

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

Modul 5 : Rekursif



Oleh:

ANGGUN WAHYU WIDIYANA

103112480280

12-IF-01

**S1 TEKNIK INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
TAHUN AJARAN 2025/2026**

I. DASAR TEORI

Rekursif adalah teknik pemrograman di mana sebuah fungsi atau prosedur memanggil dirinya sendiri untuk menyelesaikan masalah. Teknik ini sering digunakan untuk menyelesaikan masalah yang memiliki pola berulang atau dapat dipecah menjadi sub-masalah yang lebih kecil.

A. Definisi Rekursif

Rekursif bekerja dengan prinsip pemanggilan diri, di mana sebuah fungsi terus memanggil dirinya sendiri hingga mencapai kondisi tertentu yang disebut base-case. Base-case adalah kondisi penghentian rekursi, yang memastikan bahwa proses tidak berjalan tanpa henti. Setelah base-case tercapai, fungsi akan kembali (backward) ke pemanggilan sebelumnya untuk menyelesaikan instruksi yang tersisa.

Misalnya, prosedur cetak berikut ini menunjukkan bagaimana sebuah fungsi dapat memanggil dirinya sendiri:

```
package main
import "fmt"
func main() {
    cetak(5)
}
func cetak(x int) {
    fmt.Println(x)
    cetak(x + 1) // Pemanggilan diri
}
```

Ketika fungsi cetak(5) dieksekusi, ia akan terus mencetak angka yang bertambah tanpa henti. Untuk menghentikan proses ini, perlu ditambahkan struktur kontrol percabangan (if-then) yang dikenal sebagai base-case.

Base-case: Kondisi yang menghentikan proses rekursif. Misalnya, ketika x mencapai 10.

Recursive-case: Kondisi di mana fungsi memanggil dirinya sendiri. Ini terjadi ketika x tidak sama dengan 10.

B. Contoh Program Menggunakan Rekursif

Berikut adalah beberapa contoh program menggunakan rekursif:

Deret Fibonacci

Deret Fibonacci adalah deret angka di mana setiap angka merupakan jumlah dari dua angka sebelumnya.

```
package main

import "fmt"

// Fungsi rekursif untuk menghitung Fibonacci
func fibonacci(n int) int {
    if n == 0 {
        return 0
    }
}
```

```

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

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

    fmt.Println("Deret Fibonacci:")
    for i := 0; i < n; i++ {
        fmt.Printf("%d ", fibonacci(i))
    }
}

```

Tabel 1.1 Deret Fibonacci

Perhitungan Faktorial

Faktorial dari sebuah bilangan n adalah hasil perkalian semua bilangan bulat positif dari 1 hingga n

```

package main

import "fmt"

// Fungsi rekursif untuk menghitung faktorial
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 bilangan untuk menghitung faktorial: ")
    fmt.Scan(&n)

    fmt.Printf("Faktorial dari %d adalah %d\n", n, faktorial(n))
}

```

Tabel 1.2 Notasi dalam Go

Menentukan Bilangan Terbesar dalam Array

Rekursif juga dapat digunakan untuk mencari elemen terbesar dalam array.

```

package main

import "fmt"

// Fungsi rekursif untuk menemukan bilangan terbesar dalam array
func maxArray(arr []int, n int) int {
    if n == 1 {
        return arr[0]
    }
}

```

```

    max := maxArray(arr, n-1)
    if arr[n-1] > max {
        return arr[n-1]
    }
    return max
}

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

    arr := make([]int, n)
    fmt.Println("Masukkan elemen array:")
    for i := 0; i < n; i++ {
        fmt.Scan(&arr[i])
    }

    fmt.Printf("Elemen terbesar dalam array adalah %d\n",
maxArray(arr, n))
}

```

C. Kelebihan dan Kekurangan Rekursif

Kelebihan: Sederhana dan Elegan: Kode lebih ringkas dan mudah dipahami untuk masalah berulang, Cocok untuk Masalah Tertentu: Sangat efektif untuk algoritma seperti pencarian biner dan perhitungan deret Fibonacci.

Kekurangan: Penggunaan Memori Tinggi: Rekursi menggunakan stack untuk setiap pemanggilan fungsi sehingga dapat menyebabkan stack overflow jika terlalu dalam.

II. GUIDED

Contoh Soal 1

```
//Nama: Anggun Wahyu Widiyana (103112480280)
package main
import "fmt"
//Fungsi iteratif untuk menghitung pangkat (base^exp)
func pangkatIteratif(base, exp int) int {
    hasil:= 1
    for i := 0; i < exp; i++ {
        hasil *= base
    }
    return hasil
}
//Fungsi iteratif untuk menghitung faktorial (n!)
func faktorialIteratif(n int) int {
    hasil := 1
    for i := 2; i <= n; i++ {
        hasil *= i
    }
    return hasil
}
func main() {
    var base, exp, n int
    // Input pangkat
    fmt.Print("Masukkan bilangan: ")
    fmt.Scanln(&base)
    fmt.Print("Masukkan pangkat: ")
    fmt.Scanln(&exp)

    fmt.Printf("%d^%d = %d\n", base, exp, pangkatIteratif(base,
exp))
    // Input faktorial
    fmt.Print("Masukkan angka untuk faktorial: ")
    fmt.Scanln(&n)

    fmt.Printf("%d! = %d\n", n, faktorialIteratif(n))
}
```

Output:

```
PS C:\Users\User\Documents\ALPRO 2\103112480280_MODULE5> go run "c:\Users\User\Documents\ALPRO 2\103112480280_guided5\ContohSoal1.go"
Masukkan bilangan: 6
Masukkan pangkat: 2
6^2 = 36
Masukkan angka untuk faktorial: 6
6! = 720
PS C:\Users\User\Documents\ALPRO 2\103112480280_MODULE5> go run "c:\Users\User\Documents\ALPRO 2\103112480280_guided5\ContohSoal1.go"
Masukkan bilangan: 5
Masukkan pangkat: 3
5^3 = 125
Masukkan angka untuk faktorial: 25
25! = 7034535277573963776
PS C:\Users\User\Documents\ALPRO 2\103112480280_MODULE5> go run "c:\Users\User\Documents\ALPRO 2\103112480280_guided5\ContohSoal1.go"
Masukkan bilangan: █
```

Deskripsi Program (Iteratif)

Program ini berfungsi untuk menghitung dua operasi matematika dasar: pangkat dan faktorial. Program menggunakan loop seperti `for` atau `while`. Menghitung pangkat dengan cara mengalikan bilangan dasar (**base**) sebanyak nilai eksponen (**exp**) menggunakan loop. Menghitung faktorial dengan cara mengalikan angka dari 1 hingga **n** menggunakan loop.

pangkatIteratif (base, exp int) int: Fungsi ini menghitung hasil dari **base** dipangkatkan **exp** menggunakan pendekatan iteratif. Hasilnya dihitung dengan mengalikan **base** sebanyak **exp** kali.

faktorialIteratif (n int) int: Fungsi ini menghitung faktorial dari bilangan bulat positif **n** menggunakan pendekatan iteratif. Faktorial dihitung dengan mengalikan semua bilangan bulat dari 1 hingga **n**.

Contoh Soal 2

```
//Nama: Anggun Wahyu Widiyana (103112480280)
package main
import "fmt"
//pangkat 0 adalah 1
func pangkatRekursif(base, exp int) int {
    if exp == 0 {
        return 1
    }
    return base * pangkatRekursif(base, exp-1)
}
func faktorialRekursif(n int) int {
    if n == 0 || n == 1 {
        return 1
    }
    return n * faktorialRekursif(n-1)
}
func main() {
    var base, exp, n int
    // Input pangkat
    fmt.Print("Masukkan bilangan: ")
    fmt.Scanln(&base)
    fmt.Print("Masukkan pangkat: ")
    fmt.Scanln(&exp)

    fmt.Printf("%d^%d = %d\n", base, exp, pangkatRekursif(base,
exp))

    // Input faktorial
    fmt.Print("Masukkan angka untuk faktorial: ")
    fmt.Scanln(&n)
```

```
    fmt.Printf("%d!=%d\n", n, faktorialRekursif(n))
}
```

Output:

```
PS C:\Users\User\Documents\ALPRO 2\103112480280_MODULE5> go run "c:\Users\User\Documents\ALPRO 2\103112480280_MODULE5\103112480280_Guided5\ContohSoal2.go"
Masukkan bilangan: 6
Masukkan pangkat: 2
6^2 = 36
Masukkan angka untuk faktorial: 6
6!=720
PS C:\Users\User\Documents\ALPRO 2\103112480280_MODULE5> go run "c:\Users\User\Documents\ALPRO 2\103112480280_MODULE5\103112480280_Guided5\ContohSoal2.go"
Masukkan bilangan: 5
Masukkan pangkat: 3
5^3 = 125
Masukkan angka untuk faktorial: 25
25!=7034535277573963776
PS C:\Users\User\Documents\ALPRO 2\103112480280_MODULE5> go run "c:\Users\User\Documents\ALPRO 2\103112480280_MODULE5\103112480280_Guided5\ContohSoal2.go"
Masukkan bilangan: 
```

Deskripsi Program (Rekursif)

Program ini memiliki fungsi yang sama dengan Progrm Contoh Soal 1. Program menggunakan fungsi yang memanggil dirinya sendiri untuk menyelesaikan masalah secara bertahap. pangkat dengan cara mengalikan bilangan dasar (**base**) dengan hasil pangkat sebelumnya (**base^(exp-1)**) hingga (**exp**) mencapai 0. Menghitung faktorial dengan cara mengalikan angka **n** dengan faktorial dari angka sebelumnya (**(n-1) !**) hingga **n** mencapai 0 atau 1.

pangkatRekursif (base, exp int) int: Fungsi ini menghitung hasil dari **base** dipangkatkan **exp** menggunakan pendekatan rekursif. Jika eksponen (**exp**) adalah 0, fungsi mengembalikan 1 (karena pangkat 0 adalah 1). Jika tidak, fungsi mengalikan (**base**) dengan hasil dari pemanggilan diri fungsi dengan eksponen yang dikurangi 1.

faktorialRekursif (n int) int: Fungsi ini menghitung faktorial dari bilangan bulat positif **n** menggunakan pendekatan rekursif. Jika **n** adalah 0 atau 1, fungsi mengembalikan 1 (karena faktorial dari 0 dan 1 adalah 1). Jika tidak, fungsi mengalikan **n** dengan hasil dari pemanggilan diri fungsi dengan **n** yang dikurangi 1.

III. UNGUIDED

Latihan Soal 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

```
// Nama: Anggun Wahyu Widiyana (103112480280)
package main

import "fmt"

func fibonacci(n int) int {
    if n == 0 { // Base-case pertama
        return 0
    } else if n == 1 { // Base-case kedua
        return 1
    } else { // Recursive-case
        return fibonacci(n-1) + fibonacci(n-2)
    }
}

func main() {
    var n int
    fmt.Scan(&n)

    fmt.Printf("Nilai suku ke-%d dari deret Fibonacci adalah %d\n",
n, fibonacci(n))
}
```

Output:

```
PS C:\Users\User\Documents\ALPRO 2\103112480280_MODULE5> go run "c:\Users\User\Documents\ALPRO 2\103112480280_MODULE5\103112480280_Unguided5\LatihanSoal1.go"
0
Nilai suku ke-0 dari deret Fibonacci adalah 0
PS C:\Users\User\Documents\ALPRO 2\103112480280_MODULE5> go run "c:\Users\User\Documents\ALPRO 2\103112480280_MODULE5\103112480280_Unguided5\LatihanSoal1.go"
1
Nilai suku ke-1 dari deret Fibonacci adalah 1
PS C:\Users\User\Documents\ALPRO 2\103112480280_MODULE5> go run "c:\Users\User\Documents\ALPRO 2\103112480280_MODULE5\103112480280_Unguided5\LatihanSoal1.go"
2
Nilai suku ke-2 dari deret Fibonacci adalah 1
PS C:\Users\User\Documents\ALPRO 2\103112480280_MODULE5> go run "c:\Users\User\Documents\ALPRO 2\103112480280_MODULE5\103112480280_Unguided5\LatihanSoal1.go"
3
Nilai suku ke-3 dari deret Fibonacci adalah 2
PS C:\Users\User\Documents\ALPRO 2\103112480280_MODULE5> go run "c:\Users\User\Documents\ALPRO 2\103112480280_MODULE5\103112480280_Unguided5\LatihanSoal1.go"
4
Nilai suku ke-4 dari deret Fibonacci adalah 3
PS C:\Users\User\Documents\ALPRO 2\103112480280_MODULE5> go run "c:\Users\User\Documents\ALPRO 2\103112480280_MODULE5\103112480280_Unguided5\LatihanSoal1.go"
5
Nilai suku ke-5 dari deret Fibonacci adalah 5
PS C:\Users\User\Documents\ALPRO 2\103112480280_MODULE5> go run "c:\Users\User\Documents\ALPRO 2\103112480280_MODULE5\103112480280_Unguided5\LatihanSoal1.go"
6
Nilai suku ke-6 dari deret Fibonacci adalah 8
PS C:\Users\User\Documents\ALPRO 2\103112480280_MODULE5> go run "c:\Users\User\Documents\ALPRO 2\103112480280_MODULE5\103112480280_Unguided5\LatihanSoal1.go"
7
Nilai suku ke-7 dari deret Fibonacci adalah 13
PS C:\Users\User\Documents\ALPRO 2\103112480280_MODULE5> go run "c:\Users\User\Documents\ALPRO 2\103112480280_MODULE5\103112480280_Unguided5\LatihanSoal1.go"
8
```



```

PS C:\Users\User\Documents\ALPRO 2\103112480280_MODULE5> go run "c:\Users\User\Documents\ALPRO 2\103112480280_MODULE5\103112480280_Unguided5\LatihanSoal1.go"
8
Nilai suku ke-8 dari deret Fibonacci adalah 21
PS C:\Users\User\Documents\ALPRO 2\103112480280_MODULE5> go run "c:\Users\User\Documents\ALPRO 2\103112480280_MODULE5\103112480280_Unguided5\LatihanSoal1.go"
9
Nilai suku ke-9 dari deret Fibonacci adalah 34
PS C:\Users\User\Documents\ALPRO 2\103112480280_MODULE5> go run "c:\Users\User\Documents\ALPRO 2\103112480280_MODULE5\103112480280_Unguided5\LatihanSoal1.go"
10
Nilai suku ke-10 dari deret Fibonacci adalah 55
PS C:\Users\User\Documents\ALPRO 2\103112480280_MODULE5> go run "c:\Users\User\Documents\ALPRO 2\103112480280_MODULE5\103112480280_Unguided5\LatihanSoal1.go"

```

Deskripsi Program

Program ini menghitung nilai suku ke-n dari deret Fibonacci menggunakan pendekatan rekursif. Deret Fibonacci adalah urutan angka yang dimulai dengan 0 dan 1, di mana setiap suku berikutnya merupakan jumlah dari dua suku sebelumnya.

Formula umum untuk menghitung suku ke-n adalah: $S_n = S_{n-1} + S_{n-2}$

Fungsi **fibonacci (n int) int**: untuk menghitung nilai suku ke-n dari deret Fibonacci. Jika n sama dengan 0, fungsi mengembalikan nilai 0, Jika n sama dengan 1, fungsi mengembalikan nilai 1. Untuk nilai n yang lebih besar, fungsi memanggil dirinya sendiri untuk menghitung dua suku sebelumnya, yaitu **fibonacci (n-1)** dan **fibonacci (n-2)**, dan menjumlahkan hasilnya.

Latihan Soal 2

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

No	Masukan	Keluaran
1.	5	<pre> * ** *** **** ***** </pre>
2.	1	<pre> * </pre>
3.	3	<pre> * ** *** </pre>

```

//Nama: Anggun Wahyu Widiyana (103112480280)
package main

import "fmt"

func cetakBintang(n int, i int) {
    if i > n {
        return
    }
    for j := 0; j < i; j++ {
        fmt.Print("*")
    }
    fmt.Println()
    cetakBintang(n, i+1)
}

```

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

Output:

```
PS C:\Users\User\Documents\ALPRO 2\103112480280_MODULES> go run "c:\Users\User\Documents\ALPRO 2\103112480280_MODULES\103112480280_Unguided5\LatihanSoal2.go"
5
*
**
***
****
*****
PS C:\Users\User\Documents\ALPRO 2\103112480280_MODULES> go run "c:\Users\User\Documents\ALPRO 2\103112480280_MODULES\103112480280_Unguided5\LatihanSoal2.go"
1
*
PS C:\Users\User\Documents\ALPRO 2\103112480280_MODULES> go run "c:\Users\User\Documents\ALPRO 2\103112480280_MODULES\103112480280_Unguided5\LatihanSoal2.go"
3
*
**
***
PS C:\Users\User\Documents\ALPRO 2\103112480280_MODULES> go run "c:\Users\User\Documents\ALPRO 2\103112480280_MODULES\103112480280_Unguided5\LatihanSoal2.go"
```

Deskripsi Program

Program ini berfungsi untuk mencetak pola bintang berdasarkan jumlah baris (N) yang dimasukkan oleh pengguna. Pola bintang ditampilkan secara bertingkat, di mana baris pertama memiliki satu bintang, baris kedua memiliki dua bintang, dan seterusnya hingga baris ke- N memiliki N bintang.

Fungsi **Rekursif cetakBintang (n int, i int)**: mencetak pola bintang secara bertahap menggunakan pendekatan rekursif. Parameter n adalah jumlah baris yang ingin dicetak, sedangkan i adalah nomor baris saat ini (dimulai dari 1). Base-case terjadi ketika $i > n$, di mana fungsi berhenti. Recursive-case terjadi ketika fungsi memanggil dirinya sendiri dengan nilai $i+1$ untuk mencetak baris berikutnya.

Latihan Soal 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

```
//Nama: Anggun Wahyu Widiyana (103112480280)

package main

import "fmt"
```

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

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

Output:

```
PS C:\Users\User\Documents\ALPRO 2\103112480280_MODULE5> go run "c:\Users\User\Documents\ALPRO 2\103112480280_MODULE5\103112480280_Unguided5\LatihanSoal3.go"
5
PS C:\Users\User\Documents\ALPRO 2\103112480280_MODULE5> go run "c:\Users\User\Documents\ALPRO 2\103112480280_MODULE5\103112480280_Unguided5\LatihanSoal3.go"
1 2 3 4 6 12
PS C:\Users\User\Documents\ALPRO 2\103112480280_MODULE5> go run "c:\Users\User\Documents\ALPRO 2\103112480280_MODULE5\103112480280_Unguided5\LatihanSoal3.go"
1 5 25
PS C:\Users\User\Documents\ALPRO 2\103112480280_MODULE5> go run "c:\Users\User\Documents\ALPRO 2\103112480280_MODULE5\103112480280_Unguided5\LatihanSoal3.go"
```

Deskripsi Program

Program ini berfungsi untuk menghitung dan mencetak semua faktor dari bilangan bulat positif N yang dimasukkan oleh pengguna. Faktor adalah bilangan bulat yang dapat membagi N tanpa menyisakan sisa.

Fungsi cetakFaktor (n int, i int): untuk mencari dan mencetak faktor dari N. Parameter n adalah bilangan yang ingin dicari faktornya, sedangkan i adalah angka yang sedang diperiksa sebagai faktor, dimulai dari 1. Fungsi memeriksa apakah i adalah faktor dari N menggunakan operasi modulus ($n \% i == 0$). Jika benar, i dicetak sebagai faktor. Fungsi memanggil dirinya sendiri dengan nilai $i+1$ untuk memeriksa angka berikutnya hingga mencapai nilai lebih besar dari N.

Latihan Soal 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

```
//Nama: Anggun Wahyu Widiyana (103112480280)
package main

import "fmt"

func cetakBilangan(n int, angkaSaatIni int) {
    if angkaSaatIni < 1 {
        return
    }
    fmt.Print(angkaSaatIni, " ")
    cetakBilangan(n, angkaSaatIni-1)
    if angkaSaatIni != n {
        fmt.Print(angkaSaatIni, " ")
    }
}

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

Output:

```
PS C:\Users\User\Documents\ALPRO 2\103112480280_MODULES> go run "c:\Users\User\Documents\ALPRO 2\103112480280_MODULES\103112480280_Unguided5\LatihanSoal4.go"
5
5 4 3 2 1 2 3 4
PS C:\Users\User\Documents\ALPRO 2\103112480280_MODULES> go run "c:\Users\User\Documents\ALPRO 2\103112480280_MODULES\103112480280_Unguided5\LatihanSoal4.go"
9
9 8 7 6 5 4 3 2 1 2 3 4 5 6 7 8
PS C:\Users\User\Documents\ALPRO 2\103112480280_MODULES> go run "c:\Users\User\Documents\ALPRO 2\103112480280_MODULES\103112480280_Unguided5\LatihanSoal4.go"
15
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 2 3 4 5 6 7 8 9 10 11 12 13 14
PS C:\Users\User\Documents\ALPRO 2\103112480280_MODULES> go run "c:\Users\User\Documents\ALPRO 2\103112480280_MODULES\103112480280_Unguided5\LatihanSoal4.go"
```

Deskripsi Program

Program ini mencetak barisan bilangan dari N hingga 1, kemudian kembali ke N, menggunakan fungsi rekursif. Pengguna memasukkan bilangan bulat positif N, dan program menghasilkan output sesuai dengan pola tersebut.

Cara Kerja: Fungsi rekursif mencetak angka dari N menurun hingga 1, Setelah mencapai 1, fungsi mencetak angka kembali dari 2 hingga N.

Latihan Soal 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

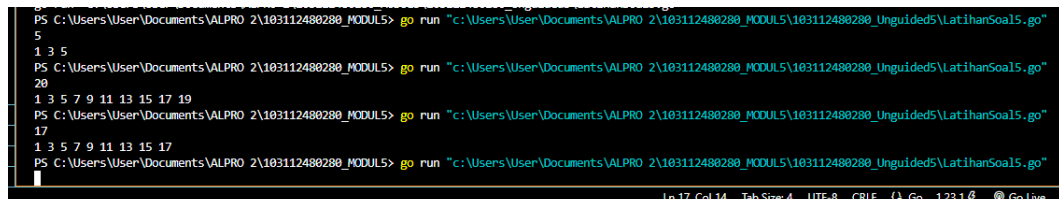
```
//Nama: Anggun Wahyu Widiyana (103112480280)
package main

import "fmt"

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

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

Output:



Deskripsi Program

Program ini dirancang untuk mencetak semua bilangan ganjil dari 1 hingga N, di mana N adalah bilangan bulat positif yang dimasukkan oleh pengguna.

Cara Kerja: Program menggunakan fungsi rekursif untuk memeriksa setiap angka dari 1 hingga N. Jika angka tersebut adalah bilangan ganjil (diperiksa dengan operasi modulus), angka tersebut dicetak. Proses ini berlanjut hingga semua angka dari 1 hingga N telah diperiksa.

Latihan Soal 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

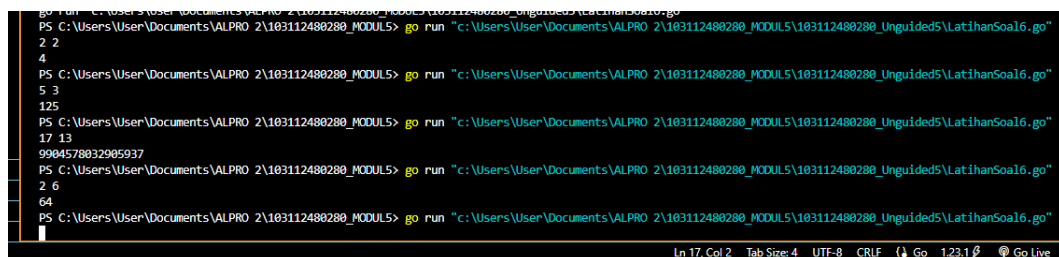
```
//Nama: Anggun Wahyu Widiyana (103112480280)
package main

import "fmt"

func pangkat(x int, 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))
}
```

Output:



```
PS C:\Users\User\Documents\ALPRO 2\103112480280\Modules\103112480280\UnGuided5\LatihanSoal6.go> go run "c:\Users\User\Documents\ALPRO 2\103112480280\Modules\103112480280\UnGuided5\LatihanSoal6.go"
2 2
4
PS C:\Users\User\Documents\ALPRO 2\103112480280\Modules\103112480280\UnGuided5\LatihanSoal6.go> go run "c:\Users\User\Documents\ALPRO 2\103112480280\Modules\103112480280\UnGuided5\LatihanSoal6.go"
5 3
125
PS C:\Users\User\Documents\ALPRO 2\103112480280\Modules\103112480280\UnGuided5\LatihanSoal6.go> go run "c:\Users\User\Documents\ALPRO 2\103112480280\Modules\103112480280\UnGuided5\LatihanSoal6.go"
17 13
9904578032905937
PS C:\Users\User\Documents\ALPRO 2\103112480280\Modules\103112480280\UnGuided5\LatihanSoal6.go> go run "c:\Users\User\Documents\ALPRO 2\103112480280\Modules\103112480280\UnGuided5\LatihanSoal6.go"
2 6
64
PS C:\Users\User\Documents\ALPRO 2\103112480280\Modules\103112480280\UnGuided5\LatihanSoal6.go>
```

Deskripsikan Program

Program ini dirancang untuk menghitung hasil dari bilangan x dipangkatkan y menggunakan fungsi rekursif. Pengguna memasukkan dua bilangan bulat positif x (basis) dan y (eksponen), lalu program menghasilkan hasil pangkatnya.

Cara Kerja: Fungsi rekursif memanfaatkan sifat matematika bahwa:

$x^y = x \times x^{y-1}$ Kondisi penghentian terjadi ketika eksponen (y) bernilai 0, di mana hasilnya adalah 1. Fungsi terus memanggil dirinya sendiri dengan mengurangi nilai

eksponen (y) hingga mencapai 0. Program mencetak hasil akhir setelah semua pemanggilan rekursif selesai.

IV. KESIMPULAN

Laporan ini membahas penerapan algoritma pemrograman dengan pendekatan rekursif menggunakan bahasa Go. Pendekatan rekursif digunakan untuk menyelesaikan berbagai masalah komputasi dasar, seperti pencetakan pola bilangan, pencarian faktor bilangan, perhitungan pangkat, dan lainnya. Rekursi merupakan konsep penting dalam algoritma pemrograman yang memungkinkan sebuah fungsi memanggil dirinya sendiri untuk menyelesaikan masalah secara bertahap. Pendekatan rekursif memberikan solusi yang bagus dan terstruktur untuk berbagai masalah komputasi dasar. Rekursi membantu menyederhanakan logika program dengan memanfaatkan sifat berulang dari masalah tersebut. Namun, perlu diperhatikan bahwa rekursi memiliki batasan seperti potensi penggunaan memori besar pada pemanggilan fungsi berulang dengan input yang besar.

V. REFERENSI

School of Computing. *Modul Praktikum 5 – Rekursif Algoritma dan Pemrograman 2 SI Informatika*. 2025

Munir, Rinaldi. *Algoritma dan Pemrograman dalam Bahasa Go*. Bandung: Informatika. 2020

Cormen, T.H., Leiserson, C.E., Rivest, R.L., & Stein, C. *Introduction to Algorithms*. The MIT Press. 2009