

**LAPORAN PRAKTIKUM
ALGORITMA PEMROGRAMAN 2
MODUL 5
REKURSIF**



Oleh:

MUHAMMAD AGHA ZULFADHLI

2311102015

S1-IF11-02

**S1 TEKNIK INFORMATIKA
INSTITUT TEKNOLOGI TELKOM PURWOKERTO
2024**

I. DASAR TEORI

Rekursif adalah sebuah teknik di pemrograman, di mana sebuah fungsi memanggil dirinya sendiri. Mirip dengan loop (pengulangan), yang kedua tujuannya memanggil aksi yang sama berkali-kali.

	Notasi Algoritma	Notasi dalam bahasa GO
1	procedure cetak(in x:integer)	func cetak(x int){
2	algoritma	fmt.Println(x)
3	output(x)	cetak(x+1)
4	cetak(x+1)	}
5	endprocedure	

Catatan:

- Teknik rekursif ini merupakan salah satu alternatif untuk mengganti struktur kontrol perulangan dengan memanfaatkan subprogram (bisa fungsi ataupun prosedur).
- Untuk menghentikan proses rekursif digunakan percabangan (if then).
- Base-case adalah kondisi proses rekursif berhenti. Base-case merupakan hal terpenting dan pertama yang harus diketahui ketika akan membuat program rekursif. Mustahil membuat program rekursif tanpa mengetahui base-case terlebih dahulu.
- Recursive-case adalah kondisi dimana proses pemanggilan dirinya sendiri dilakukan. Kondisi recursive-case adalah komplemen atau negasi dari base-case.
- Setiap algoritma rekursif selalu memiliki padanan dalam bentuk algoritma iteratif.

Algoritma rekursif terdiri dari dua komponen utama:

- Base-case (Basis), yaitu bagian untuk menghentikan proses rekursif dan menjadi komponen terpenting di dalam sebuah rekursif.
- Recursive-case, yaitu bagian pemanggilan subprogramnya.

II. GUIDED

1. Guided 1

Source code

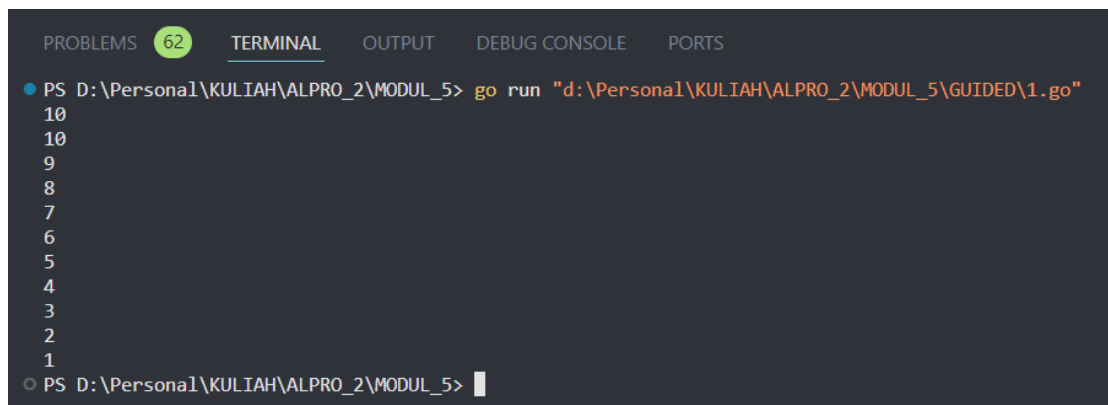
```
package main

import "fmt"

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

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

Screenshoot program



```
PROBLEMS 62 TERMINAL OUTPUT DEBUG CONSOLE PORTS
PS D:\Personal\KULIAH\ALPRO_2\MODUL_5> go run "d:\Personal\KULIAH\ALPRO_2\MODUL_5\GUIDED\1.go"
10
10
9
8
7
6
5
4
3
2
1
PS D:\Personal\KULIAH\ALPRO_2\MODUL_5> █
```

Deskripsi program

Kode ini adalah program rekursif yang mencetak bilangan dari nilai n yang dimasukkan pengguna hingga 1, satu per baris. Fungsi utama main meminta input dari pengguna untuk nilai n, kemudian memanggil fungsi baris dengan parameter tersebut. Fungsi baris mengecek apakah nilai bilangan sama dengan 1, dan jika ya,

mencetak 1. Jika tidak, fungsi mencetak nilai bilangan saat ini dan memanggil dirinya sendiri dengan mengurangi nilai tersebut sebesar 1, sehingga mencetak nilai dari n hingga 1 secara menurun.

2. Guided 2

Source code

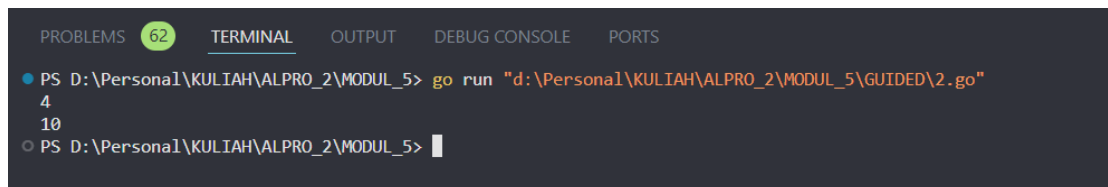
```
package main

import "fmt"

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

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

Screenshoot program



```
PROBLEMS 62 TERMINAL OUTPUT DEBUG CONSOLE PORTS
PS D:\Personal\KULIAH\ALPRO_2\MODUL_5> go run "d:\Personal\KULIAH\ALPRO_2\MODUL_5\GUIDED\2.go"
4
10
PS D:\Personal\KULIAH\ALPRO_2\MODUL_5> |
```

Deskripsi program

Kode ini adalah program rekursif dalam bahasa Go yang menghitung penjumlahan bilangan dari 1 hingga n, di mana n merupakan nilai yang dimasukkan oleh pengguna. Fungsi penjumlahan menerima satu parameter n dan memeriksa apakah nilai n sama dengan 1. Jika ya, fungsi mengembalikan nilai 1 sebagai dasar rekursi. Jika tidak, fungsi menambahkan n dengan hasil dari pemanggilan rekursif fungsi penjumlahan(n-1), sehingga secara bertahap menjumlahkan semua bilangan dari n hingga 1. Hasil akhir adalah total penjumlahan dari 1 sampai n, yang kemudian dicetak oleh fungsi main.

III. UNGUIDED

1. UNGUIDED 1

Source code

```
package main

import "fmt"

func rekursif(prev1, prev2, max, iterasi int) {
    if iterasi <= max {
        fmt.Print(prev2+prev1, " ")
        prev1 = prev2 + prev1
        rekursif(prev2, prev1, max, iterasi+1)
    }
}

func main() {
    var prev2, prev1 int = 0, 1
    var max int
    fmt.Print("Jumlah Iterasi : ")
    fmt.Scan(&max)
    rekursif(prev1, prev2, max, 1)
}
```

Screenshoot program



```
PS D:\Personal\KULIAH\ALPRO_2\MODUL_5> go run "d:\Personal\KULIAH\ALPRO_2\MODUL_5\UNGUIDED\1.go"
Jumlah Iterasi : 20
1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597 2584 4181 6765
PS D:\Personal\KULIAH\ALPRO_2\MODUL_5>
```

Deskripsi program

Kode menjalankan fungsi rekursif yang menghasilkan deret Fibonacci. Fungsi utama rekursif menerima empat parameter: dua angka sebelumnya dalam deret Fibonacci (prev1 dan prev2), batas maksimal iterasi (max), dan nomor iterasi saat ini (iterasi). Di dalam fungsi tersebut, deret Fibonacci dihitung dengan menjumlahkan dua angka sebelumnya (prev1 + prev2) dan mencetak hasilnya. Kemudian, fungsi dipanggil kembali (rekursif) dengan memperbarui nilai prev1 menjadi prev2 + prev1, sampai iterasi mencapai batas yang ditentukan (max). Program ini dimulai dari nilai awal 0 dan 1 untuk prev2 dan prev1 masing-masing,

yang diinisialisasi di main. Setelah pengguna memasukkan jumlah iterasi, program akan mencetak deret Fibonacci hingga jumlah yang diminta.

2. UNGUIDED 2

Source code

```
package main

import "fmt"

func cetak(n int) {
    if n > 0 {
        fmt.Print("*")
        cetak(n - 1)
    }
}

func rekursif(n int) {
    if n == 1 {
        fmt.Println("*")
    } else {
        rekursif(n - 1)
        cetak(n)
        fmt.Println()
    }
}

func main() {
    var a int
    fmt.Scan(&a)
    rekursif(a)
}
```

Screenshoot program



```
PROBLEMS 62 TERMINAL OUTPUT DEBUG CONSOLE PORTS
PS D:\Persona1\KULIAH\ALPRO_2\MODUL_5> go run "d:\Persona1\KULIAH\ALPRO_2\MODUL_5\UNGUIDED\2.go"
5
*
**
***
****
*****
PS D:\Persona1\KULIAH\ALPRO_2\MODUL_5>
```

Deskripsi program

Kode menjalankan fungsi rekursif yang mencetak pola bintang menggunakan dua fungsi rekursif: cetak dan rekursif. Fungsi rekursif menerima satu parameter *n*, yang menentukan jumlah baris yang akan dicetak. Jika *n* sama dengan 1, fungsi mencetak satu bintang di baris baru. Jika *n* lebih besar dari 1, fungsi memanggil dirinya sendiri dengan mengurangi *n* sebesar 1 untuk mencetak pola untuk baris sebelumnya, kemudian memanggil fungsi cetak(*n*) untuk mencetak *n* bintang pada baris saat ini, diikuti dengan mencetak baris baru. Fungsi cetak bertanggung jawab untuk mencetak bintang secara rekursif hingga jumlah bintang yang diminta tercapai. Program dimulai dengan meminta input dari pengguna untuk jumlah baris, kemudian mencetak pola bintang berbentuk segitiga dengan jumlah baris sesuai input.

3. UNGUIDED 3

Source code

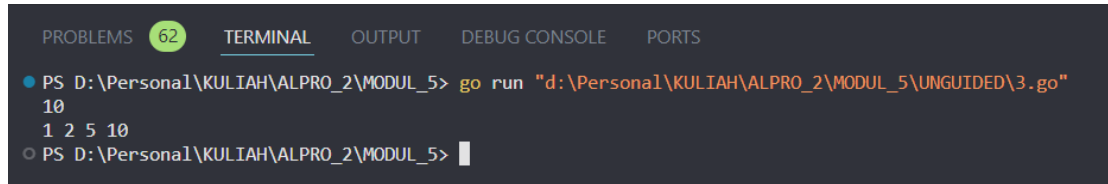
```
package main

import "fmt"

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

func main() {
    var a int
    fmt.Scan(&a)
    rekursif(a, 1)
}
```

Screenshoot program



```
PROBLEMS 62 TERMINAL OUTPUT DEBUG CONSOLE PORTS
● PS D:\Personal\KULIAH\ALPRO_2\MODUL_5> go run "d:\Personal\KULIAH\ALPRO_2\MODUL_5\UNGUIDED\3.go"
10
1 2 5 10
○ PS D:\Personal\KULIAH\ALPRO_2\MODUL_5> |
```

Deskripsi program

Kode menjalankan fungsi rekursif yang bertujuan untuk mencetak semua faktor dari suatu bilangan. Fungsi rekursif menerima dua parameter: bilangan n yang faktornya akan dicari dan $current$, yang digunakan untuk mengecek setiap angka dari 1 hingga n . Jika n habis dibagi oleh $current$ (artinya $current$ adalah faktor dari n), program akan mencetak angka tersebut. Fungsi kemudian dipanggil kembali secara rekursif dengan menambah nilai $current$ hingga mencapai nilai n . Program dimulai dengan meminta input dari pengguna, kemudian memanggil fungsi rekursif dengan parameter awal n (input) dan $current$ mulai dari 1.

4. UNGUIDED 4

Source code

```
package main

import "fmt"

var reverse bool

func rekursif(n int) {
    if n > 1 {
        fmt.Print(n, " ")
        rekursif(n - 1)
    }
    fmt.Print(n, " ")
}

func main() {
    var a int
    fmt.Scan(&a)
    rekursif(a)
}
```


Screenshoot program



```
PROBLEMS 62 TERMINAL OUTPUT DEBUG CONSOLE PORTS
PS D:\Personal\KULIAH\ALPRO_2\MODUL_5> go run "d:\Personal\KULIAH\ALPRO_2\MODUL_5\UNGUIDED\4.go"
10
10 9 8 7 6 5 4 3 2 1 2 3 4 5 6 7 8 9 10
PS D:\Personal\KULIAH\ALPRO_2\MODUL_5>
```

Deskripsi program

Kode menjalankan fungsi rekursi yang mencetak bilangan dari nilai input n hingga 1 secara menurun, kemudian mencetaknya kembali secara menaik hingga n . Fungsi rekursif memeriksa apakah nilai n lebih besar dari 1, jika ya, maka mencetak nilai n dan memanggil fungsi secara rekursif dengan $n-1$, sehingga menghasilkan urutan menurun. Setelah mencapai nilai 1, fungsi melanjutkan mencetak kembali nilai n saat kembali dari setiap pemanggilan rekursif, sehingga menciptakan pola cetakan yang menurun dan kemudian menaik. Program ini meminta input dari pengguna dan memanggil fungsi rekursif dengan input tersebut.

5. UNGUIDED 5

Source code

```
package main

import "fmt"

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

func main() {
    var a int
    fmt.Scan(&a)
    rekursif(a, 1)
}
```

Screenshoot program



```
PROBLEMS 62 TERMINAL OUTPUT DEBUG CONSOLE PORTS
PS D:\Personal\KULIAH\ALPRO_2\MODUL_5> go run "d:\Personal\KULIAH\ALPRO_2\MODUL_5\UNGUIDED\5.go"
23
1 3 5 7 9 11 13 15 17 19 21 23
PS D:\Personal\KULIAH\ALPRO_2\MODUL_5>
```

Deskripsi program

Kode menjalankan fungsi rekursif yang mencetak semua bilangan ganjil dari 1 hingga n, di mana n merupakan nilai yang dimasukkan oleh pengguna. Fungsi rekursif menerima dua parameter: n (batas atas) dan current (bilangan yang akan dicek). Jika current merupakan bilangan ganjil (dicek dengan $\text{current} \% 2 \neq 0$), maka angka tersebut dicetak. Fungsi kemudian memanggil dirinya sendiri dengan menaikkan current sebesar 1 hingga current mencapai nilai n. Program dimulai dengan meminta input dari pengguna, lalu mencetak bilangan ganjil antara 1 sampai nilai input tersebut.

6. UNGUIDED 6

Source code

```
package main

import "fmt"

func rekursif(x, y, current int) int {
    hasil := 1
    if current <= y {
        hasil *= (x * (rekursif(x, y, current+1)))
    }
    return hasil
}

func main() {
    var x, y int
    fmt.Scan(&x, &y)
    fmt.Println(rekursif(x, y, 1))
}
```

Screenshoot program



```
PROBLEMS 62 TERMINAL OUTPUT DEBUG CONSOLE PORTS
PS D:\Personal\KULIAH\ALPRO_2\MODUL_5> go run "d:\Personal\KULIAH\ALPRO_2\MODUL_5\UNGUIDED\6.go"
5 3
125
PS D:\Personal\KULIAH\ALPRO_2\MODUL_5> 
```

Deskripsi program

Kode menjalankan fungsi rekursif yang menghitung hasil perpangkatan dari dua bilangan, di mana x adalah bilangan pokok dan y adalah pangkatnya. Fungsi rekursif menerima tiga parameter: x (bilangan pokok), y (pangkat), dan $current$ (untuk melacak iterasi saat ini). Pada setiap pemanggilan, jika $current$ masih lebih kecil atau sama dengan y , maka fungsi mengalikan x dengan hasil dari pemanggilan rekursif berikutnya, menaikkan nilai $current$ sampai mencapai nilai y . Hasil akhir dari seluruh rekursi adalah x pangkat y . Program ini meminta dua input dari pengguna, lalu memanggil fungsi rekursif untuk mencetak hasil perpangkatan tersebut.