

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL V
REKURSIF**



Disusun Oleh :

Rendi Widya Anggita/2311102278

S1IF-11-06

Dosen Pengampu :

ABEDNEGO DWI SEPTIADI

PROGRAM STUDI S1 TEKNIK INFORMATIKA

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2024

I. DASAR TEORI

Dasar Teori

Rekursif adalah metode dalam pemrograman yang memungkinkan sebuah fungsi untuk memanggil dirinya sendiri, metode ini digunakan untuk menyelesaikan masalah dengan membaginya menjadi sub-masalah yang lebih kecil.

Metode ini memiliki dua kondisi yaitu *base case* dan *recursive case*. Kondisi *base case* merupakan kondisi di mana fungsi rekursif berhenti memanggil dirinya sendiri dan memberikan hasil akhir. Tanpa kondisi dasar, program akan terus memanggil dirinya sendiri tanpa berhenti, yang akhirnya menyebabkan error karena memori yang dipakai terus bertambah. Misalnya, pada perhitungan faktorial, *base case* terjadi saat nilai faktorial dari 0 atau 1 diketahui (faktorial dari 0 dan 1 adalah 1), sehingga perhitungan bisa berhenti di sana.

Sedangkan *recursive case* adalah kondisi di mana fungsi tersebut memanggil dirinya sendiri untuk menyelesaikan masalah yang lebih kecil. Setiap kali fungsi dijalankan, ia akan memecah masalah yang ada menjadi sub-masalah yang lebih sederhana hingga mencapai kondisi yang dapat diselesaikan langsung, yaitu *base case*. Contoh, dalam perhitungan faktorial, misal $n! = n \times (n-1)!$. Di sini, *recursive case* terjadi pada ekspresi $n \times (n-1)!$, di mana fungsi faktorial memanggil dirinya sendiri dengan parameter yang lebih kecil (yaitu $n-1$) hingga akhirnya mencapai *base case* saat $n=1$ atau $n=0$.

II. GUIDED

1. Soal Studi Case

Membuat baris bilangan dari n hingga 1

Base case : bilangan == 1

Sourcecode

```
package main

import "fmt"

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

func main() {
    var n int
    fmt.Print("Masukkan bilangan : ")
    fmt.Scan(&n)
    fmt.Print("Bilangan n ke 1 : ")
    baris(n)
}
```

Screenshoot Output



Deskripsi Program

Program ini adalah program untuk mencetak bilangan dari n hingga 1. program ini mempunyai fungsi baris untuk memeriksa apakah bilangan n == 1?, jika iya, maka program akan mencetak 1, jika tidak, maka program akan mencetak bilangan n, kemudian n-1, hingga hasilnya 1.

2. Soal Studi Case

Menghitung hasil penjumlahan 1 hingga n

Base case : $n == 1$

Sourcecode

```
package main

import "fmt"

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

func main() {
    var n int
    fmt.Print("Masukkan bilangan : ")
    fmt.Scan(&n)
    fmt.Print("Hasil : ")
    fmt.Println(penjumlahan(n))
}
```

Screenshoot Output



Deskripsi Program

Program ini merupakan penjumlahan dari n hingga 1 menggunakan rekursif. Fungsi ini akan menerima satu parameter, jika $n == 1$, maka program akan mengembalikan nilai 1. Jika tidak, fungsi akan mengembalikan hasil dari $n + \text{penjumlahan}(n-1)$, menambahkan nilai n saat ini dengan hasil rekursif dari $\text{penjumlahan}(n-1)$.

3. Soal Studi Case

Mencari 2 pangkat n atau

Base case : $n == 0$

Sourcecode

```
package main

import "fmt"

func pangkat(n int) int{
    if n == 0 {
        return 1
    }else{
        return 2 * pangkat(n-1)
    }
}

func main() {
    var n int
    fmt.Print("Masukkan bilangan : ")
    fmt.Scan(&n)
    fmt.Print("Hasil : ")
    fmt.Println(pangkat(n))
}
```

Screenshoot Output



Deskripsi Program

Program tersebut merupakan program untuk menghitung hasil dari 2 pangkat n, dimana n adalah bilangan bulat yang diinputkan user, pertama program akan meminta inputan dari user, lalu program akan menghitung 2 pangkat dari n tersebut dan menampilkan hasilnya.

4. Soal Studi Case

Mencari nilai faktorial atau $n!$

Base Case : $n==0$ atau $n==1$

Sourcecode

```
package main

import "fmt"

func faktorial(n int) int {
    if n == 0 || n == 1 {
        return 1
    } else {
        return n * faktorial(n-1)
    }
}

func main() {
    var n int
    fmt.Print("Masukkan bilangan : ")
    fmt.Scan(&n)
    fmt.Print("Hasil : ")
    fmt.Println(faktorial(n))
}
```

Screenshoot Output



Deskripsi Program

Program tersebut adalah program untuk menghitung bilangan faktorial dengan inputan dari user, dengan base case $n==0$ atau $n==1$.

III. UNGUIDED

1. Soal Studi Case

- 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. Secara umum dapat diformulasikan $S_n = S_{n-1} + S_{n-2}$. Berikut ini adalah contoh nilai deret fibonacci hingga suku ke-10. Buatlah program yang mengimplementasikan fungsi rekursif pada deret fibonacci tersebut.

n	0	1	2	3	4	5	6	7	8	9	10
S_n	0	1	1	2	3	5	8	13	21	34	55

Sourcecode

```
package main

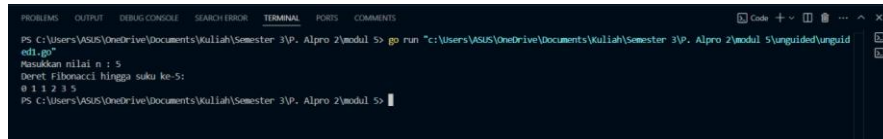
import "fmt"

func fibonacci(n int) int {
    if n == 0 {
        return 0
    } else if n == 1 {
        return 1
    } else {
        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 suku ke-%d:\n", n)
    for i := 0; i <= n; i++ {
        fmt.Printf("%d ", fibonacci(i))
    }
    fmt.Println()
}
```

Screenshoot Output



```
PS C:\Users\ASUS\OneDrive\Documents\Kuliah\Semester 3\P. Alpro 2\modul 5> go run "c:\Users\ASUS\OneDrive\Documents\Kuliah\Semester 3\P. Alpro 2\modul 5\unguides\unguided.go"
Masukkan nilai n : 5
Deret Fibonacci hingga suku ke-5:
0 1 1 2 3 5
PS C:\Users\ASUS\OneDrive\Documents\Kuliah\Semester 3\P. Alpro 2\modul 5>
```

Deskripsi Program

Program ini adalah program untuk menghitung dan mencetak deret Fibonacci hingga suku ke-n secara rekursif. Fungsi utama dalam program ini adalah `fibonacci(n int) int`, yang berfungsi menghitung nilai suku Fibonacci pada posisi ke-n. Fungsi ini memiliki base case : jika $n==0$, maka akan mengembalikan 0, jika $n==1$, maka akan mengembalikan 1. Untuk nilai n yang lebih besar dari 1, fungsi akan mengembalikan penjumlahan dari dua suku sebelumnya, yaitu `fibonacci(n-1) + fibonacci(n-2)`. Pada fungsi `main()`, program meminta input dari pengguna. Setelah pengguna memasukkan nilai n , program akan mencetak setiap nilai dari suku ke-0 hingga suku ke-n dengan memanggil fungsi `fibonacci` secara rekursif untuk setiap suku. Sebagai contoh, jika pengguna memasukkan angka 10, maka program akan menampilkan deret Fibonacci dari 0 hingga 55, sesuai dengan format yang diminta pada soal.

2. Soal Studi Case

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

Contoh masukan dan keluaran:

No	Masukan	Keluaran
1	5	* ** *** **** *****
2	1	*
3	3	* ** ***

Sourcecode

```
package main

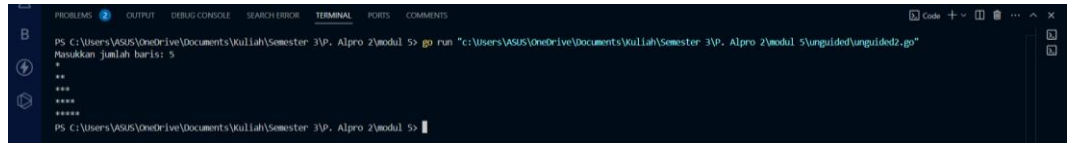
import "fmt"

func bintang(hitung int) {
    if hitung > 0 {
        fmt.Print("*")
        bintang(hitung - 1)
    }
}

func cetak(n int, baris int) {
    if baris <= n {
        bintang(baris)
        fmt.Println()
        cetak(n, baris + 1)
    }
}

func main() {
    var n int
    fmt.Print("Masukkan jumlah baris: ")
    fmt.Scan(&n)
    cetak(n, 1)
}
```

Screenshoot Output



```
PS C:\Users\ASUS\OneDrive\Documents\Kuliah\Semester 3\Pro. Alpro 2\modul 5> go run "c:\Users\ASUS\OneDrive\Documents\Kuliah\Semester 3\Pro. Alpro 2\modul 5\unguided2.go"
Masukkan jumlah baris: 5
*
**
***
****
*****
PS C:\Users\ASUS\OneDrive\Documents\Kuliah\Semester 3\Pro. Alpro 2\modul 5>
```

Deskripsi Program

Program tersebut adalah program untuk membuat pola bintang sesuai inputan dari pengguna. Program ini menggunakan 2 fungsi yaitu fungsi bintang dan fungsi cetak. Fungsi bintang mencetak jumlah bintang yang ada pada 1 baris, dan fungsi cetak digunakan untuk mencetak pola hingga mencapai jumlah baris yang diinginkan.

3. Soal Studi Case

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

Masukan terdiri dari sebuah bilangan bulat positif N.

Keluaran terdiri dari barisan bilangan yang menjadi faktor dari N (terurut dari 1 hingga N ya).

Contoh masukan dan keluaran:

No	Masukan	Keluaran
1	5	1 5
2	12	1 2 3 4 6 12

Sourcecode

```
package main

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

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

Screenshoot Output



```
PS C:\Users\ASUS\OneDrive\Documents\Kuliah\Semester 3\Pro. Alpro 2\modul 5> go run "c:\Users\ASUS\OneDrive\Documents\Kuliah\Semester 3\Pro. Alpro 2\modul 5\unguided3.go"
Masukkan bilangan: 10
Faktor dari 10: 1 2 5 10
PS C:\Users\ASUS\OneDrive\Documents\Kuliah\Semester 3\Pro. Alpro 2\modul 5>
```

Deskripsi Program

Program tersebut adalah program untuk menampilkan factor-faktor dari bilangan positif n yang diinputkan user. Program ini memiliki fungsi faktor yang menerima 2 parameter. Jika bilangan lebih besar dari n maka fungsi akan berhenti. Jika n habis dibagi bil , maka bil dicetak sebagai faktor dari n .

4. Soal Studi Case

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

Masukan terdiri dari sebuah bilangan bulat positif N.

Keluaran terdiri dari barisan bilangan dari N hingga 1 dan kembali ke N.

Contoh masukan dan keluaran:

No	Masukan	Keluaran
1	5	5 4 3 2 1 2 3 4 5
2	9	9 8 7 6 5 4 3 2 1 2 3 4 5 6 7 8 9

Sourcecode

```
package main

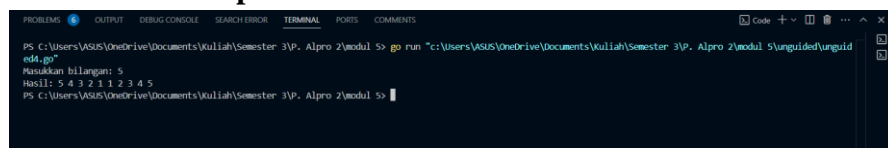
import "fmt"

func tampilBilangan(n int, m int) {
    if m < 1 {
        return
    }
    fmt.Print(m, " ")
    tampilBilangan(n, m-1)
    fmt.Print(m, " ")
}

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

    fmt.Print("Hasil: ")
    tampilBilangan(n, n)
    fmt.Println()
}
```

Screenshoot Output



```
PS C:\Users\VASUS\OneDrive\Documents\kuliah\Semester 3\P. Alpro 2\modul 5> go run "c:\Users\VASUS\OneDrive\Documents\kuliah\Semester 3\P. Alpro 2\modul 5\unguided\unguided.go"
Masukkan bilangan: 5
Hasil: 5 4 3 2 1 2 3 4 5
PS C:\Users\VASUS\OneDrive\Documents\kuliah\Semester 3\P. Alpro 2\modul 5>
```

Deskripsi Program

Program tersebut berfungsi untuk menampilkan bilangan dari n ke 1, lalu kembali lagi ke n. Program ini memiliki fungsi tampilBilangan yang memiliki 2 parameter yaitu n dan m. Program ini meminta user untuk memasukkan bilangan n dan m. Jika nilai m kurang dari 1, fungsi akan berhenti, jika tidak maka program akan menampilkan nilai m, lalu menampilkan m-1 dan seterusnya hingga 1, kemudian kembali menampilkan bilangannya meningkat +1 hingga m.

5. Soal Studi Case

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

Masukan terdiri dari sebuah bilangan bulat positif N.

Keluaran terdiri dari barisan bilangan ganjil dari 1 hingga N.

Contoh masukan dan keluaran:

No	Masukan	Keluaran
1	5	1 3 5
2	20	1 3 5 7 9 11 13 15 17 19

Sourcecode

```
package main

import "fmt"

func tampilGanjil(n int, i int) {
    if i > n {
        return
    }
    fmt.Print(i, " ")
    tampilGanjil(n, i+2)
}

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

    fmt.Print("Bilangan ganjil dari 1 hingga ", n, ": ")
    tampilGanjil(n, 1)
    fmt.Println()
}
```

Screenshoot Output



```
PS C:\Users\ASUS\OneDrive\Documents\Kuliah\Semester 3\Pro. Alpro 2\modul 5> go run "c:\Users\ASUS\OneDrive\Documents\Kuliah\Semester 3\Pro. Alpro 2\modul 5\unguided\unguid
eds.go"
Masukkan bilangan: 10
Bilangan ganjil dari 1 hingga 10: 1 3 5 7 9
PS C:\Users\ASUS\OneDrive\Documents\Kuliah\Semester 3\Pro. Alpro 2\modul 5>
```

Deskripsi Program

Program tersebut adalah program untuk menampilkan bilangan ganjil. Dimana n adalah bilangan maksimum yang diinputkan oleh pengguna, dan i adalah bilangan ganjil yang akan ditampilkan. Pertama program akan meminta user untuk menginputkan sebuah bilangan bulat, lalu program akan menampilkan bilangan ganjil dari 1 hingga n .

6. Soal Studi Case

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

Masukan terdiri dari bilangan bulat x dan y.

Keluaran terdiri dari hasil x dipangkatkan y.

Catatan: diperbolehkan menggunakan asterik "*", tapi dilarang menggunakan import "math".

Contoh masukan dan keluaran:

No	Masukan	Keluaran
1	2 2	4
2	5 3	125

Sourcecode

```
package main

import "fmt"

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

func main() {
    var x, y int
    fmt.Print("Masukkan basis (x): ")
    fmt.Scan(&x)
    fmt.Print("Masukkan pangkat (y): ")
    fmt.Scan(&y)

    hasil := pangkat(x, y)
    fmt.Printf("Hasil dari %d^%d adalah: %d\n", x, y, hasil)
}
```


Screenshoot Output



```
PS C:\Users\ASUS\OneDrive\Documents\Kuliah\Semester 3\P. Alpro 2(modul 5)> go run "c:\Users\ASUS\OneDrive\Documents\Kuliah\Semester 3\P. Alpro 2(modul 5)\unguided\unguided.go"
Masukkan basis (x): 3
Masukkan pangkat (y): 3
Hasil dari 3^3 adalah: 27
PS C:\Users\ASUS\OneDrive\Documents\Kuliah\Semester 3\P. Alpro 2(modul 5)>
```

Deskripsi Program

Program tersebut merupakan program pangkat, dimana x dipangkatkan dengan y dari inputan user. Dimana x adalah base bilangan yang akan dipangkatkan, sedangkan y adalah pangkat dari x tersebut.