

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2
REKURSIF
MODUL V**



Disusun Oleh :

Rakha Arbiyandanu / 2311102263

IF-11-6

Dosen Pengampu :

ABEDNEGO DWI SEPTIADI

PROGRAM STUDI S1 TEKNIK INFORMATIKA

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2024

I. DASAR TEORI

Rekursif Function

Apa itu recursive function? Recursive function adalah sebuah function yang memanggil/mengeksekusi dirinya sendiri. Recursive function bisa dikatakan salah satu yang bisa kita gunakan untuk melakukan perulangan. Ketika menulis kode aplikasi, terkadang ada kasus dimana akan lebih mudah jika dilakukan dengan recursive function. Contoh penggunaan sederhana recursive function adalah ketika melakukan operasi factorial.

Saat membuat recursive function, kita harus memastikan function tersebut dapat berhenti. Jika tidak, maka akan terkena error stack overflow atau melebihi limit stack karena function terus memanggil dirinya sendiri. Oleh karena itu, biasanya recursive function tidak langsung memanggil dirinya sendiri tetapi bergantung pada kondisi tertentu.

Cara kerja

Rekursi bekerja dengan membagi masalah menjadi bagian-bagian yang lebih kecil, di mana setiap bagian diselesaikan secara mandiri oleh pemanggilan fungsi itu sendiri. Rekursi selalu memiliki dua komponen utama:

1. Base Case (Kasus Dasar)

Kasus dasar adalah kondisi di mana fungsi rekursif berhenti memanggil dirinya sendiri. Ini adalah titik henti yang memastikan rekursi tidak berjalan tanpa batas. Tanpa kasus dasar, fungsi akan terus memanggil dirinya sendiri, menghasilkan loop tak terbatas yang menyebabkan stack overflow error.

2. Recursive Case (Kasus Rekursif)

Kasus rekursif adalah bagian di mana fungsi memanggil dirinya sendiri dengan parameter atau kondisi yang lebih kecil atau lebih sederhana dari panggilan sebelumnya, sehingga secara bertahap menuju ke base case.

I. GUIDED

Soal Studi Case

Membuat baris bilangan dan n hingga 1 base case : bilangan == 1

Sourcecode

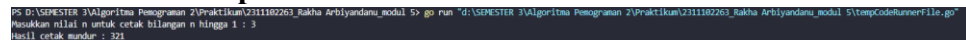
```
package main

import "fmt"

func cetakMundur(n int) {
    if n == 1 {
        fmt.Println(n)
        return
    }
    fmt.Print(n, " ")
    cetakMundur(n - 1)
}

func main() {
    var n int
    fmt.Print("Masukkan nilai n untuk cetak bilangan n hingga 1 : ")
    fmt.Scanln(&n)
    fmt.Print("Hasil cetak mundur : ")
    cetakMundur(n)
}
```

Screenshoot Output



```
PS D:\SEPESTER 3\Algoritma Pemrograman 2\Praktikum\2311180263_Rakha Arbiyandanu_modul 5> go run "d:\SEPESTER 3\Algoritma Pemrograman 2\Praktikum\2311180263_Rakha Arbiyandanu_modul 5\TempCodeRunnerFile.go"
Masukkan nilai n untuk cetak bilangan n hingga 1 : 3
Hasil cetak mundur : 3 2 1
```

Deskripsi Program

Program Go ini mencetak angka mundur dari nnn sampai 1 secara otomatis menggunakan rekursi. Di fungsi cetakMundur, kalau nnn sudah mencapai 1, program bakal cetak angka itu dan berhenti; tapi kalau nnn masih lebih dari 1, program akan cetak angka nnn, terus panggil dirinya sendiri dengan nilai $n-1$ untuk lanjut mundur. Di fungsi main, pengguna diminta memasukkan angka nnn, lalu program akan tampilkan hasilnya. Misalnya, kalau $n=5$, hasilnya jadi "5 4 3 2 1".

II. GUIDED

Soal Studi Case

Menghitung hasil penjumlahan 1 hingga n Base-case $n == 1$.

Sourcecode

```
package main

import "fmt"

func jumlahRekursif(n int) int {
    if n == 1 {
        return 1
    }

    return n + jumlahRekursif(n-1)
}

func main() {
    var n int

    fmt.Print("Masukkan nilai n untuk penjumlahan 1 hingga n: ")
    fmt.Scanln(&n)
    fmt.Println("Hasil penjumlahan:", jumlahRekursif(n))
}
```

Screenshoot Output



```
PS D:\SEMESTER 3\Algoritma Pemrograman 2\Praktikum\2311180263_Rakha Arbiyandaru_modul 5> go run "d:\SEMESTER 3\Algoritma Pemrograman 2\Praktikum\2311180263_Rakha Arbiyandaru_modul 5\tempCodeRunnerFile.go"
Masukkan nilai n untuk penjumlahan 1 hingga n: 5
Hasil penjumlahan: 15
PS D:\SEMESTER 3\Algoritma Pemrograman 2\Praktikum\2311180263_Rakha Arbiyandaru_modul 5>
```

Deskripsi Program

Program Go ini menghitung jumlah bilangan dari 1 hingga n menggunakan fungsi rekursif `jumlahRekursif`. Dalam fungsi tersebut, jika `n` bernilai 1, fungsi mengembalikan 1 sebagai hasil akhir. Namun, jika `n` lebih besar dari 1, fungsi akan menambah nilai `n` dengan hasil dari pemanggilan fungsi `jumlahRekursif` dengan nilai `n-1`. Sehingga proses rekursi akan terus berjalan sampai `n` mencapai 1. Di fungsi `main`, pengguna diminta memasukkan nilai `n`, dan hasil dari penjumlahan akan ditampilkan.

III. GUIDED

Soal Studi Case

Mencari dua pangkat Base – Case: $n == 0$

Sourcecode

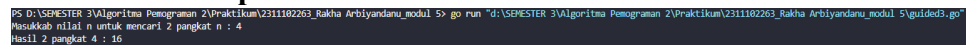
```
package main

import "fmt"

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

func main(){
    var n int
    fmt.Print("Masukkan nilai n untuk mencari 2 pangkat n
: ")
    fmt.Scanln(&n)
    fmt.Println("Hasil 2 pangkat", n, ":", pangkatDua(n))
}
```

Screenshoot Output



```
PS D:\SEMESTER 3\Algoritma Pemrograman 2\Praktikum\2311182263_Rakha Arbiyandaru_modul 5> go run "d:\SEMESTER 3\Algoritma Pemrograman 2\Praktikum\2311182263_Rakha Arbiyandaru_modul 5\guided3.go"
Masukkan nilai n untuk mencari 2 pangkat n : 4
Hasil 2 pangkat 4 : 16
```

Deskripsi Program

Program Go ini menghitung 2 pangkat n dengan menggunakan fungsi rekursif `pangkatDua`. Di fungsi ini, kalau n sama dengan 0, program langsung mengembalikan 1, karena $2^0 = 1$. Tapi kalau n lebih dari 0, program akan mengalikan 2 dengan hasil panggilan fungsi `pangkatDua` dengan $n-1$, jadi hasilnya akan dihitung bertahap. Di fungsi `main`, pengguna diminta memasukkan nilai n , dan program akan menampilkan hasil 2 pangkat n . Misalnya, kalau $n=3$, hasilnya adalah "8" (karena $2^3 = 8$).

IV. GUIDED

Soal Studi Case

Mencari nilai faktorial atau $n!$. Base-Case: $n == 0$ atau $n == 1$

Sourcecode

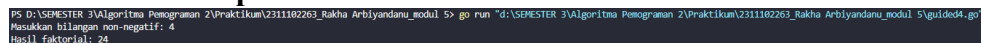
```
package main
import "fmt"

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

    if n < 0 {
        fmt.Println("Faktorial tidak didefinisikan untuk bilangan negatif")
    } else {
        fmt.Println("Hasil faktorial:", faktorial(n))
    }
}

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

Screenshoot Output



```
PS D:\SEMESTER 3\Algoritma Pemrograman 2\Praktikum\2311102263_Rakha Arbiyandani_modul 5> go run "d:\SEMESTER 3\Algoritma Pemrograman 2\Praktikum\2311102263_Rakha Arbiyandani_modul 5\guided4.go"
Masukkan bilangan non-negatif: 4
Hasil faktorial: 24
```

Deskripsi Program

Program Go ini menghitung faktorial dari angka yang dimasukkan pengguna, selama angkanya tidak negatif. Di fungsi main, pengguna diminta memasukkan angka. Kalau angkanya negatif, program langsung memberi tahu bahwa faktorial tidak bisa dihitung untuk angka negatif. Tapi kalau angkanya nol atau lebih, program akan menghitung faktorialnya dengan memanggil fungsi faktorial. Di fungsi ini, kalau nnn sama dengan 0 atau 1, langsung dikembalikan 1, karena $0!0!0!$ dan $1!1!1!$ sama dengan 1. Kalau nnn lebih dari 1, program akan mengalikan nnn dengan faktorial dari angka sebelumnya hingga selesai. Misalnya, kalau pengguna memasukkan angka 4, hasilnya jadi "24" (karena $4!=4 \times 3 \times 2 \times 1=24$ $4! = 4 \times 3 \times 2 \times 1 = 24$ $4! = 4 \times 3 \times 2 \times 1 = 24$).

I. UNGUIDED

Soal Studi Case

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.

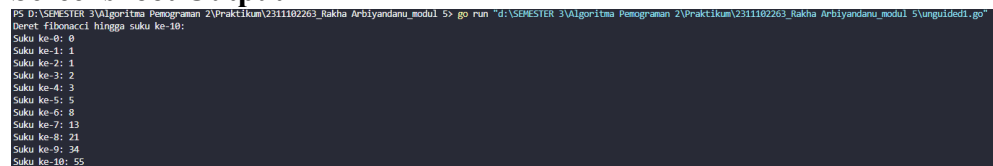
Sourcecode

```
package main
import "fmt"

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

func main(){
    var n int = 10
    fmt.Printf("Deret fibonacci hingga suku ke-%d:\n",
n)
    for i :=0; i <=n; i++){
        fmt.Printf("Suku ke-%d: %d\n", i, fibonacci(i))
    }
}
```

Screenshot Output



```
PS D:\SDPESIER 3\Algoritma Pemrograman 2\Praktikum\2311182263_Rakha Arbiyandani_modul 5> go run "d:\SDPESIER 3\Algoritma Pemrograman 2\Praktikum\2311182263_Rakha Arbiyandani_modul 5\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

Program ini menghitung deret Fibonacci sampai suku ke-10 menggunakan cara rekursif. Fungsi `fibonacci(n int) int` bekerja dengan menghitung suku ke-n dalam deret Fibonacci, yaitu menambahkan dua suku sebelumnya. Kalau n sama dengan 0 atau 1, hasilnya langsung dikembalikan (ini batas biar programnya nggak looping terus). Di bagian main, variabel n diset ke 10, artinya kita mau deret Fibonacci sampai suku ke-10. Lalu, ada for loop

yang ngulang dari suku ke-0 sampai ke-10 dan mencetak setiap suku tersebut satu per satu dengan cara manggil fungsi fibonacci tadi.

II. UNGUIDED

Soal Studi Case

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

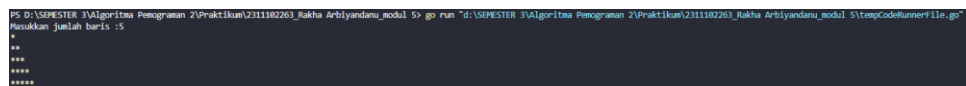
Sourcecode

```
package main
import "fmt"

func cetakBintang(n, barisdicetak int){
    if barisdicetak > n {
        return
    }
    for i := 0; i < barisdicetak; i++){
        fmt.Print("*")
    }
    fmt.Println()
    cetakBintang(n, barisdicetak+1)
}

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

Screenshoot Output



```
PS D:\SEMESTER 3\Algoritma Pemrograman 2\Praktikum\2311180263_Rakha Arbiyandaru_modul 5> go run "d:\SEMESTER 3\Algoritma Pemrograman 2\Praktikum\2311180263_Rakha Arbiyandaru_modul 5\tempcodeunnerfile.go"
Masukkan jumlah baris :5
*
**
***
****
*****
```

Deskripsi Program

Program ini akan mencetak segitiga bintang sesuai dengan jumlah baris yang dimasukkan oleh pengguna. Fungsi cetakBintang ini dipanggil secara berulang (rekursif) untuk mencetak bintang di setiap baris. Kalau nilai barisdicetak (jumlah baris yang sudah dicetak) lebih besar dari n (jumlah baris yang diminta), fungsi akan berhenti. Di setiap baris, jumlah bintang yang dicetak sesuai dengan angka barisdicetak, jadi baris pertama ada satu bintang, baris kedua dua bintang, dan seterusnya. Setelah mencetak bintang di satu baris, fungsi akan lompat ke baris berikutnya dan menambah jumlah bintang, sampai akhirnya tercetak segitiga bintang sesuai jumlah baris yang diminta pengguna.

III. UNGUIDED

Soal Studi Case

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).

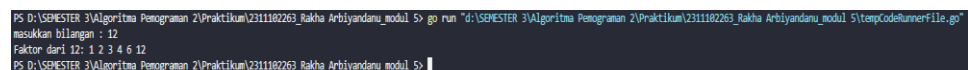
Sourcecode

```
package main
import "fmt"

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

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

Screenshoot Output



```
PS D:\SEMESTER 3\Algoritma Pemrograman 2\Praktikum\2311182263_Rakha Arbiyandaru_modul 5> go run "d:\SEMESTER 3\Algoritma Pemrograman 2\Praktikum\2311182263_Rakha Arbiyandaru_modul 5\tempCodeRunnerFile.go"
masukkan bilangan : 12
Faktor dari 12: 1 2 3 4 6 12
PS D:\SEMESTER 3\Algoritma Pemrograman 2\Praktikum\2311182263_Rakha Arbiyandaru_modul 5>
```

Deskripsi Program

Program ini mencetak semua faktor dari sebuah bilangan yang dimasukkan oleh pengguna. Fungsi `cetakFaktor(n, i int)` bekerja secara rekursif untuk mengecek dan mencetak setiap faktor `i` dari bilangan `n`. Jika nilai `i` lebih besar dari `n`, fungsi berhenti. Untuk setiap nilai `i` yang merupakan faktor dari `n` (jika `n % i == 0`), nilai `i` dicetak. Setelah itu, fungsi memanggil dirinya sendiri dengan nilai `i` bertambah 1, sehingga program akan terus mengecek dan mencetak semua faktor dari 1 hingga `n`. Di bagian `main`, pengguna diminta memasukkan bilangan, dan hasilnya ditampilkan dengan memanggil `cetakFaktor`.

IV. UNGUIDED

Soal Studi Case

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`. **Sourcecode**

```
package main
import "fmt"

func deretBilangan(n, bilangan int){
    if bilangan > n {
        return
    }

    fmt.Print(bilangan, " ")
    if bilangan == 1 {
        balikPola(2, n)
    }else {
        deretBilangan(n, bilangan-1)
    }
}

func balikPola(bilangan, n int){
    if bilangan > n {
        return
    }

    fmt.Print(bilangan, " ")
```

```

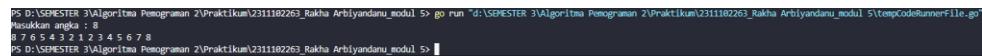
        balikPola(bilangan+1, n)
    }

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

        deretBilangan(n, n)
        fmt.Println()
    }

```

Screenshoot Output



```

PS D:\SEMESTER 3\Algoritma Pemrograman 2\Praktikum\2311180263_Rakha Arbiyandaru_modul 5> go run -d:\SEMESTER 3\Algoritma Pemrograman 2\Praktikum\2311180263_Rakha Arbiyandaru_modul 5\tempcodeidnerfile.go
Masukkan angka : 8
8 7 6 5 4 3 2 1 2 3 4 5 6 7 8
PS D:\SEMESTER 3\Algoritma Pemrograman 2\Praktikum\2311180263_Rakha Arbiyandaru_modul 5>

```

Deskripsi Program

Program ini mencetak deretan angka yang turun dari n ke 1, lalu naik lagi dari 1 ke n . Fungsi `deretBilangan` mulai mencetak dari angka n , lalu terus mengurangi angkanya satu per satu sampai mencapai 1. Saat sudah sampai di angka 1, fungsi `balikPola` dipanggil untuk mencetak angka dari 2 sampai n , sehingga deretnya naik lagi. Di bagian `main`, diminta sebuah angka n , lalu pola ini dicetak dengan memanggil `deretBilangan`—hasilnya adalah angka yang turun dan kembali naik sesuai angka yang dimasukkan.

V. UNGUIDED

Soal Studi Case

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.

Sourcecode

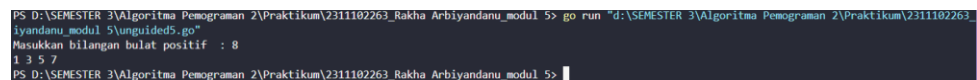
```
package main
import "fmt"

func printBarisanBilanganGanjil(n int, bilangan int){
    if bilangan > n{
        return
    }

    fmt.Print(bilangan, " ")
    printBarisanBilanganGanjil(n, bilangan+2)
}

func main() {
    var N int
    fmt.Print("Masukkan bilangan bulat positif : ")
    fmt.Scan(&N)
    printBarisanBilanganGanjil(N, 1)
    fmt.Println()
}
```

Screenshoot Output



```
PS D:\SEMESTER 3\Algoritma Pemrograman 2\Praktikum\2311102263_Rakha Arbiyandanu_modul 5> go run "d:\SEMESTER 3\Algoritma Pemrograman 2\Praktikum\2311102263_Rakha Arbiyandanu_modul 5\unguided5.go"
Masukkan bilangan bulat positif : 8
1 3 5 7
PS D:\SEMESTER 3\Algoritma Pemrograman 2\Praktikum\2311102263_Rakha Arbiyandanu_modul 5>
```

Deskripsi Program

Program Go ini digunakan untuk mencetak angka ganjil mulai dari 1 hingga angka yang dimasukkan. Di bagian utama program, diminta untuk memasukkan angka positif N, yang menjadi batas atas deretan angka ganjil yang akan dicetak. Fungsi printBarisanBilanganGanjil akan mulai

mencetak angka 1 dan terus menambahkan 2 untuk mencetak angka ganjil berikutnya. Proses ini akan berhenti jika angka yang dicetak sudah lebih dari NNN. Jadi, jika memasukkan $N=10$, program akan menampilkan "1 3 5 7 9".

VI. UNGUIDED

Soal Studi Case

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".

Sourcecode

```
package main

import "fmt"

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

func main() {
    var x, y int

    fmt.Print("Masukkan nilai x dan y: ")

    fmt.Scan(&x, &y)

    hasil := hasilpangkat(x, y)

    fmt.Printf("%d pangkat %d adalah %d\n", x, y, hasil)
}
```

Screenshoot Output

```
PS D:\SEMESTER 3\Algoritma Pemograman 2\Praktikum\2311102263_Rakha Arbiyandanu_modul 5> go run "d:\SEMESTER 3\Algoritma Pemograman 2\Praktikum\2311102263_Rakha Arbiyandanu_modul 5\tempCodeRunnerFile.go"
Masukkan nilai x dan y: 3 4
3 pangkat 4 adalah 81
PS D:\SEMESTER 3\Algoritma Pemograman 2\Praktikum\2311102263_Rakha Arbiyandanu_modul 5>
```

Deskripsi Program

Program ini menghitung hasil dari sebuah angka x yang dipangkatkan dengan angka y . Fungsi hasilpangkat bekerja dengan cara rekursif, di mana jika y sama dengan 0, hasilnya adalah 1 (karena angka berapa pun dipangkatkan 0 hasilnya 1). Jika y lebih besar dari 0, program akan mengalikan x dengan hasil pangkat dari x dengan $y-1$, sampai akhirnya mencapai kondisi di mana y menjadi 0. Di bagian main, program meminta pengguna untuk memasukkan nilai x dan y , kemudian menghitung hasil pangkat menggunakan fungsi hasilpangkat dan menampilkan hasilnya.