

LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2

MODUL V
REKURSIF



Disusun Oleh:

Muhammad Djulianoor / 2311102253

Kelas

IF-11-06

Dosen Pengampu:

Abednego Dwi Septiadi

PROGRAM STUDI S1 TEKNIK INFORMATIKA
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2024

I. DASAR TEORI

Dasar Teori

Fungsi rekursif ialah proses fungsi yang mana fungsi tersebut memanggil dirinya sendiri secara langsung maupun tidak langsung. Fungsi rekursif memanggil dirinya sendiri secara berulang dalam proses pengolahan data atau pemanggilan fungsi. Pada beberapa kasus fungsi rekursif bisa lebih mudah dipahami dan sederhana daripada dengan solusi iteratif.

Contoh penggunaan rekursif

```
package main

import "fmt"

func ganjil(n int) {
    if n == 1 {
        fmt.Print(1, " ")
    } else {
        if n%2 == 0 {
            ganjil(n - 1)
        } else {
            ganjil(n - 2)
            fmt.Print(n, " ")
        }
    }
}

func main() {
    var n int

    fmt.Scan(&n)

    ganjil(n)
}
```

Contoh program di atas adalah program yang akan menampilkan deret bilangan ganjil dari 1 hingga n. Prosedur ganjil memiliki parameter n yang mana n akan menjadi *input*. Kemudian terdapat percabangan untuk memeriksa apakah bilangan yang di-*input*-kan adalah 1 atau bukan. Jika

bukan maka *else* akan dieksekusi. Pada percabangan *else* terdapat percabangan lagi untuk memeriksa apakah bilangan *input* adalah genap atau ganjil. Jika kemudian bilangan *input* adalah genap maka akan dilakukan pengurangan 1 dengan cara memanggil prosedur itu sendiri: ganjil($n-1$). Jika bilangan *input* adalah ganjil, maka akan dilakukan pengurangan 2 dengan cara memanggil prosedur itu sendiri: ganjil($n-2$). Kemudian program akan terus memeriksa bilangan hingga bilangan itu menjadi 1 dan berhenti.

Itulah contoh dari program yang menggunakan fungsi atau prosedur rekursif untuk melakukan perhitungan. Dengan menggunakan rekursif, tidak perlu lagi menggunakan perulangan.

II. GUIDED

1. Soal Studi Case

Source code

```
package main

import "fmt"

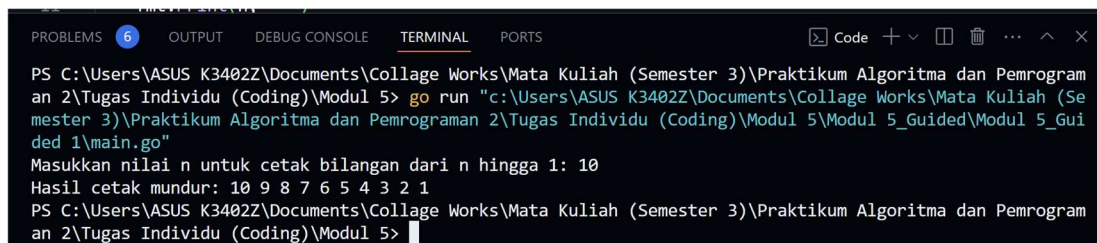
func cetakMundur(n int) {
    if n == 1 {
        fmt.Println(n)
        return
    }

    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)
}
```

Screenshoot Output



```
PS C:\Users\ASUS K3402Z\Documents\Collage Works\Mata Kuliah (Semester 3)\Praktikum Algoritma dan Pemrograman 2\Tugas Individu (Coding)\Modul 5> go run "c:\Users\ASUS K3402Z\Documents\Collage Works\Mata Kuliah (Semester 3)\Praktikum Algoritma dan Pemrograman 2\Tugas Individu (Coding)\Modul 5\Modul 5_Guided\Modul 5_Guided 1\main.go"
Masukkan nilai n untuk cetak bilangan dari n hingga 1: 10
Hasil cetak mundur: 10 9 8 7 6 5 4 3 2 1
PS C:\Users\ASUS K3402Z\Documents\Collage Works\Mata Kuliah (Semester 3)\Praktikum Algoritma dan Pemrograman 2\Tugas Individu (Coding)\Modul 5>
```

Deskripsi Program

Program di atas adalah program yang mencetak mundur bilangan dari bilangan n hingga bilangan terkecil, yaitu 1. Program ini menggunakan fungsi rekursif untuk memproses perulangan. Pada prosedur rekursif cetakMundur terdapat percabangan *if* untuk mengetahui apakah bilangan yang di-*input*-kan adalah bilangan 1 atau tidak. Jika tidak maka tidak akan memproses apa yang ada pada *statement if*. Kemudian terdapat *syntax* untuk mengeluarkan bilangan, juga terdapat prosedur itu sendiri, yaitu cetakMundur. Pada rekursif cetakMundur, dilakukan pengurangan dengan tujuan untuk mencetak mundur bilangan dari n hingga 1.

2. Soal Studi Case

Source Code

```
package main

import "fmt"

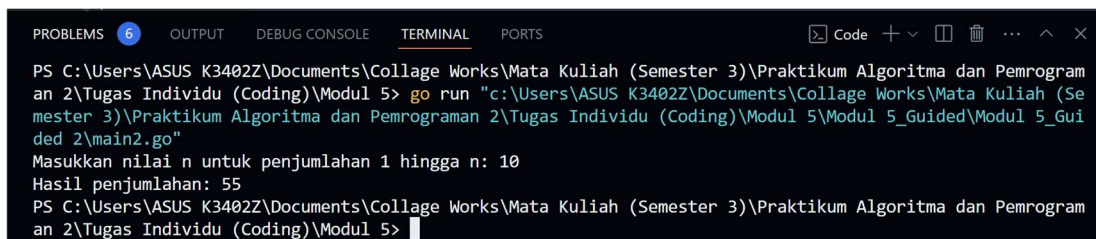
func jumlahRekursif(n int) int {
    if n == 1 {
        return 1
    }

    return n + jumlahRekursif(n-1)
}

func main() {
    var n int

    fmt.Print("Masukkan nilai n untuk penjumlahan 1 hingga n: ")
    fmt.Scanln(&n)
    fmt.Println("Hasil penjumlahan:", jumlahRekursif(n))
}
```

Screenshoot Output



```
PROBLEMS 6 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\ASUS K3402Z\Documents\Collage Works\Mata Kuliah (Semester 3)\Praktikum Algoritma dan Pemrograman 2\Tugas Individu (Coding)\Modul 5> go run "c:\Users\ASUS K3402Z\Documents\Collage Works\Mata Kuliah (Semester 3)\Praktikum Algoritma dan Pemrograman 2\Tugas Individu (Coding)\Modul 5\Modul 5_Guided\Modul 5_Guided 2\main2.go"
Masukkan nilai n untuk penjumlahan 1 hingga n: 10
Hasil penjumlahan: 55
PS C:\Users\ASUS K3402Z\Documents\Collage Works\Mata Kuliah (Semester 3)\Praktikum Algoritma dan Pemrograman 2\Tugas Individu (Coding)\Modul 5> 
```

Deskripsi Program

Penjumlahan di atas dilakukan dengan cara rekursif dengan menggunakan fungsi rekursif `jumlahRekursif`. Fungsi `jumlahRekursif` memiliki parameter `n`, yang mana `n` adalah *input* dari *user*. Di dalam rekursif terdapat percabangan *if* untuk memeriksa apakah *input*-an dari *user* adalah 1 atau bukan. Percabangan *if* akan dieksekusi jika *input*-an dari *user* adalah 1. Jika *input*-an dari *user* bukan 1, maka akan dilakukan pengembalian nilai $n + \text{jumlahRekursif}(n-1)$.

3. Soal Studi Case

Source Code

```
package main

import "fmt"


func pangkatDua(n int) int {
    if n == 0 {
        return 1
    }

    return 2 * pangkatDua(n-1)
}

func main() {
    var n int

    fmt.Print("Masukkan nilai n untuk mencari 2 pangkat n: ")
    fmt.Scanln(&n)
    fmt.Println("Hasil 2 pangkat", n, ":", pangkatDua(n))
}
```

Screenshoot Output



```
PS C:\Users\ASUS K3402Z\Documents\Collage Works\Mata Kuliah (Semester 3)\Praktikum Algoritma dan Pemrograman 2\Tugas Individu (Coding)\Modul 5>
go run "c:\Users\ASUS K3402Z\Documents\Collage Works\Mata Kuliah (Semester 3)\Praktikum Algoritma dan Pemrograman 2\Tugas Individu (Coding)\Modul 5\Modul 5_Guided\Modul 5_Guided 3\main3.go"
Masukkan nilai n untuk mencari 2 pangkat n: 4
Hasil 2 pangkat 4 : 16
PS C:\Users\ASUS K3402Z\Documents\Collage Works\Mata Kuliah (Semester 3)\Praktikum Algoritma dan Pemrograman 2\Tugas Individu (Coding)\Modul 5> |
```


Deskripsi Program

Pada program di atas dilakukan perhitungan pangkat. 2 adalah bilangan konstan yang akan dipangkatkan berdasarkan *input* dari *user*. Pada fungsi rekursif pangkatDua terdapat parameter untuk menerima *input*. Di dalam fungsi tersebut terdapat percabangan untuk memeriksa apakah *input* yang dimasukkan adalah angka 1 atau bukan. Jika angka 1, maka akan dilakukan *return* 1. Pada fungsi tersebut dilakukan perhitungan dengan cara *return* $2 * \text{pangkatDua}(n-1)$.

4. Soal Studi Case

Source Code

```
package main

import "fmt"

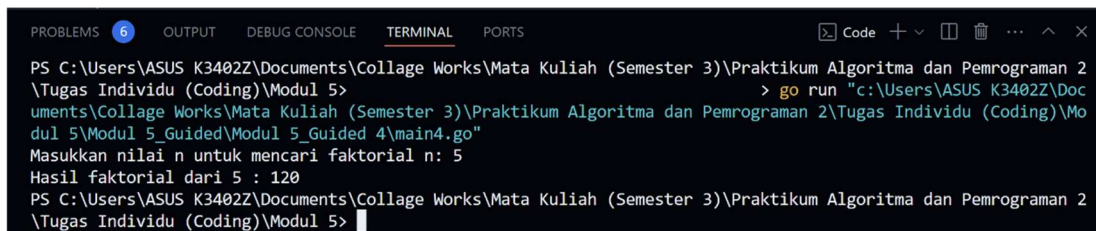
func faktorial(n int) int {
    if n == 0 || n == 1 {
        return 1
    }

    return n * faktorial(n-1)
}

func main() {
    var n int

    fmt.Print("Masukkan nilai n untuk mencari faktorial n: ")
    fmt.Scanln(&n)
    fmt.Println("Hasil faktorial dari", n, ":", faktorial(n))
}
```

Screenshoot Output



```
PS C:\Users\ASUS K3402Z\Documents\Collage Works\Mata Kuliah (Semester 3)\Praktikum Algoritma dan Pemrograman 2
\tugas Individu (Coding)\Modul 5> go run "c:\Users\ASUS K3402Z\Documents\Collage Works\Mata Kuliah (Semester 3)\Praktikum Algoritma dan Pemrograman 2\tugas Individu (Coding)\Modul 5\Modul 5_Guided\Modul 5_Guided 4\main4.go"
Masukkan nilai n untuk mencari faktorial n: 5
Hasil faktorial dari 5 : 120
PS C:\Users\ASUS K3402Z\Documents\Collage Works\Mata Kuliah (Semester 3)\Praktikum Algoritma dan Pemrograman 2
\tugas Individu (Coding)\Modul 5> █
```

Deskripsi Program

Program faktorial di atas dilakukan dengan cara rekursif. Terdapat fungsi rekursif dengan nama faktorial yang memiliki parameter n untuk menerima *input*. Terdapat percabangan di dalam fungsi tersebut untuk memeriksa apakah input berupa bilangan 0 atau 1, jika iya maka *statement if* akan dijalankan oleh program, jika tidak maka akan dilewati terlebih dahulu. Pada program terdapat proses perhitungan faktorial jika *input* bukan 0 atau 1, dengan cara *return* $n * \text{faktorial}(n - 1)$.

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 $S_n = S_{n-1} + S_{n-2}$. 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 |

Source code

```
package main

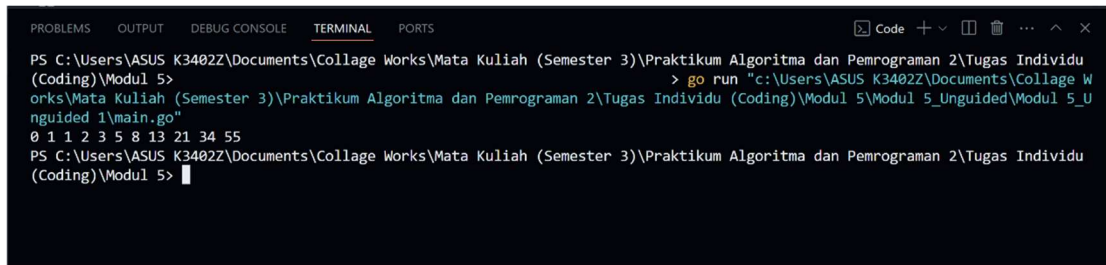
import "fmt"

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

func printFibonacci(n, current int) {
    if current > n {
        return
    }
    fmt.Printf("%d ", fibonacci(current))
    printFibonacci(n, current+1)
}

func main() {
    printFibonacci(10, 0)
    fmt.Println()
}
```

Screenshoot Output



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\ASUS K3402Z\Documents\Collage Works\Mata Kuliah (Semester 3)\Praktikum Algoritma dan Pemrograman 2\Tugas Individu (Coding)\Modul 5>
PS C:\Users\ASUS K3402Z\Documents\Collage Works\Mata Kuliah (Semester 3)\Praktikum Algoritma dan Pemrograman 2\Tugas Individu (Coding)\Modul 5> go run "c:\Users\ASUS K3402Z\Documents\Collage Works\Mata Kuliah (Semester 3)\Praktikum Algoritma dan Pemrograman 2\Tugas Individu (Coding)\Modul 5\Modul 5_Unguided 1\main.go"
0 1 1 2 3 5 8 13 21 34 55
PS C:\Users\ASUS K3402Z\Documents\Collage Works\Mata Kuliah (Semester 3)\Praktikum Algoritma dan Pemrograman 2\Tugas Individu (Coding)\Modul 5>
```

Deskripsi Program

Program fibonacci di atas mengeluarkan *output* dari suku 0 hingga suku 10. Program fibonacci ini menggunakan cara rekursif untuk melakukan perhitungan. Pada fungsi rekursif fibonacci terdapat percabangan *if* untuk memeriksa apakah n adalah 1 atau di bawah 1, jika iya, maka akan dilakukan pengembalian pada *input* itu sendiri. Jika tidak, maka akan dilewati terlebih dahulu dan dilakukan perhitungan dengan cara rekursif, *return* fibonacci($n-1$) + fibonacci($n-2$). Setiap *input* yang bukan 1 atau kurang dari 1 akan dikurangi 1 dan juga dikurangi 2, lalu hasil pengurangan tadi ditambahkan. Dengan cara tersebut program dapat melakukan perhitungan fibonacci.

2. Soal Studi Case

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

Contoh masukan dan keluaran:

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

Source Code

```
package main

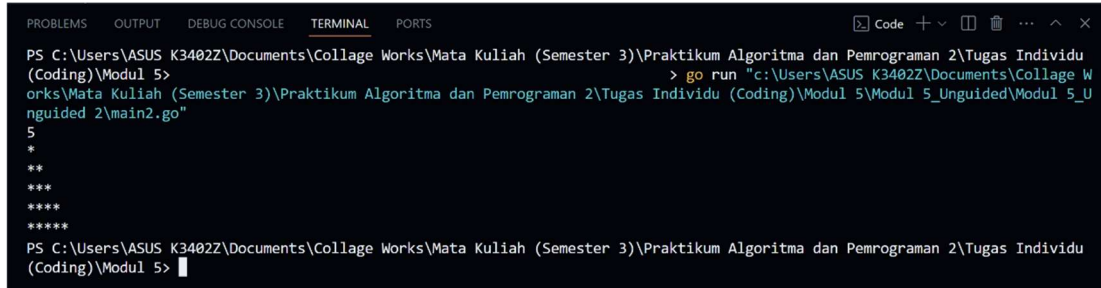
import "fmt"

func pola(i, N int, b string) {
    if i <= N {
        b = "*" + b
        fmt.Println(b)
        pola(i+1, N, b)
    }
}

func main() {
    var N int
    var b string
```

```
    fmt.Scanln(&N)
    pola(1, N, b)
}
```

Screenshoot Output



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\ASUS K3402Z\Documents\Collage Works\Mata Kuliah (Semester 3)\Praktikum Algoritma dan Pemrograman 2\Tugas Individu (Coding)\Modul 5> go run "c:\Users\ASUS K3402Z\Documents\Collage Works\Mata Kuliah (Semester 3)\Praktikum Algoritma dan Pemrograman 2\Tugas Individu (Coding)\Modul 5\Modul 5_Unguided 2\main2.go"
5
*
**
***
****
*****
PS C:\Users\ASUS K3402Z\Documents\Collage Works\Mata Kuliah (Semester 3)\Praktikum Algoritma dan Pemrograman 2\Tugas Individu (Coding)\Modul 5>
```

Deskripsi Program

Program pola bintang di atas dilakukan dengan cara rekursif. Prosedur pola memiliki 3 parameter, yaitu *i*, *N*, dan *b*. Terdapat percabangan untuk memeriksa apakah *i* lebih kecil atau sama dengan *N*. Pada program pola ini, nilai *i* adalah konstan 1. Pada *statement* percabangan terdapat variabel *b* yang di dalamnya terdapat penambahan string untuk melakukan penambahan pola bintang berdasarkan *input*. Kemudian variabel *b* di-*print* atau dikeluarkan, lalu prosedur rekursif pola dipanggil yang mana di dalamnya berupa variabel *i* yang ditambahkan dengan angka 1, kemudian memasukkan variabel *N*, dan variabel *b*.

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.

Masukan terdiri dari sebuah bilangan bulat positif N.

Keluaran terdiri dari barisan bilangan yang menjadi faktor dari N.

Contoh masukan dan keluaran:

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

Source Code

```
package main

import "fmt"

func faktor(n, i int) {
    if i <= n {
        if n%i == 0 {
            fmt.Print(i, " ")
            faktor(n, i+1)
        } else {
            faktor(n, i+1)
        }
    }
}

func main() {
    var n, i int

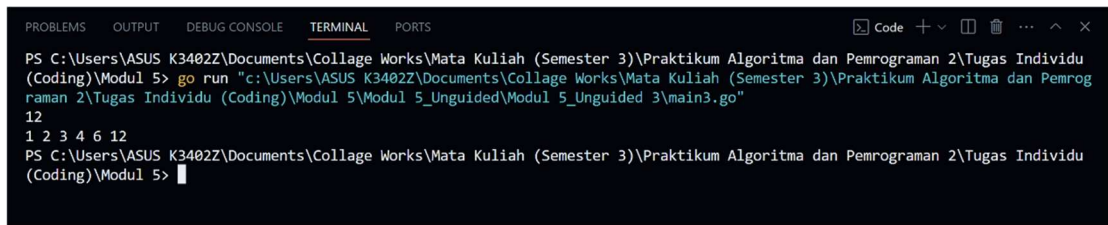
    i = 1

    fmt.Scanln(&n)
```



```
faktor(n, i)
}
```

Screenshoot Output



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
PS C:\Users\ASUS K34022\Documents\Collage Works\Mata Kuliah (Semester 3)\Praktikum Algoritma dan Pemrograman 2\Tugas Individu (Coding)\Modul 5> go run "c:\Users\ASUS K34022\Documents\Collage Works\Mata Kuliah (Semester 3)\Praktikum Algoritma dan Pemrograman 2\Tugas Individu (Coding)\Modul 5\Modul 5_Unguided\Modul 5_Unguided 3\main3.go"
12
1 2 3 4 6 12
PS C:\Users\ASUS K34022\Documents\Collage Works\Mata Kuliah (Semester 3)\Praktikum Algoritma dan Pemrograman 2\Tugas Individu (Coding)\Modul 5>
```

Deskripsi Program

Program faktor di atas menggunakan percabangan di dalam percabangan untuk melakukan perhitungan. Pada prosedur rekrusif faktor, terdapat percabangan yang pertama adalah untuk memeriksa apakah i kurang dari atau sama dengan n . Kemudian di dalam percabangan *if* tersebut, terdapat percabangan *if* lagi untuk memeriksa apakah n jika dibagi i akan habis. Jika iya, maka *statement if* akan dijalankan. Jika tidak, maka *statement else* yang akan dijalankan.

4. Soal Studi Case

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 |

Source Code

```
package main

import "fmt"

func printNtoN(n, current int) {
    fmt.Printf("%d ", current)
    if current == 1 {
        return
    }
    printNtoN(n, current-1)
    fmt.Printf("%d ", current)
}

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

    fmt.Print("Keluaran: ")
    printNtoN(n, n)
    fmt.Println()
}
```

Screenshoot Output



```
PS C:\Users\ASUS K3402Z\Documents\Collage Works\Mata Kuliah (Semester 3)\Praktikum Algoritma dan Pemrograman 2\Tugas Individu (Coding)\Modul 5>
go run "c:\Users\ASUS K3402Z\Documents\Collage Works\Mata Kuliah (Semester 3)\Praktikum Algoritma dan Pemrograman 2\Tugas Individu (Coding)\Modul 5\Modul 5 Unguided\Modul 5 Unguided 4\main4.go"
Masukkan bilangan bulat positif N: 5
Keluaran: 5 4 3 2 1 2 3 4 5
PS C:\Users\ASUS K3402Z\Documents\Collage Works\Mata Kuliah (Semester 3)\Praktikum Algoritma dan Pemrograman 2\Tugas Individu (Coding)\Modul 5>
```

Deskripsi Program

Untuk membuat bilangan dari N hingga 1 dan kembali ke N dilakukan cara rekursif untuk melakukan perulangan. Pada prosedur rekursif printNtoN terdapat 2 parameter, yaitu n dan current (current adalah n juga). Sebelum ke tahap percabangan, current akan dikeluarkan terlebih dahulu, kemudian akan diperiksa dipercabangan. Pada percabangan terdapat kondisi di mana current sama dengan 1 dan di *statement*-nya dilakukan pengembalian atau *return*. Lalu prosedur printNtoN dipanggil lagi yang di dalamnya terdapat variabel n yang menjadi parameter dan current yang dikurangi 1. Setelah itu, current akan dikeluarkan lagi.

5. Soal Studi Case

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 |

Source Code

```
package main

import "fmt"

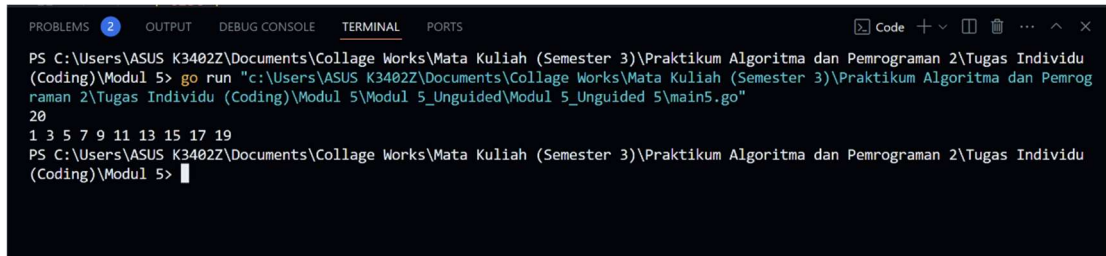
func ganjil(n int) {
    if n == 1 {
        fmt.Print(1, " ")
    } else {
        if n%2 == 0 {
            ganjil(n - 1)
        } else {
            ganjil(n - 2)
            fmt.Print(n, " ")
        }
    }
}

func main() {
    var n int

    fmt.Scan(&n)
```

```
ganjil(n)  
}
```

Screenshoot Output



```
PS C:\Users\ASUS K3402Z\Documents\Collage Works\Mata Kuliah (Semester 3)\Praktikum Algoritma dan Pemrograman 2\Tugas Individu (Coding)\Modul 5> go run "c:\Users\ASUS K3402Z\Documents\Collage Works\Mata Kuliah (Semester 3)\Praktikum Algoritma dan Pemrograman 2\Tugas Individu (Coding)\Modul 5\Modul 5_Unguided\Modul 5_Unguided 5\main5.go"  
20  
1 3 5 7 9 11 13 15 17 19  
PS C:\Users\ASUS K3402Z\Documents\Collage Works\Mata Kuliah (Semester 3)\Praktikum Algoritma dan Pemrograman 2\Tugas Individu (Coding)\Modul 5> █
```

Deskripsi Program

Program bilangan ganjil adalah program yang menampilkan *output* dari 1 hingga bilangan ganjil yang di-*input*-kan. Jika bilangan yang di-*input*-kan adalah bilangan genap, maka bilangan *input* tersebut tidak akan ditampilkan pada *output*, bilangan yang ditampilkan adalah bilangan ganjil sebelum bilangan genap yang telah di-*input*-kan. Pada program ini, tiap *input* yang genap akan dikurangi satu, sedangkan *input* ganjil akan dikurangi dua. Program ini menggunakan prosedur rekursif ganjil dengan parameter *n* yang berupa *input*. Terdapat percabangan pada prosedur tersebut untuk memeriksa apakah *input* adalah bilangan 1 atau bukan. Jika *input* adalah bilangan 1, maka program hanya akan menampilkan *output* 1 saja. Jika tidak, maka *else* akan diperiksa. Pada *else* terdapat percabangan lagi yang memeriksa, apakah bilangan *input* adalah bilangan genap atau ganjil. Jika *input* adalah bilangan genap, maka akan dikurangi 1. Jika *input* adalah bilangan ganjil, maka akan dikurangi 2. Kemudian program akan terus memeriksa hasil dari perhitungan hingga bilangan *n*.

6. Soal Studi Case

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 |

Source Code

```
package main

import "fmt"

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

func main() {
    var x, y, hasil int

    fmt.Scan(&x, &y)

    hasil = pangkat(x, y)
```

```
    fmt.Print(hasil)
}
```

Screenshoot Output



```
PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\ASUS K3402Z\Documents\Collage Works\Mata Kuliah (Semester 3)\Praktikum Algoritma dan Pemrograman 2\Tugas Individu (Coding)\Modul 5>
go run "c:\Users\ASUS K3402Z\Documents\Collage Works\Mata Kuliah (Semester 3)\Praktikum Algoritma dan Pemrograman 2\Tugas Individu (Coding)\Modul 5\Modul 5_Unguided\Modul 5_Unguided 6\main6.go"
4 4
256
PS C:\Users\ASUS K3402Z\Documents\Collage Works\Mata Kuliah (Semester 3)\Praktikum Algoritma dan Pemrograman 2\Tugas Individu (Coding)\Modul 5>
```

Deskripsi Program

Program perpangkatan di atas dilakukan dengan cara rekursif tanpa menggunakan import “math”. Fungsi rekursif pangkat memiliki 2 parameter. Variabel x adalah angka yang akan dipangkatkan, sedangkan y adalah angka pangkat dari x. Terdapat percabangan untuk melakukan perhitungan. Percabangan pertama memeriksa apakah pangkat 0 atau tidak. Jika pangkat adalah 0, maka akan dilakukan pengembalian nilai 1 atau return 1. Jika tidak maka akan dilakukan pengembalian nilai terhadap x dikalikan dengan fungsi rekursif: return x * pangkat(x, y-1). Program akan melakukan perhitungan hingga pangkat menjadi 0.