**LAPORAN PRAKTIKUM   
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL VII**

**STRUCK & ARRAY**

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

Deskripsi dibuat secara otomatis**

**Disusun Oleh :**

**Rasyid Nafsyarie / 2311102011**

**IF-11-06**

**Dosen Pengampu :**

**Abednego Dwi Septiadi, S.Kom., M.Kom**

**PROGRAM STUDI S1 TEKNIK INFORMATIKA**

**FAKULTAS INFORMATIKA**

**TELKOM UNIVERSITY PURWOKERTO**

**2024**

1. **DASAR TEORI**

**Struct** adalah tipe data komposit di Golang yang digunakan untuk mengelompokkan beberapa nilai data yang memiliki tipe yang berbeda menjadi satu kesatuan. Struct sangat berguna untuk merepresentasikan entitas yang kompleks dengan beberapa atribut/properti.

Deklarasi Struct

type NamaStruct struct {

Field1 TipeData

Field2 TipeData

FieldN TipeData

}

Fitur Struct

1. Embedded Struct: Struct dapat menyimpan struct lain.
2. Pointer Struct: Struct dapat diakses menggunakan pointer untuk efisiensi.
3. Method Struct: Struct dapat memiliki fungsi/metode.

**Array** adalah struktur data yang digunakan untuk menyimpan sekumpulan elemen dengan tipe data yang sama. Panjang array bersifat tetap dan didefinisikan saat deklarasi.

Deklarasi Array

var namaArray [ukuran]TipeData

Fitur Array

1. Panjang Tetap: Ukuran array tidak dapat diubah setelah dideklarasikan.
2. Multi-Dimensi: Golang mendukung array multi-dimensi.
3. Inisialisasi Default: Nilai elemen array default tergantung tipe datanya (0 untuk numerik, "" untuk string, dll.).
4. **GUIDED**
   * + 1. **Guided 1**

**Sourcecode**

|  |
| --- |
| package main  import (      "fmt"      "math"  )  // struktur untuk menyimpan titik dengan koordinat (x, y)  type Titik struct {      x int      y int  }  // struktur untuk menyimpan lingkaran dengan pusat dan radius  type Lingkaran struct {      pusat  Titik      radius int  }  // fungsi untuk menghitung jarak antara dua titik  func hitungJarak(a, b Titik) float64 {      return math.Sqrt(float64((a.x-b.x)(a.x-b.x) + (a.y-b.y)(a.y-b.y)))  }  // fungsi untuk memeriksa apakah titik berada di dalam lingkaran  func titikDiDalamLingkaran(t Titik, l Lingkaran) bool {      jarak := hitungJarak(t, l.pusat)      return jarak <= float64(l.radius)  }  func main() {      //input untuk lingkaran 1      var cx1, cy1, r1 int      fmt.Print("Masukkan koordinat pusat dan radius lingkaran 1 (cx1 cy1 r1): ")      fmt.Scanln(&cx1, &cy1, &r1)      lingkaran1 := Lingkaran{pusat: Titik{x: cx1, y: cy1}, radius: r1}      // input untuk lingkaran 2      var cx2, cy2, r2 int      fmt.Print("Masukkan koordinat pusat dan radius lingkaran 2 (cx2 cy2 r2): ")      fmt.Scanln(&cx2, &cy2, &r2)      lingkaran2 := Lingkaran{pusat: Titik{x: cx2, y: cy2}, radius: r2}      // input untuk titik sembarang      var x, y int      fmt.Print("Masukkan koordinat titik sembarang (x y): ")      fmt.Scanln(&x, &y)      titik := Titik{x: x, y: y}      // pengecekan posisi titik      diDalamL1 := titikDiDalamLingkaran(titik, lingkaran1)      diDalamL2 := titikDiDalamLingkaran(titik, lingkaran2)      // menampilkan hasil sesuai kondisi      if diDalamL1 && diDalamL2 {          fmt.Println("Titik di dalam lingkaran 1 dan 2")      } else if diDalamL1 {          fmt.Println("Titik di dalam lingkaran 1")      } else if diDalamL2 {          fmt.Println("Titik di dalam lingkaran 2")      } else {          fmt.Println("Titik berada di luar lingkaran")      }  } |

**Screenshoot Output**

A screenshot of a computer

Description automatically generated

**Deskripsi Program**

Fungsi hitungJarak: Fungsi ini menerima dua objek Titik dan menghitung jarak antara keduanya menggunakan rumus Pythagoras. Fungsi titikDiDalamLingkaran: Fungsi ini memeriksa apakah jarak dari titik ke pusat lingkaran kurang dari atau sama dengan radius lingkaran.

1. **UNGUIDED**

**Unguided 1**

**Sourcecode**

|  |
| --- |
| package main  //Rasyid Nafsyarie 2311102011 IF 11 06  import (      "fmt"      "math"  )  func tampilSemua(array []int) {      fmt.Println("Isi array:", array)  }  func tampilGanjil(array []int) {      fmt.Print("Elemen dengan indeks ganjil: ")      for i := 1; i < len(array); i += 2 {          fmt.Print(array[i], " ")      }      fmt.Println()  }  func tampilGenap(array []int) {      fmt.Print("Elemen dengan indeks genap: ")      for i := 0; i < len(array); i += 2 {          fmt.Print(array[i], " ")      }      fmt.Println()  }  func tampilKelipatan(array []int, x int) {      fmt.Printf("Elemen dengan indeks kelipatan %d: ", x)      for i := x; i < len(array); i += x {          fmt.Print(array[i], " ")      }      fmt.Println()  }  func hapusIndeks(array []int, indeks int) []int {      fmt.Printf("Menghapus elemen pada indeks %d\n", indeks)      return append(array[:indeks], array[indeks+1:]...)  }  func rataRata(array []int) float64 {      total := 0      for \_, val := range array {          total += val      }      return float64(total) / float64(len(array))  }  func standarDeviasi(array []int) float64 {      mean := rataRata(array)      var sum float64      for \_, val := range array {          sum += math.Pow(float64(val)-mean, 2)      }      return math.Sqrt(sum / float64(len(array)))  }  func frekuensi(array []int, nilai int) int {      count := 0      for \_, val := range array {          if val == nilai {              count++          }      }      return count  }  func main() {      var N, x, hapusIdx, cariFrekuensi int      fmt.Print("Masukkan jumlah elemen array (N): ")      fmt.Scan(&N)      array := make([]int, N)      fmt.Println("Masukkan elemen array:")      for i := 0; i < N; i++ {          fmt.Printf("Elemen ke-%d: ", i)          fmt.Scan(&array[i])      }      tampilSemua(array)      tampilGanjil(array)      tampilGenap(array)      fmt.Print("Masukkan nilai x untuk menampilkan elemen dengan indeks kelipatan x: ")      fmt.Scan(&x)      tampilKelipatan(array, x)      fmt.Print("Masukkan indeks yang ingin dihapus: ")      fmt.Scan(&hapusIdx)      if hapusIdx >= 0 && hapusIdx < len(array) {          array = hapusIndeks(array, hapusIdx)          tampilSemua(array)      } else {          fmt.Println("Indeks yang dimasukkan tidak valid!")      }      fmt.Printf("Rata-rata elemen array: %.2f\n", rataRata(array))      fmt.Printf("Standar deviasi elemen array: %.2f\n", standarDeviasi(array))      fmt.Print("Masukkan bilangan yang ingin dicari frekuensinya: ")      fmt.Scan(&cariFrekuensi)      fmt.Printf("Frekuensi %d di dalam array: %d\n", cariFrekuensi, frekuensi(array, cariFrekuensi))  } |

**Screenshoot Output**

A screenshot of a computer

Description automatically generated

**Deskripsi Program**

Fungsi untuk menampilkan elemen array: Menampilkan semua elemen, elemen dengan indeks ganjil, dan genap. Fungsi untuk menghapus elemen: Menghapus elemen pada indeks tertentu. Statistik dasar: Menghitung rata-rata dan standar deviasi dari elemen array. Frekuensi elemen: Menghitung berapa kali suatu nilai muncul dalam array.

**Unguided 2**

**Sourcecode**

|  |
| --- |
| package main  //Rasyid Nafsyarie 2311102011 IF 11 06  import (      "fmt"  )  func main() {      var klubA, klubB string      var skorA, skorB int      var pemenang []string      fmt.Print("Masukkan nama Klub A: ")      fmt.Scanln(&klubA)      fmt.Print("Masukkan nama Klub B: ")      fmt.Scanln(&klubB)      for i := 1; ; i++ {          fmt.Printf("Pertandingan %d - Masukkan skor %s: ", i, klubA)          fmt.Scan(&skorA)          fmt.Printf("Pertandingan %d - Masukkan skor %s: ", i, klubB)          fmt.Scan(&skorB)          if skorA < 0 || skorB < 0 {              fmt.Println("Skor tidak valid. Pertandingan selesai.")              break          }          if skorA > skorB {              pemenang = append(pemenang, klubA)              fmt.Printf("Hasil %d: %s\n", i, klubA)          } else if skorB > skorA {              pemenang = append(pemenang, klubB)              fmt.Printf("Hasil %d: %s\n", i, klubB)          } else {              fmt.Printf("Hasil %d: Draw\n", i)          }      }      fmt.Println("\nDaftar klub yang memenangkan pertandingan:")      for \_, klub := range pemenang {          fmt.Println(klub)      }  } |

**Screenshoot Output**

A screenshot of a computer

Description automatically generated

**Deskripsi Program**

Program ini adalah contoh yang baik untuk memahami dasar-dasar pemrograman dalam Go, termasuk penggunaan variabel, loop, dan pengkondisian. Dengan program ini, pengguna dapat dengan mudah mencatat hasil pertandingan sepak bola dan melihat siapa yang menjadi pemenang.

**Unguided 3**

**Sourcecode**

|  |
| --- |
| package main  //Rasyid Nafsyarie 2311102011 IF 11 06  import (      "fmt"  )  const NMAX int = 127  type tabel [NMAX]rune  // Fungsi untuk mengisi array dengan karakter dari input user  func isiArray(t \*tabel, n \*int) {      var ch rune      \*n = 0      for {          fmt.Scanf("%c", &ch)          if ch == '\n' || ch == 'T' {              break          }          t[\*n] = ch          \*n++          if \*n >= NMAX {              break          }      }  }  // Fungsi untuk membalikkan isi array  func balikanArray(t \*tabel, n int) {      for i := 0; i < n/2; i++ {          t[i], t[n-1-i] = t[n-1-i], t[i]      }  }  // Fungsi untuk mencetak isi array  func cetakArray(t tabel, n int) {      for i := 0; i < n; i++ {          fmt.Printf("%c", t[i])      }      fmt.Println()  }  // Fungsi untuk memeriksa apakah array membentuk palindrom  func palindrom(t tabel, n int) bool {      for i := 0; i < n/2; i++ {          if t[i] != t[n-1-i] {              return false          }      }      return true  }  func main() {      var tab tabel      var n int      // Mengisi array dengan input dari user      fmt.Print("Masukkan teks: ")      isiArray(&tab, &n)      // Mencetak array asli      fmt.Print("Teks: ")      cetakArray(tab, n)      // Membalikkan isi array      balikanArray(&tab, n)      fmt.Print("Reverse teks: ")      cetakArray(tab, n)      // Memeriksa apakah array membentuk palindrom      if palindrom(tab, n) {          fmt.Println("Teks ini adalah palindrom.")      } else {          fmt.Println("Teks ini bukan palindrom.")      }  } |

**Screenshoot Output**

A screenshot of a computer

Description automatically generated

**Deskripsi Program**

**Tipe Data**: Mendefinisikan tipe tabel sebagai array dari rune. **Fungsi**: isiArray: Mengisi array dengan karakter dari input pengguna, balikanArray: Membalikkan isi array, cetakArray: Mencetak isi array ke layer, palindrom: Memeriksa apakah array membentuk palindrom.