

**LAPORAN PRAKTIKUM  
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL 6  
REKURSIF**



**Disusun Oleh :**

**Afif Rijal Azzami / 2311102235**

**IF-11-05**

**Dosen Pengampu :**

**Arif Amrulloh, S.Kom., M.Kom.**

**PROGRAM STUDI S1 TEKNIK INFORMATIKA**

**FAKULTAS INFORMATIKA**

**TELKOM UNIVERSITY PURWOKERTO**

**2024**

## **I. DASAR TEORI**

### **A. Rekursif**

Rekursif adalah teknik dalam pemrograman di mana sebuah fungsi memanggil dirinya sendiri untuk menyelesaikan versi lebih kecil dari masalah asli. Metode ini digunakan untuk menangani masalah yang bisa dipecah menjadi sub-masalah yang lebih sederhana dan memiliki pola serupa, hingga kondisi dasar tercapai. Cara Pemanggilan Procedure

### **B. Komponen Rekursif**

Rekursif memiliki dua komponen utama, yaitu :

#### **1. Base-case:**

Base-case adalah kondisi dasar atau titik akhir yang digunakan untuk menghentikan pemanggilan rekursif. Komponen ini sangat penting karena tanpanya, rekursi akan terus berlanjut tanpa batas, menyebabkan program mengalami error atau kehabisan memori. Dengan kata lain, Base-case adalah syarat yang harus dipenuhi agar fungsi berhenti memanggil dirinya sendiri dan mengembalikan hasil.

#### **2. Recursive-case:**

Recursive-case adalah bagian dari algoritma rekursif di mana fungsi memanggil dirinya sendiri untuk mendekati Base-case. Pada setiap pemanggilan ulang, fungsi mempersempit masalah hingga mencapai kondisi dasar (Base-case). Recursive-case memungkinkan pembagian masalah yang kompleks menjadi bagian yang lebih kecil dan sederhana.

## II. GUIDED 1

### Sourcecode

```
package main

import (
    "fmt"
)

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

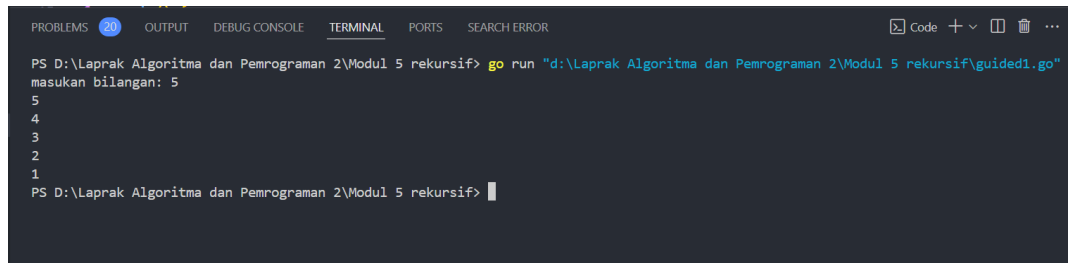
func main() {
    var n int

    fmt.Print("masukan bilangan: ")

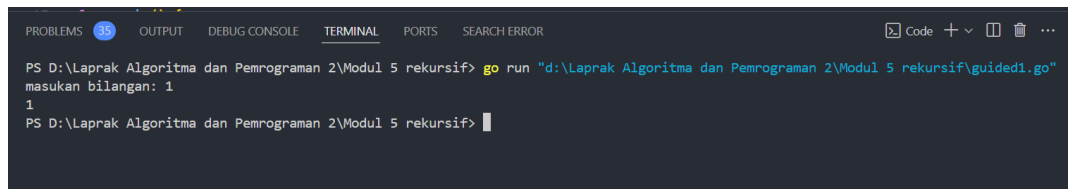
    fmt.Scan(&n)

    baris(n)
}
```

## Screenshoot Output



```
PROBLEMS 20 OUTPUT DEBUG CONSOLE TERMINAL PORTS SEARCH ERROR
PS D:\Laprak Algoritma dan Pemrograman 2\Modul 5 rekursif> go run "d:\Laprak Algoritma dan Pemrograman 2\Modul 5 rekursif\guided1.go"
masukan bilangan: 5
5
4
3
2
1
PS D:\Laprak Algoritma dan Pemrograman 2\Modul 5 rekursif> |
```



```
PROBLEMS 35 OUTPUT DEBUG CONSOLE TERMINAL PORTS SEARCH ERROR
PS D:\Laprak Algoritma dan Pemrograman 2\Modul 5 rekursif> go run "d:\Laprak Algoritma dan Pemrograman 2\Modul 5 rekursif\guided1.go"
masukan bilangan: 1
1
PS D:\Laprak Algoritma dan Pemrograman 2\Modul 5 rekursif> |
```

## Deskripsi Program

Sintaks tersebut adalah program untuk menampilkan baris angka yang di-inputkan oleh user secara menurun menggunakan prosedur rekursif, pertama kita membuat prosedur yang dinamai baris, di dalam prosedur tersebut terdapat percangan jika bilangan yang diinputkan 1 maka program akan menampilkan 1, dan pada else (yaitu bilangan bukan 1) terdapat rekursif, yaitu memanggil prosedur itu sendiri untuk menampilkan baris bilangan.

### III. GUIDED 2

```
package main

import (
    "fmt"
)

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

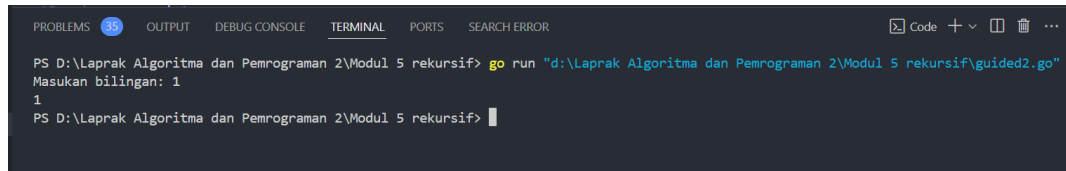
func main() {
    var n int

    fmt.Print("Masukan bilangan: ")

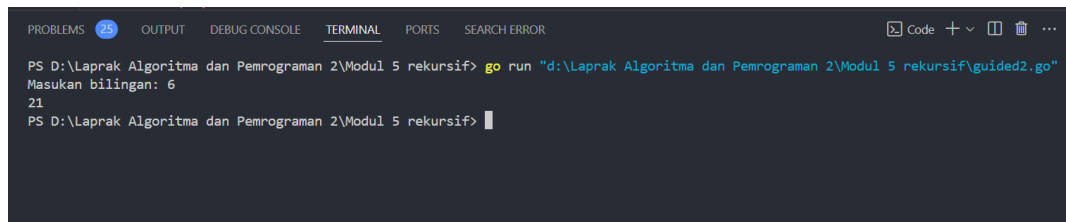
    fmt.Scan(&n)

    fmt.Println(penjumlahan(n))
}
```

## Screenshoot Output



```
PROBLEMS 35 OUTPUT DEBUG CONSOLE TERMINAL PORTS SEARCH ERROR
PS D:\Laprak Algoritma dan Pemrograman 2\Modul 5 rekursif> go run "d:\Laprak Algoritma dan Pemrograman 2\Modul 5 rekursif\guided2.go"
Masukan bilangan: 1
1
PS D:\Laprak Algoritma dan Pemrograman 2\Modul 5 rekursif> |
```



```
PROBLEMS 25 OUTPUT DEBUG CONSOLE TERMINAL PORTS SEARCH ERROR
PS D:\Laprak Algoritma dan Pemrograman 2\Modul 5 rekursif> go run "d:\Laprak Algoritma dan Pemrograman 2\Modul 5 rekursif\guided2.go"
Masukan bilangan: 6
21
PS D:\Laprak Algoritma dan Pemrograman 2\Modul 5 rekursif> |
```

## Deskripsi Program

Sintaks tersebut adalah program untuk menghitung penjumlahan dari n sampai 1 menggunakan fungsi rekursif, pertama kita membuat fungsi yang dinamai penjumlahan dan kita membuat percabangan jika bilangan yang diinputkan = 1 maka nilai akan dikembalikan jadi 1, dan pada elsenya (jika bilangan bukan 1) terdapat rekursif yaitu penjumlahan(n-1) untuk menghitung bilangan dari n sampe 1.

#### IV. GUIDED 3

```
package main

import (
    "fmt"
)

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

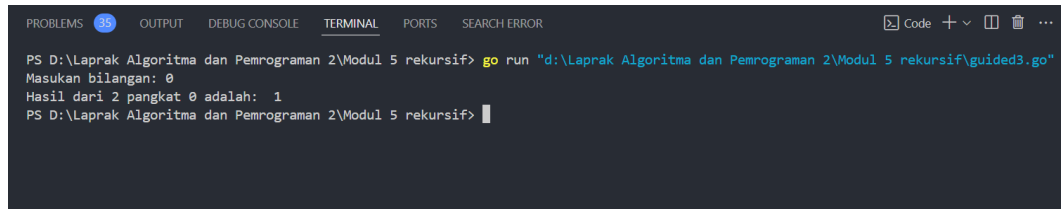
func main() {
    var n int

    fmt.Print("Masukan bilangan: ")

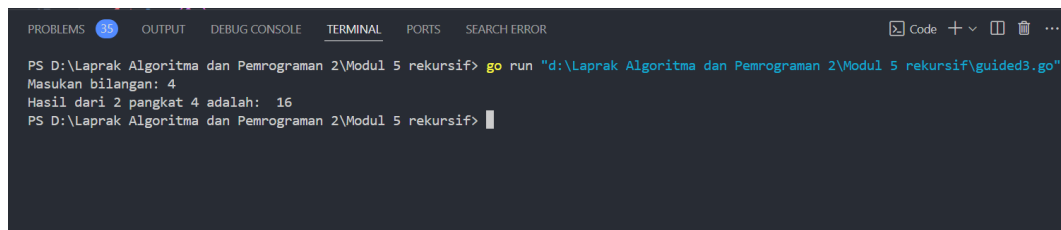
    fmt.Scan(&n)

    fmt.Println("Hasil dari 2 pangkat", n, "adalah: ",
pangkat(n))
}
```

## Screenshoot Output



```
PROBLEMS 35 OUTPUT DEBUG CONSOLE TERMINAL PORTS SEARCH ERROR
PS D:\Laprak Algoritma dan Pemrograman 2\Modul 5 rekursif> go run "d:\Laprak Algoritma dan Pemrograman 2\Modul 5 rekursif\guided3.go"
Masukan bilangan: 0
Hasil dari 2 pangkat 0 adalah: 1
PS D:\Laprak Algoritma dan Pemrograman 2\Modul 5 rekursif>
```



```
PROBLEMS 35 OUTPUT DEBUG CONSOLE TERMINAL PORTS SEARCH ERROR
PS D:\Laprak Algoritma dan Pemrograman 2\Modul 5 rekursif> go run "d:\Laprak Algoritma dan Pemrograman 2\Modul 5 rekursif\guided3.go"
Masukan bilangan: 4
Hasil dari 2 pangkat 4 adalah: 16
PS D:\Laprak Algoritma dan Pemrograman 2\Modul 5 rekursif>
```

## Deskripsi Program

Sintaks tersebut adalah program untuk menghitung 2 pangkat n menggunakan fungsi rekursif, pertama kita membuat fungsi yang dinamai pangkat, lalu kita membuat percabangan jika  $n = 0$  maka nilai akan dikembalikan jadi 1, dan pada else-nya (bilang tidak sama dengan 0) terdapat rekursif yaitu  $\text{pangkat}(n-1)$  untuk menghitung 2 pangkat dari n.



## I. GUIDED 4

### Sourcecode

```
package main

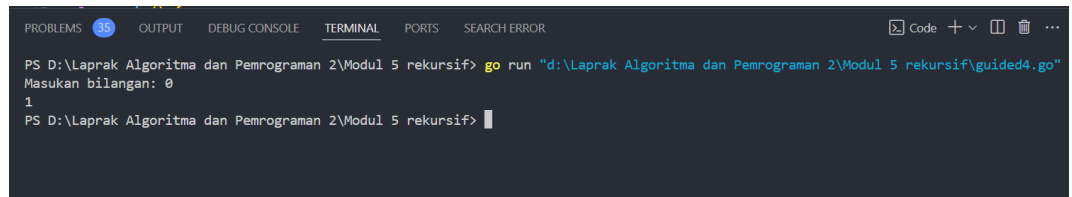
import (
    "fmt"
)

var n int

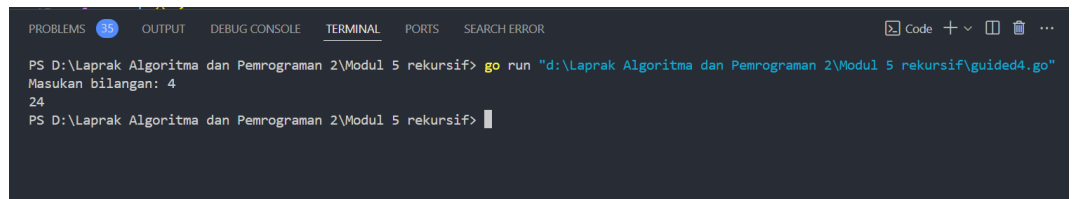
func faktorial (n int) int {
    if n == 0 || n == 1 {
        return 1
    } else {
        return n * faktorial(n - 1)
    }
}

func main() {
    fmt.Print("Masukan bilangan: ")
    fmt.Scan(&n)
    fmt.Println(faktorial(n))
}
```

## Screenshoot Output



```
PROBLEMS 35 OUTPUT DEBUG CONSOLE TERMINAL PORTS SEARCH ERROR
PS D:\Laprak Algoritma dan Pemrograman 2\Modul 5 rekursif> go run "d:\Laprak Algoritma dan Pemrograman 2\Modul 5 rekursif\guided4.go"
Masukan bilangan: 0
1
PS D:\Laprak Algoritma dan Pemrograman 2\Modul 5 rekursif>
```



```
PROBLEMS 35 OUTPUT DEBUG CONSOLE TERMINAL PORTS SEARCH ERROR
PS D:\Laprak Algoritma dan Pemrograman 2\Modul 5 rekursif> go run "d:\Laprak Algoritma dan Pemrograman 2\Modul 5 rekursif\guided4.go"
Masukan bilangan: 4
24
PS D:\Laprak Algoritma dan Pemrograman 2\Modul 5 rekursif>
```

## Deskripsi Program

Sintaks tersebut adalah program untuk menghitung faktorial dari  $n$  menggunakan fungsi rekursif, pertama kita membuat fungsi yang dinamai dengan faktorial, kemudian kita membuat percabangan jika  $n = 0$  atau  $1$  maka program akan mengembalikan  $1$ , dan pada else ( $n$  bukan  $= 0$  atau  $1$ ) terdapat rekursif yaitu faktorial( $n-1$ ), yaitu fungsi mengalikan  $n$  dengan hasil pemanggilan fungsi faktorial( $n - 1$ ), di mana nilai  $n$  dikurangi  $1$  setiap kali pemanggilan hingga mencapai kasus dasar.

## II. UNGUIDED 1

```
package main

import "fmt"

// Fungsi rekursif untuk menghitung Fibonacci ke-n
func fibonasi(n int) int {
    if n == 0 {
        return 0
    } else if n == 1 {
        return 1
    } else {
        return fibonasi(n-1) + fibonasi(n-2)
    }
}

func main() {
    // Menampilkan deret Fibonacci hingga suku ke-n
    var n int
    print("masukan bilangan: ")
    fmt.Scanln(&n)
    for i := 0; i <= n; i++ {
        fmt.Printf("Fibonaci ke-%d: %d\n", i, fibonasi(i))
    }
}
```

## Screenshoot Output

```
PROBLEMS 59 OUTPUT DEBUG CONSOLE TERMINAL PORTS SEARCH ERROR
PS D:\Laprak Algoritma dan Pemrograman 2\Modul 5 rekursif> go run "d:\Laprak Algoritma dan Pemrograman 2\Modul 5 rekursif\unguided1.go"
masukan bilangan: 0
Fibonacci ke-0: 0
PS D:\Laprak Algoritma dan Pemrograman 2\Modul 5 rekursif>
```

```
PROBLEMS 59 OUTPUT DEBUG CONSOLE TERMINAL PORTS SEARCH ERROR
PS D:\Laprak Algoritma dan Pemrograman 2\Modul 5 rekursif> go run "d:\Laprak Algoritma dan Pemrograman 2\Modul 5 rekursif\unguided1.go"
masukan bilangan: 1
Fibonacci ke-0: 0
Fibonacci ke-1: 1
PS D:\Laprak Algoritma dan Pemrograman 2\Modul 5 rekursif>
```

```
PROBLEMS 59 OUTPUT DEBUG CONSOLE TERMINAL PORTS SEARCH ERROR
PS D:\Laprak Algoritma dan Pemrograman 2\Modul 5 rekursif> go run "d:\Laprak Algoritma dan Pemrograman 2\Modul 5 rekursif\unguided1.go"
masukan bilangan: 10
Fibonacci ke-0: 1
Fibonacci ke-1: 1
Fibonacci ke-2: 2
Fibonacci ke-3: 3
Fibonacci ke-4: 5
Fibonacci ke-5: 8
Fibonacci ke-6: 13
Fibonacci ke-7: 21
Fibonacci ke-8: 34
Fibonacci ke-9: 55
Fibonacci ke-10: 89
```

## Deskripsi Program

Sintaks tersebut adalah program untuk mencetak deret fibonasi berdasarkan formula  $S_n = S_{n-1} + S_{n-2}$  menggunakan fungsi rekursif, pertama kita membuat fungsi yang dinamai fibonasi, di dalam fungsi tersebut terdapat percabangan, percabangan pertama jika nilai  $n = 0$  maka program akan mengembalikan 0, percabangan kedua jika  $n = 1$  maka nilai akan mengenbalikan nilai 1, dam pada percabangan terakhir jika nilai selain 0 dan 1 maka program akan mengolah nilai  $n$  untuk mencari fibonasi dengan rekursif ( $\text{fibonasi}(n-1) + \text{fibonasi}(n-2)$ ). Fungsi ini memanggil dirinya sendiri secara rekursif untuk menghitung nilai fibonasi sebelumnya hingga mencapai kasus dasar yaitu  $n = 1$  dan  $n = 0$ .

### III. UNGUDED 2

#### Sourcecode

```
package main

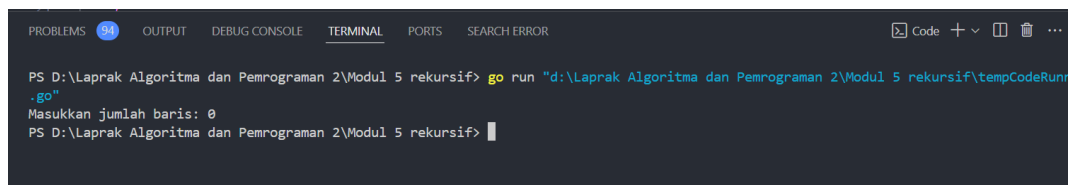
import (
    "fmt"
)

// Fungsi rekursif untuk mencetak sejumlah bintang dalam satu baris
func cetakBintang(n int) {
    if n > 0 {
        fmt.Print("*")
        cetakBintang(n - 1)
    }
}

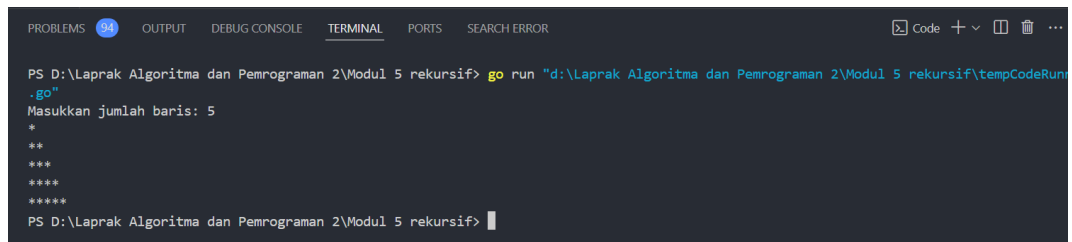
// Fungsi rekursif untuk mencetak pola bintang per baris
func cetakPola(rows int) {
    if rows > 0 {
        cetakPola(rows - 1)
        cetakBintang(rows)
        fmt.Println()
    }
}
```

```
func main() {  
  
    var N int  
  
    fmt.Print("Masukkan jumlah baris: ")  
  
    fmt.Scan(&N)  
  
    cetakPola(N)  
  
}
```

## Screenshoot Output



```
PS D:\Laprak Algoritma dan Pemrograman 2\Modul 5 rekursif> go run "d:\Laprak Algoritma dan Pemrograman 2\Modul 5 rekursif\tempCodeRun.go"  
Masukkan jumlah baris: 0  
PS D:\Laprak Algoritma dan Pemrograman 2\Modul 5 rekursif>
```



```
PS D:\Laprak Algoritma dan Pemrograman 2\Modul 5 rekursif> go run "d:\Laprak Algoritma dan Pemrograman 2\Modul 5 rekursif\tempCodeRun.go"  
Masukkan jumlah baris: 5  
*  
**  
***  
****  
*****  
PS D:\Laprak Algoritma dan Pemrograman 2\Modul 5 rekursif>
```

## Deskripsi Program

Sintaks tersebut adalah program untuk mencetak pola bintang dalam bentuk segitiga menggunakan fungsi rekursif. Pertama kita membuat fungsi cetakbintang yang bersifat rekursif dan mencetak satu bintang untuk setiap pemanggilan hingga jumlah yang diinginkan tercapai dalam satu baris. Setelah bintang pada satu baris selesai dicetak, cetakpola berpindah ke baris berikutnya hingga semua baris selesai dicetak, menghasilkan pola segitiga yang bertambah banyak bintangnya pada setiap baris.

#### IV. UNGUDED 3

##### Sourcecode

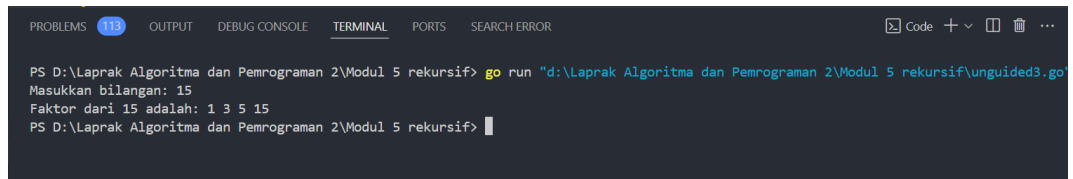
```
package main

import (
    "fmt"
)

// Fungsi rekursif untuk mencari faktor dari N
func cariFaktor(n, i int) {
    if i > n {
        return
    }
    if n%i == 0 {
        fmt.Print(i, " ")
    }
    cariFaktor(n, i+1)
}

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

## Screenshoot Output



```
PROBLEMS 113 OUTPUT DEBUG CONSOLE TERMINAL PORTS SEARCH ERROR
PS D:\Laprak Algoritma dan Pemrograman 2\Modul 5 rekursif> go run "d:\Laprak Algoritma dan Pemrograman 2\Modul 5 rekursif\unguided3.go"
Masukkan bilangan: 15
Faktor dari 15 adalah: 1 3 5 15
PS D:\Laprak Algoritma dan Pemrograman 2\Modul 5 rekursif> |
```

## Deskripsi Program

Sintaks tersebut adalah program untuk mencari faktor sebuah bilangan menggunakan fungsi rekursif. Pertama kita membuat fungsi carifaktor yang bersifat rekursif untuk mencari dan mencetak faktor-faktor dari bilangan n. Fungsi ini menerima dua parameter yaitu n, bilangan yang ingin dicari faktornya, dan i, yang berfungsi sebagai penghitung untuk memeriksa setiap angka dari 1 hingga n. Pada setiap pemanggilan, fungsi mengecek apakah i adalah faktor dari n dengan cara apakah hasil modulus n dan i = 0, jika iya nilai i akan menjadi faktor dari n.



## V. UNGUDED 4

### Sourcecode

```
package main

import (
    "fmt"
)

// Fungsi rekursif untuk mencetak bilangan dari N ke 1 dan
// kembali ke N
func cetakTurunNaik(n int) int {
    if n == 0 {
        return 0
    } else {
        fmt.Print(n, " ")
        cetakTurunNaik(n - 1)
        fmt.Print(n, " ")
    }
    return 0
}

func main() {
    var N int

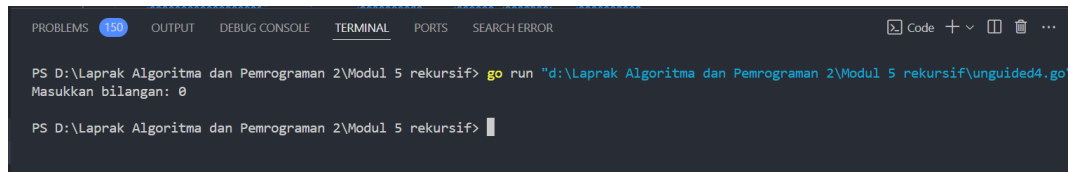
    fmt.Print("Masukkan bilangan: ")

    fmt.Scan(&N)

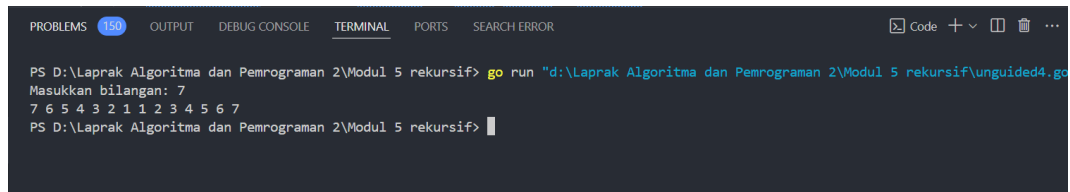
    cetakTurunNaik(N)

    fmt.Println()
}
```

## Screenshoot Output



```
PROBLEMS 150 OUTPUT DEBUG CONSOLE TERMINAL PORTS SEARCH ERROR
PS D:\Laprak Algoritma dan Pemrograman 2\Modul 5 rekursif> go run "d:\Laprak Algoritma dan Pemrograman 2\Modul 5 rekursif\unguided4.go"
Masukkan bilangan: 0
PS D:\Laprak Algoritma dan Pemrograman 2\Modul 5 rekursif>
```



```
PROBLEMS 150 OUTPUT DEBUG CONSOLE TERMINAL PORTS SEARCH ERROR
PS D:\Laprak Algoritma dan Pemrograman 2\Modul 5 rekursif> go run "d:\Laprak Algoritma dan Pemrograman 2\Modul 5 rekursif\unguided4.go"
Masukkan bilangan: 7
7 6 5 4 3 2 1 1 2 3 4 5 6 7
PS D:\Laprak Algoritma dan Pemrograman 2\Modul 5 rekursif>
```

## Deskripsi Program

Sintaks tersebut adalah program untuk mencetak barisan bilangan dari  $n$  ke 1 dan naik lagi ke  $n$  menggunakan fungsi rekursif. Pertama kita mencetak fungsi cetakturunnaik, fungsi ini memiliki base case saat  $n$  sama dengan 0, yang mengembalikan 0 dan menghentikan rekursi tanpa mencetak apa pun. Jika  $n$  tidak sama dengan 0 atau pada elsenya fungsi mencetak nilai  $n$ , kemudian memanggil dirinya sendiri dengan  $n - 1$  untuk mencetak nilai-nilai berikutnya secara berurutan turun, setelah mencetak nilai 1 fungsi mencetak kembali ke nilai awal  $n$ .

## VI. UNGUDED 5

### Sourcecode

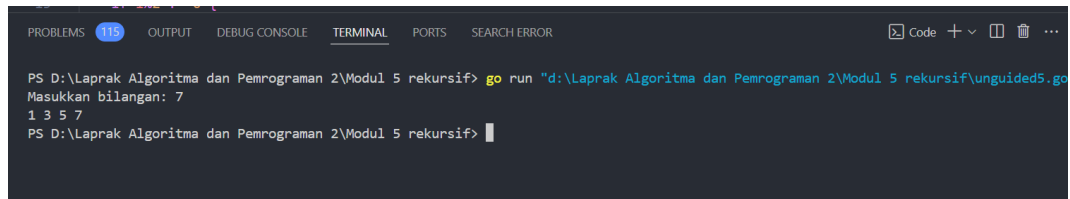
```
package main

import (
    "fmt"
)

//Fungsi rekursif untuk mencetak bilangan ganjil dari 1 hingga
n
func cetakGanjil(n, i int) {
    if i > n {
        return
    }
    if i%2 != 0 {
        fmt.Print(i, " ")
    }
    cetakGanjil(n, i+1)
}

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

## Screenshoot Output



```
PS D:\Laprak Algoritma dan Pemrograman 2\Modul 5 rekursif> go run "d:\Laprak Algoritma dan Pemrograman 2\Modul 5 rekursif\unguided5.go"
Masukkan bilangan: 7
1 3 5 7
PS D:\Laprak Algoritma dan Pemrograman 2\Modul 5 rekursif>
```

## Deskripsi Program

Sintaks tersebut adalah program untuk mencetak bilangan ganjil yang terdapat pada nilai  $n$  menggunakan fungsi rekursif. Pertama kita membuat fungsi cetakganjil yang akan mencetak semua bilangan ganjil dari 1 hingga  $n$ . Fungsi dimulai dengan memeriksa apakah  $i$  lebih besar dari  $n$ , jika ya maka rekursi dihentikan (base case), dan jika  $i$  adalah bilangan ganjil, yaitu pada operasi modulus  $n \% 2 = 1$ , fungsi akan mencetak  $i$  kemudian melanjutkan proses untuk bilangan berikutnya hingga mencapai batas  $n$  dengan rekursif cetakganjil( $i+1$ ).

## VII. UNGUDED 5

### Sourcecode

```
package main

import (
    "fmt"
)

// Fungsi rekursif untuk menghitung pangkat
func pangkat(x, y int) int {
    if y == 0 {
        return 1
    } else {
        return x * pangkat(x, y-1)
    }
}

func main() {
    var x, y int

    fmt.Print("Masukkan bilangan (x): ")

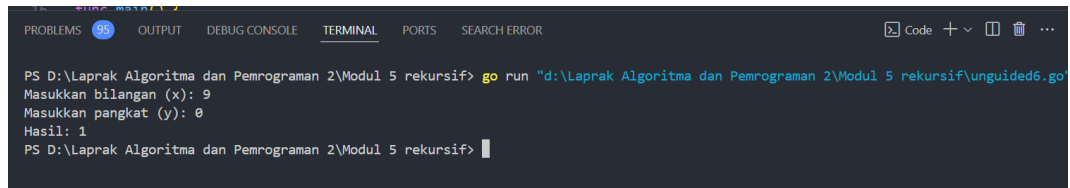
    fmt.Scan(&x)

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

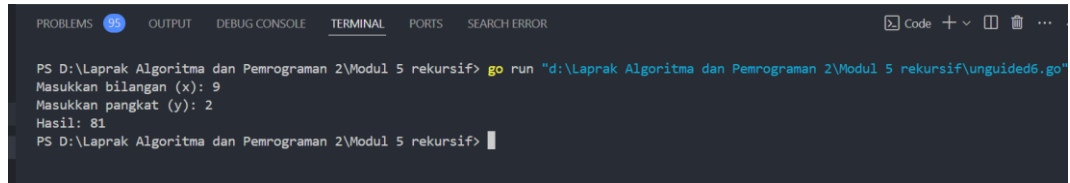
    fmt.Scan(&y)

    fmt.Println("Hasil:", pangkat(x, y))
}
```

## Screenshoot Output



```
PS D:\Laprak Algoritma dan Pemrograman 2\Modul 5 rekursif> go run "d:\Laprak Algoritma dan Pemrograman 2\Modul 5 rekursif\unguided6.go"
Masukkan bilangan (x): 9
Masukkan pangkat (y): 0
Hasil: 1
PS D:\Laprak Algoritma dan Pemrograman 2\Modul 5 rekursif>
```



```
PS D:\Laprak Algoritma dan Pemrograman 2\Modul 5 rekursif> go run "d:\Laprak Algoritma dan Pemrograman 2\Modul 5 rekursif\unguided6.go"
Masukkan bilangan (x): 9
Masukkan pangkat (y): 2
Hasil: 81
PS D:\Laprak Algoritma dan Pemrograman 2\Modul 5 rekursif>
```

## Deskripsi Program

Sintaks tersebut adalah program untuk menghitung hasil dari bilangan x yang dipangkatkan oleh bilangan y menggunakan fungsi rekursif. Pertama kita membuat fungsi pangkat yang menghitung hasil dari bilangan x yang dipangkatkan dengan y, base case dari fungsi tersebut adalah jika y bernilai 0, maka fungsi mengembalikan nilai 1 karena hasil pangkat dari nol adalah satu, kemudian Jika y lebih besar dari 0, fungsi akan mengalikan x dengan hasil pemanggilan fungsi itu sendiri (rekursif) yang berkurang satu sampai y mencapai 0, sehingga menghasilkan hasil perkalian berulang dari x sebanyak y kali.