorN, 1)
}
```

```
λ MacBook-Air-Daffa Modul5 → go run "/Users/daffahakim/Doc
12
1 2 3 4 6 12 %
λ MacBook-Air-Daffa Modul5 →
```

Program ini digunakan untuk mencari faktor dari bilangan menggunakan rekursif, sub program yang rekursif disini adalah faktor parameter faktual disini adalah n dan i, n untuk menyimpan bilangan yang faktornya akan dicari, dan i digunakan untuk iterasi ketika mengecek bilangan faktorial atau bukan. Base case dari sub program ini adalah ketika $n == i$ menghasilkan 0. Bilangan hanya dicetak jika n dimodulo i menghasilkan 0. pada fungsi utama setelah pengguna memasukkan angka yang mau dicari faktornya nilai akan disimpan pada faktorN dan fungsi faktor akan dipanggil dengan parameter aktual

nilai faktorN dan 1, rekursif akan terus melakukan forward sampai nilai i (1) pada parameter faktual sama dengan n (5).

4. Buatlah program yang mengimplementasikan rekursif untuk menampilkan barisan bilangan tertentu.

```
package main

import "fmt"

func printPattern(n, current int) {
    if current == 1 {
        fmt.Printf("%d ", current)
        return
    }
    fmt.Printf("%d ", current)
    printPattern(n, current-1)
    fmt.Printf("%d ", current)
}

func main() {
    var nVerse int
    fmt.Scan(&nVerse)
    printPattern(nVerse, nVerse)
}
```

Program ini mencetak rekursif menurun dan kembali lagi ke angka semula. Sub program yang melakukan rekursi adalah printPattern, fungsi printPattern memiliki 2 parameter faktual n dan current, n digunakan untuk menampung bilangan yang mau dicetak dimulai dan berakhir dimana dan current menampung bilangan yang sekarang dicetak. Base case dari sub program ini adalah ketika current sama dengan 1, ketika mencapai base case maka rekursi "naik" kembali dengan mencetak angka dalam urutan menaik, membentuk pola simetris. Penyebab naik ini karena stack sehingga setiap pemanggilan disimpan di dalam stack sampai mencapai base case. Setelah base case tercapai, fungsi mulai kembali ke level sebelumnya dalam tumpukan, mencetak angka dalam urutan yang berlawanan. Lalu pada fungsi main user menginputkan angka disimpan pada variable nVerse, lalu

dipanggil sub program printPattern dengan parameter actual yaitu nilai dari nVerse dan nVerse

5. Buatlah program yang mengimplementasikan rekursif untuk menampilkan barisan bilangan ganjil.

```
package main

import "fmt"

func cetakGanjil(n, current int) {
    if current > n {
        return
    } else {
        fmt.Print(current, " ")
        cetakGanjil(n, current+2)
    }
}

func main() {
    var angkaGanjil int
    fmt.Scan(&angkaGanjil)
    cetakGanjil(angkaGanjil, 1)
    fmt.Println()
}
```

```
λ MacBook-Air-Daffa Modul5 → go run "/Users/daffahakim/Docume
13
1 3 5 7 9 11 13
λ MacBook-Air-Daffa Modul5 → █
```

Program ini digunakan untuk mencetak bilangan ganjil dimulai dari satu sampai n, disini digunakan sub program rekursif cetakGanjil. Base-case dari sub program ini adalah ketika $current \geq n$ maka rekursif akan mencetak current dan melakukan backward. Selain itu sub-program akan mencetak current dan melakukan forward dengan memanggil cetakGanjil dengan parameter aktual n dan $current + 2$. pada main pengguna akan diminta menginputkan angka dan akan disimpan pada variable angkaGanjil lalu sub-program cetakGanjil akan dipanggil dengan parameter aktual nilai dari variable angkaGanjil dan 1. `fmt.Println()` digunakan untuk menghilangkan “%” di akhir.

6. Buatlah program yang mengimplementasikan rekursif untuk mencari hasil pangkat dari dua buah bilangan.

```

package main

import "fmt"

func pangkat(numPangkat, pangkatke int) int {
    if pangkatke == 1 {
        return numPangkat
    }
    return numPangkat * pangkat(numPangkat,
pangkatke-1)
}

func main() {
    var n, pngktke int
    fmt.Scan(&n, &pngktke)
    fmt.Println(pangkat(n, pngktke))
}

```

```

λ MacBook-Air-Daffa Modul5 → go run "/Users/daffa
2 3
8
λ MacBook-Air-Daffa Modul5 → █

```

Program ini merupakan program untuk menghitung bilangan dengan pangkat menggunakan rekursif. Ini memanfaatkan sub program pangkat dengan parameter factual numpangkat sebagai bilangan yang akan dipangkatkan dan pangkatke sebagai pangkat dari bilangan. Nilai dari pangkatke akan berkurang satu setiap forward rekursif terjadi, base-case dari sub program ini Ketika nilai pangkatke sama dengan 1 dan akan dilakukan backward yang akan mengalikan bilangan sebanyak nilai "pangkatke" kali. Pada fungsi main akan menerima dua inputan yang masing masing akan disimpan divariable n dan pngktke. Lalu dilakukan pencetakan nilai dengan fmt.Println setelah pemanggilan fungsi pangkat dengan parameter actual nilai dari n dan pngktke.