

**LAPORAN PRAKTIKUM  
ALGORITMA PEMROGRAMAN 2**

**MODUL V  
REKURSIF**



Oleh:

Mansyuroh

NIM:

2311102234

**S1 TEKNIK INFORMATIKA  
TELKOM UNIVERSITY PURWOKERTO  
2024**

## I. DASAR TEORI

### A. Definisi Rekursif

Rekursi adalah sebuah teknik dalam pemrograman di mana sebuah fungsi memanggil dirinya sendiri untuk menyelesaikan tugas tertentu. 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. Sebagai contoh perhatikan prosedur cetak berikut ini!

	Notasi Algoritma	Notasi dalam bahasa GO
1	procedure cetak(in x:integer)	func cetak(x int){
2	algoritma	fmt.Println(x)
3	output(x)	cetak(x+1)
4	cetak(x+1)	}
5	endprocedure	

Catatan:

- Teknik rekursif ini merupakan salah satu alternatif untuk mengganti struktur kontrol perulangan dengan memanfaatkan subprogram (bisa fungsi ataupun prosedur).
- Untuk menghentikan proses rekursif digunakan percabangan (if-then).
- Base-case adalah kondisi proses rekursif berhenti. Base-case merupakan hal terpenting dan pertama yang harus diketahui ketika akan membuat program rekursif. Mustahil membuat program rekursif tanpa mengetahui base-case terlebih dahulu.
- Recursive-case adalah kondisi dimana proses pemanggilan dirinya sendiri dilakukan. Kondisi recursive-case adalah komplemen atau negasi dari base-case.
- Setiap algoritma rekursif selalu memiliki padanan dalam bentuk algoritma iteratif.

### B. Komponen Rekursif

Algoritma rekursif terdiri dari dua komponen utama:

- Base-case (Basis), yaitu bagian untuk menghentikan proses rekursif dan menjadi komponen terpenting di dalam sebuah rekursif.
- Recursive-case, yaitu bagian pemanggilan subprogramnya.

## II. GUIDED

### 1. Source code

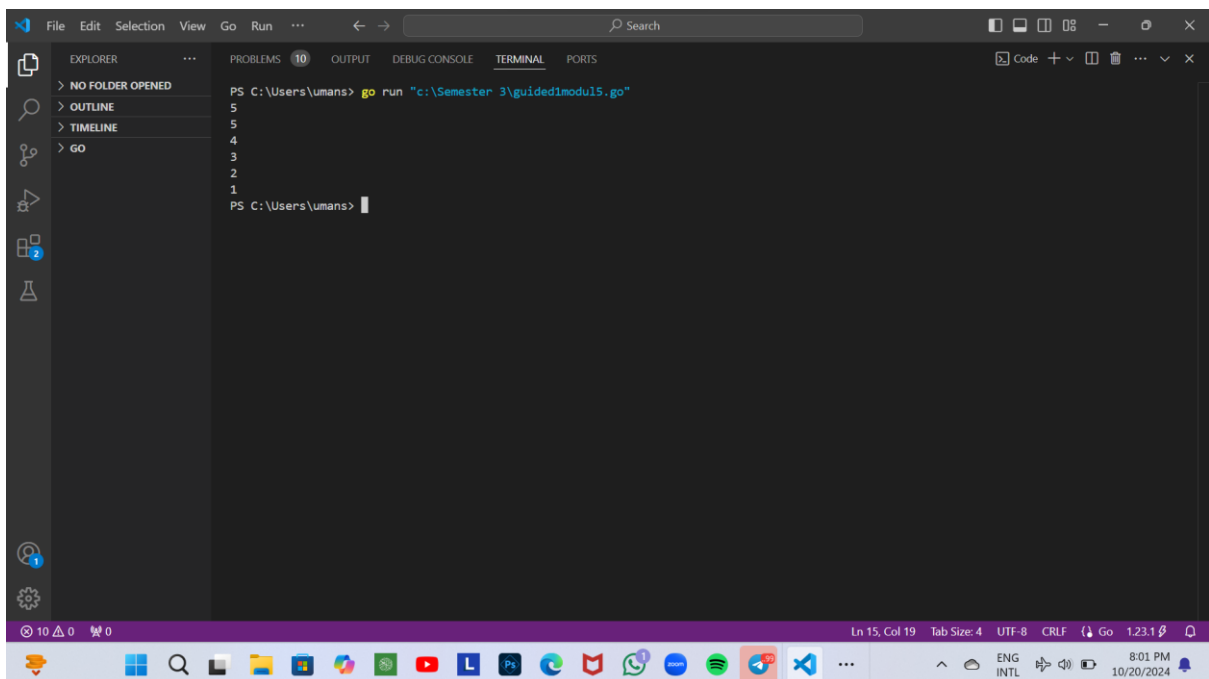
```
package main

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)
    }
}
```

### Screenshot program

A screenshot of the Visual Studio Code interface showing a Go program being executed. The Explorer pane on the left shows 'NO FOLDER OPENED'. The Output pane shows the execution of 'go run "c:\Semester 3\guidedmodul5.go"'. The Terminal pane shows the output of the program, which is a sequence of numbers from 1 to 5, printed on separate lines. The status bar at the bottom indicates the file is at line 15, column 19, with a tab size of 4, UTF-8 encoding, and CRLF line endings. The system tray at the bottom shows the date and time as 8:01 PM on 10/20/2024.

```
File Edit Selection View Go Run ... Search
EXPLORER PROBLEMS 10 OUTPUT DEBUG CONSOLE TERMINAL PORTS
> NO FOLDER OPENED
> OUTLINE
> TIMELINE
> GO
PS C:\Users\umans> go run "c:\Semester 3\guidedmodul5.go"
5
5
4
3
2
1
PS C:\Users\umans>
Ln 15, Col 19 Tab Size: 4 UTF-8 CRLF Go 1.23.1
```

### Deskripsi program

Program di atas membaca input bilangan bulat  $n$  dan mencetaknya secara menurun hingga 1 menggunakan fungsi rekursif `baris()`. Jika bilangan sama dengan 1, program mencetak 1 dan menghentikan rekursi; jika lebih, program mencetak bilangan dan memanggil `baris(bilangan - 1)`.

## 2. Source code

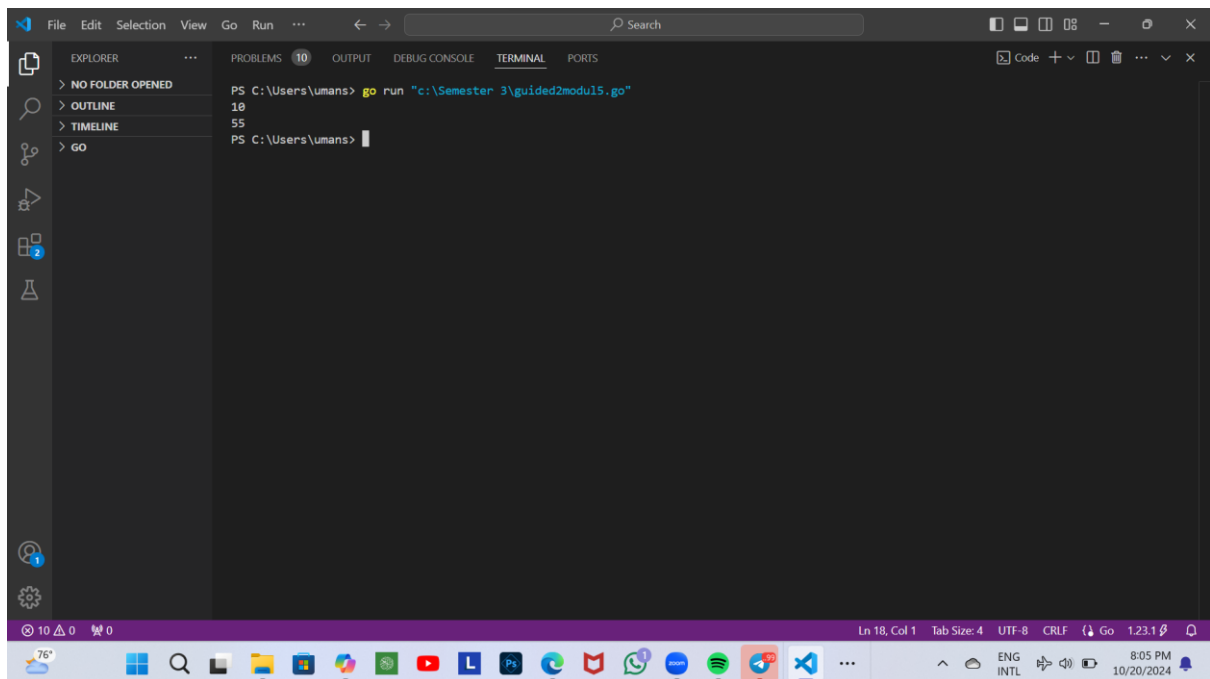
```
package main

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)
    }
}
```

## Screenshot program



## Deskripsi program

Program di atas membaca input bilangan bulat  $n$  dan menghitung jumlah bilangan dari 1 hingga  $n$  menggunakan fungsi rekursif penjumlahan(). Jika  $n$  sama dengan 1, fungsi mengembalikan 1; jika lebih, fungsi mengembalikan  $n$  ditambah hasil pemanggilan penjumlahan( $n-1$ ). Hasil akhirnya dicetak di layar.

## III. UNGUIDED

### 1. Source code

```
package main

import "fmt"

func hitungFibonacci234(angka234 int) int {
    if angka234 <= 1 {
        return angka234
    }
    return hitungFibonacci234(angka234-1) +
        hitungFibonacci234(angka234-2)
}

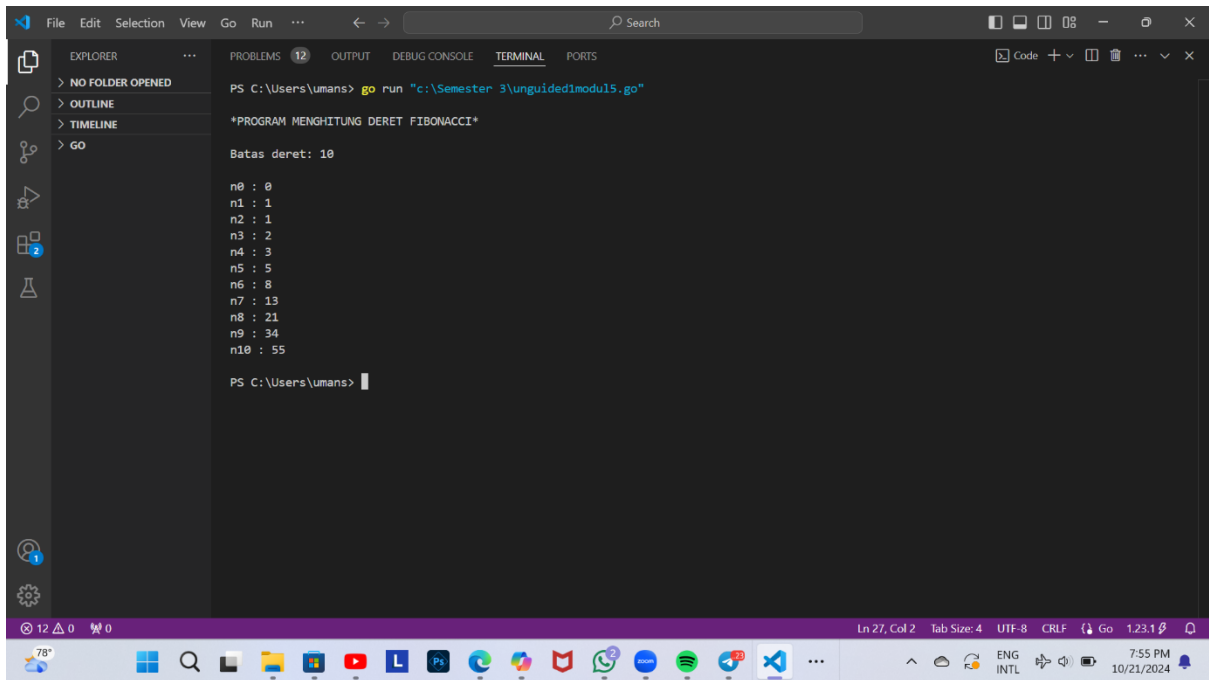
func main() {
    var batasDeret234 int
    indeks234 := 0

    fmt.Println("\n*PROGRAM MENGHITUNG DERET FIBONACCI*")
    fmt.Print("\nBatas deret: ")
    fmt.Scan(&batasDeret234)
    fmt.Println()

    for indeks := 0; indeks <= batasDeret234; indeks++ {
        fmt.Printf("n%d : %d\n", indeks,
            hitungFibonacci234(indeks234))
        indeks234++
    }

    fmt.Println()
}
```

## Screenshot program



```
PS C:\Users\umans> go run "c:\Semester 3\unguided1modul5.go"

*PROGRAM MENGHITUNG DERET FIBONACCI*

Batas deret: 10

n0 : 0
n1 : 1
n2 : 1
n3 : 2
n4 : 3
n5 : 5
n6 : 8
n7 : 13
n8 : 21
n9 : 34
n10 : 55

PS C:\Users\umans>
```

## Deskripsi program

Program di atas untuk menghitung dan menampilkan deret Fibonacci hingga batas yang ditentukan oleh pengguna. Berikut langkah-langkah kerjanya:

1. Pengguna diminta memasukkan batas deret Fibonacci yang ingin dihitung.
2. Program akan menghitung dan mencetak nilai Fibonacci untuk setiap indeks mulai dari 0 hingga batas yang dimasukkan.
3. Fungsi `hitungFibonacci234` digunakan untuk menghitung nilai Fibonacci secara rekursif.

## 2. Source code

```
package main

import "fmt"

func tampilkanBintang234(hitung234 int) {

    if hitung234 <= 0 {
```

```
        return

    }

    fmt.Print("*")

    tampilkanBintang234(hitung234 - 1)

}

func tampilkanBaris234(totalBaris234, urutan234 int) {

    if urutan234 > totalBaris234 {

        return

    }

    tampilkanBintang234(urutan234)

    fmt.Println()

    tampilkanBaris234(totalBaris234, urutan234+1)
}

func main() {

    var jumlahBaris234 int

    fmt.Print("\nMasukkan jumlah baris(n) : ")

    fmt.Scan(&jumlahBaris234)

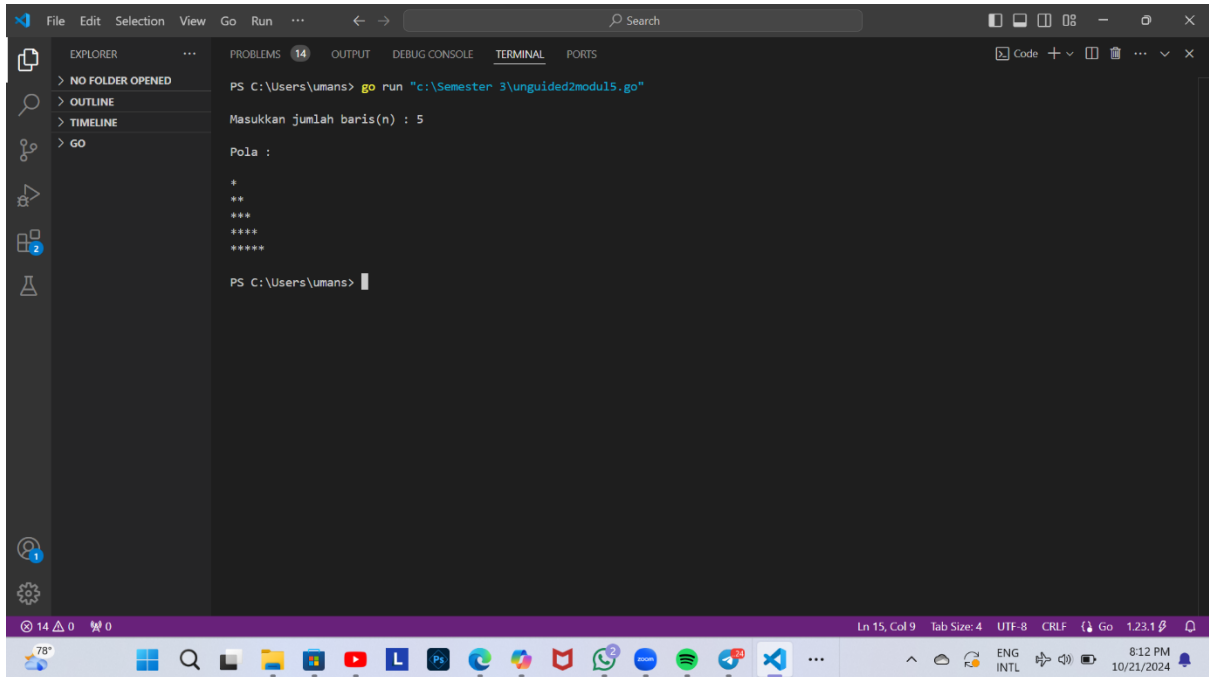
    fmt.Println("\nPola : \n")

    tampilkanBaris234(jumlahBaris234, 1)

    fmt.Println()

}
```

## Screenshot program



```
File Edit Selection View Go Run ... < -> Search
EXPLORER PROBLEMS 14 OUTPUT DEBUG CONSOLE TERMINAL PORTS
> NO FOLDER OPENED
> OUTLINE
> TIMELINE
> GO

PS C:\Users\umans> go run "c:\Semester 3\unguided2modul5.go"
Masukkan jumlah baris(n) : 5

Pola :

*
**
***
****
*****

PS C:\Users\umans>
```

## Deskripsi program

Program di atas mencetak pola segitiga bintang yang terdiri dari beberapa baris sesuai dengan input pengguna. Berikut langkah-langkah kerjanya:

1. Pengguna diminta memasukkan jumlah baris bintang yang diinginkan.
2. Program akan mencetak pola bintang, dimulai dari 1 bintang pada baris pertama, kemudian bertambah 1 bintang di setiap baris berikutnya.
3. Fungsi `tampilkanBintang234` mencetak bintang pada satu baris, sedangkan fungsi `tampilkanBaris234` memanggil fungsi tersebut dan mencetak baris secara berulang hingga mencapai jumlah baris yang diminta.



### 3. Source code

```
package main

import "fmt"

func faktorBilangan234(bilangan234, i234 int) {
    if i234 > bilangan234 {
        return
    }

    if bilangan234%i234 == 0 {
        fmt.Print(i234, " ")
    }

    faktorBilangan234(bilangan234, i234+1)
}

func main() {
    var inputBilangan234 int

    fmt.Print("\nMasukkan bilangan : ")

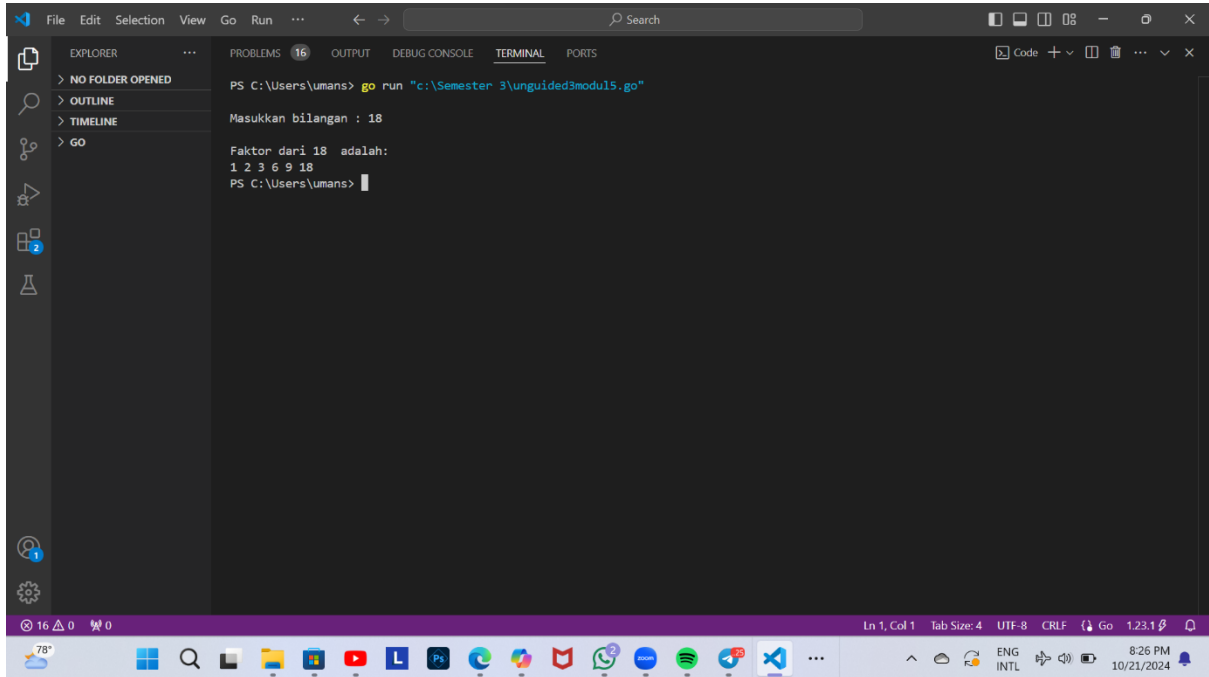
    fmt.Scan(&inputBilangan234)

    fmt.Println("\nFaktor dari", inputBilangan234, " adalah:")

    faktorBilangan234(inputBilangan234, 1)

    fmt.Println()
}
```

## Screenshot program



The screenshot shows the Visual Studio Code interface with a terminal window open. The terminal displays the following output:

```
PS C:\Users\umans> go run "c:\Semester 3\unguided3modul5.go"
Masukkan bilangan : 18

Faktor dari 18 adalah:
1 2 3 6 9 18
PS C:\Users\umans>
```

The terminal window is titled "TERMINAL" and shows the execution of a Go program. The user has entered the number 18, and the program has output the factors of 18: 1, 2, 3, 6, 9, and 18.

## Deskripsi Program

Program di atas untuk mencari dan menampilkan faktor-faktor dari sebuah bilangan yang dimasukkan oleh pengguna. Berikut langkah-langkah kerjanya:

1. Pengguna diminta memasukkan sebuah bilangan.
2. Program akan mencari semua bilangan yang bisa membagi habis bilangan tersebut (faktor) mulai dari 1 hingga bilangan itu sendiri.
3. Fungsi `faktorBilangan234` melakukan pengecekan faktor dan mencetaknya satu per satu.

#### 4. Source code

```
package main

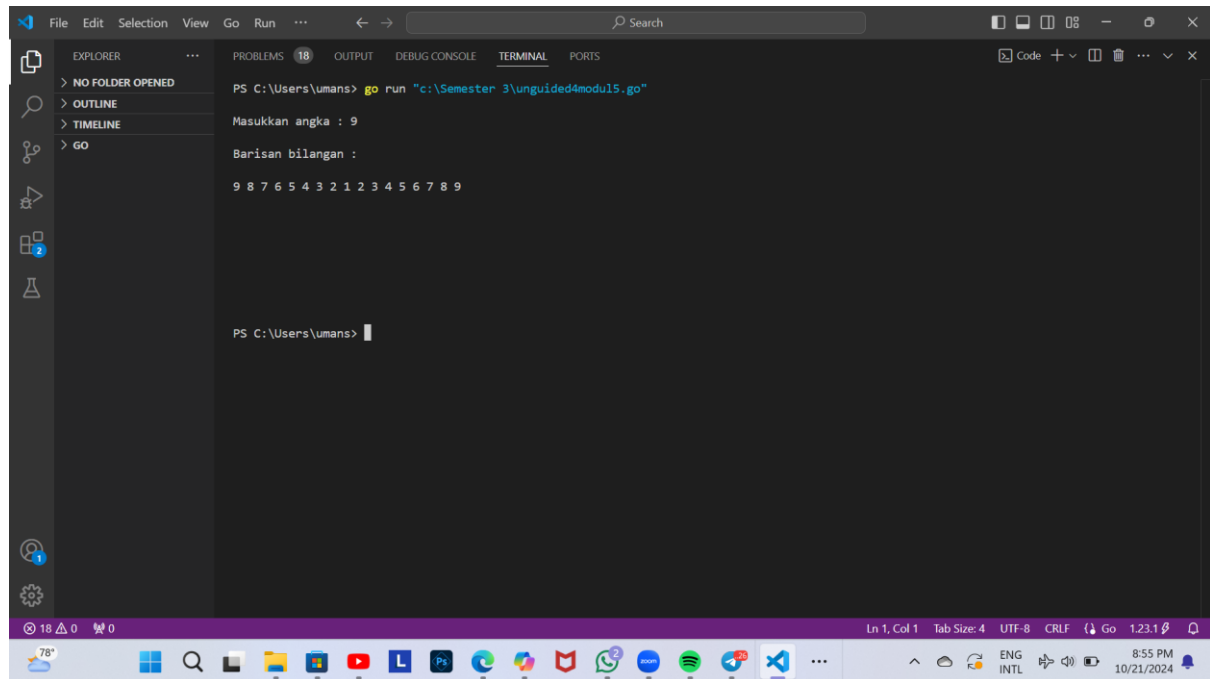
import "fmt"

func cetakTurun234(angka int) {
    if angka < 1 {
        return
    }
    fmt.Print(angka, " ")
    cetakTurun234(angka - 1)
}

func cetakNaik234(batas, sekarang int) {
    if sekarang > batas {
        return
    }
    fmt.Print(sekarang, " ")
    cetakNaik234(batas, sekarang+1)
    fmt.Println()
}

func main() {
    var inputAngka234 int
    fmt.Print("\nMasukkan angka : ")
    fmt.Scan(&inputAngka234)
    fmt.Println()
    fmt.Println("Barisan bilangan : \n")
    cetakTurun234(inputAngka234)
    cetakNaik234(inputAngka234, 2)
    fmt.Println()
}
```

## Screenshot program



## Deskripsi program

Program di atas meminta pengguna memasukkan sebuah angka, lalu mencetak dua pola angka. Pertama, mencetak angka turun dari angka yang dimasukkan sampai 1, kemudian mencetak angka naik mulai dari 2 hingga angka yang sama. Berikut langkah-langkah kerjanya:

1. Pengguna diminta memasukkan sebuah angka.
2. Program mencetak angka dari input ke 1 secara menurun (fungsi `cetakTurun234`).
3. Lalu, program mencetak angka dari 2 sampai input secara menaik (fungsi `cetakNaik234`).

## 5. Source code

```
package main

import "fmt"

func tampilkanGanjil234(angka, batas int) {
    if angka > batas {
```

```

        return
    }
    if angka%2 != 0 {
        fmt.Print(angka, " ")
    }
    tampilkanGanjil234(angka+1, batas)
}

func main() {
    var maxBilangan234 int
    fmt.Print("\nMasukkan jumlah bilangan: ")
    fmt.Scan(&maxBilangan234)
    fmt.Println("\nBilangan yang ganjil : ")
    tampilkanGanjil234(0, maxBilangan234)
    fmt.Println()
}

```

## Screenshot program

The screenshot shows a Windows terminal window with the following content:

```

PS C:\Users\umans> go run "c:\Semester 3\unguided5modul5.go"
Masukkan jumlah bilangan: 12
Bilangan yang ganjil :
1 3 5 7 9 11
PS C:\Users\umans>

```

The terminal window is titled "Terminal" and shows the execution of a Go program. The user has entered the command to run the program, and the program has successfully executed, displaying the first 12 odd numbers as requested.

## Deskripsi program

Program di atas untuk menampilkan semua bilangan ganjil dari 0 hingga batas yang dimasukkan oleh pengguna. Berikut langkah-langkah kerjanya:

1. Pengguna diminta memasukkan batas bilangan.
2. Program akan mencetak semua bilangan ganjil mulai dari 0 hingga batas yang diberikan.
3. Fungsi `tampilkanGanjil234` memeriksa setiap angka, dan jika angka tersebut ganjil, maka akan dicetak.

## 6. Source code

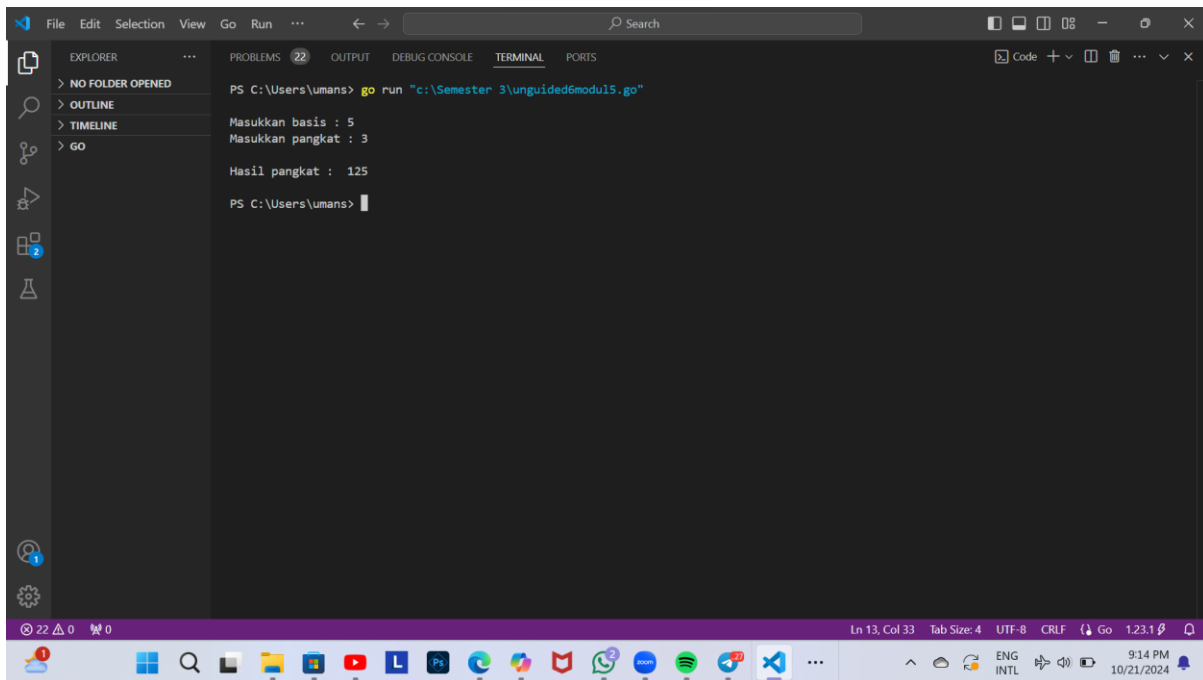
```
package main

import "fmt"

func hitungPangkat234(basis, eksponen int) int {
    if eksponen == 0 {
        return 1
    }
    return basis * hitungPangkat234(basis, eksponen-1)
}

func main() {
    var basis234, pangkat234 int
    fmt.Print("\nMasukkan basis : ")
    fmt.Scan(&basis234)
    fmt.Print("Masukkan pangkat : ")
    fmt.Scan(&pangkat234)
    hasil := hitungPangkat234(basis234, pangkat234)
    fmt.Println("\nHasil pangkat : ", hasil)
    fmt.Println()
}
```

## Screenshot program



The screenshot shows the Visual Studio Code interface with the Terminal panel active. The terminal displays the execution of a Go program. The user has run the command `go run "c:\Semester 3\unguided6modul5.go"` in the PowerShell prompt. The program prompts for a base value and a power value, which are entered as 5 and 3 respectively. The program then calculates and displays the result as 125.

```
PS C:\Users\umans> go run "c:\Semester 3\unguided6modul5.go"
Masukkan basis : 5
Masukkan pangkat : 3

Hasil pangkat : 125
PS C:\Users\umans>
```

## Deskripsi program

Program di atas untuk menghitung hasil perpangkatan dari sebuah bilangan (basis) dengan pangkat yang ditentukan oleh pengguna. Berikut langkah-langkah kerjanya:

1. Pengguna diminta memasukkan bilangan dasar (basis) dan nilai pangkat.
2. Program kemudian menghitung hasil perpangkatan menggunakan fungsi rekursif `'hitungPangkat234'`.
3. Hasil perhitungan pangkat tersebut akan ditampilkan di layar.

