

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

MODUL V

REKURSIF



Disusun Oleh :

Haposan Felix Marcel Siregar / 2311102210

IF_11_06

Dosen Pengampu :

Abednego Dwi Septiadi, S.Kom., M.Kom

PROGRAM STUDI S1 TEKNIK INFORMATIKA

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2024

I. DASAR TEORI

a. Definisi

Rekursi adalah teknik dalam pemrograman di mana sebuah fungsi memanggil dirinya sendiri untuk menyelesaikan suatu masalah. Teknik ini sering digunakan untuk menyelesaikan masalah yang dapat dipecah menjadi sub-masalah yang lebih kecil dan serupa. Rekursi memanfaatkan prinsip Divide and Conquer, di mana masalah besar dibagi menjadi masalah yang lebih kecil, diselesaikan secara individu, kemudian hasilnya digabungkan untuk mendapatkan solusi akhir.

b. Contoh Implementasi Rekursif

- **Cetak Bilangan Ganjil:** Fungsi untuk mencetak bilangan ganjil dari 1 hingga (n):

```
func cetakBilanganGanjil(n int, current int) {  
    if current > n {  
        return  
    }  
    if current%2 != 0 {  
        fmt.Print(current, " ")  
    }  
    cetakBilanganGanjil(n, current+1)  
}
```

- **Penjumlahan Bilangan:** Fungsi untuk menjumlahkan bilangan dari 1 hingga (n):

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

- **Pangkat Dua:** Fungsi untuk menghitung (2^n):

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

II. GUIDED

1. Soal Studi Case

menyusun sebuah program yang dapat mencetak bilangan secara mundur dari (n) hingga 1, di mana (n) adalah input yang diberikan oleh pengguna.

Sourcecode

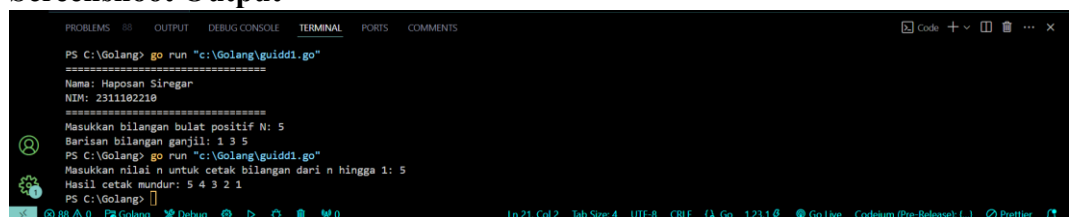
```
package main

import "fmt"

// Fungsi untuk mencetak bilangan dari n hingga 1
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)
}
```

Screenshot Output

A screenshot of a terminal window showing the execution of a Go program. The terminal has tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, PORTS, and COMMENTS. The command 'PS C:\Golang> go run "c:\Golang\guidd1.go"' is entered. The program prompts for a value 'n' and the user enters '5'. It then prints 'Masukkan bilangan bulat positif N: 5' and 'Barisan bilangan ganjil: 1 3 5'. The user enters '5' again, and the program prompts for 'n' and the user enters '5'. The program then prints 'Masukkan nilai n untuk cetak bilangan dari n hingga 1: 5' and 'Hasil cetak mundur: 5 4 3 2 1'. The terminal status bar at the bottom shows 'Ln 21, Col 2', 'Tab Size: 4', 'UTF-8', 'CRLF', 'Go 1.23.1', 'Go Live', 'Codeium (Pre-Release): [...]', and 'Prettier'.

Deskripsi Program

Program dimulai dengan mendeklarasikan paket utama dan mengimpor paket **fmt** untuk input-output. Di dalam fungsi **main()**, program meminta pengguna untuk memasukkan nilai bilangan bulat (*n*) yang akan digunakan sebagai batas atas untuk pencetakan. Setelah pengguna memasukkan nilai tersebut, program menampilkan pesan "Hasil cetak mundur:" dan memanggil fungsi **cetakMundur(*n*)** untuk mencetak bilangan dari (*n*) hingga 1. Fungsi **cetakMundur** bekerja secara rekursif: jika (*n*) sama dengan 1, fungsi akan mencetak 1 dan kemudian berhenti. Jika (*n*) lebih besar dari 1, fungsi akan mencetak nilai (*n*) diikuti dengan pemanggilan dirinya sendiri dengan argumen (*n*-1). Proses ini terus berlanjut sampai semua bilangan tercetak secara berurutan dari (*n*) hingga 1. Program pun berakhir setelah semua bilangan berhasil ditampilkan.

2. Soal Studi Case

menjumlahkan seluruh bilangan bulat dari 1 hingga (*n*), di mana (*n*) adalah nilai yang diinput oleh pengguna

Sourcecode

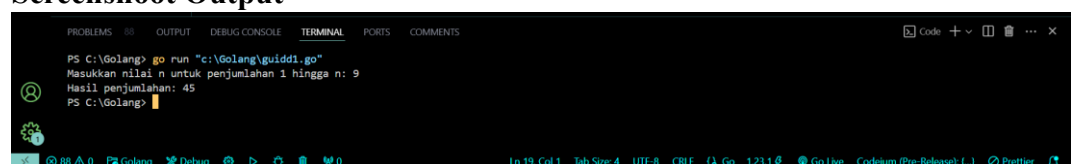
```
package main

import "fmt"

// Fungsi untuk menghitung penjumlahan 1 hingga n
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))
}
```

Screenshoot Output



The screenshot shows a Go IDE interface with a terminal window. The terminal output is as follows:

```
PS C:\Golang> go run "c:\Golang\guiddi.go"
Masukkan nilai n untuk penjumlahan 1 hingga n: 9
Hasil penjumlahan: 45
PS C:\Golang>
```

The IDE status bar at the bottom indicates the file is `Ln 19, Col 1`, the tab size is `4`, the encoding is `UTF-8`, and the line ending is `CRLF`. It also shows the Go version as `1.23.1` and the current package as `Codeium (Pre-Release)`.

Deskripsi Program

Program dimulai dengan mendeklarasikan paket utama dan mengimpor paket **fmt** untuk melakukan input dan output. Di dalam fungsi **main()**, program meminta pengguna untuk memasukkan nilai bilangan bulat (n) yang akan digunakan sebagai batas penjumlahan. Setelah pengguna memberikan nilai tersebut, program akan mencetak "Hasil penjumlahan:" diikuti dengan memanggil fungsi **jumlahRekursi(n)**, yang bertugas menghitung jumlah seluruh bilangan dari 1 hingga (n). Fungsi **jumlahRekursi** berfungsi secara rekursif: jika (n) sama dengan 1, maka fungsi akan mengembalikan 1. Jika tidak, fungsi akan mengembalikan (n) ditambah hasil dari pemanggilan fungsi yang sama dengan argumen ($n-1$). Proses ini akan berlanjut hingga mencapai base case di mana (n) bernilai 1. Setelah penjumlahan selesai, program menampilkan hasil penjumlahan kepada pengguna dan berakhir.

3. Soal Studi Case

untuk menghitung nilai 2^n (dua pangkat (n)), di mana (n) adalah nilai yang diberikan oleh pengguna

Sourcecode

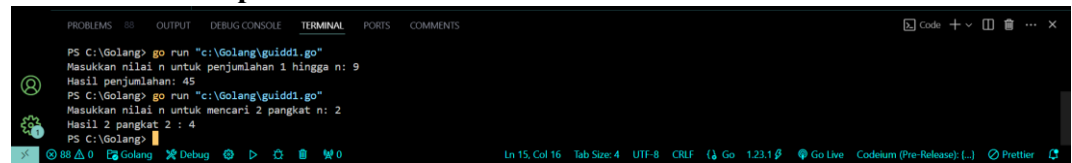
```
package main

import "fmt"

// Fungsi untuk menghitung penjumlahan 1 hingga n
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))
}
```

Screenshoot Output

A screenshot of a Go IDE terminal window. The terminal shows the execution of a Go program. The first run prompts for a value 'n' and calculates the sum of numbers from 1 to 'n'. The second run prompts for a value 'n' and calculates 2 raised to the power of 'n'. The output shows the results of these calculations.

```
PS C:\Golang> go run "c:\Golang\guiddi.go"
Masukkan nilai n untuk penjumlahan 1 hingga n: 9
Hasil penjumlahan: 45
PS C:\Golang> go run "c:\Golang\guiddi.go"
Masukkan nilai n untuk mencari 2 pangkat n: 2
Hasil 2 pangkat 2 : 4
PS C:\Golang>
```

Deskripsi Program

Program dimulai dengan mendeklarasikan paket utama dan mengimpor paket `fmt` untuk melakukan input dan output. Di dalam fungsi `main()`, program meminta pengguna untuk memasukkan nilai bilangan bulat (n), yang akan digunakan untuk menghitung (2^n). Setelah menerima input dari pengguna, program mencetak "Hasil 2 pangkat" diikuti dengan nilai (n) dan hasil dari pemanggilan fungsi `pangkatDua(n)`, yang bekerja untuk menghitung nilai tersebut. Fungsi `pangkatDua` berfungsi secara rekursif: jika (n) sama dengan 0, fungsi akan mengembalikan 1 (karena (2^0) sama dengan 1). Jika tidak, fungsi akan mengalikan 2 dengan hasil dari pemanggilan fungsi yang sama dengan argumen ($n-1$). Proses ini akan terus berlangsung hingga mencapai base case di mana (n) menjadi 0.

4. Soal Studi Case

Program ini dirancang untuk menghitung faktorial dari suatu bilangan bulat positif n . Faktorial dari n , dinyatakan sebagai $n!$, adalah hasil perkalian semua bilangan bulat dari 1 hingga n .

Sourcecode


```
package main

import "fmt"

// Fungsi untuk menghitung faktorial n!
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))
}
```

Screenshoot Output

A screenshot of a Go IDE terminal window. The terminal shows the command 'go run "c:\Golang\guiddi.go"' being executed. The output displays the author's name and NIM, followed by a prompt for a value 'n'. The user enters '6', and the program outputs 'Hasil faktorial dari 6 : 720'. The terminal interface includes tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, PORTS, and COMMENTS. The status bar at the bottom shows 'Ln 18, Col 2 (341 selected)', 'Tab Size 4', 'UTF-8', 'CR/LF', 'Go 1.23.1', 'Go Live', 'Codeium (Pre-Release) L...', and 'Prettier'.

Deskripsi Program

Program ini menghitung faktorial dari sebuah bilangan bulat positif n , menggunakan pendekatan rekursif. Saat program dimulai, ia mencetak identitas penulis dengan menggunakan fungsi identitas, yang menampilkan nama dan NIM penulis serta memisahkannya dengan garis pembatas. Setelah itu, program meminta pengguna untuk memasukkan nilai n dan membaca input tersebut. Untuk menghitung faktorial dari n , program memanggil fungsi rekursif faktorial. Fungsi ini memiliki basis kasus yang mengembalikan nilai 1 jika n sama dengan 0 atau 1, karena secara definisi, $0!$ dan $1!$ sama dengan 1. Untuk nilai n yang lebih besar dari 1, fungsi ini memanggil dirinya sendiri dengan parameter $n-1$ dan mengalikan hasilnya dengan n . Proses ini berlanjut hingga mencapai basis kasus.

III. UNGUIDED

1. Soal Studi Case

Mengimplementasikan fungsi rekursif pada deret Fibonacci

Sourcecode

```
package main

import (
    "fmt"
)

// Fungsi rekursif --> menghitung fibonacci
func fibonacci(n int) int {
    if n <= 1 {
        return n
    }
    return fibonacci(n-1) + fibonacci(n-2)
}

func identitas() {
    fmt.Println("=====")
    fmt.Println("Nama: Haposan Siregar")
    fmt.Println("NIM: 2311102210")
    fmt.Println("=====")
}

func main() {

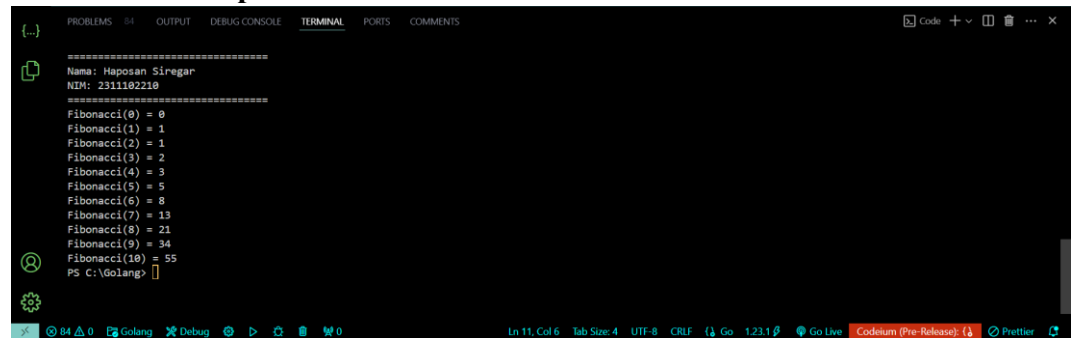
    identitas() // Menampilkan identitas

    for i := 0; i <= 10; i++ {

        fmt.Printf("Fibonacci(%d) = %d\n", i, fibonacci(i))
    }

}
```

Screenshoot Output



```
=====
Nama: Haposan Siregar
NIM: 2311102210
=====
Fibonacci(0) = 0
Fibonacci(1) = 1
Fibonacci(2) = 1
Fibonacci(3) = 2
Fibonacci(4) = 3
Fibonacci(5) = 5
Fibonacci(6) = 8
Fibonacci(7) = 13
Fibonacci(8) = 21
Fibonacci(9) = 34
Fibonacci(10) = 55
PS C:\Golang>
```

Deskripsi Program

a) Menghitung Angka Fibonacci:

- Untuk setiap angka dari 0 hingga 10, program akan menghitung angka Fibonacci dengan memanggil fungsi fibonacci(n).
- Fungsi fibonacci(n) berjalan secara rekursif untuk menghitung angka Fibonacci berdasarkan definisi deret.

2. Soal Studi Case

Mengimplementasikan fungsi rekursif untuk pencetakan pola bintang sesuai dengan inputan pengguna.

Source code

```
package main

import (
    "fmt"
)

// Fungsi rekursif untuk mencetak bintang
func CetakBintang(n int) {
    if n == 0 {
        return
    }
    CetakBintang(n - 1)
    fmt.Print("*")
}

// Fungsi rekursif untuk mencetak pola
```

```

func cetakPola(n, current int) {
    if current > n {
        return
    }
    CetakBintang(current)
    fmt.Println()
    cetakPola(n, current+1)
}

func identitas() {
    fmt.Println("=====")
    fmt.Println("Nama: Haposan Siregar")
    fmt.Println("NIM: 2311102210")
    fmt.Println("=====")
}

func main() {
    identitas() // Menampilkan identitas
    var n int
    fmt.Print("Masukkan angka: ")
    fmt.Scan(&n)
    cetakPola(n, 1)
}

```

Screenshoot Output

```

PS C:\Golang> go run "c:\Golang\tempCodeRunnerFile.go"
=====
Nama: Haposan Siregar
NIM: 2311102210
=====
Masukkan angka: 5
*
**
***
****
*****
PS C:\Golang>

```

Deskripsi Program

- Input Angka: Program meminta pengguna untuk memasukkan angka (n), yang menentukan jumlah baris bintang yang akan dicetak.
- Cetak Pola Bintang:
 - Fungsi cetakPola(n, 1) dipanggil untuk mencetak pola bintang.

- Fungsi ini menggunakan rekursi untuk mencetak dari 1 bintang hingga n bintang.
- c) Fungsi Cetak Bintang:
- Fungsi CetakBintang(n) mencetak n bintang dengan cara rekursif.

3. Soal Studi Case

Mengimplementasikan rekursif untuk mencari faktor dari sebuah N (bilangan)

Sourcecode

```
package main

import (
    "fmt"
)

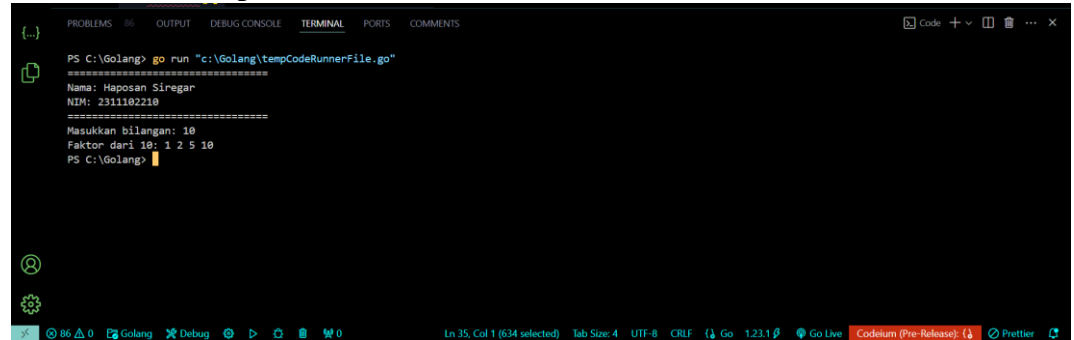
// Fungsi rekursif untuk mencetak faktor
func printFactors(n, divisor int) {
    if divisor > n {
        return
    }
    if n%divisor == 0 {
        fmt.Printf("%d ", divisor)
    }
    printFactors(n, divisor+1)
}

func identitas() {
    fmt.Println("=====")
    fmt.Println("Nama: Haposan Siregar")
    fmt.Println("NIM: 2311102210")
    fmt.Println("=====")
}

func main() {
    identitas() // Menampilkan identitas
    var n int
    fmt.Print("Masukkan bilangan: ")
    fmt.Scan(&n)
    fmt.Printf("Faktor dari %d: ", n)
    printFactors(n, 1)
    fmt.Println()
}
```

```
}
```

Screenshoot Output

A screenshot of a Go IDE terminal window. The terminal shows the command 'go run "c:\Golang\tempCodeRunnerFile.go"' being executed. The output displays the program's execution: it prints a separator line, then the name 'Haposan Siregar' and NIM '2311102210', followed by another separator line. It then prompts 'Masukkan bilangan: 10' and prints 'Faktor dari 10: 1 2 5 10'. The terminal interface includes tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, PORTS, and COMMENTS. The status bar at the bottom shows 'Ln 35, Col 1 (634 selected)', 'Tab Size: 4', 'UTF-8', 'CRLF', 'Go 1.23.1', 'Go Live', 'Codeium (Pre-Release)', and 'Prettier'.

Deskripsi Program

- a) fungsi utama printFactors
 - printFactors adalah fungsi rekursif yang mencetak faktor dari angka n dengan membaginya dengan divisor.
- b) Eksekusi Program Utama:
 - Fungsi main pertama memanggil identitas untuk menampilkan informasi identitas.
 - Kemudian meminta pengguna memasukkan angka n dan membaca input dari pengguna.
 - Menggunakan printFactors, program mencetak semua faktor dari n

4. Soal Studi Case

Mengimplementasikan rekursif untuk menampilkan barisan bilangan tertentu.

Sourcecode

```
package main

import "fmt"

func identitas() {
    fmt.Println("=====")
    fmt.Println("Nama: Haposan Siregar")
    fmt.Println("NIM: 2311102210")
    fmt.Println("=====")
}

func main() {
```

```

    identitas() // Menampilkan identitas
    fmt.Println("Buatlah program yang mengimplementasikan rekursif
    untuk menampilkan barisan bilangan tertentu.")

    fmt.Println("Masukan terdiri dari sebuah bilangan bulat positif N.")
    var N int
    fmt.Print("Masukkan nilai N: ")
    fmt.Scanln(&N)

    fmt.Println("Keluaran terdiri dari barisan bilangan dari N hingga 1
    dan kembali ke N.")
    printSequence(N)
}

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

    fmt.Print(n, " ")
    if n > 1 {
        printSequence(n - 1)
    }
    fmt.Print(n, " ")
}

```

Screenshoot Output

```

Nama: Haposan Siregar
NIM: 2311102210
=====
Masukan terdiri dari sebuah bilangan bulat positif N.
Masukkan nilai N: go run "c:\Golang\unguided64.go"
Keluaran terdiri dari barisan bilangan dari N hingga 1 dan kembali ke N.
PS C:\Golang> go run "c:\Golang\unguided64.go"
=====
Nama: Haposan Siregar
NIM: 2311102210
=====
Masukkan nilai N: 10
Keluaran terdiri dari barisan bilangan dari N hingga 1 dan kembali ke N.
10 9 8 7 6 5 4 3 2 1 1 2 3 4 5 6 7 8 9 10
PS C:\Golang>

```

Deskripsi Program

a) Program Utama:

- Cetak instruksi program dan apa yang perlu dimasukkan oleh pengguna.
- Minta pengguna memasukkan angka N.

b) Rekursif untuk Cetak Bilangan:

- Cetak nilai N.

- Panggil fungsi ini lagi dengan N yang lebih kecil sampai 1.
- Cetak nilai N lagi saat kembali.

5. Soal Studi Case

Mengimplementasikan rekursif untuk menampilkan barisan bilangan ganjil.

Sourcecode

```
package main

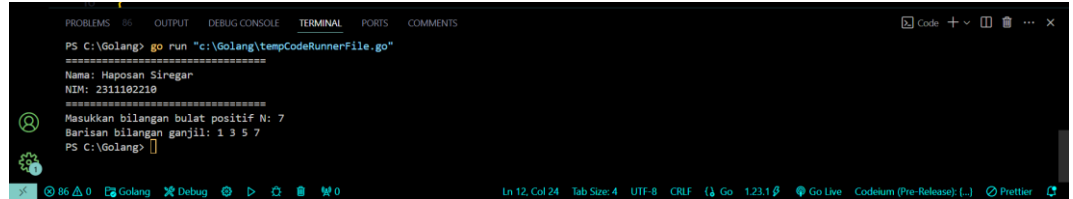
import (
    "fmt"
)

// Fungsi untuk mencetak barisan bilangan ganjil
func cetakBilanganGanjil(n int, current int) {
    if current > n {
        return
    }
    if current%2 != 0 {
        fmt.Print(current, " ")
    }
    cetakBilanganGanjil(n, current+1)
}

func identitas() {
    fmt.Println("=====")
    fmt.Println("Nama: Haposan Siregar")
    fmt.Println("NIM: 2311102210")
    fmt.Println("=====")
}

func main() {
    identitas() // Menampilkan identitas
    var n int
    fmt.Print("Masukkan bilangan bulat positif N: ")
    fmt.Scan(&n)
    fmt.Print("Barisan bilangan ganjil: ")
    cetakBilanganGanjil(n, 1)
    fmt.Println()
}
```

Screenshot Output

A screenshot of a Go IDE terminal window. The terminal shows the command 'go run "c:\Golang\tempCodeRunnerFile.go"' being executed. The output displays the program's header information: 'Name: Haposan Siregar' and 'NIM: 2311102210'. It then prompts the user to 'Masukkan bilangan bulat positif N: 7'. The program outputs the sequence of odd numbers '1 3 5 7' and returns to the prompt 'PS C:\Golang>'. The IDE interface includes tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, PORTS, and COMMENTS. The status bar at the bottom shows 'Ln 12, Col 24', 'Tab Size: 4', 'UTF-8', 'CRLF', 'Go 1.23.1', 'Go Live', 'Codeium (Pre-Release): [...]', and 'Prettier'.

Deskripsi Program

Program memanggil fungsi cetakBilanganGanjil(n , 1) untuk mencetak bilangan ganjil dari 1 hingga n menggunakan rekursi. Di setiap langkah, fungsi memeriksa apakah current adalah bilangan ganjil. Jika ya, bilangan tersebut dicetak.

Rekursi Berhenti:

- Fungsi akan berhenti ketika current lebih besar dari n , menandakan bahwa semua bilangan ganjil telah dicetak.

6. Soal Studi Case

Mengimplementasikan rekursif untuk mencari hasil pangkat dari dua buah bilangan.

Sourcecode

```
package main

import (
    "fmt"
)

// Recursive function to calculate  $x^y$ 
func pangkat(x int, y int) int {
    if y == 0 {
        return 1
    }
    return x * pangkat(x, y-1)
}

func identitas() {
    fmt.Println("=====")
    fmt.Println("Nama: Haposan Siregar")
    fmt.Println("NIM: 2311102210")
    fmt.Println("=====")
}
```



```

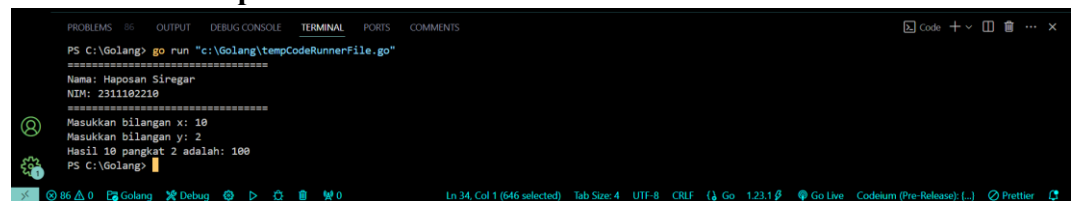
}

func main() {
    identitas() // Menampilkan identitas
    var x, y int
    fmt.Print("Masukkan bilangan x: ")
    fmt.Scan(&x)
    fmt.Print("Masukkan bilangan y: ")
    fmt.Scan(&y)

    result := pangkat(x, y)
    fmt.Printf("Hasil %d pangkat %d adalah: %d\n", x, y, result)
}

```

Screenshoot Output



```

PS C:\Golang> go run "c:\Golang\tempCodeRunnerFile.go"
=====
Name: Haposan Siregar
NIM: 2311102210
=====
Masukkan bilangan x: 10
Masukkan bilangan y: 2
Hasil 10 pangkat 2 adalah: 100
PS C:\Golang>

```

Deskripsi Program

Program ini menghitung nilai (x) pangkat (y) dengan menggunakan fungsi rekursif yang disebut pangkat. Saat fungsi dipanggil, ia akan memeriksa apakah nilai (y) sama dengan 0; jika ya, fungsi akan mengembalikan 1 karena nilai apa pun pangkat 0 adalah 1. Jika tidak, fungsi akan melakukan perhitungan dengan mengalikan (x) dengan hasil dari pemanggilan fungsi pangkat dengan parameter (x) dan (y-1). Proses ini akan berlanjut hingga (y) mencapai 0, pada titik di mana semua perkalian dilakukan untuk menghasilkan nilai akhir. Setelah perhitungan pangkat selesai, program akan menampilkan hasilnya di layar, menunjukkan nilai (x), (y), dan hasil akhirnya dalam format yang jelas dan mudah dibaca.