**LAPORAN PRAKTIKUM   
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL V**

**REKURSIF**

**Sebuah gambar berisi logo, simbol, Grafis, Font

Deskripsi dibuat secara otomatis**

**Disusun Oleh :**

**Tri Marselinus Sitanggang / 2311102209**

**IF11-06**

**Dosen Pengampu :**

**ABEDNEGO DWI SEPTIADI**

**PROGRAM STUDI S1 TEKNIK INFORMATIKA**

**FAKULTAS INFORMATIKA**

**TELKOM UNIVERSITY PURWOKERTO**

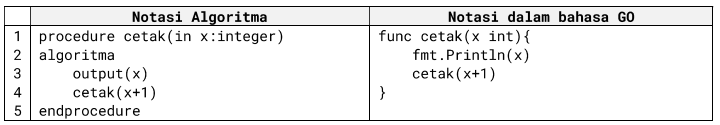
**2024**

1. **DASAR TEORI**

**Rekursif**

Rekursif adalah konsep suatu fungsi memanggil dirinya sendiri untuk menyelesaikan suatu masalah. Teknik ini sangat berguna untuk memecahkan masalah yang dapat dipecah menjadi sub-masalah yang lebih kecil dan mirip dengan masalah asli.

**Contoh**

****

Apabila diperhatikan subprogram cetak() di atas, terlihat pada baris ke-4 terdapat pemanggilan subprogram cetak() kembali. Misalnya apabila kita eksekusi perintah cetak (5) maka akan menampilkan angka 5 6 7 8 9...dst tanpa henti. Artinya setiap pemanggilan subprogram cetak() nilai x akan selalu bertambah 1 (Increment by one) secara terus menerus tanpa henti.

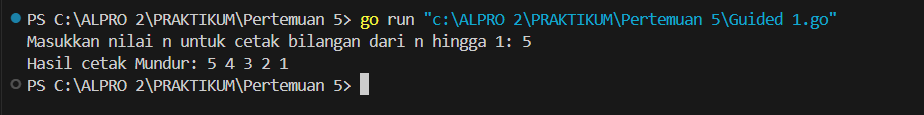
1. **GUIDED**
2. **Soal Studi Case**

Buatlah sebuah program untuk mencetak mundur bilangan n yang diinputkan oleh pengguna.

**Sourcecode**

|  |
| --- |
| 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**

****

**Deskripsi Program**

Program diatas adalah program yang berfungsi untuk mencetak bilangan mundur dari suatu nilai **n** yang diberikan oleh pengguna hingga 1. Program dimulai dengan meminta pengguna memasukkan nilai **n**. Kemudian, fungsi rekursif **cetakMundur** dipanggil dengan parameter **n**. Fungsi ini akan mencetak nilai **n** dan memanggil dirinya sendiri dengan **n-1**, terus berlanjut hingga **n** mencapai **1**. Saat **n** sama dengan **1,** fungsi mencetak **1** dan berhenti.

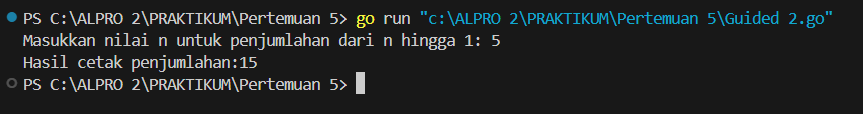
1. **Soal Studi Case**

Buatlah program untuk menjumlahkan bilangan n yang diipuntkan hingga 1 secara berurutan.

**Sourcecode**

|  |
| --- |
| 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 dari n hingga 1: ")      fmt.Scanln(&n)      fmt.Print("Hasil cetak penjumlahan:",jumlahRekursif(n))  } |

**Screenshoot Output**

****

**Deskripsi Program**

Program diatas adalah program yang bertujuan untuk menghitung jumlah bilangan dari **n** hingga **1** menggunakan **rekursi.** Program dimulai dengan meminta pengguna memasukkan nilai **n**. Kemudian, fungsi rekursif **jumlahRekursif** dipanggil dengan parameter **n.** Fungsi ini akan menjumlahkan nilai **n** dengan hasil pemanggilan dirinya sendiri dengan **n-1**, berlanjut terus hingga **n** mencapai **1**. Saat **n** sama dengan **1**, fungsi akan mengembalikan nilai **1.** Hasil akhir dari penjumlahan tersebut kemudian dicetak.

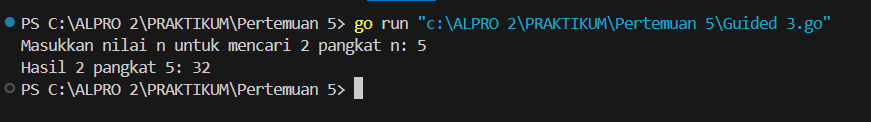
1. **Soal Studi Case**

Buatlah program yang berfungsi untuk menghitung nilai dari 2^n, nilai n tersebut diinputkan oleh pengguna.

**Sourcecode**

|  |
| --- |
| 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.Print("Hasil 2 pangkat ",n, ": ", pangkatDua(n))  } |

**Screenshoot Output**

****

**Deskripsi Program**

Program diatas adalah program yang bertujuan untuk menghitung nilai **2 pangkat n** menggunakan rekursi. Program dimulai dengan meminta pengguna memasukkan nilai **n**. Kemudian, fungsi rekursif **pangkatDua** dipanggil dengan parameter **n.** Fungsi ini akan mengalikan **2** dengan hasil pemanggilan dirinya sendiri dengan **n-1**, berlanjut terus hingga **n** mencapai **0.** Saat **n** sama dengan **0**, fungsi akan mengembalikan nilai **1**, karena **2** pangkat **0** adalah **1**. Hasil akhir dari perhitungan tersebut kemudian dicetak.

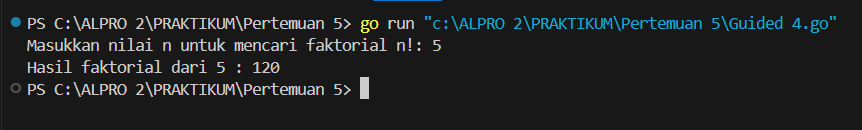
1. **Soal Studi Case**

Buatlah program yang berfungsi untuk menghitung nilai faktorial dari n, yang diinputkan oleh pengguna.

**Sourcecode**

|  |
| --- |
| 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**

****

**Deskripsi Program**

Program diatas adalah program yang bertujuan untuk menghitung nilai faktorial dari **n** menggunakan rekursi. Program dimulai dengan meminta pengguna memasukkan nilai **n**. Kemudian, fungsi rekursif **faktorial** dipanggil dengan parameter **n.** Fungsi ini akan mengembalikan nilai **1** jika **n** adalah **0** atau **1**, karena faktorial dari **0** dan **1** adalah **1.** Jika **n** lebih besar dari **1**, fungsi akan mengalikan **n** dengan hasil pemanggilan dirinya sendiri dengan **n-1**, berlanjut terus hingga **n** mencapai **1** atau **0.** Hasil akhir dari perhitungan tersebut kemudian dicetak.

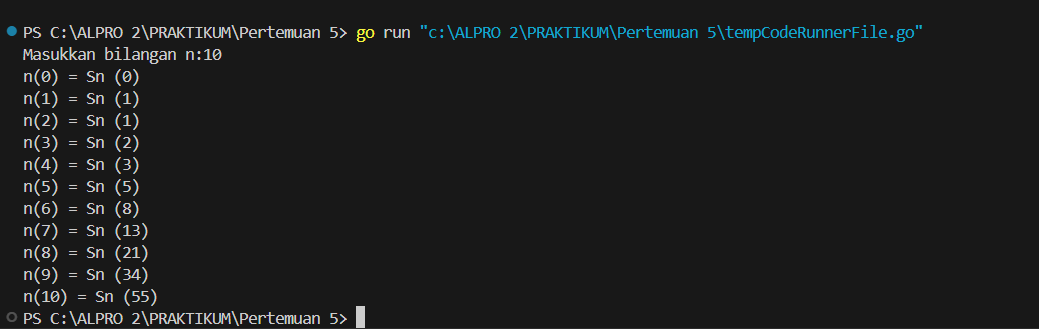
1. **UNGUIDED**
   * + 1. **Soal Studi Case**

Deret fibonacci adalah sebuah deret dengan nilai suku ke-O dan ke-1 adalah 0 dan 1, dan nilai suku ke-n selanjutnya adalah hasil penjumlahan dua suku sebelumnya. Secara umum dapat diformulasikan Sn = Sn-1 + Sn-2. Berikut ini adalah contoh nilai deret fibonacci hingga suku ke-10. Buatlah program yang mengimplementasikan fungsi rekursif pada deret fibonacci tersebut.

**Sourcecode**

|  |
| --- |
| package main  import "fmt"  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 bilangan n:")      fmt.Scan(&n)      for i := 0; i <= n; i++ {          fmt.Printf("n(%d) = Sn (%d)\n", i, fibonacci(i))      }  } |

**Screenshoot Output**

****

**Deskripsi Program**

Program diatas adalah program yang bertujuan untuk menghitung dan mencetak deret Fibonacci hingga bilangan ke-**n** menggunakan rekursi. Program dimulai dengan meminta pengguna memasukkan nilai **n**. Kemudian, fungsi rekursif **fibonacci** dipanggil untuk setiap bilangan dari **0** hingga **n** dalam sebuah **loop**. Fungsi **fibonacci** bekerja dengan dua kondisi dasar: jika **n adalah 0**, fungsi mengembalikan **0;** jika **n adalah 1**, fungsi mengembalikan **1.** Untuk nilai **n yang lebih besar**, fungsi mengembalikan hasil penjumlahan dari **fibonacci(n-1)** dan **fibonacci(n-2)**. Hasil dari setiap pemanggilan fungsi dicetak dalam format **n(i) = Sn (hasil)**.

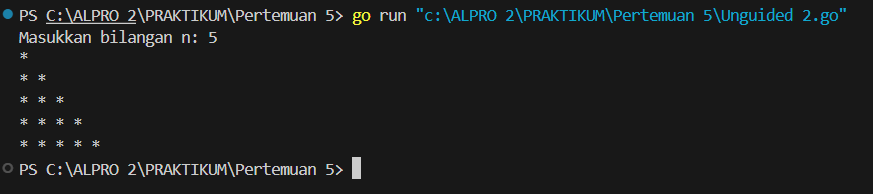
* + - 1. **Soal Studi Case**

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

**Sourcecode**

|  |
| --- |
| package main  import "fmt"    func cetakbintang(n int) {      if n <= 0 {          return      }      cetakbintang(n - 1)      for i := 0; i < n; i++ {          fmt.Print("\* ")      }      fmt.Println()  }  func main() {      var n int      fmt.Print("Masukkan bilangan n: ")      fmt.Scan(&n)      fmt.Print ("")      cetakbintang(n)  } |

**Screenshoot Output**

****

**Deskripsi Program**

Program diatas adalah program yang berfungsi untuk mencetak pola bintang sebanyak bilangan **n**, di mana baris pertama berisi satu bintang dan setiap baris berikutnya menambah satu bintang. Algoritma dimulai dengan meminta pengguna memasukkan bilangan bulat **n** yang menentukan jumlah baris. Fungsi **cetakbintang** dipanggil dengan **n**, dan jika **n lebih dari 0**, fungsi ini memanggil dirinya sendiri dengan **n-1,** mencetak baris dengan bintang lebih sedikit terlebih dahulu. Setelah itu, fungsi mencetak **n** bintang di baris saat ini.

* + - 1. **Soal Studi Case**

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

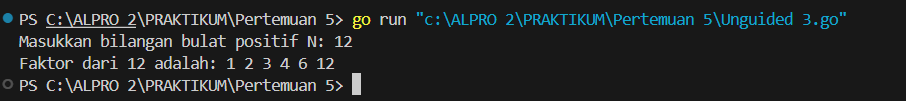
**Masukan** terdiri dari sebuah bilangan bulat positif N.

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

**Sourcecode**

|  |
| --- |
| 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.Print("Masukkan bilangan bulat positif N: ")      fmt.Scan(&n)      fmt.Print("Faktor dari ", n, " adalah: ")      cetakFaktor(n, 1)      fmt.Println()  } |

**Screenshoot Output**

****

**Deskripsi Program**

Program diatas adalah program untuk mencetak faktor dari bilangan bulat positif **n** yang dimasukkan oleh pengguna. Setelah pengguna memasukkan nilai **n**, fungsi rekursif **cetakFaktor** dipanggil dengan parameter **n** dan **i** yang dimulai dari **1**. Fungsi ini memeriksa apakah **i** lebih besar dari **n**; jika ya, proses **berhenti**. Jika **i** adalah faktor dari **n** (yaitu n dibagi i tanpa sisa), maka **i** dicetak. Fungsi kemudian memanggil dirinya sendiri dengan **i** yang ditingkatkan satu.

* + - 1. **Soal Studi Case**

Buatlah program yang mengimplementasikan rekursif untuk menampilkan barisan bilangan tertentu.

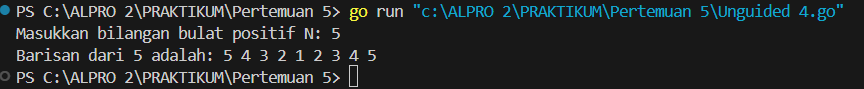
**Masukan** terdiri dari sebuah bilangan bulat positif N.

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

**Sourcecode**

|  |
| --- |
| package main  import "fmt"  func cetakBarisan(n int, batas int) {      fmt.Print(batas, " ")      if batas == 1 {          return      }      cetakBarisan(n, batas-1)      fmt.Print(batas, " ")  }  func main() {      var n int      fmt.Print("Masukkan bilangan bulat positif N: ")      fmt.Scan(&n)      fmt.Print("Barisan dari ", n, " adalah: ")      cetakBarisan(n, n)  } |

**Screenshoot Output**

****

**Deskripsi Program**

Program diatas adalah program untuk mencetak barisan bilangan dari **n hingga 1**, kemudian kembali dari **1 hingga n**. Setelah pengguna memasukkan bilangan bulat positif **n**, fungsi rekursif **cetakBarisan** dipanggil dengan batas diinisialisasi dengan **n**. Fungsi mencetak **batas**, lalu memanggil dirinya sendiri dengan batas yang dikurangi satu hingga mencapai **1.** Setelah pemanggilan rekursif selesai, fungsi mencetak kembali nilai **batas**.

* + - 1. **Soal Studi Case**

Buatlah program yang mengimplementasikan rekursif untuk menampilkan barisan bilangan ganjil.

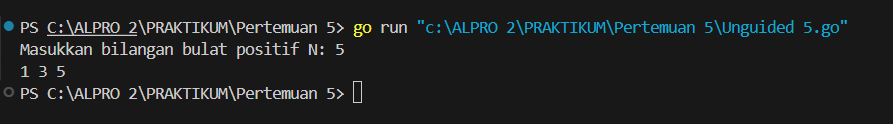
**Masukan** terdiri dari sebuah bilangan bulat positif N.

**Keluaran** terdiri dari barisan bilangan ganjil dari 1 hingga N.

**Sourcecode**

|  |
| --- |
| package main  import "fmt"  func cetakGanjil(n int, a int) {      if a > n {          return      }      if a%2 != 0 {          fmt.Print(a, " ")      }      cetakGanjil(n, a+1)  }  func main() {      var n int      fmt.Print("Masukkan bilangan bulat positif N: ")      fmt.Scan(&n)      fmt.Print("")      cetakGanjil(n, 1)  } |

**Screenshoot Output**

****

**Deskripsi Program**

Program diatas adalah program untuk mencetak **semua bilangan ganjil** dari **1 hingga bilangan bulat positif n** yang dimasukkan oleh pengguna. Setelah pengguna memasukkan nilai **n**, fungsi rekursif **cetakGanjil** dipanggil dengan **a** diinisialisasi dengan **1**. Fungsi memeriksa apakah **a** **lebih besar dari n**; jika **ya**, **proses berhenti**. Jika **a adalah bilangan ganjil**, maka **a dicetak**, dan fungsi memanggil dirinya sendiri dengan **a** yang ditingkatkan satu.

* + - 1. **Soal Studi Case**

Buatlah program yang mengimplementasikan rekursif untuk mencari hasil pangkat dari dua buah matic bilangan.

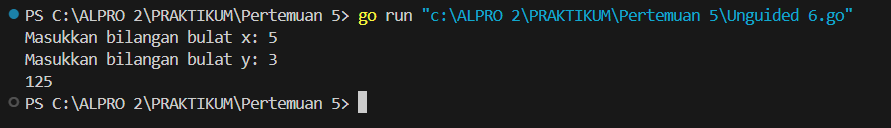
**Masukan** terdiri dari bilangan bulat x dan y.

**Keluaran** terdiri dari hasil x dipangkatkan y.

**Sourcecode**

|  |
| --- |
| package main  import "fmt"  func pangkat(x int, y int) int {      if y == 0 {          return 1      }      if y < 0 {          return 1 / pangkat(x, -y)      }      return x \* pangkat(x, y-1)  }  func main() {      var x, y int      fmt.Print("Masukkan bilangan bulat x: ")      fmt.Scan(&x)      fmt.Print("Masukkan bilangan bulat y: ")      fmt.Scan(&y)      hasil := pangkat(x, y)      fmt.Print(hasil)  } |

**Screenshoot Output**

****

**Deskripsi Program**

Program diatas adalah program untuk menghitung nilai pangkat dari bilangan bulat **x** yang dipangkatkan dengan **y**. Setelah pengguna memasukkan **nilai x dan y**, fungsi rekursif **pangkat** dipanggil. Fungsi ini **mengembalikan 1** jika **y adalah 0**, menghitung **invers** jika **y negatif**, dan **mengalikan x** dengan **hasil fungsi untuk y-1 jika y positif**. Hasil perhitungan kemudian dicetak.