**LAPORAN PRAKTIKUM**

**ALGORITMA DAN PEMROGRAMAN 2**

**MODUL VII**

**STRUCK & ARRAY**

****

**DISUSUN OLEH :**

**WILDAN MAULANA ZIDAN**

**2311102162**

**IF 11 02**

**S1 TEKNIK INFORMATIKA**

**TELKOM UNIVERSITY PURWOKERTO**

1. **DASAR TEORI**
2. **Tipe Bentukan**

Tipe bentukan memungkinkan pemrograman untuk mendefinisikan suatu tipe data baru pada suatu bahasa pemrograman. Tipė bentukan Ini dapat dibedakan atas dua jenis, yaltu Alias dan Struct.

1. **Allas (Type)**

Bahasa pemrograman pada umumnya mengizinkan pemrograman untuk mengubah nama suatu tipe data dengan nama baru yang lebih ringkas dan familiar. Sebagai contoh "Integer" dapat dirubah dengan nama alias "bilangan ". Caranya dengan menggunakan kata kunci "type".

|  |  |  |
| --- | --- | --- |
|  | Notasi Algoritma | Notasi dalam bahasa Go |
| 2  3  4  5  6  7 | kamus type <nama alias> <tipe data> algoritma | package main type <nama alias> <tipe data> func main(){ |

Sebagai contoh perhatikan program Go berikut beserta hasil eksekusinya!

|  |
| --- |
| package mainimport "fmt' type bilangan inttype pecahan float64func main() {  var a, b bilangan var hasil pecahan hasil - pecahan(a) / pecahan(b) fmt . Println(hasil) |

1. **Struct atau Record**

Stucture memungkinkan pemrograman untuk mengelompokkan beberapa data atau nilai yang memiliki relasi atau keterkaitan tertentu menjadi suatu kesatuan. Masing-masing nilai tersimpan dalam field dari stucture tersebut.

|  |  |
| --- | --- |
| Notasi Algoritma | Notasi dalam Bahasa go |
| Kamus  Type <nama struct> <  <field 1> <tipe data>  <field 2> <tipe data>  <field 3> <tipe data>  > | Package main  Type <nama struct> struct {  <field 1> <tipe data>  <field 2> <tipe data>  <field 3> <tipe data>  } |

Berbeda dengan bahasa pemrograman lain. kesamaan tipe dari dua variabel berjenis stucture bukan karena namanya tetapi karena strukturnya. Dua variabel dengan nama-nama field dan tipe field yang sama (dan dalam urutan yang sama) dianggap mempunyai tipe yang sama. Tentunya akan lebih memudahkan jika stucture tersebut didefinisikan sebagai sebuah tipe baru, sehingga deklarasi stucture tidak perlu lagi seluruh field-nya ditulis ulang berkali-kali.

|  |
| --- |
| package main  import "fmt"  type waktu struct {  jam, menit, detik int  }  func main(){  var wParkir, wPulang, durasi waktu  var dParkir, dPulang, 1Parkir int  fmt.Scan(&wParkir.jam, &wParkir.menit, &wParkir.detik  fmt.Scan(&wPulang.jam, &wPulang.menit, &wPulang.detik)  dParkir = wParkir.detik + wParkir.menit\*60 + wParkir.jam\*3600  dPulang = wPulang.detik + wPulang.menit\*60 + wPulang.jam\*3600  1Parkir dPulang dParkir  durasi.jam 1Parkir / 3600  durasi menit = 1Parkir % 3600 / 60  durasi.detik = 1Parkir % 3600% 60  fmt.Printf("Lama parkir: %d jam %d menit %d detik", durasi.jam, dûrasi.menit, durasi.detik)  } |

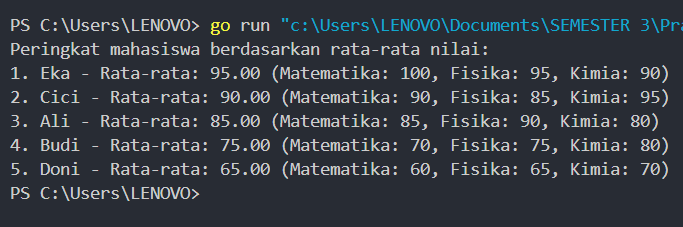
1. **Map**

Tipe array lain, sebuah array dinamik. Indeksnya (di sini disebut kunci) tidak harus berbentuk integer. Indeks dapat berasal dari tipe apa saja. Struktur ini disebut map.

1. **GUIDED**

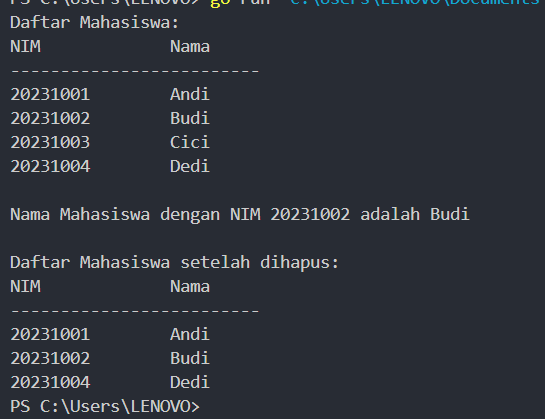
Guided 1

|  |
| --- |
| package main  import (  "fmt"  "sort"  )  // Struktur untuk menampung data mahasiswa  type Mahasiswa struct {  Nama string  Matematika int  Fisika int  Kimia int  RataRata float64  }  // Fungsi untuk menghitung rata-rata nilai tiap mahasiswa  func hitungRataRata(m \*Mahasiswa) {  total := m.Matematika + m.Fisika + m.Kimia  m.RataRata = float64(total) / 3.0  }  // Fungsi utama untuk mengelola dan mengurutkan data mahasiswa berdasarkan nilai rata-rata  func main() {  // Array untuk menampung data mahasiswa  mahasiswa := []Mahasiswa{  {"Ali", 85, 90, 80, 0},  {"Budi", 70, 75, 80, 0},  {"Cici", 90, 85, 95, 0},  {"Doni", 60, 65, 70, 0},  {"Eka", 100, 95, 90, 0},  }  // Menghitung rata-rata nilai tiap mahasiswa  for i := range mahasiswa {  hitungRataRata(&mahasiswa[i])  }  // Mengurutkan mahasiswa berdasarkan nilai rata-rata (descending)  sort.Slice(mahasiswa, func(i, j int) bool {  return mahasiswa[i].RataRata > mahasiswa[j].RataRata  })  // Menampilkan hasil  fmt.Println("Peringkat mahasiswa berdasarkan rata-rata nilai:")  for i, m := range mahasiswa {  fmt.Printf("%d. %s - Rata-rata: %.2f (Matematika: %d, Fisika: %d, Kimia: %d)\n",  i+1, m.Nama, m.RataRata, m.Matematika, m.Fisika, m.Kimia)  }  } |



Guided 2

|  |
| --- |
| package main  import "fmt"  func main() {  // Membuat map dengan NIM sebagai kunci dan Nama sebagai nilai  mahasiswa := map[string]string{  "20231001": "Andi",  "20231002": "Budi",  "20231003": "Cici",  }  // Menambahkan data baru ke map  mahasiswa["20231004"] = "Dedi"  // Menampilkan seluruh isi map dalam format kolom dan baris  fmt.Println("Daftar Mahasiswa:")  fmt.Println("NIM\t\tNama")  fmt.Println("-------------------------")  for nim, nama := range mahasiswa {  fmt.Printf("%s\t%s\n", nim, nama)  }  // Mengakses data berdasarkan NIM  nim := "20231002"  fmt.Println("\nNama Mahasiswa dengan NIM", nim, "adalah", mahasiswa[nim])  // Menghapus data berdasarkan NIM  delete(mahasiswa, "20231003")  // Menampilkan isi map setelah data dihapus dalam format kolom dan baris  fmt.Println("\nDaftar Mahasiswa setelah dihapus:")  fmt.Println("NIM\t\tNama")  fmt.Println("-------------------------")  for nim, nama := range mahasiswa {  fmt.Printf("%s\t%s\n", nim, nama)  }  } |

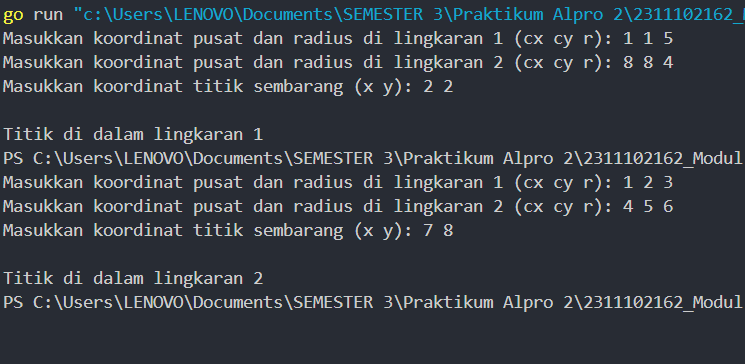


1. **UNGUIDED**

Unguided 1

**SOURCE CODE**

|  |
| --- |
| // 2311102162  package main  import (  "fmt"  "math"  )  type titik struct {  x, y int  }  type lingkaran struct {  cx, cy, r int  }  func jarak(p titik, c lingkaran) float64 {  return math.Sqrt(float64((p.x-c.cx)\*(p.x-c.cx) + (p.y-c.cy)\*(p.y-c.cy)))  }  func posisiTitik(p titik, l1, l2 lingkaran) string {  jarak1 := jarak(p, l1)  jarak2 := jarak(p, l2)  dalam1 := jarak1 < float64(l1.r)  dalam2 := jarak2 < float64(l2.r)  if dalam1 && dalam2 {  return "Titik di dalam lingkaran 1 dan 2"  } else if dalam1 {  return "Titik di dalam lingkaran 1"  } else if dalam2 {  return "Titik di dalam lingkaran 2"  } else {  return "Titik di luar lingkaran 1 dan 2"  }  }  func main() {  var l1, l2 lingkaran  var p titik  fmt.Print("Masukkan koordinat pusat dan radius di lingkaran 1 (cx cy r): ")  fmt.Scan(&l1.cx, &l1.cy, &l1.r)  fmt.Print("Masukkan koordinat pusat dan radius di lingkaran 2 (cx cy r): ")  fmt.Scan(&l2.cx, &l2.cy, &l2.r)  fmt.Print("Masukkan koordinat titik sembarang (x y): ")  fmt.Scan(&p.x, &p.y)  fmt.Print("\n")  hasil := posisiTitik(p, l1, l2)  fmt.Println(hasil)  } |

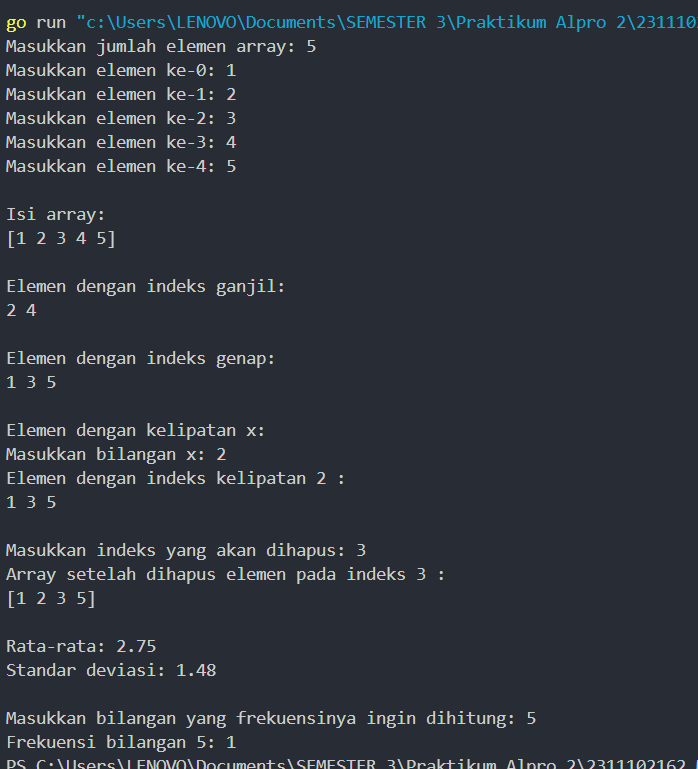


Kode di atas untuk menentukan posisi suatu titik terhadap dua lingkaran. Program ini menjelaskan dua jenis struktur: `titik`untuk menyimpan koordinat titik `(x, y)` dan `lingkaran` untuk menyimpan pusat `(cx, cy)` dan jari-jari `r` lingkaran. Fungsi `jarak` menghitung jarak dari suatu titik ke pusat lingkaran menggunakan rumus Euclidean. Fungsi `posisiTitik` dipergunakan untuk menilai posisi titik, apakah berada di dalam lingkaran pertama, kedua, antara keduanya, atau di luar keduanya. Penilaian didasarkan pada perbandingan jarak titik ke pusat lingkaran dengan jari-jari lingkaran terkait. Dalam fungsi `main`, program akan meminta pengguna untuk memasukkan koordinat pusat dan jari-jari dua lingkaran, serta koordinat titik. Kemudian, hasil posisi titik terhadap lingkaran-lingkaran tersebut akan dicetak.

Unguided 2

**SOURCE CODE**

|  |
| --- |
| // 2311102162  package main  import (  "fmt"  "math"  )  func main() {  var N int  fmt.Print("Masukkan jumlah elemen array: ")  fmt.Scan(&N)  array := make([]int, N)  for i := 0; i < N; i++ {  fmt.Printf("Masukkan elemen ke-%d: ", i)  fmt.Scan(&array[i])  }  fmt.Println("\nIsi array:")  fmt.Println(array)  fmt.Println("\nElemen dengan indeks ganjil:")  for i := 1; i < N; i += 2 {  fmt.Print(array[i], " ")  }  fmt.Println("\n\nElemen dengan indeks genap:")  for i := 0; i < N; i += 2 {  fmt.Print(array[i], " ")  }  var x int  fmt.Println("\n\nElemen dengan kelipatan x:")  fmt.Print("Masukkan bilangan x: ")  fmt.Scan(&x)  fmt.Println("Elemen dengan indeks kelipatan", x, ":")  for i := 0; i < N; i++ {  if i%x == 0 {  fmt.Print(array[i], " ")  }  }  var indeksHapus int  fmt.Print("\n\nMasukkan indeks yang akan dihapus: ")  fmt.Scan(&indeksHapus)  array = append(array[:indeksHapus], array[indeksHapus+1:]...)  fmt.Println("Array setelah dihapus elemen pada indeks", indeksHapus, ":")  fmt.Println(array)  var total int  for \_, value := range array {  total += value  }  rataRata := float64(total) / float64(len(array))  fmt.Printf("\nRata-rata: %.2f\n", rataRata)  var variance float64  for \_, value := range array {  variance += math.Pow(float64(value)-rataRata, 2)  }  variance /= float64(len(array))  standarDeviasi := math.Sqrt(variance)  fmt.Printf("Standar deviasi: %.2f\n", standarDeviasi)  var bilangan int  fmt.Print("\nMasukkan bilangan yang frekuensinya ingin dihitung: ")  fmt.Scan(&bilangan)  frekuensi := 0  for \_, value := range array {  if value == bilangan {  frekuensi++  }  }  fmt.Printf("Frekuensi bilangan %d: %d\n", bilangan, frekuensi)  } |

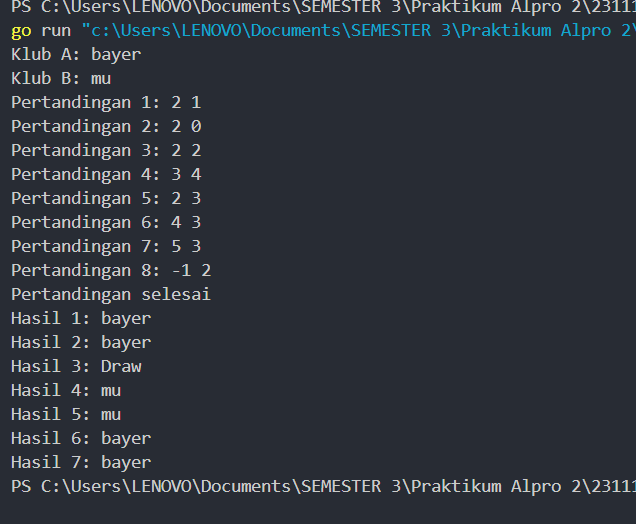


Kode di atas meminta pengguna untuk menentukan jumlah elemen array, lalu mengisi array tersebut dengan angka-angka yang dimasukkan. Program ini menampilkan elemen-elemen array berdasarkan indeks ganjil, genap, serta elemen pada indeks yang merupakan kelipatan dari bilangan yang diinput pengguna. Kemudian, dalam program, pengguna diminta untuk memasukkan indeks tertentu yang ingin dihapus dari array. Setelah itu, dilakukan perhitungan rata-rata serta standar deviasi dari elemen array yang tersisa, yang kemudian ditampilkan. Selanjutnya, program akan meminta pengguna mengetikkan angka tertentu untuk dihitung berapa kali munculnya dalam larik data.

Unguided 3

**SOURCE CODE**

|  |
| --- |
| // 2311102162  package main  import "fmt"  func main() {  var klubA, klubB string  var skorA, skorB int  var pemenang []string  fmt.Print("Klub A: ")  fmt.Scanln(&klubA)  fmt.Print("Klub B: ")  fmt.Scanln(&klubB)  pertandingan := 1  for {  fmt.Printf("Pertandingan %d: ", pertandingan)  fmt.Scanln(&skorA, &skorB)  if skorA < 0 || skorB < 0 {  break  }  if skorA > skorB {  pemenang = append(pemenang, klubA)  } else if skorA < skorB {  pemenang = append(pemenang, klubB)  } else {  pemenang = append(pemenang, "Draw")  }  pertandingan++  }  fmt.Println("Pertandingan selesai")  for i, p := range pemenang {  fmt.Printf("Hasil %d: %s\n", i+1, p)  }  } |

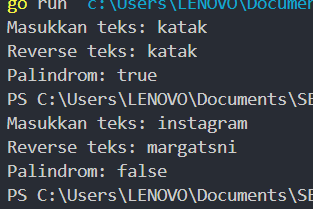
****

Kode di atas meminta input dua nama klub sepak bola (klub A dan klub B) dan skor mereka untuk setiap pertandingan secara berulang. Program akan terus meminta skor untuk setiap pertandingan, menghitung pemenang setiap pertandingan, dan menyimpan hasilnya dalam array 'pemenang' sampai skor yang dimasukkan negatif (menandakan akhir pertandingan). Apabila skor klub A melebihi klub B, maka klub A akan diumumkan sebagai pemenang. Sebaliknya, jika klub B mendapat skor lebih tinggi, klub B lah yang akan menjadi pemenang. Namun, jika keduanya mendapat skor yang sama, hasilnya akan ditetapkan sebagai "Draw". Setelah proses pengulangan berakhir, program akan menampilkan seluruh hasil pertandingan yang telah direkam.

Unguided 4

**SOURCE CODE**

|  |
| --- |
| // 2311102162  package main  import (  "fmt"  "strings"  )  func reverse(text string) string {  runes := []rune(text)  for i, j := 0, len(runes)-1; i < j; i, j = i+1, j-1 {  runes[i], runes[j] = runes[j], runes[i]  }  return string(runes)  }  func isPalindrome(text string) bool {  reversedText := reverse(text)  return strings.ToLower(text) == strings.ToLower(reversedText)  }  func main() {  var text string  fmt.Print("Masukkan teks: ")  fmt.Scanln(&text)  reversedText := reverse(text)  fmt.Println("Reverse teks:", reversedText)  if isPalindrome(text) {  fmt.Println("Palindrom: true")  } else {  fmt.Println("Palindrom: false")  }  } |

****Kode di atas untuk memeriksa apakah sebuah teks itu palindrom atau tidak. Program ini menyertakan fungsi 'reverse' yang dapat membalik urutan karakter dalam teks serta fungsi 'isPalindrome' yang mengecek apakah teks tersebut merupakan palindrom, di mana teks tersebut akan sama ketika dibaca dari depan maupun dari belakang, tanpa memperhatikan perbedaan huruf besar dan kecil. Dalam fungsi `main`, program akan meminta pengguna untuk memberikan teks, kemudian program akan mencetak teks tersebut dalam kebalikan. Apabila teksnya serupa dengan versi terbalik, program akan mencetak 'Palindrom: true'; jika tidak, program akan mencetak 'Palindrom: false'.