

**LAPORAN PRAKTIKUM
ALGORITME DAN PEMROGRAMAN 2**

**MODUL 5
REKURSIF**



Oleh:

MUHAMMAD AMIR SALEH

2311102233

IF - 11 - 06

PROGRAM STUDI S1 TEKNIK INFORMATIKA

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2024

I. Dasar Teori

Rekursif dalam Go (Golang) adalah teknik pemrograman yang memungkinkan sebuah fungsi untuk memanggil dirinya sendiri guna menyelesaikan masalah yang dapat dipecah menjadi sub-masalah yang lebih kecil. Dalam penggunaan rekursif, ada dua komponen penting: kasus basis dan kasus rekursif. Kasus basis adalah kondisi yang menghentikan pemanggilan rekursif, mencegah terjadinya loop tak berujung, sedangkan kasus rekursif adalah bagian di mana fungsi memanggil dirinya sendiri dengan argumen yang lebih kecil. Misalnya, dalam fungsi untuk menghitung faktorial, jika n sama dengan 0, fungsi akan mengembalikan 1 (kasus basis). Untuk nilai n yang lebih besar, fungsi akan mengembalikan n dikalikan dengan hasil faktorial dari $n-1$. Rekursif sangat berguna dalam banyak algoritma, seperti pencarian dalam struktur data atau pengurutan, namun penggunaannya harus hati-hati agar tidak menyebabkan stack overflow, terutama pada masalah dengan banyak lapisan rekursi.

II. Guided Guided 1

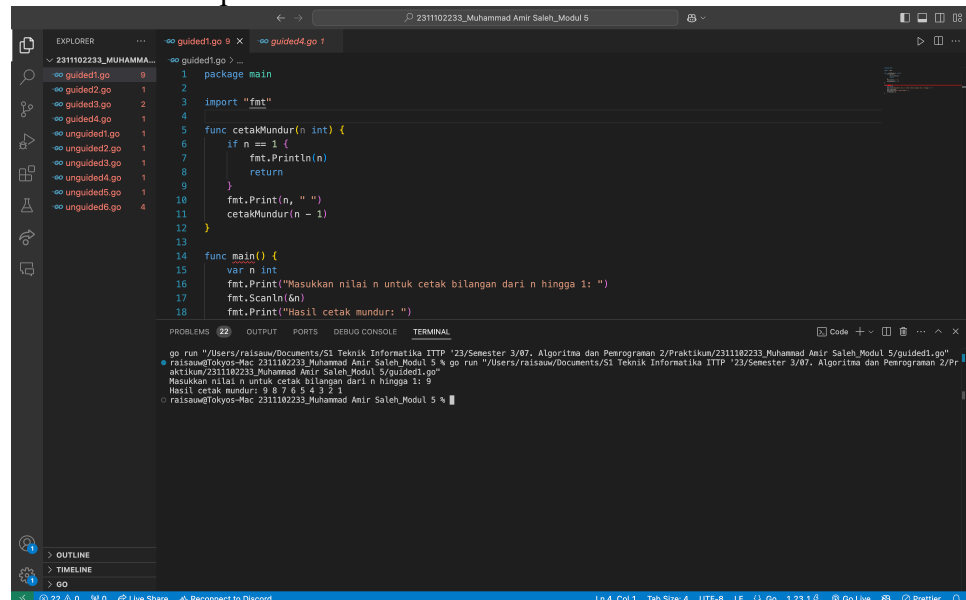
```
package main

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 dari n
hingga 1: ")
    fmt.Scanln(&n)
    fmt.Print("Hasil cetak mundur: ")
    cetakMundur(n)
}
```

Screenshots Output



Deskripsi:

Program ini meminta pengguna untuk memasukkan sebuah angka n dan kemudian mencetak bilangan dari n hingga 1 secara berurutan. Fungsi

rekursif cetakMundur mencetak angka saat ini (n), diikuti dengan memanggil dirinya sendiri dengan n dikurangi 1. Ketika n mencapai 1, fungsi mencetak angka 1 dan proses berhenti. Dengan cara ini, program menghasilkan deretan angka yang dicetak mundur dari n hingga 1.

```
package main

import "fmt"

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

func main() {
    var n int
    fmt.Print("Masukkan nilai n untuk penjumlahan 1 hingga n :")
    fmt.Scanln(&n)
    fmt.Println("Hasil penjumlahan:", jumlahRekursi(n))
}
```

Program ini meminta pengguna untuk memasukkan sebuah angka n dan menghitung penjumlahan dari 1 hingga n menggunakan fungsi rekursif `jumlahRekursi`. Jika n sama dengan 1, fungsi mengembalikan 1; jika tidak, fungsi menambahkan n dengan hasil dari pemanggilan dirinya sendiri dengan n dikurangi 1. Dengan cara ini, program akan menghasilkan jumlah total dari bilangan bulat dari 1 hingga n , yang kemudian dicetak di layar.

Guided 3

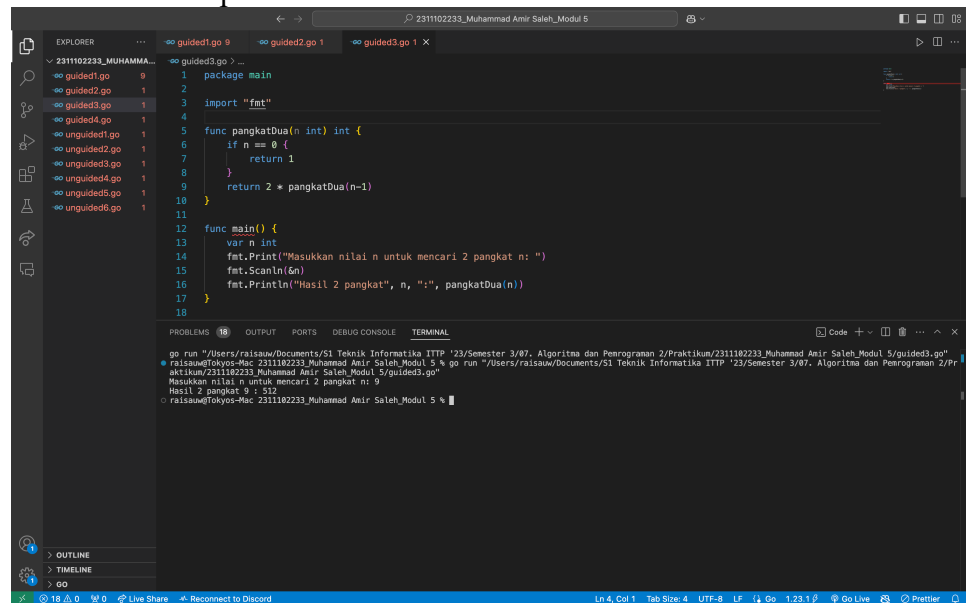
```
package main

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))
}
```

Screenshots Output



Deskripsi:

Program ini meminta pengguna untuk memasukkan sebuah angka n dan menghitung nilai 2 pangkat n (2^n) menggunakan fungsi rekursif `pangkatDua`. Jika n sama dengan 0 , fungsi mengembalikan 1 (karena $2^0 = 1$); jika tidak, fungsi mengalikan 2 dengan hasil dari pemanggilan dirinya sendiri dengan n dikurangi 1 . Hasil akhir dari perhitungan tersebut dicetak di layar, menampilkan nilai 2 pangkat n .

Guided 4

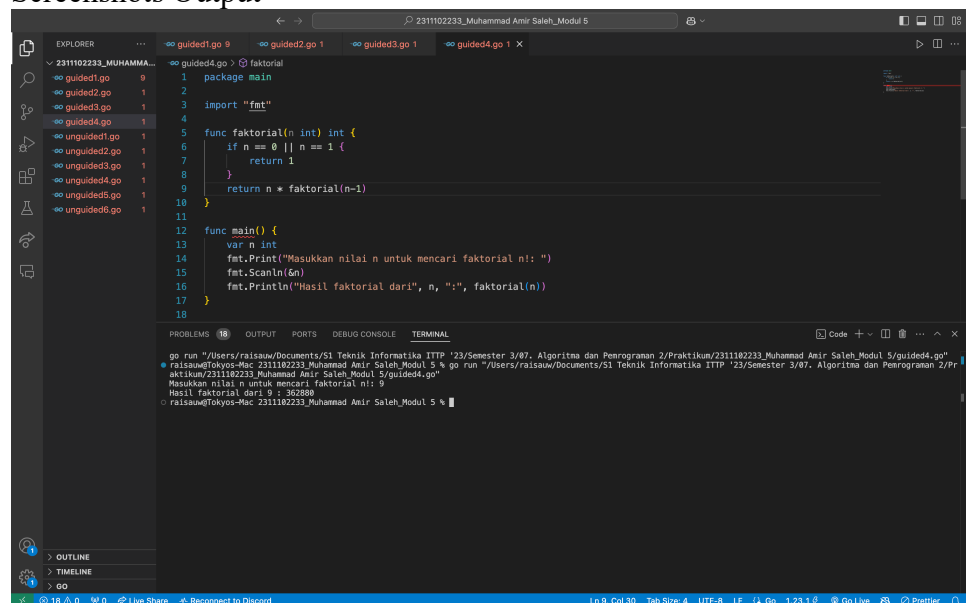
```
package main

import "fmt"

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

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

Screenshots Output



Deskripsi:

Program ini meminta pengguna untuk memasukkan sebuah angka n dan menghitung faktorial dari n ($n!$) menggunakan fungsi rekursif faktorial. Jika n adalah 0 atau 1, fungsi mengembalikan 1 (karena $0!$ dan $1!$ sama dengan 1); jika tidak, fungsi mengalikan n dengan hasil dari pemanggilan dirinya sendiri dengan n dikurangi 1. Hasil akhir dari faktorial tersebut kemudian dicetak di layar, menunjukkan nilai faktorial dari n.

III. Unguided Unguided 1

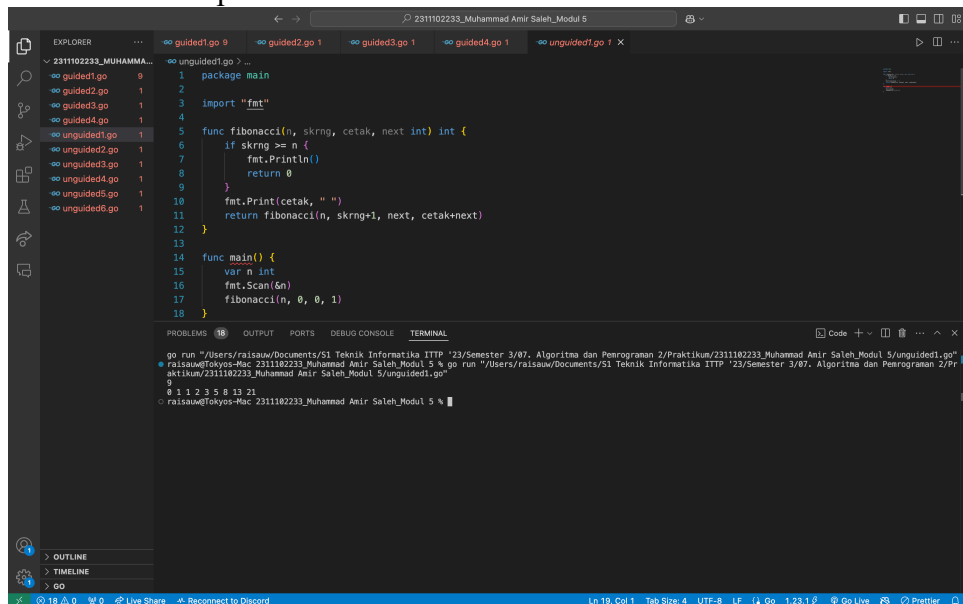
```
package main

import "fmt"

func fibonacci(n, skrng, cetak, next int) int {
    if skrng >= n {
        fmt.Println()
        return 0
    }
    fmt.Print(cetak, " ")
    return fibonacci(n, skrng+1, next, cetak+next)
}

func main() {
    var n int
    fmt.Scan(&n)
    fibonacci(n, 0, 0, 1)
}
```

Screenshots Output



Deskripsi:

Program ini membaca sebuah angka n dari input pengguna dan mencetak angka-angka dari n hingga 1 secara berurutan ke bawah. Fungsi baris menggunakan rekursi untuk mencetak angka saat ini, lalu memanggil dirinya sendiri dengan nilai bilangan yang dikurangi 1, hingga mencapai 1, yang kemudian dicetak dan menghentikan proses rekursi

Unguided 2

```
package main

import "fmt"

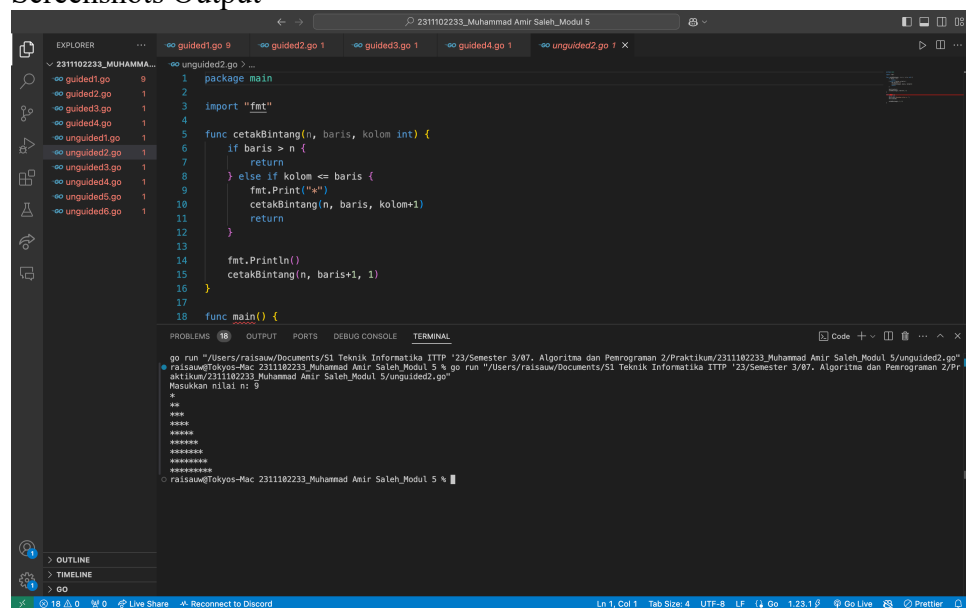
func cetakBintang(n, baris, kolom int) {
    if baris > n {
        return
    } else if kolom <= baris {
        fmt.Print("*")
        cetakBintang(n, baris, kolom+1)
        return
    }

    fmt.Println()
    cetakBintang(n, baris+1, 1)
}

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

    cetakBintang(n, 1, 1)
}
```

Screenshots Output



The screenshot shows a Go IDE with the following components:

- EXPLORER:** A list of files including `guided1.go` through `guided6.go` and `unguided1.go` through `unguided6.go`.
- EDITOR:** The code for `unguided2.go` is displayed, matching the code in the previous block.
- TERMINAL:** The output of running the program is shown. It prompts for a value of `n`, which is `9`. The output is a 9x9 grid of asterisks, with each row containing a decreasing number of asterisks from left to right, and each row on a new line.

```
go run "/Users/raisau/Documents/S1 Teknik Informatika ITPP '23/Semester 3/07. Algoritma dan Pemrograman 2/Praktikum/2311182233_Muhammad Amir Saleh_Modul 5/unguided2.go"
raissau@Tokyo-Mac: 2311182233_Muhammad Amir Saleh_Modul 5 % go run "/Users/raisau/Documents/S1 Teknik Informatika ITPP '23/Semester 3/07. Algoritma dan Pemrograman 2/Pr
aktikum/2311182233_Muhammad Amir Saleh_Modul 5/unguided2.go"
Masukkan nilai n: 9
*
**
***
****
*****
*****
*****
*****
*****
raissau@Tokyo-Mac: 2311182233_Muhammad Amir Saleh_Modul 5 %
```

Deskripsi:

Program ini membaca angka n dari input pengguna dan mencetak pola segitiga bintang sebanyak n baris. Fungsi rekursif `cetakBintang` memiliki tiga parameter: n (jumlah baris yang diinginkan), `baris` (baris saat ini), dan `kolom` (posisi kolom saat ini). Fungsi mencetak bintang sebanyak nomor baris, bertambah satu bintang setiap kali sampai baris ke- n . Jika kolom masih kurang dari atau sama dengan baris, bintang dicetak dan fungsi dipanggil lagi untuk mencetak bintang berikutnya. Ketika satu baris selesai, pindah ke baris berikutnya dan ulangi proses sampai semua baris tercetak.

Unguided 3

```
package main

import "fmt"

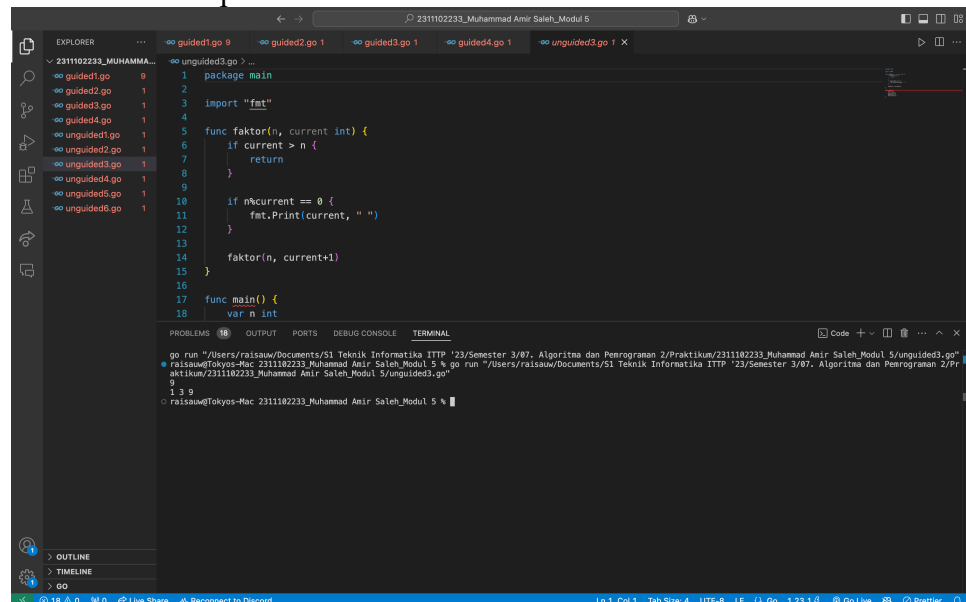
func faktor(n, current int) {
    if current > n {
        return
    }

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

    faktor(n, current+1)
}

func main() {
    var n int
    fmt.Scan(&n)
    faktor(n, 1)
    fmt.Println()
}
```

Screenshots Output



Deskripsi:

Program ini membaca angka n dari input pengguna dan mencetak semua faktor dari n. Fungsi rekursif faktor memeriksa setiap angka mulai

dari 1 hingga n (dengan parameter `current`). Jika n habis dibagi oleh `current`, angka tersebut dicetak sebagai faktor. Fungsi kemudian memanggil dirinya sendiri dengan `current` bertambah 1, dan proses berlanjut sampai semua angka telah diperiksa. Hasilnya adalah daftar faktor n yang dipisahkan oleh spasi.

Unguided 4

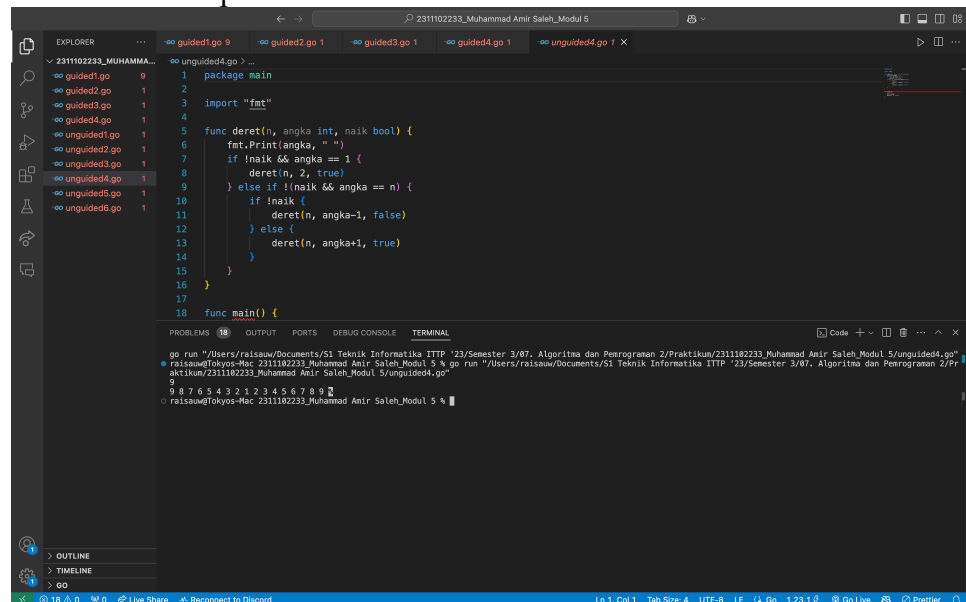
```
package main

import "fmt"

func deret(n, angka int, naik bool) {
    fmt.Print(angka, " ")
    if !naik && angka == 1 {
        deret(n, 2, true)
    } else if !(naik && angka == n) {
        if !naik {
            deret(n, angka-1, false)
        } else {
            deret(n, angka+1, true)
        }
    }
}

func main() {
    var n int
    fmt.Scan(&n)
    deret(n, n, false)
}
```

Screenshots Output



Deskripsi:

Program ini membaca angka n dari input pengguna dan mencetak deret angka yang dimulai dari n, menurun hingga 1, lalu naik kembali ke n.

Fungsi rekursif deret mencetak angka saat ini (angka), dan menggunakan parameter naik untuk menentukan apakah angka harus menurun atau naik. Jika naik bernilai false dan angka mencapai 1, deret berbalik naik, dan jika naik bernilai true dan angka mencapai n, proses berhenti.

Unguided 5

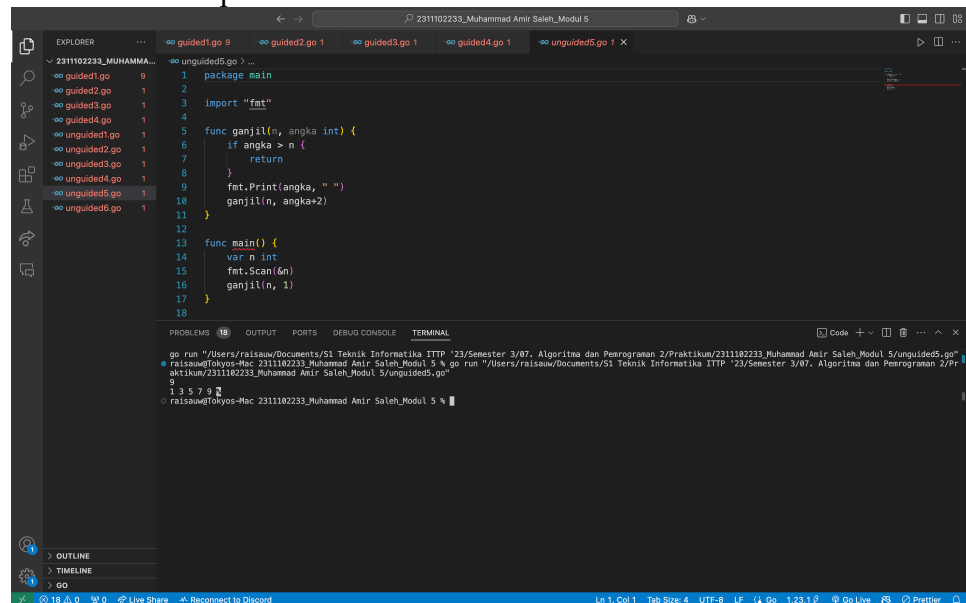
```
package main

import "fmt"

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

func main() {
    var n int
    fmt.Scan(&n)
    ganjil(n, 1)
}
```

Screenshots Output



Deskripsi:

Program ini membaca angka n dari input pengguna dan mencetak semua bilangan ganjil dari 1 hingga n. Fungsi rekursif ganjil mencetak angka saat ini (angka) dan memanggil dirinya sendiri dengan menambahkan 2 ke angka, sehingga hanya bilangan ganjil yang dicetak. Proses ini berlanjut sampai angka lebih besar dari n, dan program selesai mencetak deret bilangan ganjil.

Unguided 6

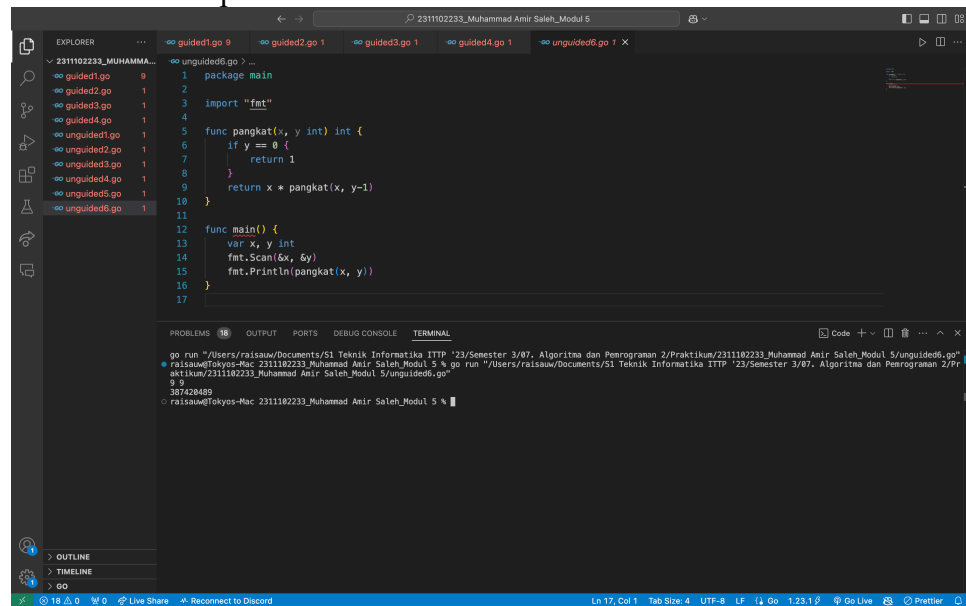
```
package main

import "fmt"

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

func main() {
    var x, y int
    fmt.Scan(&x, &y)
    fmt.Println(pangkat(x, y))
}
```

Screenshots Output



Deskripsi:

Program ini membaca dua angka, x dan y, dari input pengguna dan menghitung nilai x pangkat y (x^y) menggunakan fungsi rekursif pangkat. Jika y sama dengan 0, fungsi mengembalikan 1 (karena $x^0 = 1$); jika tidak, fungsi mengalikan x dengan hasil dari pemanggilan dirinya sendiri dengan y dikurangi 1. Hasil akhir dari perhitungan tersebut dicetak di layar.