

LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2

MODUL 5
REKURSIF



Disusun Oleh :

Muhammad Djoko Susilo / 2311102212

Dosen Pengampu :

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

PROGRAM STUDI S1 TEKNIK INFORMATIKA
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2024

I. DASAR TEORI

Pengertian Rekursif

Rekursif adalah sebuah fungsi yang dapat memanggil dirinya sendiri, juga dapat diartikan sebagai cara untuk menyelesaikan suatu masalah dengan cara menyelesaikan suatu masalah dengan cara menyelesaikan sub-masalah yang identik dari masalah utama. Biasanya didalam fungsi rekursif terdapat sebuah struktur percabangan if-else untuk menghentikan proses ini, Kondisi ini disebut dengan **Base-Case**. Ketika sedang memanggil dirinya sendiri didalam fungsi tersebut, itu dinamakan **Recursive-Case**.

Didalam fungsi rekursif ada namanya **Forward** dan **Backward**:

- Forward adalah saat subprogram dipanggil secara rekursif, maka subprogram akan terus melakukan pemanggilan sampai saat kondisi **Base-Case** terpenuhi atau **true**.
- Backward adalah ketika proses kembali ke subprogram yang sebelumnya.

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

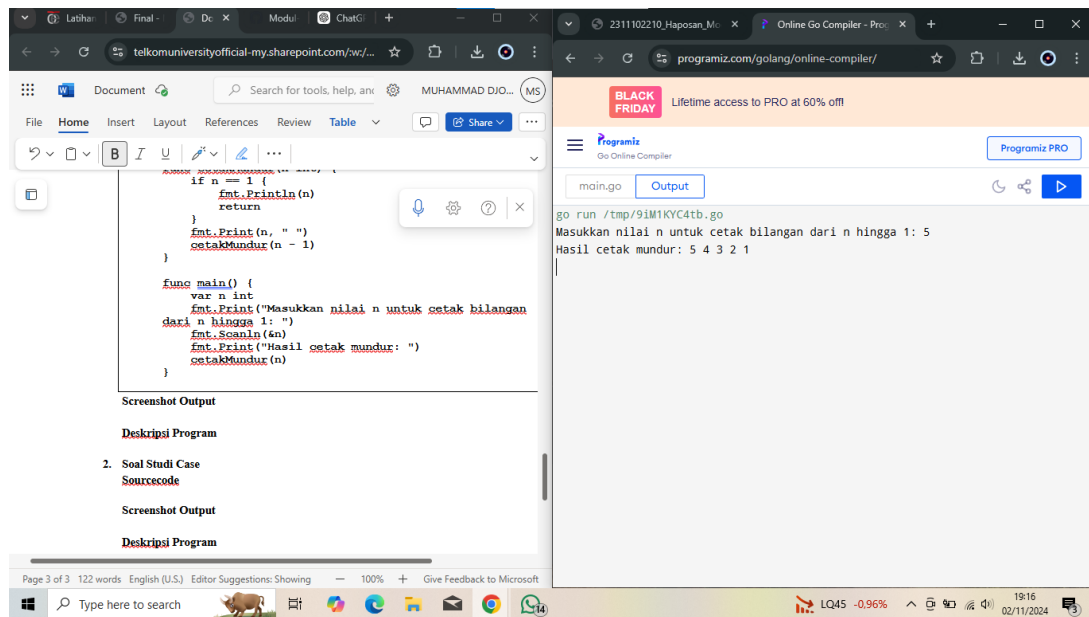
```
package main

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("Muhammad Djoko Susilo/2311102212 \n")
    fmt.Print("Masukkan nilai n untuk cetak bilangan
dari n hingga 1: ")
    fmt.Scanln(&n)
    fmt.Print("Hasil cetak mundur: ")
    cetakMundur(n)
}
```

Screenshot Output



Deskripsi Program

Program ini bertujuan untuk mencetak angka secara menurun dari nilai yang diinputkan oleh user menggunakan metode rekursif. Cara kerja dari program ini adalah program akan meminta nilai untuk dicetak menurun, setelah user memasukkan nilai maka program akan melakukan proses rekursi, dengan metode seperti ini:

jika nilai input lebih besar dari satu maka fungsi akan mencetak nilai input dan diikuti dengan pemanggilan fungsi rekursif dengan nilai yang diinputkan di kurangi satu hingga mencapai batas penghentian rekursi.

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"

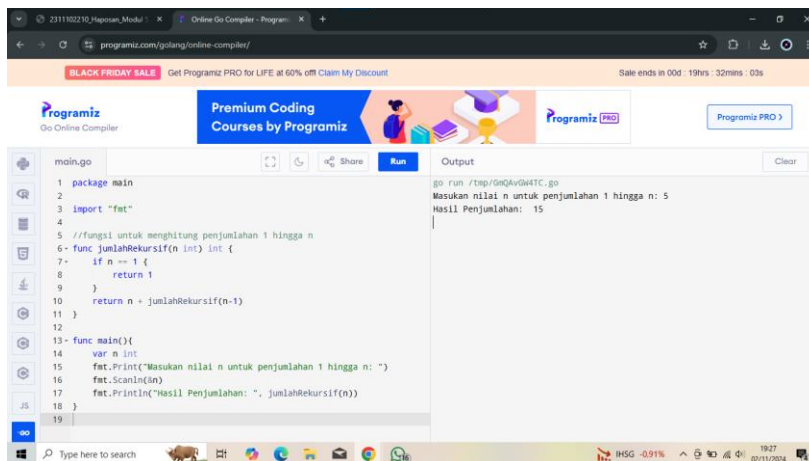
//fungsi untuk menghitung penjumlahan 1 hingga n
func jumlahRekursif(n int) int {
    if n == 1 {
        return 1
    }
    return n + jumlahRekursif(n-1)
}
```

```

func main(){
    var n int
    fmt.Print("Muhammad Djoko Susilo/2311102212 \n")
    fmt.Print("Masukan nilai n untuk penjumlahan 1
hingga n: ")
    fmt.Scanln(&n)
    fmt.Println("Hasil Penjumlahan: ",
jumlahRekursif(n))
}

```

Screenshot Output



Deskripsi Program

Program diatas adalah sebuah program yang dapat menghitung penjumlahan dari 1 hingga n menggunakan fungsi rekursif. Cara kerja dari program ini adalah pertama program meminta untuk user memasukkan nilai yang nanti akan masuk kedalam variabel n,, lalu n akan dicek apakah n sama dengan satu karena jika iya maka program akan mengembalikan nilai 1 dan akan langsung menghentikan program. Kedua jika kondisi

yang tidak terpenuhi, maka fungsi rekursif akan bekerja dengan memanggil kembali fungsi dengan memanggil fungsi dengan $n-1$ dan akan ditambahkan ke hasil hingga n mencapai 1.

3. Soal Studi Case

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

Sourcecode

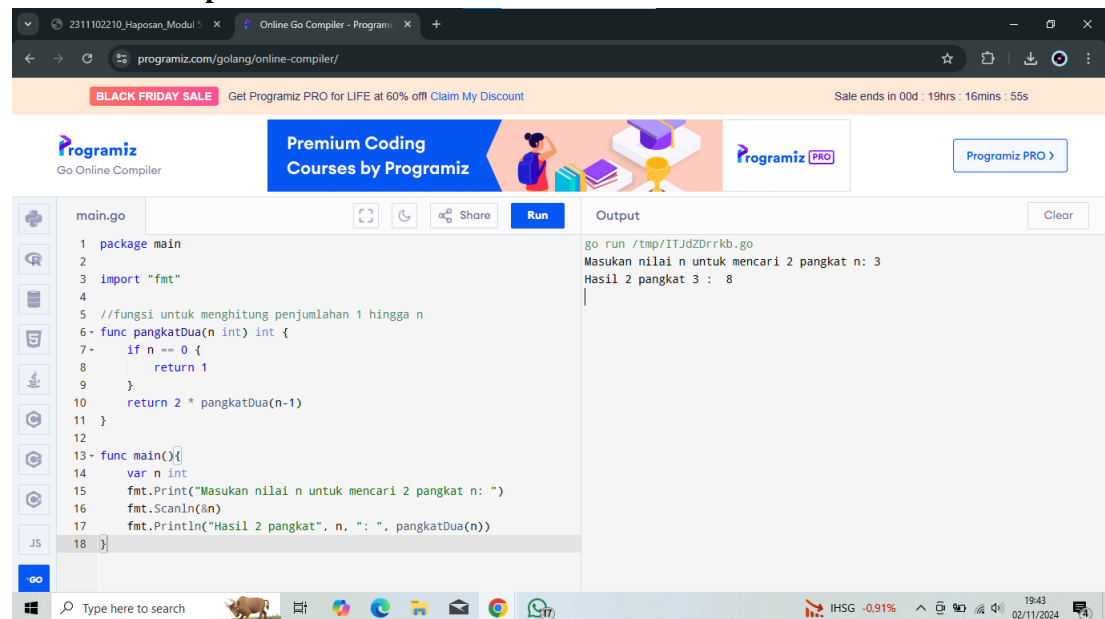
```
package main

import "fmt"

//fungsi untuk menghitung penjumlahan 1 hingga n
func pangkatDua(n int) int {
    if n == 0 {
        return 1
    }
    return 2 * pangkatDua(n-1)
}

func main(){
    var n int
    fmt.Print("Muhammad Djoko Susilo/2311102212 \n")
    fmt.Print("Masukan nilai n untuk mencari 2 pangkat
n: ")
    fmt.Scanln(&n)
    fmt.Println("Hasil 2 pangkat", n, ": ",
pangkatDua(n))
}
```

Screenshot Output



Deskripsi Program

Program merupakan suatu fungsi untuk menghitung nilai 2 pangkat n yang mana n merupakan nilai yang akan diinputkan oleh user. Cara kerja dari program ini adalah program meminta user untuk memasukkan nilai yang nanti akan masuk kedalam variabel n. Fungsi ini menggunakan rekursi untuk menghitung pangkat dua dengan cara mengalikan 2 dengan hasil pemanggilan fungsi rekursi yang menerima input n-1.

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"

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

func main(){
    var n int
    fmt.Print("Muhammad Djoko Susilo/2311102212 \n")
    fmt.Print("Masukan nilai n untuk mencari faktorial\n")
    n: "
    fmt.Scanln(&n)
    fmt.Println("Hasil faktorial dari", n, ": ",
faktorial(n))
}
```

Screenshot Output

```
1 package main
2
3 import "fmt"
4
5 func faktorial(n int) int {
6     if n == 0 || n == 1 {
7         return 1
8     }
9     return n * faktorial(n-1)
10 }
11
12 func main(){
13     var n int
14     fmt.Println("Masukan nilai n untuk mencari faktorial n: ")
15     fmt.Scanln(&n)
16     fmt.Println("Hasil faktorial dari", n, ":", faktorial(n))
17 }
```

Output

```
go run /tmp/BLzx9Asd0J.go
Masukan nilai n untuk mencari faktorial n: 5
Hasil faktorial dari 5 : 120
```

Deskripsi Program

Program ini adalah sebuah fungsi yang ditulis untuk menghasilkan nilai faktorial dari suatu bilangan n menggunakan rekursi. Cara kerja dari program ini adalah program akan meminta user untuk memasukkan nilai yang nanti akan masuk kedalam variabel n, didalam program ini terdapat kondisi dimana jika n adalah 0 atau 1 maka akan menghasilkan nilai 1. Jika n lebih besar dari 1 maka fungsi akan mengembalikan hasil perkalian antara n dan fungsi rekursi yang mengambil input n-1.

III. UNGUIDED

1. Soal Studi Case

Deret fibonacci adalah sebuah deret dengan nilai suku ke-0 dan ke-1 adalah 0 dan 1, dan nilai suku ke-n selanjutnya adalah hasil penjumlahan dua suku sebelumnya. Secara umum dapat diformulasikan $S_n = S_{n-1} + S_{n-2}$.

Sourcecode

```
package main

import "fmt"

// Fungsi untuk menghitung nilai Fibonacci ke-n
func hitungFibonacci(n int) int {
    if n == 0 {
        return 0
    }
}
```



```
        } else if n == 1 {
            return 1
        } else {
            return hitungFibonacci(n-1) +
hitungFibonacci(n-2)
        }
    }

func main() {
    var N int
    fmt.Print("Muhammad Djoko Susilo/2311102212 \n")
    fmt.Print("Masukkan nilai N untuk mencari nilai
Fibonacci hingga N: ")
    fmt.Scanln(&N)

    fmt.Println("Hasil Fibonacci:")
    for i := 1; i <= N; i++ {
        fib := hitungFibonacci(i)
        fmt.Printf("ke-%d: %d\n", i, fib)
    }
}
```

Screenshot Output

The screenshot shows the Programiz Online Go Compiler interface. The code editor on the left contains a Go program for calculating Fibonacci numbers using recursion. The program prompts the user to enter a value N, and then prints the first N Fibonacci numbers. The output window on the right shows the execution results for N=10, displaying the sequence: 1, 1, 2, 3, 5, 8, 13, 21, 34, 55.

```
main.go
7- if n == 0 {
8-     return 0
9- } else if n == 1 {
10-     return 1
11- } else {
12-     return hitungFibonacci(n-1) + hitungFibonacci(n-2)
13- }
14- }
15-
16- func main() {
17-     var N int
18-     fmt.Print("Masukkan nilai N untuk mencari nilai Fibonacci hingga N: ")
19-     fmt.Scanln(&N)
20-
21-     fmt.Println("Hasil Fibonacci:")
22-     for i := 1; i <= N; i++ {
23-         fib := hitungFibonacci(i)
24-         fmt.Printf("ke-%d: %d\n", i, fib)
25-     }
}
```

Output

```
go run /tmp/fcDZlyn63Q.go
Masukkan nilai N untuk mencari nilai Fibonacci hingga N: 10
Hasil Fibonacci:
ke-1: 1
ke-2: 1
ke-3: 2
ke-4: 3
ke-5: 5
ke-6: 8
ke-7: 13
ke-8: 21
ke-9: 34
ke-10: 55
```

Deskripsi Program

Program diatas adalah sebuah fungsi untuk menghitung nilai fibonacci ke-n menggunakan rekursi. Cara kerja dari program ini adalah program akan meminta user untuk memasukan nilai yang nanti akan masuk kedalam variabel n. dalam program ini memiliki kondisi dimana jika input merupakan 0 akan megembalikan nilai 0 begitu juga untuk inputan 1. Untuk n yang lebih dari 1 maka akan masuk kedalam fungsi menghitung nilai fibonacci ke n dengan cara menambahkan hasil dari fungsi rekursi yang menerima input n-1 + dan input n-2, sesuai definisi deret fibonacci.

2. Soal Studi Case

Buatlah sebuah program yang digunakan untuk menampilkan pola bintang berikut ini dengan meggunakan fungsi rekursif. N adalah masukan dari user.

Sourcecode

```
package main

import "fmt"

// Procedure untuk mencetak pola bintang secara rekursif
func pola(i, N int, s string) {
    if i <= N {
        s = "*" + s
        fmt.Println(s)
        pola(i+1, N, s)
    }
}

func main() {
    var N int
    fmt.Print("Muhammad Djoko Susilo/2311102212 \n")
}
```

```

        fmt.Print("Masukkan jumlah baris N untuk pola bintang:
    ")
    fmt.Scanln(&N)

    fmt.Println("Hasil Pola Bintang:")
    pola(1, N, "")
}

```

Screenshot Output

The screenshot shows the Programiz Go Online Compiler interface. On the left, the code editor displays a Go program named `main.go`. The code defines a recursive function `pola` that prints a star pattern based on the input `N`. The `main` function prompts the user to enter the number of rows, reads the input, and calls the `pola` function. On the right, the 'Output' pane shows the execution results: the prompt 'Masukkan jumlah baris N untuk pola bintang: 5', the output 'Hasil Pola Bintang:', and a star pattern consisting of 5 rows of stars (1, 2, 3, 4, and 5 stars respectively). The browser's address bar shows the URL `programiz.com/golang/online-compiler/`. A 'BLACK FRIDAY SALE' banner is visible at the top, and the Windows taskbar is visible at the bottom.

Deskripsi Program

Program ini adalah sebuah program yang bisa membuat suatu pola bintang dengan menggunakan fungsi rekursif yang akan sesuai dengan jumlah yang dimasukan oleh user. Fungsi pola menerima tiga parameter: `i` sebagai penghitung baris saat ini, `n` sebagai total jumlah baris, dan `s` sebagai string untuk menyimpan pola bintang. Cara kerja dari program ini adalah pertama program akan meminta user untuk memasukan nilai yang akan masuk kedalam variabel `n`. Di dalam pola, jika nilai `i` masih kurang atau sama dengan `n`, program menambahkan satu bintang ("`*`") ke dalam string `s` dan mencetak nilai `s`. Kemudian, fungsi pola dipanggil kembali dengan nilai `i` yang ditambah 1, yang membuat pola bertambah satu bintang pada setiap baris. Rekursi ini akan berhenti ketika nilai `i` melebihi `n`.

3. Soal Studi Case

Buatlah program yang mengimplementasikan rekursif untuk menampilkan faktor bilangan dari suatu `N`, atau bilangan yang apa saja yang habis membagi `N`.

Sourcecode

```
package main
```

```

import "fmt"

// Procedure untuk mencari dan mencetak faktor dari
bilangan n
func faktor(n, i int) {
    if i <= n {
        if n%i == 0 {
            fmt.Println(i)
        }
        faktor(n, i+1)
    }
}

func main() {
    var n int
    fmt.Print("Muhammad Djoko Susilo/2311102212 \n")
    fmt.Print("Masukkan nilai n untuk mencari faktor-
faktornya: ")
    fmt.Scanln(&n)

    fmt.Println("Faktor-faktor dari", n, "adalah:")
    faktor(n, 1)
}

```

Screenshoot Program

The screenshot shows the Programiz online Go compiler interface. The code editor on the left contains the Go program for finding factors. The output window on the right shows the program's execution results.

Code Editor (main.go):

```

2
3 import "fmt"
4
5 // Procedure untuk mencari dan mencetak faktor dari bilangan n
6 func faktor(n, i int) {
7     if i <= n {
8         if n%i == 0 {
9             fmt.Println(i)
10        }
11        faktor(n, i+1)
12    }
13 }
14
15 func main() {
16     var n int
17     fmt.Print("Masukkan nilai n untuk mencari faktor-faktornya: ")
18     fmt.Scanln(&n)
19
20     fmt.Println("Faktor-faktor dari", n, "adalah:")
21     faktor(n, 1)

```

Output:

```

go run /tmp/PIvBNfir2A.go
Masukkan nilai n untuk mencari faktor-faktornya: 12
Faktor-faktor dari 12 adalah:
1
2
3
4
6
12

```

Deskripsi Program

Program ini adalah sebuah program untuk menemukan dan mencetak semua faktor dari suatu bilangan positif N. Prosedur faktor bekerja secara rekursif, menerima dua

parameter: N sebagai bilangan yang akan dicari faktornya, dan i sebagai penghitung yang dimulai dari 1. Pada setiap pemanggilan, jika i merupakan faktor dari N (yaitu N habis dibagi i), maka i akan dicetak sebagai faktor. Proses ini berlanjut hingga i mencapai N, menghasilkan seluruh faktor dari bilangan N

4. Soal Studi Case

Buatlah program yang mengimplementasikan rekursif untuk menampilkan barisan bilangan tertentu

Sourecode Program

```
package main

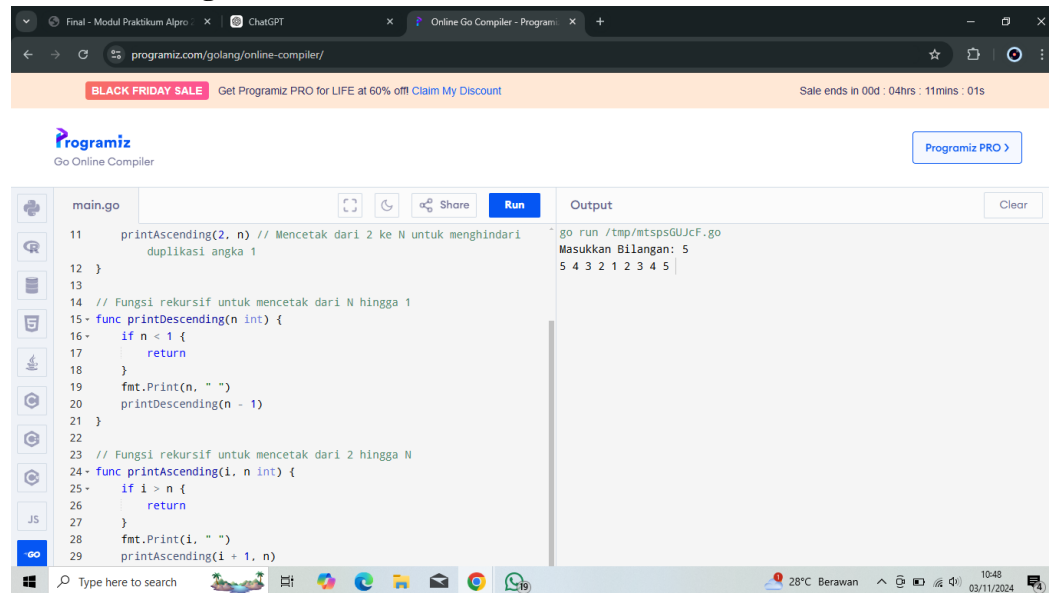
import "fmt"

// Fungsi Utama
func main() {
    var n int
    fmt.Print("Muhammad Djoko Susilo/2311102212 \n")
    fmt.Print("Masukkan Bilangan: ")
    fmt.Scan(&n)
    printDescending(n) // Mencetak dari N ke 1
    printAscending(2, n) // Mencetak dari 2 ke N untuk
    menghindari duplikasi angka 1
}

// Fungsi rekursif untuk mencetak dari N hingga 1
func printDescending(n int) {
    if n < 1 {
        return
    }
    fmt.Print(n, " ")
    printDescending(n - 1)
}

// Fungsi rekursif untuk mencetak dari 2 hingga N
func printAscending(i, n int) {
    if i > n {
        return
    }
    fmt.Print(i, " ")
    printAscending(i + 1, n)
}
```

Screenshot Program



The screenshot shows the Programiz Online Go Compiler interface. The code editor on the left contains a Go program named `main.go`. The program defines two recursive functions: `printDescending` and `printAscending`. `printDescending` prints numbers from `n` down to 1, and `printAscending` prints numbers from 2 up to `n`. The `main` function calls `printAscending(2, n)` and `printDescending(2, n)` with `n` as input. The output window on the right shows the execution results: `go run /tmp/mtsp5GJcf.go`, `Masukkan Bilangan: 5`, and the output `5 4 3 2 1 2 3 4 5`. The browser's address bar shows `programiz.com/golang/online-compiler/`. A banner at the top indicates a "BLACK FRIDAY SALE" with a 60% discount on Programiz PRO.

Deskripsi Program

Program ini bertujuan untuk mencetak deretan angka dari N hingga 1, lalu kembali dari 1 hingga N, tanpa mengulang angka 1 di tengah. Program meminta pengguna untuk memasukkan bilangan bulat positif N, kemudian menggunakan dua fungsi rekursif: `printDescending` untuk mencetak angka dari N ke 1, dan `printAscending` untuk mencetak angka dari 2 ke N. Dengan cara ini, program menghasilkan pola angka yang berurutan dari N ke 1 dan kembali lagi ke N.

5. Soal Studi Case

Bualah program yang mengimplementasikan rekursif untuk menampilkan barisan bilangan ganjil.

Sourcecode Program

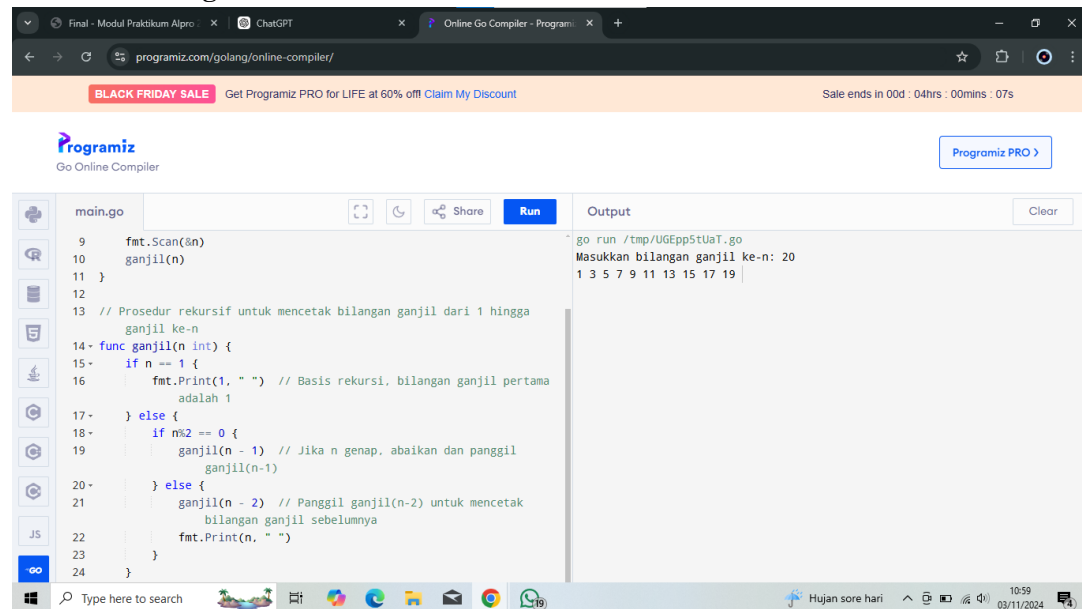
```
package main

import "fmt"

// Fungsi Utama
func main() {
    var n int
    fmt.Print("Muhammad Djoko Susilo/2311102212 \n")
    fmt.Print("Masukkan bilangan ganjil ke-n: ")
    fmt.Scan(&n)
    ganjil(n)
}
```

```
// Prosedur rekursif untuk mencetak bilangan ganjil
dari 1 hingga ganjil ke-n
func ganjil(n int) {
    if n == 1 {
        fmt.Print(1, " ") // Basis rekursi, bilangan
ganjil pertama adalah 1
    } else {
        if n%2 == 0 {
            ganjil(n - 1) // Jika n genap, abaikan dan
panggil ganjil(n-1)
        } else {
            ganjil(n - 2) // Panggil ganjil(n-2) untuk
mencetak bilangan ganjil sebelumnya
            fmt.Print(n, " ")
        }
    }
}
```

Screenshot Program



Deskripsi Program

Program ini menampilkan barisan bilangan ganjil hingga bilangan ganjil ke-n, menggunakan pendekatan rekursif. Program menerima masukan dari pengguna, yaitu bilangan bulat positif nnn, dan kemudian memanggil prosedur ganjil untuk mencetak bilangan ganjil dari yang terkecil hingga ganjil ke-n.

6. Soal Studi Case

Bualah program yang mengimplementasikan rekursif untuk mencari hasil pangkat dari dua buah bilangan.

Sourcecode Program

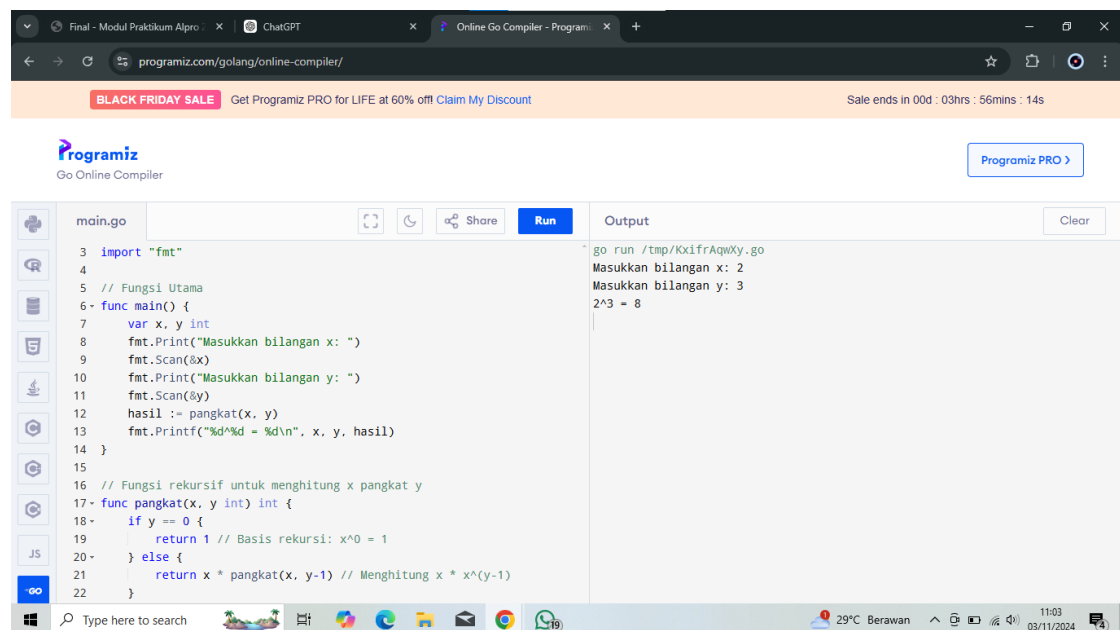
```
package main

import "fmt"

// Fungsi Utama
func main() {
    var x, y int
    fmt.Print("Muhammad Djoko Susilo/2311102212 \n")
    fmt.Print("Masukkan bilangan x: ")
    fmt.Scan(&x)
    fmt.Print("Masukkan bilangan y: ")
    fmt.Scan(&y)
    hasil := pangkat(x, y)
    fmt.Printf("%d^%d = %d\n", x, y, hasil)
}

// Fungsi rekursif untuk menghitung x pangkat y
func pangkat(x, y int) int {
    if y == 0 {
        return 1 // Basis rekursi: x^0 = 1
    } else {
        return x * pangkat(x, y-1) // Menghitung x *
x^(y-1)
    }
}
```

Screenshot Program



Deskripsi Program

Program meminta pengguna memasukkan dua bilangan bulat: x (basis) dan y (eksponen). Fungsi pangkat kemudian menghitung x^y secara rekursif. Jika $y=0$, fungsi mengembalikan 1. Jika $y>0$, fungsi mengembalikan x dikali hasil pangkat(x , $y-1$). Hasil akhirnya ditampilkan di layar.