

**LAPORAN PRAKTIKUM  
ALGORITMA PEMROGRAMAN 2**

**MODUL V  
REKURSIF**



Oleh:

**ALIFATUS SHABRINA AMALIA**

NIM:

**2311102225**

**S1 TEKNIK INFORMATIKA  
TELKOM UNIVERSITY PURWOKERTO  
2024**

## **I. DASAR TEORI**

Pada 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.

## II. GUIDED

### 1. Guided 1

Source code

```
package main

import "fmt"

func main() {
    var n int

    fmt.Scan(&n) // Membaca input pengguna

    baris(n)     // Memanggil fungsi rekursif 'baris'
}

func baris(bilangan int) {
    if bilangan == 1 { // Base case: jika bilangan sama dengan 1
        fmt.Println(1) // Cetak angka 1
    } else { // Jika bilangan lebih besar dari 1
        fmt.Println(bilangan) // Cetak bilangan saat ini
        baris(bilangan - 1)    // Panggil fungsi 'baris' dengan
        bilangan dikurangi 1
    }
}
```

### Screenshoot program

```
PS C:\Users\acer> go run "d:\Praktikum\Alpro2\Modul5\Guided\guided1.go"
9
9
8
7
6
5
4
3
2
1
PS C:\Users\acer> |
```

### Deskripsi program

Program di atas meminta input angka n dari pengguna, kemudian mencetak angka-angka dari n hingga 1 secara berurutan menggunakan fungsi rekursif bernama baris. Fungsi ini akan mencetak bilangan saat ini dan memanggil dirinya sendiri dengan bilangan yang dikurangi 1, hingga mencapai kondisi dasar saat bilangan sama dengan 1, di mana angka 1 dicetak dan rekursi berhenti.

## 2. Guided 2

### Source code

```
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

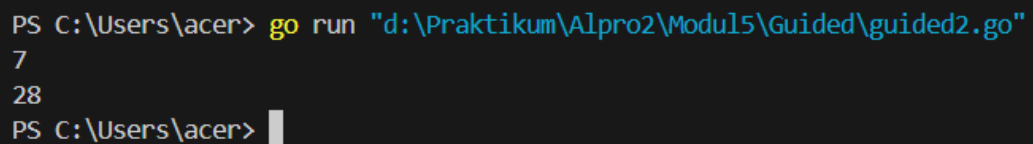
    } else {

        return n + penjumlahan(n-1)

    }

}
```

### Screenshoot program



```
PS C:\Users\acer> go run "d:\Praktikum\Alpro2\Modul5\Guided\guided2.go"
7
28
PS C:\Users\acer> |
```

### Deskripsi program

Program di atas meminta input angka  $n$  dari pengguna, kemudian menghitung penjumlahan semua angka dari 1 hingga  $n$  menggunakan fungsi rekursif bernama `penjumlahan`. Fungsi ini akan terus menambah angka dari  $n$  sampai mencapai 1, di mana kondisi dasar rekursinya ( $n == 1$ ) akan menghentikan pemanggilan berulang. Hasil akhirnya adalah penjumlahan dari 1 hingga  $n$ , yang ditampilkan ke layar.

### III. UNGUIDED

#### 1. Unguided 1

##### Source code

```
package main

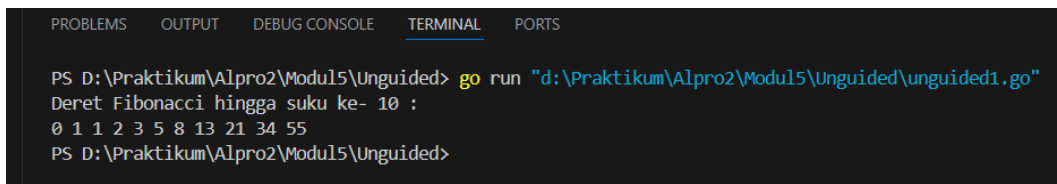
import "fmt"

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

func main() {
    n := 10

    fmt.Println("Deret Fibonacci hingga suku ke-", n, ":")
    for i := 0; i <= n; i++ {
        fmt.Print(fibonacci(i), " ")
    }
}
```

##### Screenshoot program



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS D:\Praktikum\Alpro2\Modul5\Unguided> go run "d:\Praktikum\Alpro2\Modul5\Unguided\unguided1.go"
Deret Fibonacci hingga suku ke- 10 :
0 1 1 2 3 5 8 13 21 34 55
PS D:\Praktikum\Alpro2\Modul5\Unguided>
```

## Deskripsi program

- Fungsi fibonacci(n) akan mengembalikan nilai Fibonacci pada indeks ke-n.
  - Jika  $n \leq 1$ , maka fungsi langsung mengembalikan nilai n (0 untuk  $n = 0$ , dan 1 untuk  $n = 1$ ).
  - Jika  $n > 1$ , fungsi akan memanggil dirinya sendiri dengan fibonacci(n-1) dan fibonacci(n-2), yang merupakan dua suku sebelumnya.
- Pada fungsi main(), program menghitung dan menampilkan deret Fibonacci dari suku ke-0 hingga ke-10.

## 2. Unguided 2

### Source code

```
package main

import "fmt"

func printStars(n int) {
    if n == 0 {
        return
    }

    fmt.Print("*")

    printStars(n - 1)
}

func printPattern(n int, current int) {
    if current > n {
        return
    }

    printStars(current)

    fmt.Println()
}
```

```

        printPattern(n, current+1)
    }

func main() {
    var n int

    fmt.Print("Masukkan jumlah baris: ")

    fmt.Scan(&n)

    fmt.Println("Pola bintang:")

    printPattern(n, 1)
}

```

### Screenshoot program

```

PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS D:\Praktikum\Alpro2\Modul5\Unguided> go run "d:\Praktikum\Alpro2\Modul5\Unguided\unguided2.go"
Masukkan jumlah baris: 7
Pola bintang:
*
**
***
****
*****
*****
*****
PS D:\Praktikum\Alpro2\Modul5\Unguided>

```

### Deskripsi program

- Fungsi printStars(n int):
  - Fungsi ini digunakan untuk mencetak n bintang pada satu baris.
  - Jika n == 0, maka fungsi berhenti (basis kasus rekursi).
  - Jika tidak, fungsi mencetak satu bintang, lalu memanggil dirinya sendiri dengan n - 1 hingga n menjadi nol.
- Fungsi printPattern(n int, current int):
  - Fungsi ini mencetak pola bintang baris demi baris.
  - Argumen current digunakan untuk menentukan berapa banyak bintang yang akan dicetak di baris tersebut.



- Jika `current` lebih besar dari `n`, maka fungsi berhenti (basis kasus rekursi).
- Jika tidak, ia mencetak bintang sebanyak `current` menggunakan fungsi `printStars()`, lalu berpindah ke baris berikutnya dengan memanggil dirinya sendiri untuk baris berikutnya (`current + 1`).
- Fungsi `main()`:
  - Menerima input dari pengguna untuk menentukan jumlah baris (`n`).
  - Memanggil fungsi `printPattern()` untuk mencetak pola bintang dari baris 1 hingga baris ke-`n`.

### 3. Unguided 3

#### Source code

```
package main

import "fmt"

func findFactors(n, divisor int) {
    if divisor > n {
        return
    }

    if n%divisor == 0 {
        fmt.Print(divisor, " ")
    }

    findFactors(n, divisor+1)
}

func main() {
    var n int
```

```
    fmt.Print("Masukkan bilangan bulat positif: ")

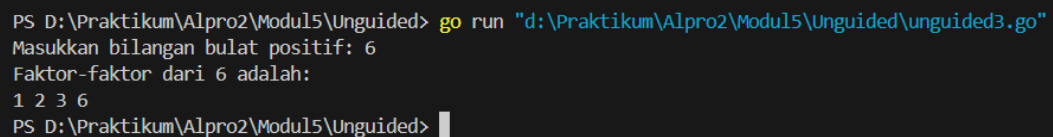
    fmt.Scan(&n)

    fmt.Println("Faktor-faktor dari", n, "adalah:")

    findFactors(n, 1)

}
```

### Screenshoot program



```
PS D:\Praktikum\Alpro2\Modul5\Unguided> go run "d:\Praktikum\Alpro2\Modul5\Unguided\unguided3.go"
Masukkan bilangan bulat positif: 6
Faktor-faktor dari 6 adalah:
1 2 3 6
PS D:\Praktikum\Alpro2\Modul5\Unguided> █
```

### Deskripsi program

- Fungsi findFactors(n, divisor int):
  - Fungsi ini mencari faktor-faktor dari n secara rekursif.
  - Basis kasus: Jika divisor lebih besar dari n, maka proses rekursif berhenti.
  - Jika n habis dibagi oleh divisor (dengan operasi  $n \% divisor == 0$ ), maka divisor adalah faktor dari n, dan akan dicetak. Fungsi memanggil dirinya sendiri dengan divisor + 1, artinya akan mencoba pembagi berikutnya hingga mencapai n.
- Fungsi main():
  - Menerima input bilangan bulat positif n dari pengguna.
  - Memanggil fungsi findFactors(n, 1) untuk mencari faktor-faktor dari n, dimulai dari 1 sebagai divisor awal.

## 4. Unguided 4

### Source code

```
package main

import "fmt"

func printDescending(n int) {

    if n == 0 {

        return

    }

    fmt.Print(n, " ")

    printDescending(n - 1)

}

func printAscending(n, current int) {

    if current > n {

        return

    }

    fmt.Print(current, " ")

    printAscending(n, current+1)

}

func main() {

    var n int

    fmt.Print("Masukkan bilangan bulat positif: ")

    fmt.Scan(&n)
```

```
printDescending(n)

printAscending(n, 2)

}
```

### Screenshoot program

```
PS D:\Praktikum\Alpro2\Modul5\Unguided> go run "d:\Praktikum\Alpro2\Modul5\Unguided\unguided4.go"
Masukkan bilangan bulat positif: 9
9 8 7 6 5 4 3 2 1 2 3 4 5 6 7 8 9
PS D:\Praktikum\Alpro2\Modul5\Unguided> █
```

### Deskripsi program

- Fungsi printDescending(n int):
  - Fungsi ini mencetak bilangan dari n ke 1 secara rekursif.
  - Basis kasus: Jika  $n == 0$ , fungsi berhenti.
  - Jika n lebih besar dari 0, angka n akan dicetak, kemudian fungsi memanggil dirinya sendiri dengan  $n - 1$ , sehingga menghasilkan urutan yang menurun.
- Fungsi printAscending(n, current int):
  - Fungsi ini mencetak bilangan dari 1 hingga n secara rekursif.
  - Basis kasus: Jika  $current > n$ , fungsi berhenti.
  - Jika tidak, angka current akan dicetak, dan fungsi memanggil dirinya sendiri dengan  $current + 1$  untuk mencetak angka berikutnya.
  - Fungsi ini dimulai dari 2 pada bagian barisan yang meningkat untuk menghindari mencetak angka 1 dua kali.
- Fungsi main():
  - Menerima input bilangan bulat positif n dari pengguna.
  - Memanggil fungsi printDescending(n) untuk mencetak bilangan dari n hingga 1.
  - Kemudian memanggil fungsi printAscending(n, 2) untuk mencetak bilangan dari 2 hingga n kembali.

## 5. Unguided 5

### Source code

```
package main

import "fmt"

func printOddNumbers(n, current int) {

    if current > n {

        return

    }

    if current%2 != 0 {

        fmt.Print(current, " ")

    }

    printOddNumbers(n, current+1)

}

func main() {

    var n int

    fmt.Print("Masukkan bilangan bulat positif: ")

    fmt.Scan(&n)

    fmt.Println("Bilangan ganjil dari 1 hingga", n, "adalah:")

    printOddNumbers(n, 1)

}
```

## Screenshoot program

```
PS D:\Praktikum\Alpro2\Modul5\Unguided> go run "d:\Praktikum\Alpro2\Modul5\Unguided\unguided5.go"
Masukkan bilangan bulat positif: 11
Bilangan ganjil dari 1 hingga 11 adalah:
1 3 5 7 9 11
PS D:\Praktikum\Alpro2\Modul5\Unguided> █
```

## Deskripsi program

- Fungsi printOddNumbers(n, current int):
  - Fungsi ini mencetak bilangan ganjil dari 1 hingga n.
  - Basis kasus: Jika current melebihi n, rekursi berhenti.
  - Jika current adalah bilangan ganjil (dengan kondisi  $\text{current} \% 2 \neq 0$ ), maka bilangan tersebut dicetak.
  - Setelah mencetak bilangan, fungsi memanggil dirinya sendiri dengan  $\text{current} + 1$  untuk memeriksa bilangan berikutnya.
- Fungsi main():
  - Menerima input bilangan bulat positif n dari pengguna.
  - Memanggil fungsi printOddNumbers(n, 1) untuk mulai mencetak bilangan ganjil dari 1 hingga n.

## 6. Unguided 6

### Source code

```
package main

import "fmt"

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

```

func main() {

    var x, y int

    fmt.Print("Masukkan nilai x (basis): ")

    fmt.Scan(&x)

    fmt.Print("Masukkan nilai y (pangkat): ")

    fmt.Scan(&y)

    result := power(x, y)

    fmt.Printf("%d dipangkatkan %d adalah: %d\n", x, y, result)

}

```

### Screenshoot program

```

PS D:\Praktikum\Alpro2\Modul5\Unguided> go run "d:\Praktikum\Alpro2\Modul5\Unguided\unguided6.go"
Masukkan nilai x (basis): 3
Masukkan nilai y (pangkat): 7
3 dipangkatkan 7 adalah: 2187
PS D:\Praktikum\Alpro2\Modul5\Unguided>

```

### Deskripsi program

- Fungsi power(x, y int):
  - Fungsi ini menghitung  $x^y$  secara rekursif.
  - Basis kasus: Jika  $y == 0$ , hasil dari setiap bilangan yang dipangkatkan 0 adalah 1.
  - Jika  $y > 0$ , fungsi mengembalikan  $x * \text{power}(x, y-1)$ , yang berarti mengalikan x dengan hasil pangkat sebelumnya ( $x^{(y-1)}$ ), dan berlanjut hingga  $y == 0$ .
- Fungsi main():
  - Menerima dua input dari pengguna, yaitu x (basis) dan y (pangkat).
  - Memanggil fungsi power(x, y) untuk menghitung hasil dari x dipangkatkan y.
  - Menampilkan hasilnya.