**LAPORAN PRAKTIKUM  
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL 6**

**STRUCK DAN ARRAY**



Oleh:

NAMA : SETYO NUGROHO

NIM : 103112400024

KELAS : S1IF-12-01

**S1 TEKNIK INFORMATIKA**

**TELKOM UNIVERSITY PURWOKERTO**

**2025**

1. **DASAR TEORI**

### ****1. Array dalam Bahasa Go****

Array adalah struktur data yang digunakan untuk menyimpan sekumpulan data dengan tipe yang sama dan memiliki jumlah elemen yang tetap. Setiap elemen dalam array diakses menggunakan indeks yang dimulai dari 0. Dalam bahasa Go, array dideklarasikan dengan menyebutkan jumlah elemen dan tipe datanya.

### ****2. Struct dalam Bahasa Go****

Struct (struktur) adalah tipe data bentukan yang memungkinkan pengelompokan beberapa variabel (field) yang memiliki tipe data berbeda ke dalam satu kesatuan. Struct sangat berguna untuk merepresentasikan objek dalam dunia nyata, seperti data mahasiswa, produk, kendaraan, dan lain sebagainya.

1. **GUIDED**

1

Source code:

package main

*//Setyo Nugroho*

*//103112400024*

import (

    "fmt"

    "time"

)

*// Struct untuk barang dalam struk belanja*

type Item struct {

    Name     string

    Price    float64

    Quantity int

}

*// Struct untuk struk belanja*

type Receipt struct {

    StoreInfo   string

    Date        time.Time

    Items       []Item

    TotalAmount float64

}

*// Method untuk menghitung total harga semua item*

func (r \*Receipt) CalculateTotal() {

    var total float64

    for \_, item := range r.Items {

        total += item.Price \* float64(item.Quantity)

    }

    r.TotalAmount = total

}

*// Method untuk mencetak struk belanja*

func (r Receipt) PrintReceipt() {

    fmt.Println("==========================================")

    fmt.Println(r.StoreInfo)

    fmt.Println("Tanggal:", r.Date.Format("02-01-2006 15:04"))

    fmt.Println("==========================================")

    fmt.Printf("%-15s %-10s %-8s %-10s\n", "Item", "Harga", "Jumlah", "Total")

    fmt.Println("------------------------------------------")

    for \_, item := range r.Items {

        itemTotal := item.Price \* float64(item.Quantity)

        fmt.Printf("%-15s Rp%-9.2f %-8d Rp%-9.2f\n", item.Name, item.Price, item.Quantity, itemTotal)

    }

    fmt.Println("==========================================")

    fmt.Printf("%-35s Rp%-9.2f\n", "Total Belanja:", r.TotalAmount)

    fmt.Println("==========================================")

    fmt.Println("Terima kasih telah berbelanja!")

}

func main() {

    receipt := Receipt{

        StoreInfo: "Toko Sembako Makmur\nJl. Raya No. 123, Jakarta",

        Date:      time.Now(),

        Items: []Item{

            {Name: "Beras", Price: 12000, Quantity: 5},

            {Name: "Gula", Price: 15000, Quantity: 2},

            {Name: "Minyak", Price: 20000, Quantity: 1},

            {Name: "Telur", Price: 2000, Quantity: 10},

        },

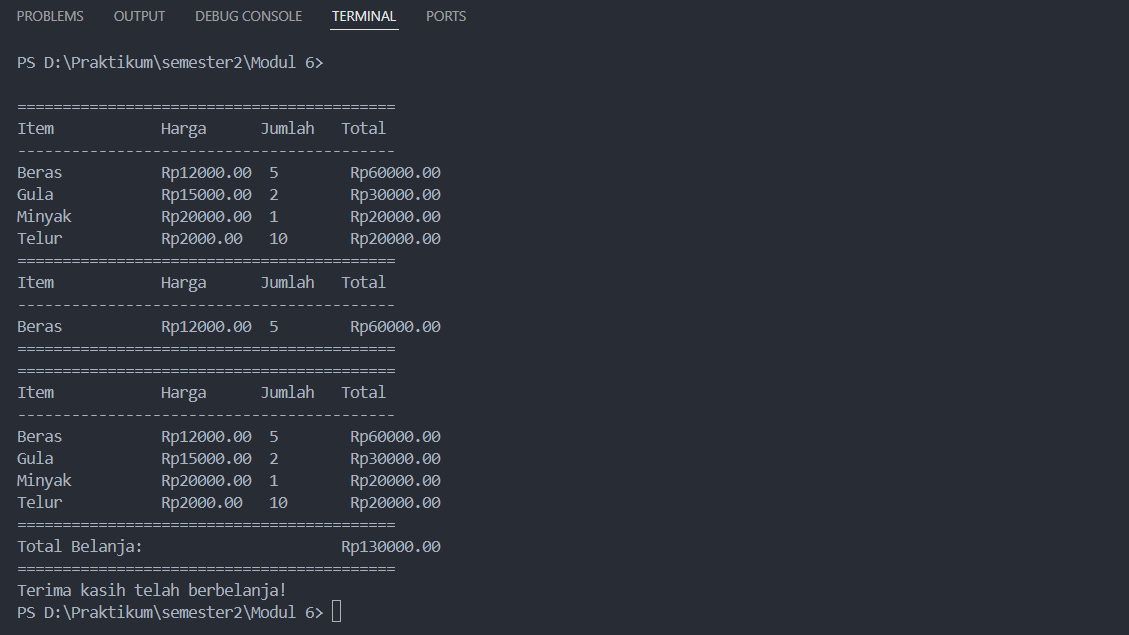
    }

    receipt.CalculateTotal()

    receipt.PrintReceipt()

}

Output:



Deskripsi:  
Program ini dibuat menggunakan bahasa pemrograman Go (Golang) dengan tujuan untuk **mensimulasikan pencetakan struk belanja** pada sebuah toko. Program memanfaatkan konsep **struct** untuk merepresentasikan data barang dan struk belanja secara terstruktur.

Program ini menunjukkan penggunaan praktis struct, slice, dan method pada Go untuk menyelesaikan permasalahan dunia nyata, dalam hal ini pembuatan struk belanja secara otomatis.

1.1

Source code:

package main

*//Setyo Nugroho*

*//103112400024*

import (

    "fmt"

)

func main() {

*// Deklarasi dan inisialisasi array nilai mahasiswa*

    nilaiMahasiswa := [5]int{85, 90, 78, 88, 95}

    fmt.Println("Data Nilai Mahasiswa:")

    fmt.Println("=====================")

*// Menampilkan nilai per mahasiswa*

    for i, nilai := range nilaiMahasiswa {

        fmt.Printf("Mahasiswa %d: %d\n", i+1, nilai)

    }

*// Menghitung rata-rata nilai*

    var total int

    for \_, nilai := range nilaiMahasiswa {

        total += nilai

    }

    rataRata := float64(total) / float64(len(nilaiMahasiswa))

    fmt.Println("=====================")

    fmt.Printf("Rata-rata nilai: %.2f\n", rataRata)

*// Mencari nilai tertinggi dan terendah*

    tertinggi := nilaiMahasiswa[0]

    terendah := nilaiMahasiswa[0]

    for \_, nilai := range nilaiMahasiswa {

        if nilai > tertinggi {

            tertinggi = nilai

        }

        if nilai < terendah {

            terendah = nilai

        }

    }

    fmt.Printf("Nilai tertinggi: %d\n", tertinggi)

    fmt.Printf("Nilai terendah: %d\n", terendah)

*// Contoh array 2 dimensi*

    fmt.Println("\nContoh Array 2 Dimensi:")

    fmt.Println("=====================")

*// Nilai ujian mahasiswa dalam 2 mata kuliah (Matematika, Bahasa)*

    nilaiUjian := [3][2]int{

        {80, 85},

        {90, 75},

        {70, 95},

    }

*// Menampilkan nilai ujian per mahasiswa*

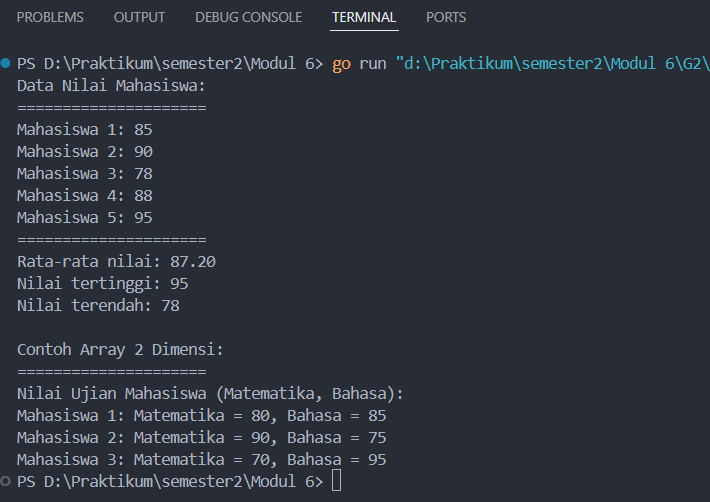
    fmt.Println("Nilai Ujian Mahasiswa (Matematika, Bahasa):")

    for i, nilai := range nilaiUjian {

        fmt.Printf("Mahasiswa %d: Matematika = %d, Bahasa = %d\n", i+1, nilai[0], nilai[1])

    }

}

Output:  


Deskripsi:  
Program ini dibuat menggunakan bahasa pemrograman Go (Golang) untuk mengelola dan menampilkan data nilai mahasiswa. Program ini mencakup pengolahan array satu dimensi dan dua dimensi untuk menghitung statistik sederhana serta menampilkan informasi nilai dari beberapa mata kuliah.

Program ini menunjukkan penerapan dasar struktur data **array satu dimensi dan dua dimensi** dalam bahasa Go. Selain itu, program ini juga melibatkan perulangan, perhitungan statistik dasar seperti rata-rata, nilai maksimum, dan minimum, yang sering dibutuhkan dalam pengolahan data sederhana.

1. **UNGUIDED**

1

Source code:

package main

*//Setyo Nugroho*

*//103112400024*

import (

    "fmt"

    "math"

)

type Titik struct {

    x, y int

}

type Lingkaran struct {

    pusat Titik

    r     int

}

func hitungJarak(a, b Titik) float64 {

    selisihX := float64(a.x - b.x)

    selisihY := float64(a.y - b.y)

    return math.Sqrt(selisihX\*selisihX + selisihY\*selisihY)

}

func cekDalam(l Lingkaran, t Titik) bool {

    return hitungJarak(l.pusat, t) <= float64(l.r)

}

func main() {

    var ling1, ling2 Lingkaran

    var titik Titik

    fmt.Scan(&ling1.pusat.x, &ling1.pusat.y, &ling1.r)

    fmt.Scan(&ling2.pusat.x, &ling2.pusat.y, &ling2.r)

    fmt.Scan(&titik.x, &titik.y)

    dalam1 := cekDalam(ling1, titik)

    dalam2 := cekDalam(ling2, titik)

    if dalam1 && dalam2 {

        fmt.Println("Titik di dalam lingkaran 1 dan 2")

    } else if dalam1 {

        fmt.Println("Titik di dalam lingkaran 1")

    } else if dalam2 {

        fmt.Println("Titik di dalam lingkaran 2")

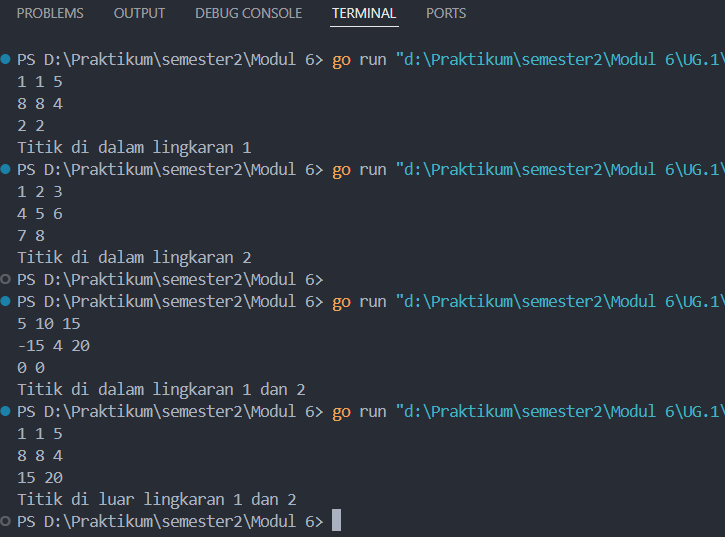
    } else {

        fmt.Println("Titik di luar lingkaran 1 dan 2")

    }

}

Output:



Deskripsi:  
Program ini dibuat menggunakan bahasa pemrograman Go (Golang) untuk **mengecek posisi suatu titik terhadap dua buah lingkaran** dalam bidang kartesius. Program memanfaatkan konsep **struct** untuk merepresentasikan data titik dan lingkaran, serta fungsi untuk menghitung jarak dan memeriksa keberadaan titik di dalam lingkaran.

Program ini menunjukkan implementasi **geometri dasar dalam pemrograman**, dengan memanfaatkan **struct, fungsi, serta perhitungan matematis** dalam bahasa Go. Cocok digunakan untuk pemahaman konsep jarak antar titik dan penerapan logika kondisi dalam kasus dua buah area lingkaran.

2

Source code:

package main

*//Setyo Nugroho*

*//103112400024*

import "fmt"

func hitungRataRata(kaen []int) float64 {

    var total int

    for \_, num := range kaen {

        total += num

    }

    return float64(total) / float64(len(kaen))

}

func hitungSimpanganBaku(kaen []int) float64 {

    rata := hitungRataRata(kaen)

    var kuadratSelisih float64

    for \_, num := range kaen {

        selisih := float64(num) - rata

        kuadratSelisih += selisih \* selisih

    }

    return kuadratSelisih / float64(len(kaen))

}

func hitungFrekuensi(kaen []int, target int) int {

    count := 0

    for \_, num := range kaen {

        if num == target {

            count++

        }

    }

    return count

}

func tampilkanIndeks(kaen []int) {

    fmt.Print("Indeks ganjil: ")

    for i := 1; i < len(kaen); i += 2 {

        fmt.Print(kaen[i], " ")

    }

    fmt.Println()

    fmt.Print("Indeks genap: ")

    for i := 0; i < len(kaen); i += 2 {

        fmt.Print(kaen[i], " ")

    }

    fmt.Println()

}

func main() {

    var jumlahElemen int

    fmt.Print("Masukkan jumlah elemen: ")

    fmt.Scan(&jumlahElemen)

    kaen := make([]int, jumlahElemen)

    for i := 0; i < jumlahElemen; i++ {

        fmt.Printf("Masukkan elemen ke-%d: ", i+1)

        fmt.Scan(&kaen[i])

    }

    fmt.Println("\nIsi array:", kaen)

    tampilkanIndeks(kaen)

    fmt.Printf("Rata-rata: %.2f\n", hitungRataRata(kaen))

    fmt.Printf("Simpangan baku (tanpa akar): %.2f\n", hitungSimpanganBaku(kaen))

    var angkaUntukFrekuensi int

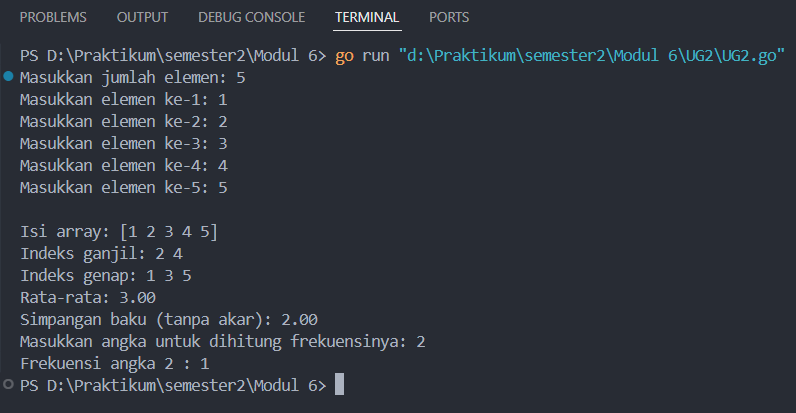
    fmt.Print("Masukkan angka untuk dihitung frekuensinya: ")

    fmt.Scan(&angkaUntukFrekuensi)

    fmt.Println("Frekuensi angka", angkaUntukFrekuensi, ":", hitungFrekuensi(kaen, angkaUntukFrekuensi))

}

Output:



Deskripsi:  
Program ini merupakan implementasi sederhana dalam bahasa Go yang digunakan untuk melakukan **analisis data dasar pada array bilangan bulat**. Operasi yang dilakukan meliputi:

* Menghitung rata-rata.
* Menghitung simpangan baku (tanpa akar).
* Menghitung frekuensi kemunculan angka tertentu.
* Menampilkan elemen array berdasarkan indeks ganjil dan genap.

Program ini cocok digunakan untuk latihan pemrosesan array dan analisis statistik dasar dalam Go. Program ini juga menekankan pentingnya pemahaman looping, indexing, serta perhitungan matematis sederhana dalam dunia pemrograman

3

Source code:

package main

*//Setyo Nugroho*

*//103112400024*

import "fmt"

const NMAX = 100

func main() {

    var klubA, klubB string

    var hasilPertandingan [NMAX]string

    var skorA, skorB int

    var matchCount int = 0

    fmt.Print("Klub A : ")

    fmt.Scan(&klubA)

    fmt.Print("Klub B : ")

    fmt.Scan(&klubB)

    for {

        matchCount++

        fmt.Printf("Pertandingan %d : ", matchCount)

        fmt.Scan(&skorA, &skorB)

        if skorA < 0 || skorB < 0 {

            matchCount--

            break

        }

        if skorA > skorB {

            hasilPertandingan[matchCount-1] = klubA

        } else if skorB > skorA {

            hasilPertandingan[matchCount-1] = klubB

        } else {

            hasilPertandingan[matchCount-1] = "Draw"

        }

    }

    for i := 0; i < matchCount; i++ {

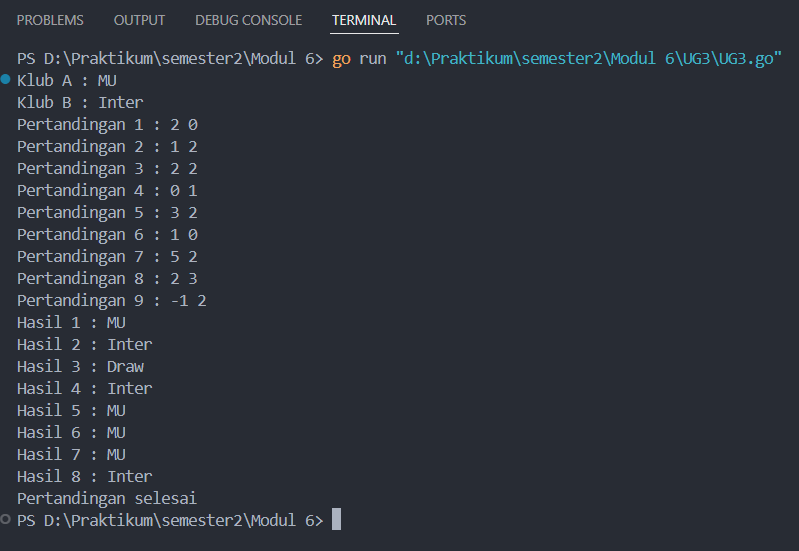
        fmt.Printf("Hasil %d : %s\n", i+1, hasilPertandingan[i])

    }

    fmt.Println("Pertandingan selesai")

}

Output:



Deskripsi:

Program ini digunakan untuk **mencatat hasil pertandingan antara dua klub sepak bola**. Program akan meminta input nama dua klub dan skor pertandingan yang dilakukan secara berulang hingga pengguna memasukkan skor negatif sebagai tanda berhenti.

Program ini menunjukkan implementasi array satu dimensi dan penggunaan perulangan dalam pencatatan data. Dengan kontrol input yang sederhana (menggunakan skor negatif untuk berhenti), program ini cocok sebagai contoh aplikasi praktis array dalam dunia olahraga.

4

Source Code

package main

*//Setyo Nugroho*

*//103112400024*

import "fmt"

const NMAX int = 127

type tabel [NMAX]rune

func isiArray(t \*tabel, n \*int) {

    var Teks rune

    \*n = 0

    fmt.Print("Teks: ")

    for {

        fmt.Scanf("%c", &Teks)

        if Teks == '.' || \*n > NMAX {

            break

        }

        if Teks != ' ' && Teks != '\n' {

            (\*t)[\*n] = Teks

            \*n++

        }

    }

}

func cetakArray(t tabel, n int) {

    fmt.Print("Reverse teks: ")

    for i := 0; i < n; i++ {

        fmt.Printf("%c ", t[i])

    }

    fmt.Println()

}

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]

    }

}

func palindrom(t tabel, n int) bool {

    var tabClone tabel

    copy(tabClone[:], t[:])

    balikanArray(&tabClone, n)

    for i := 0; i < n; i++ {

        if t[i] != tabClone[i] {

            return false

        }

    }

    return true

}

func main() {

    var tab tabel

    var m int

    isiArray(&tab, &m)

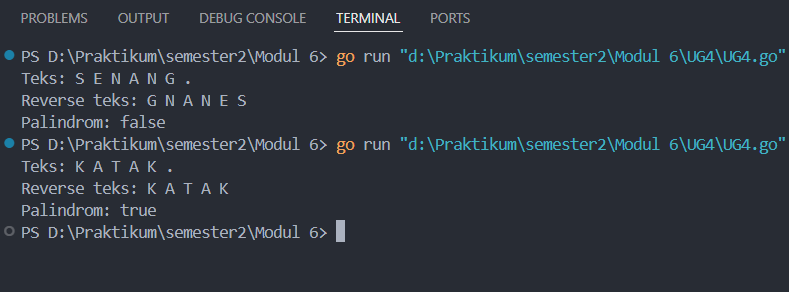
    balikanArray(&tab, m)

    cetakArray(tab, m)

    fmt.Print("Palindrom: ", palindrom(tab, m))

}

Output



Kesimpulan

Program ini digunakan untuk:

1. **Membaca karakter dari input hingga bertemu tanda titik (.)**
2. **Menampilkan hasil pembalikan teks** (reverse)
3. **Memeriksa apakah input tersebut adalah palindrom**

**Program ini menunjukkan penggunaan:**

* **Array bertipe rune untuk membaca karakter**
* **Operasi pembalikan string**
* **Pengecekan palindrom**
* **Input karakter satu per satu menggunakan Scanf("%c")**

**Cocok digunakan untuk memahami manipulasi string dasar dan logika dalam array karakter di bahasa Go**

### ****Kesimpulan**** Berdasarkan hasil praktikum yang telah dilakukan, dapat disimpulkan bahwa struct dan array merupakan dua konsep penting dalam pemrograman yang sering digunakan secara bersamaan untuk mengelola data yang kompleks. Struct memungkinkan pengelompokan berbagai tipe data dalam satu wadah, sedangkan array memudahkan penyimpanan banyak data dengan tipe yang sama secara terstruktur. Dalam praktiknya, penggunaan array of struct sangat membantu dalam penyimpanan dan pengolahan data-data yang saling berkaitan, seperti data mahasiswa, data barang, dan sebagainya. Melalui praktikum ini, pemahaman tentang cara deklarasi, inisialisasi, dan manipulasi data menggunakan struct dan array dapat diterapkan dengan baik. Selain itu, pemahaman logika program dan manajemen data dalam skala kecil juga semakin meningkat.REFERENSI

Modul 6