

**LAPORAN PRAKTIKUM PEMROGRAMAN
BERORIENTASI OBJEK**

MODUL V

REKURSIF



Oleh:

NAMA : AJI TRI PRASETYO

NIM : 2311102064

KELAS : IF 11 02

S1 TEKNIK INFORMATIKA

INSTITUT TEKNOLOGI TELKOM PURWOKERTO

2024

I. DASAR TEORI

Rekursif adalah teknik di mana sebuah fungsi memanggil dirinya sendiri untuk menyelesaikan masalah. Fungsi rekursif terdiri dari dua bagian utama: basis kasus (base case), yang menentukan kapan fungsi berhenti memanggil dirinya sendiri, dan rekursi (recursive case), yang membagi masalah menjadi sub-masalah yang lebih kecil. Teknik ini berguna untuk masalah yang bersifat berulang atau dapat disederhanakan, seperti perhitungan faktorial, bilangan Fibonacci, dan traversing struktur data seperti pohon atau graf.

	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 adalah salah satu cara alternatif untuk menggantikan struktur kontrol perulangan dengan memanfaatkan subprogram, baik itu fungsi atau prosedur.

- Proses rekursif dapat dihentikan dengan menggunakan percabangan (if then).
- Base-case adalah kondisi di mana proses rekursif berakhir. Base-case merupakan elemen terpenting yang harus dipahami sebelum membuat program rekursif, karena program rekursif tidak dapat dibuat tanpa mengetahui base-case terlebih dahulu.

Recursive-case adalah kondisi di mana fungsi memanggil dirinya sendiri. Kondisi ini merupakan pelengkap atau negasi dari base-case.

Setiap algoritma rekursif memiliki padanan dalam bentuk algoritma iteratif.

Algoritma rekursif terdiri dari dua komponen utama:

- Base-case (Basis), yaitu bagian yang berfungsi untuk menghentikan proses rekursif dan merupakan komponen terpenting dalam rekursi.
- Recursive-case, yaitu bagian yang melakukan pemanggilan subprogram.

II. GUIDED

Source Code + Screenshot hasil program beserta penjelasan

SOAL

NO 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)
    }
}
```

Output :

```
PS C:\Users\MyBook Z Series\Desktop\modul 5>
go run "d:\LAPRAK MODUL 5\GUIDED MODUL 5\GUIDED 1 MODUL 5.go"
5
5
4
3
2
1
PS C:\Users\MyBook Z Series\Desktop\modul 5> |
```

Penjelasan :

Program di atas menerima input berupa bilangan bulat `n` dari pengguna, kemudian mencetak bilangan tersebut secara berurutan menurun hingga mencapai 1 menggunakan fungsi rekursif bernama `baris`. Fungsi `baris(bilangan int)` akan mencetak nilai dari `bilangan`, lalu memanggil dirinya sendiri dengan parameter `bilangan - 1`. Rekursi ini akan berhenti ketika `bilangan` bernilai 1, di mana angka 1 dicetak dan rekursi dihentikan. Program ini menunjukkan cara penggunaan rekursi untuk mencetak bilangan dalam urutan menurun.

NO 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)
    }
}
```

Output :

```
PS C:\Users\MyBook Z Series\Desktop\modul 5>  
go run "d:\LAPRAK MODUL 5\GUIDED MODUL 5\guided 2 modul 5.go"  
10  
55  
PS C:\Users\MyBook Z Series\Desktop\modul 5> |
```

Penjelasan :

Program di atas menghitung total penjumlahan dari 1 sampai `n` menggunakan metode rekursi. Setelah pengguna memasukkan nilai `n`, fungsi rekursif `penjumlahan(n)` akan menambahkan nilai `n` dengan hasil penjumlahan dari `n-1`, dan proses ini berlanjut hingga mencapai nilai 1 sebagai kondisi dasar. Hasil akhir penjumlahan kemudian dicetak sebagai output

III. UNGUIDED

Source Code + Screenshot hasil program beserta penjelasan

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.

Source code :

```
1  package main
2
3  import "fmt"
4
5  func fibonacci(n int) int {
6      if n <= 1 {
7          return n
8      }
9      return fibonacci(n-1) + fibonacci(n-2)
10 }
11
12 func main() {
13     var n int
14     fmt.Print("Masukkan nilai n: ")
15     fmt.Scanln(&n)
16
17     for i := 0; i <= n; i++ {
18         fmt.Printf("Fibonacci(%d) = %d\n", i, fibonacci(i))
19     }
20 }
```

Output :

```
PS C:\Users\MyBook Z Series\Desktop\modul 5>
go run "c:\Users\MyBook Z Series\Desktop\modul 5\ungguide
Masukkan nilai n: 10
Fibonacci(0) = 0
Fibonacci(1) = 1
Fibonacci(2) = 1
Fibonacci(3) = 2
Fibonacci(4) = 3
Fibonacci(5) = 5
Fibonacci(6) = 8
Fibonacci(7) = 13
Fibonacci(8) = 21
Fibonacci(9) = 34
Fibonacci(10) = 55
PS C:\Users\MyBook Z Series\Desktop\modul 5> |
```

Penjelasan :

Program di atas menghitung dan menampilkan deret Fibonacci hingga nilai ke-`n` yang dimasukkan oleh pengguna. Fungsi rekursif `fibonacci(n)` digunakan untuk menghitung nilai Fibonacci, di mana jika `n` kurang dari atau sama dengan 1, fungsi mengembalikan `n`. Jika tidak, fungsi akan memanggil dirinya sendiri untuk menghitung dua angka Fibonacci sebelumnya (`n-1` dan `n-2`), dan menjumlahkannya. Pada fungsi `main`, program meminta pengguna untuk memasukkan nilai `n` dan kemudian mencetak hasil perhitungan Fibonacci untuk setiap nilai dari 0 hingga `n`.

SOAL 2

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

Source code :

```
1 package main
2
3 import (
4     "fmt"
5     "strings"
6 )
7
8 func main() {
9     var n int
10    fmt.Print("Masukkan nilai N: ")
11    fmt.Scan(&n)
12    fmt.Println(printbintang(n))
13 }
14
15 func printbintang(n int) string {
16     if n == 1 {
17         return "*"
18     }
19     return strings.Join([]string{printbintang(n-1), cetakBintang(n)}, "\n")
20 }
21
22 func cetakBintang(n int) string {
23     return strings.Repeat("*", n)
24 }
```

Output :

```
PS C:\Users\MyBook Z Series\Desktop\modul 5>
go run "c:\Users\MyBook Z Series\Desktop\modul 5\te
Masukkan nilai N: 5
*
**
***
****
*****
```

Penjelasan :

Program di atas meminta input berupa bilangan bulat `n` dari pengguna, kemudian mencetak pola bintang secara bertingkat dari 1 hingga `n` baris menggunakan fungsi rekursif `printbintang(n)`. Fungsi ini memanggil dirinya sendiri untuk mencetak baris sebelumnya sebelum menambahkan baris dengan jumlah bintang yang sesuai dengan nilai `n`, yang dihasilkan oleh fungsi `cetakBintang(n)`. Fungsi `cetakBintang(n)` menggunakan `strings.Repeat` untuk mencetak baris berisi bintang sebanyak nilai `n`.

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).

Source code :

```
1  package main
2
3  import (
4      "fmt"
5  )
6
7  func main() {
8      var n int
9      fmt.Scan(&n)
10     fmt.Printf("Faktor dari %d adalah:\n", n)
11     cetakFaktor(n)
12 }
13
14 func cetakFaktor(n int) {
15     for i := 1; i <= n; i++ {
16         if n%i == 0 {
17             fmt.Print(i, " ")
18         }
19     }
20 }
```

Output :

```
PS C:\Users\MyBook Z Series\Desktop\modul 5>
go run "c:\Users\MyBook Z Series\Desktop\modul 5\tempCodeR
12
Faktor dari 12 adalah:
1 2 3 4 6 12
PS C:\Users\MyBook Z Series\Desktop\modul 5> █
```

Penjelasan :

Program di atas menerima input bilangan bulat `n` dari pengguna dan mencetak faktor-faktornya. Pada fungsi `main()`, variabel `n` diambil dari input, lalu program mencetak teks yang menginformasikan bilangan yang akan dicari faktornya. Fungsi `cetakFaktor(n)` bertanggung jawab menghitung faktor-faktor dari `n` dengan cara memeriksa setiap bilangan dari 1 hingga `n` dan mencetak bilangan yang habis membagi `n` (artinya, `n % i == 0`).

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

Source code :

```
1  package main
2
3  import "fmt"
4
5  func main() {
6      var n int
7      fmt.Scan(&n)
8      baris(n)
9      fmt.Println()
10 }
11
12 func baris(bilangan int) {
13     if bilangan > 1 {
14         fmt.Print(bilangan, " ")
15         baris(bilangan - 1)
16         fmt.Print(bilangan, " ")
17     } else {
18         fmt.Print(1, " ")
19     }
20 }
```

Output :

```
PS C:\Users\MyBook Z Series\Desktop\modul 5>
go run "c:\Users\MyBook Z Series\Desktop\modul 5\tempCodeRun
5
5 4 3 2 1 2 3 4 5
PS C:\Users\MyBook Z Series\Desktop\modul 5> go run "c:\User
9
9 8 7 6 5 4 3 2 1 2 3 4 5 6 7 8 9
PS C:\Users\MyBook Z Series\Desktop\modul 5> █
```

Penjelasan :

Program di atas menggunakan fungsi rekursif untuk mencetak pola angka secara terbalik hingga 1, lalu kembali naik ke nilai semula. Pada fungsi `main`, pengguna memasukkan sebuah angka `n`, kemudian memanggil fungsi `baris(n)` yang bekerja secara rekursif. Fungsi `baris` memeriksa jika nilai bilangan lebih besar dari 1, maka ia mencetak angka tersebut, memanggil dirinya sendiri dengan mengurangi nilai bilangan satu per satu hingga mencapai 1, mencetak angka yang sama lagi setelah rekursi selesai. Pola ini menghasilkan tampilan angka yang menurun dan naik secara simetris di layar.

SOAL 5

Source code :

```
1 package main
2
3 import "fmt"
4
5 func main() {
6     var n int
7     fmt.Scan(&n)
8     barisGanjil(n)
9     fmt.Println()
10 }
11
12 func barisGanjil(bilangan int) {
13     if bilangan > 0 {
14         barisGanjil(bilangan - 1)
15         if bilangan%2 != 0 {
16             fmt.Print(bilangan, " ")
17         }
18     }
19 }
```

Output :

```
PS C:\Users\MyBook Z Series\Desktop\modul 5>
go run "c:\Users\MyBook Z Series\Desktop\modul 5\tempCodeRun
5
1 3 5
PS C:\Users\MyBook Z Series\Desktop\modul 5> go run "c:\User
20
1 3 5 7 9 11 13 15 17 19
PS C:\Users\MyBook Z Series\Desktop\modul 5> █
```

Penjelasan :

Program di atas bertujuan untuk mencetak bilangan ganjil dari 1 hingga n secara terbalik. Program dimulai dengan meminta pengguna untuk memasukkan nilai integer n. Fungsi `barisGanjil` kemudian dipanggil dengan parameter n. Di dalam fungsi tersebut, terdapat logika rekursif yang pertama-tama mengurangi nilai `bilangan` sampai mencapai 0. Ketika `bilangan` lebih besar dari 0, fungsi tersebut memanggil dirinya sendiri. Setelah kembali dari pemanggilan rekursif, jika `bilangan` adalah bilangan ganjil (yaitu jika `bilangan%2` tidak sama dengan 0), maka bilangan tersebut akan dicetak. Dengan cara ini, program akan mencetak semua bilangan ganjil dari n hingga 1.

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".

Source code :

```
1  package main
2
3  import "fmt"
4
5  func main() {
6      var base, exponent int
7      fmt.Scan(&base)
8      fmt.Scan(&exponent)
9
10     result := calculatePower(base, exponent)
11     fmt.Println(result)
12 }
13
14 func calculatePower(base, exponent int) int {
15     if exponent == 0 {
16         return 1
17     }
18     return base * calculatePower(base, exponent-1)
19 }
20
```

Output :

```
PS C:\Users\MyBook Z Series\Desktop\modul 5>
go run "c:\Users\MyBook Z Series\Desktop\modul 5\tempCode
2 2
4
PS C:\Users\MyBook Z Series\Desktop\modul 5> go run "c:\U
5 3
125
PS C:\Users\MyBook Z Series\Desktop\modul 5> |
```

Penjelasan :

Program di atas adalah implementasi sederhana dari fungsi untuk menghitung pangkat (eksponen) suatu bilangan. Di dalam fungsi `main`, program meminta pengguna untuk memasukkan dua bilangan bulat: `base` (basis) dan `exponent` (pangkat). Setelah itu, program memanggil fungsi `calculatePower`, yang menggunakan rekursi untuk menghitung hasil dari `base` dipangkatkan dengan `exponent`. Jika nilai `exponent` adalah 0, fungsi mengembalikan 1 (sesuai dengan aturan bahwa setiap bilangan pangkat nol adalah satu). Hasil perhitungan kemudian ditampilkan di layar.