

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL VI
REKURSIF**



Disusun Oleh :

Afra Lintang Maharani (2311102258)

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

Rekursif adalah teknik dalam pemrograman di mana sebuah fungsi memanggil dirinya sendiri secara langsung atau tidak langsung untuk menyelesaikan suatu masalah. Rekursif memungkinkan suatu masalah besar atau kompleks untuk dipecah menjadi sub-masalah yang lebih kecil dan lebih mudah dikelola. Dalam pemrograman, penggunaan rekursif harus selalu disertai dengan kondisi dasar (base case) untuk mencegah fungsi memanggil dirinya sendiri tanpa henti, yang akan menyebabkan stack overflow.

Rekursif biasa digunakan dalam program yang menggunakan fungsi faktorial, fibonacci, dan juga bilangan berpangkat.

Adapun kelebihan dan kekurangan rekursif sendiri yaitu,

- Sederhana dan Elegan, Rekursif terbukti memberikan solusi yang lebih sederhana dan lebih elegant daripada iterasi khususnya pada suatu masalah yang stereotipe berulang.
- Proses pengurangan dan peningkatan kompleksitas masalah, yaitu mengubah suatu persoalan besar ke dalam beberapa persoalan yang lebih kecil.

Untuk kekurangan nya sendiri rekursif,

- Rekursif berpotensi memerlukan overhead lebih tinggi daripada iterasi karena setiap kali fungsional dipanggil, akan memerlukan memori stack tambahan.
- Tidak adanya satu base case yang tepat dapat merubah kondisi ini menjadi stack overflow jika terlalu sering melakukan pemanggilan fungsi yang ada.

Optimisasi Rekursif, rekursif dapat di optimasi dengan menerapkan beberapa teknik pada fungsi rekursif itu sendiri, contohnya :

- Memoization, Sisa dari hasil hitung sub-masalah yang dapat di ulang penggunaannya mengurangi jumlah panggilan fungsi.
- Tail Recursion, Konversi fungsi rekursif menjadi terurai rekursif (apabila memungkinkan) di mana pemanggilan fungsi rekursif adalah tahap terakhir dalam suatu fungsi. Go tidak secara otomatis optimize tail-recursive tapi ini dapat membantu dalam beberapa kasus.

Rekursif adalah konsep power dalam pemrograman, mulai dari pemrograman Go. Sebagai konsekuensi dari mempelajari dasar dasar tentang rekursif programmer dapat mengatasi berbagai masalah rumit dalam cara yang lebih enteng dan teratur. Namun, tipe data ini harus digunakan secara rekursif sangat hati-hati karena perhatian terhadap waktu eksekusi dan stack overflow.

II. GUIDED

1. Soal Studi Case

Sourcecode

```
package main

import "fmt"

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

//
func main() {
    var n int
    fmt.Scan(&n)
    baris(n)
}
```

Output

```
PS D:\Dokument\Semester 2\.vscode> go run "d:\Dokument\Semester 2\.vscode\Semester3.go\Modul 6\guided1.go"
2
2
1
PS D:\Dokument\Semester 2\.vscode> go run "d:\Dokument\Semester 2\.vscode\Semester3.go\Modul 6\guided2.go"
```

Deskripsi

Program ini menggunakan fungsi rekursif untuk mencetak deretan bilangan dari suatu bilangan yang diberikan user hingga 1. Fungsi baris menerima satu argumen integer bilangan. Jika nilai bilangan adalah 1 maka, fungsi akan mencetak angka 1 dan berhenti. Jika tidak, fungsi akan mencetak nilai bilangan saat ini dan kemudian memanggil dirinya sendiri dengan bilangan yang dikurangi 1. Pada fungsi main, program meminta user untuk memasukkan sebuah angka n, dan kemudian memanggil fungsi baris dengan n sebagai argumennya. Hasilnya adalah deretan bilangan yang dicetak dari n hingga 1 secara menurun.

2. Soal Studi Case

Sourcecode

```
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.Scan(&n)
    fmt.Print(penjumlahan(n))
}
```

Screenshoot Output

```
PS D:\Dokument\Semester 2\.vscode> go run "d:\Dokument\Semester 2\.vscode\Semester3.go\Modul 6\guided2.go"
7
28
PS D:\Dokument\Semester 2\.vscode> go run "d:\Dokument\Semester 2\.vscode\Semester3.go\Modul 6\guided4.go"
```

Deskripsi Program

Program diatas adalah program dengan fungsi rekursif untuk menghitung penjumlahan deretan bilangan dari 1 hingga bilangan n yang diberikan oleh pengguna. Fungsi penjumlahan menerima satu argumen integer n. Jika nilai n adalah 1, fungsi akan mengembalikan nilai 1. Jika tidak, fungsi akan mengembalikan nilai n ditambah dengan hasil pemanggilan fungsi penjumlahan dengan n dikurangi 1. Pada fungsi main, program meminta pengguna untuk memasukkan sebuah angka n, dan kemudian memanggil fungsi penjumlahan dengan n sebagai argumennya. Hasil dari pemanggilan fungsi tersebut akan dicetak ke layar. Fungsi ini secara rekursif menjumlahkan semua bilangan dari 1 hingga n.

3. Soal Studi Case

Sourcecode

```
package main

import "fmt"

//fungsi rekursif untuk menghitung 2^n
func pangkat(n int) int {
    if n == 0 {
        return 1
    } else {
        return 2 * pangkat(n-1)
    }
}

func main() {
    var n int
    fmt.Print("Masukkan Nilai n : ")
    fmt.Scan(&n)
    fmt.Println("Hasil dari 2 pangkat", n, "adalah",
pangkat(n))
}
```

Output

```
1 PS D:\Dokument\Semester 2\.vscode> go run "d:\Dokument\Semester 2\.vscode\Semester3.go\Modul 6\guided3.go"
1 Masukkan Nilai n : 4
file... Hasil dari 2 pangkat 4 adalah 16
PS D:\Dokument\Semester 2\.vscode> go run "d:\Dokument\Semester 2\.vscode\Semester3.go\Modul 6\guided1.go"
```

Deskripsi

Program ini menggunakan fungsi rekursif untuk menghitung nilai 2 pangkat n, di mana n adalah bilangan bulat yang dimasukkan oleh pengguna. Fungsi pangkat menerima satu argumen integer n. Jika nilai n adalah 0, fungsi akan mengembalikan nilai 1, sesuai dengan aturan bahwa setiap bilangan pangkat 0 adalah 1. Jika tidak, fungsi akan mengembalikan hasil kali dari 2 dengan pemanggilan fungsi pangkat dengan n dikurangi 1. Pada fungsi main, program meminta pengguna untuk memasukkan sebuah angka n, dan kemudian mencetak hasil dari 2 pangkat n dengan memanggil fungsi pangkat tersebut. Fungsi ini secara rekursif mengalikan 2 sebanyak n kali untuk mendapatkan hasil akhirnya.

4. Soal Studi Case

Sourcecode

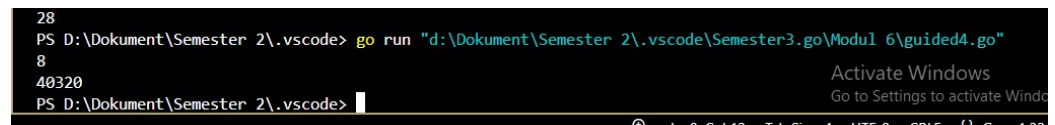
```
package main

import "fmt"

//n = 5
//5*4*3*2*1
func faktorial(n int) int {
    if n == 0 || n == 1 {
        return 1
    } else {
        return n * faktorial(n-1)
    }
}

//
func main() {
    var n int
    fmt.Scan(&n)
    fmt.Println(faktorial(n))
}
```

Screenshoot Output



```
PS D:\Dokument\Semester 2\.vscode> go run "d:\Dokument\Semester 2\.vscode\Semester3.go\Modul 6\guided4.go"
28
8
40320
PS D:\Dokument\Semester 2\.vscode>
```

Deskripsi Program

Program ini menggunakan fungsi rekursif untuk menghitung faktorial dari sebuah bilangan n yang dimasukkan oleh pengguna. Fungsi faktorial menerima satu argumen integer n . Jika nilai n adalah 0 atau 1, fungsi akan mengembalikan nilai 1, sesuai dengan definisi faktorial bahwa $0!$ dan $1!$ adalah 1. Jika tidak, fungsi akan mengembalikan hasil kali dari n dengan pemanggilan fungsi faktorial dengan n dikurangi 1. Pada fungsi main, program meminta pengguna untuk memasukkan sebuah angka n , dan kemudian mencetak hasil dari pemanggilan fungsi faktorial tersebut. Fungsi ini secara rekursif mengalikan semua bilangan dari n hingga 1 untuk mendapatkan nilai faktorial dari n . Sebagai contoh, jika n adalah 5, fungsi akan menghitung $5 * 4 * 3 * 2 * 1$ dan mengembalikan hasilnya.

III. UNGUIDED

1. Soal Studi Case

6.4 Soal Latihan Modul 6

- 1) 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 $S_n = S_{n-1} + S_{n-2}$. Berikut ini adalah contoh nilai deret fibonacci hingga suku ke-10. Buatlah program yang mengimplementasikan fungsi rekursif pada deret fibonacci tersebut.

| | | | | | | | | | | | |
|-------|---|---|---|---|---|---|---|----|----|----|----|
| n | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| S_n | 0 | 1 | 1 | 2 | 3 | 5 | 8 | 13 | 21 | 34 | 55 |

Sourcecode

```
package main

import (
    "fmt"
)

func FIBONACCI(n int) int {
    if n <= 1 {
        return n
    }
    return FIBONACCI(n-1) + FIBONACCI(n-2)
}

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

    fmt.Print("n\t| ")
    for i := 0; i <= n; i++ {
        fmt.Printf("%d\t", i)
    }
    fmt.Println()

    fmt.Print("Sn\t| ")
    for i := 0; i <= n; i++ {
        fmt.Printf("%d\t", FIBONACCI(i))
    }
    fmt.Println()
}
```




Screenshoot Output

```
PS D:\Dokument\Semester 2\.vscode> go run "d:\Dokument\Semester 2\.vscode\Semester3.go\Modul 6\ungui1.go"
Masukkan nilai n untuk batas deret FIBONACCI: 0
n
| 0
Sn
| 0
PS D:\Dokument\Semester 2\.vscode> go run "d:\Dokument\Semester 2\.vscode\Semester3.go\Modul 6\ungui1.go"
Masukkan nilai n untuk batas deret FIBONACCI: 1
n
| 0 1
Sn
| 0 1
PS D:\Dokument\Semester 2\.vscode>
```

Deskripsi Program

Program ini merupakan program yang menghitung dan menampilkan deret Fibonacci hingga nilai n yang diberikan oleh pengguna. Program ini menggunakan fungsi rekursif untuk menghitung nilai Fibonacci dari suatu bilangan. Fungsi FIBONACCI akan menerima satu integer n , Jika nilai n kurang dari atau sama dengan 1, fungsi akan mengembalikan nilai n (yaitu 0 atau 1, yang merupakan dua angka pertama dalam deret Fibonacci).

Jika tidak, fungsi akan mengembalikan hasil penjumlahan dari pemanggilan fungsi FIBONACCI dengan argumen $n-1$ dan $n-2$, sesuai dengan rumus deret Fibonacci ($F(n) = F(n-1) + F(n-2)$). Pada fungsi main, program meminta pengguna untuk memasukkan sebuah angka n yang akan digunakan sebagai batas deret Fibonacci yang akan dihitung.

Program mencetak judul " n " diikuti dengan deretan angka dari 0 hingga n .

Program kemudian mencetak judul " Sn " diikuti dengan nilai-nilai deret Fibonacci dari 0 hingga n , yang dihitung menggunakan fungsi FIBONACCI.

2. Soal Studi Case

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

Contoh masukan dan keluaran:

| No | Masukan | Keluaran |
|----|---------|---------------------------------|
| 1 | 5 | * ** *** **** ***** |
| 2 | 1 | * |
| 3 | 3 | * ** *** |

Sourcecode

```
package main

import (
    "fmt"
)

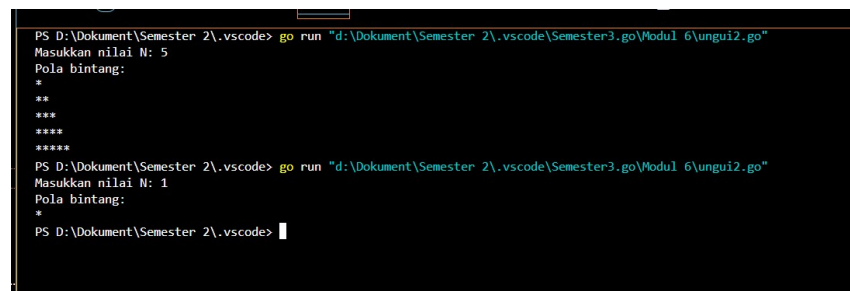
func printBintang(n int) {
    if n == 0 {
        return
    }
    fmt.Print("*")
    printBintang(n - 1)
}

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

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

    fmt.Println("Pola bintang:")
    printPola(n, 1)
}
```

Screenshoot Output



```
PS D:\Dokument\Semester 2\.vscode> go run "d:\Dokument\Semester 2\.vscode\Semester3.go\Modul 6\ungui2.go"
Masukkan nilai N: 5
Pola bintang:
*
**
***
****
*****
PS D:\Dokument\Semester 2\.vscode> go run "d:\Dokument\Semester 2\.vscode\Semester3.go\Modul 6\ungui2.go"
Masukkan nilai N: 1
Pola bintang:
*
PS D:\Dokument\Semester 2\.vscode> |
```

Deskripsi Program

Program diatas adalah program yang menggunakan fungsi rekursif untuk mencetak pola bintang bertingkat sesuai dengan nilai n yang dimasukkan oleh pengguna.

Fungsi printBintang menerima satu argumen integer n, jika n bernilai 0, fungsi akan berhenti (mengembalikan return).

Jika tidak, fungsi akan mencetak satu karakter bintang (*) dan kemudian memanggil dirinya sendiri dengan argumen n-1. Fungsi ini digunakan untuk mencetak deretan bintang sebanyak n. Kemudian ada fungsi printPola, fungsi ini menerima dua argumen integer (n dan current), jika nilai current lebih besar dari n, fungsi akan berhenti (mengembalikan return).

Jika tidak, fungsi akan memanggil printBintang dengan argumen current untuk mencetak deretan bintang sebanyak current.

Setelah mencetak deretan bintang, fungsi mencetak baris baru (fmt.Println()) dan kemudian memanggil dirinya sendiri dengan argumen current+1. Fungsi ini digunakan untuk mencetak pola bintang bertingkat dari 1 hingga n.

Lalu dalam fungsi main, program akan meminta pengguna untuk memasukkan sebuah angka n, yang akan digunakan sebagai jumlah baris pola bintang yang akan dicetak. Program kemudian mencetak judul "Pola bintang:" dan memanggil fungsi printPola dengan argumen n dan current dimulai dari 1.

3. Soal Studi Case

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

Masukan terdiri dari sebuah bilangan bulat positif N.

Keluaran terdiri dari barisan bilangan yang menjadi faktor dari N (terurut dari 1 hingga N ya).

Contoh masukan dan keluaran:

| No | Masukan | Keluaran |
|----|---------|--------------|
| 1 | 5 | 1 5 |
| 2 | 12 | 1 2 3 4 6 12 |

Sourcecode

```
package main

import (
    "fmt"
)

func printFaktor(n, i int) {
    if i > n {
        return
    }
    if n%i == 0 {
        fmt.Print(i, " ")
    }
    printFaktor(n, i+1)
}
```

```

    }
    printFaktor(n, i+1)
}

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

    fmt.Printf("Faktor dari %d adalah: ", n)
    printFaktor(n, 1)
    fmt.Println()
}

```

Screenshoot Output

```

PS D:\Dokument\Semester 2\.vscode> go run "d:\Dokument\Semester 2\.vscode\Semester3.go\Modul 6\ungui3.go"
Masukkan bilangan N: 5
Faktor dari 5 adalah: 1 5
PS D:\Dokument\Semester 2\.vscode> go run "d:\Dokument\Semester 2\.vscode\Semester3.go\Modul 6\ungui3.go"
Masukkan bilangan N: 12
Faktor dari 12 adalah: 1 2 3 4 6 12
PS D:\Dokument\Semester 2\.vscode>

```

Deskripsi Program

Program ini adalah program yang menggunakan fungsi rekursif untuk mencetak semua faktor dari suatu bilangan n yang dimasukkan oleh pengguna. Fungsi `printFaktor` akan menerima dua argumen integer, yaitu n (bilangan yang akan dicari faktornya) dan i (bilangan pembagi yang diuji).

Jika nilai i lebih besar dari n , fungsi akan berhenti (return), namun jika tidak, fungsi akan memeriksa apakah n habis dibagi i ($n \% i == 0$).

Jika n habis dibagi i , fungsi akan mencetak nilai i diikuti oleh spasi.

Fungsi kemudian memanggil dirinya sendiri dengan argumen n dan $i+1$, melanjutkan pencarian faktor dengan bilangan pembagi berikutnya.

Fungsi main:

Pada fungsi main, program meminta pengguna untuk memasukkan sebuah bilangan n yang akan dicari faktornya.

Program kemudian mencetak teks "Faktor dari n adalah: " dan memanggil fungsi `printFaktor` dengan argumen n dan i dimulai dari 1.

Setelah semua faktor dicetak, program mencetak baris baru (`fmt.Println()`) untuk mengakhiri output.

4. Soal Studi Case

- 4) Buatlah program yang mengimplementasikan rekursif untuk menampilkan barisan bilangan tertentu.

Masukan terdiri dari sebuah bilangan bulat positif N.

Keluaran terdiri dari barisan bilangan dari N hingga 1 dan kembali ke N.

Contoh masukan dan keluaran:

| No | Masukan | Keluaran |
|----|---------|-----------------------------------|
| 1 | 5 | 5 4 3 2 1 2 3 4 5 |
| 2 | 9 | 9 8 7 6 5 4 3 2 1 2 3 4 5 6 7 8 9 |

Sourcecode

```
package main

import (
    "fmt"
)

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

func printASCENDING(n, current int) {
    if current > n {
        return
    }
    fmt.Print(current, " ")
    printASCENDING(n, current+1)
}

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

    fmt.Printf("Hasil: ")
    printDESCENDING(n)
    printASCENDING(n, 1)
    fmt.Println()
```



Screenshoot Output

```
Faktor dari 12 adalah: 1 2 3 4 6 12
PS D:\Dokument\Semester 2\.vscode> go run "d:\Dokument\Semester 2\.vscode\Semester3.go\Modul 6\ungui4.go"
Masukkan nilai N: 5
Hasil: 5 4 3 2 1 1 2 3 4 5
PS D:\Dokument\Semester 2\.vscode> go run "d:\Dokument\Semester 2\.vscode\Semester3.go\Modul 6\ungui4.go"
Masukkan nilai N: 9
Hasil: 9 8 7 6 5 4 3 2 1 1 2 3 4 5 6 7 8 9
PS D:\Dokument\Semester 2\.vscode>
Activate Windows
Go to Settings to activate Windows.
```

Deskripsi Program

Program ini merupakan program yang menggunakan dua fungsi rekursif untuk mencetak bilangan dalam urutan menurun (descending) dan naik (ascending) dari sebuah bilangan n yang dimasukkan oleh pengguna. Fungsi printDESCENDING menerima satu argumen integer n. Jika nilai n kurang dari 1, fungsi akan berhenti (return). Jika tidak, fungsi akan mencetak nilai n diikuti oleh spasi dan kemudian memanggil dirinya sendiri dengan argumen n-1, melanjutkan pencetakan bilangan dalam urutan menurun. Fungsi printASCENDING menerima dua argumen integer: n dan current. Jika nilai current lebih besar dari n, fungsi akan berhenti (return). Jika tidak, fungsi akan mencetak nilai current diikuti oleh spasi dan kemudian memanggil dirinya sendiri dengan argumen n dan current+1, melanjutkan pencetakan bilangan dalam urutan naik. Pada fungsi main, program meminta pengguna untuk memasukkan sebuah bilangan n, kemudian mencetak teks "Hasil:" sebagai pembuka. Program memanggil fungsi printDESCENDING dengan argumen n untuk mencetak bilangan dari n ke 1. Setelah itu, program memanggil fungsi printASCENDING dengan argumen n dan current dimulai dari 1 untuk mencetak bilangan dari 1 ke n, dan akhirnya mencetak baris baru (fmt.Println()) untuk mengakhiri output.

5. Soal Studi Case

5) Buatlah program yang mengimplementasikan rekursif untuk menampilkan barisan bilangan ganjil.

Masukan terdiri dari sebuah bilangan bulat positif N.

Keluaran terdiri dari barisan bilangan ganjil dari 1 hingga N.

Contoh masukan dan keluaran:

| No | Masukan | Keluaran |
|----|---------|--------------------------|
| 1 | 5 | 1 3 5 |
| 2 | 20 | 1 3 5 7 9 11 13 15 17 19 |

Sourcecode

```
package main

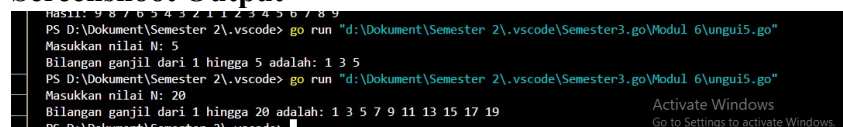
import (
    "fmt"
)

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

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

    fmt.Printf("Bilangan ganjil dari 1 hingga %d adalah: ",
n)
    printBilanganGanjil(n, 1)
    fmt.Println()
}
```

Screenshoot Output



```
PS D:\Dokument\Semester 2\.vscode> go run "d:\Dokument\Semester 2\.vscode\Semester3.go\Modul 6\ungui5.go"
Masukkan nilai N: 5
Bilangan ganjil dari 1 hingga 5 adalah: 1 3 5
PS D:\Dokument\Semester 2\.vscode> go run "d:\Dokument\Semester 2\.vscode\Semester3.go\Modul 6\ungui5.go"
Masukkan nilai N: 20
Bilangan ganjil dari 1 hingga 20 adalah: 1 3 5 7 9 11 13 15 17 19
PS D:\Dokument\Semester 2\.vscode>
```

Deskripsi Program

Program diatas adalah program yang menggunakan fungsi rekursif untuk mencetak semua bilangan ganjil dari 1 hingga sebuah bilangan n yang dimasukkan oleh pengguna. Fungsi printBilanganGanjil menerima dua argumen integer: n dan current. Jika nilai current lebih besar dari n, fungsi akan berhenti (return). Jika tidak, fungsi akan memeriksa apakah current adalah bilangan ganjil dengan menggunakan operator modulus ($\text{current} \% 2 \neq 0$). Jika current adalah bilangan ganjil, fungsi akan mencetak nilai current diikuti oleh spasi. Fungsi kemudian memanggil dirinya sendiri dengan argumen n dan current+1, melanjutkan pencetakan bilangan ganjil berikutnya hingga current melebihi n.

Pada fungsi main, program meminta pengguna untuk memasukkan sebuah bilangan n. Program kemudian mencetak teks "Bilangan ganjil dari 1 hingga n adalah: " sebagai pembuka. Setelah itu, program memanggil fungsi printBilanganGanjil dengan argumen n dan current dimulai dari 1 untuk mencetak semua bilangan ganjil dari 1 hingga n, dan akhirnya mencetak baris baru (fmt.Println()) untuk mengakhiri output.

6. Soal Studi Case

- 6) Buatlah program yang mengimplementasikan rekursif untuk mencari hasil pangkat dari dua buah bilangan.

Masukan terdiri dari bilangan bulat x dan y.

Keluaran terdiri dari hasil x dipangkatkan y.

Catatan: diperbolehkan menggunakan asterik "*", tapi dilarang menggunakan import "math".

Contoh masukan dan keluaran:

| No | Masukan | Keluaran |
|----|---------|----------|
| 1 | 2 2 | 4 |
| 2 | 5 3 | 125 |

A
G

Sourcecode

```
package main

import (
    "fmt"
)

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

func main() {
    var x, y int
    fmt.Print("Masukkan bilangan x (basis): ")
    fmt.Scan(&x)
    fmt.Print("Masukkan bilangan y (pangkat): ")
    fmt.Scan(&y)

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


Screenshoot Output

```
Bilangan ganjil dari 1 hingga 20 adalah: 1 3 5 7 9 11 13 15 17 19
PS D:\Dokument\Semester 2\.vscode> go run "d:\Dokument\Semester 2\.vscode\Semester3.go\Modul 6\ungui6.go"
Masukkan bilangan x (basis): 2
Masukkan bilangan y (pangkat): 2
Hasil 2 pangkat 2 adalah: 4
PS D:\Dokument\Semester 2\.vscode> go run "d:\Dokument\Semester 2\.vscode\Semester3.go\Modul 6\ungui6.go"
Masukkan bilangan x (basis): 5
Masukkan bilangan y (pangkat): 3
Hasil 5 pangkat 3 adalah: 125
PS D:\Dokument\Semester 2\.vscode>
```

Deskripsi Program

Program ini adalah program yang menggunakan fungsi rekursif untuk menghitung hasil perpangkatan dari sebuah bilangan x yang dipangkatkan dengan y. Fungsi POWER menerima dua argumen integer: x (basis) dan y (pangkat). Jika nilai y sama dengan 0, fungsi akan mengembalikan nilai 1, sesuai dengan aturan matematika bahwa setiap bilangan yang dipangkatkan 0 adalah 1. Jika tidak, fungsi akan mengembalikan hasil kali dari x dengan pemanggilan fungsi POWER dengan argumen x dan y-1. Fungsi ini secara rekursif mengalikan x dengan dirinya sendiri y kali.

Pada fungsi main, program meminta pengguna untuk memasukkan dua bilangan: x sebagai basis dan y sebagai pangkat. Program kemudian memanggil fungsi POWER dengan argumen x dan y untuk menghitung hasil perpangkatan tersebut, dan menyimpan hasilnya dalam variabel result. Setelah itu, program mencetak hasil perhitungan dengan format "Hasil x pangkat y adalah: result".

IV. DAFTAR PUSTAKA

Spark Code Hub. (n.d.). *Golang recursion explained*. Retrieved November 3, 2024, from <https://www.sparkcodehub.com/golang-recursion-explained>