LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2

MODUL 6

DISUSUN OLEH:

RIZKINA AZIZAH

103112400082

S1 IF-12-01

PROGRAM STUDI S1 TEKNIK INFORMATIKA

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2025

DASAR TEORI

A. Struct

Struct adalah kumpulan definisi variabel (atau property) dan atau fungsi (atau method), yang dibungkus sebagai tipe data baru dengan nama tertentu. Property dalam struct, tipe datanya bisa bervariasi. Mirip seperti map, hanya saja key-nya sudah didefinisikan di awal, dan tipe data tiap itemnya bisa berbeda. Kombinasi keyword type dan struct digunakan untuk deklarasi struct. Cara inisialisasi variabel objek adalah dengan menuliskan nama struct yang telah dibuat diikuti dengan kurung kurawal. Nilai masing-masing property bisa diisi pada saat inisialisasi. Objek yang dibuat dari tipe struct bisa diambil nilai pointer-nya, dan bisa disimpan pada variabel objek yang bertipe struct pointer. Embedded struct adalah mekanisme untuk menempelkan sebuah struct sebagai properti struct lain. Jika salah satu nama properti sebuah struct memiliki kesamaan dengan properti milik struct lain yang di-embed, maka pengaksesan property-nya harus dilakukan secara eksplisit atau jelas. Pengisian nilai property sub-struct bisa dilakukan dengan langsung memasukkan variabel objek yang tercetak dari struct yang sama Anonymous struct adalah struct yang tidak dideklarasikan di awal sebagai tipe data baru, melainkan langsung ketika pembuatan objek. Teknik ini cukup efisien digunakan pada *use case* pembuatan variabel objek yang struct-nya hanya dipakai sekali.

Slice dan struct bisa dikombinasikan seperti pada slice dan map, caranya penggunaannya-pun mirip, cukup tambahkan tanda [] sebelum tipe data pada saat deklarasi. Anonymous struct bisa dijadikan sebagai tipe sebuah slice. Dan nilai awalnya juga bisa diinisialisasi langsung pada saat deklarasi.

B. Array

Array adalah kumpulan data bertipe sama, yang disimpan dalam sebuah variabel. Array memiliki kapasitas yang nilainya ditentukan pada saat pembuatan, menjadikan elemen/data yang disimpan di array tersebut jumlahnya tidak boleh melebihi yang sudah dialokasikan. Pengisian elemen array pada indeks yang tidak sesuai dengan jumlah alokasi menghasilkan error. Pengisian elemen array bisa dilakukan pada saat deklarasi variabel. Caranya dengan menuliskan data elemen dalam kurung kurawal setelah tipe data, dengan pembatas antar elemen adalah tanda koma (,). Deklarasi array yang nilainya diset di awal, boleh tidak dituliskan jumlah

lebar array-nya, cukup ganti dengan tanda 3 titik (...). Metode penulisan ini membuat kapasitas array otomatis dihitung dari jumlah elemen array yang ditulis.

Array multidimensi adalah array yang tiap elemennya juga berupa array. Cara deklarasi array multidimensi secara umum sama dengan array biasa, bedanya adalah pada array biasa, setiap elemen berisi satu nilai, sedangkan pada array multidimensi setiap elemen berisi array.

GUIDED

1. Guided 1

Source Code:

```
package main

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()
}
```

Deskripsi Program:

- Program ini digunakan **untuk mencetak struk belanja otomatis** yang menampilkan informasi toko, tanggal dan waktu transaksi, daftar item belanja (nama, harga, jumlah, total per item), dan total keseluruhan belanja.
- *Struct Item* berfungsi untuk menyimpan informasi barang belanja:
  - *Nama* barang
  - *Price* harga satuan.
  - *Quantity* Jumlah barang dibeli
  - .
- *Struct Receipt* berfungsi untuk menyimpan data struk belanja:
  - *StoreInfo* Informasi nama dan alamat toko.
  - *Date* Waktu transaksi belanja (dalam format waktu sistem).
  - *Items* Slice berisi kumpulan data Item.
  - *TotalAmount* Total belanja (dihitung dari semua item)
- *CalculateTotal* berfungsi untuk
  - Menghitung total harga dari seluruh item dalam *Items.*
  - Melakukan perkalian antara harga dan jumlah tiap item.
  - Menyimpan hasil total ke dalam *TotalAmount.*
- *PrintReceipt* berfungsi untuk:
  - Menampilkan struk belanja ke layar dalam format rapi.

- Menampilkan informasi toko dan tanggal transaksi.
- Menampilkan daftar barang, harga, jumlah, dan total per barang.
- Menampilkan total belanja secara keseluruhan.
- Menampilkan ucapan terima kasih.

- *func main* berfungsi untuk:
  - Membuat objek *Receipt* berisi data toko, waktu saat ini, dan daftar barang belanja.
  - Memanggil *CalculateTotal* untuk menghitung total belanja dari data item.
  - Memanggil *PrintReceipt* untuk mencetak struk ke layar.

2. Guided 2

Source Code:

```
package main

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])
    }
}
```

Deskripsi Program:

- Program ini digunakan untuk menampilkan data nilai mahasiswa, menghitung rata-rata, serta mencari nilai tertinggi dan terendah. Program juga menampilkan penggunaan array 2 dimensi untuk menyimpan nilai beberapa mata kuliah.
- Fitur Program :
    a. Menampilkan Nilai Mahasiswa (Array 1 Dimensi)
    b. Menghitung rata-rata
    c. Mennetukan nilai tertinggi dan terendah
    d. Penggunaan array 2 dimensi
- *func main* berfungsi untuk:

- Menyimpan dan mengelola data nilai mahasiswa.
- Menampilkan nilai, menghitung rata-rata, nilai maksimum, dan minimum.
- Menampilkan penggunaan array 2 dimensi dalam menyimpan nilai antar mata kuliah.

UNGUIDED

1. Unguided 1

Source Code:

```
package main

import (
    "fmt"
    "math"
)

type Titik struct {
    x int
    y int
}

type Lingkaran struct {
    pusat  Titik
    radius int
}

func jarak(p, q Titik) float64 {
    return math.Sqrt(float64((p.x-q.x)*(p.x-q.x) + (p.y-q.y)*(p.y-q.y)))
}

func didalam(c Lingkaran, p Titik) bool {
    return jarak(c.pusat, p) < float64(c.radius)
}

func main() {
    var cx1, cy1, r1 int
    var cx2, cy2, r2 int
    var x, y int
```

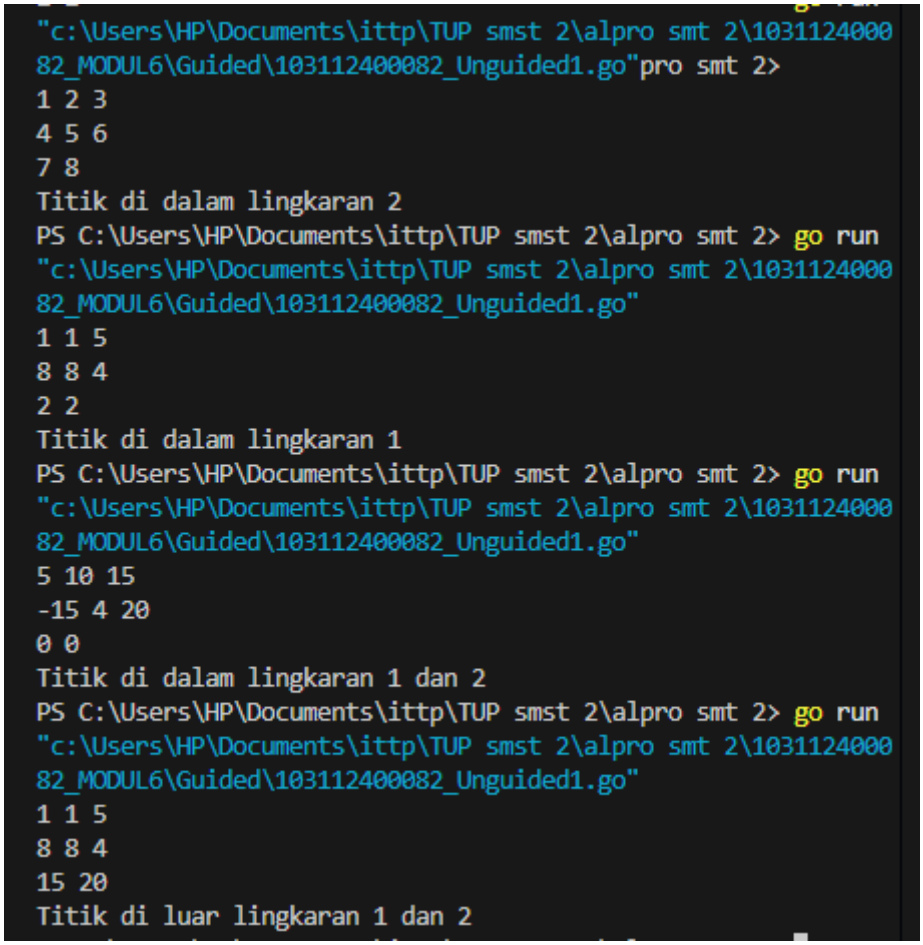
```
    fmt.Scan(&cx1, &cy1, &r1)
    fmt.Scan(&cx2, &cy2, &r2)
    fmt.Scan(&x, &y)

    lingkaran1 := Lingkaran{Titik{cx1, cy1}, r1}
    lingkaran2 := Lingkaran{Titik{cx2, cy2}, r2}
    titikSembarang := Titik{x, y}

    inLingkaran1 := didalam(lingkaran1, titikSembarang)
    inLingkaran2 := didalam(lingkaran2, titikSembarang)

    if inLingkaran1 && inLingkaran2 {
       fmt.Println("Titik di dalam lingkaran 1 dan 2")
    } else if inLingkaran1 {
       fmt.Println("Titik di dalam lingkaran 1")
    } else if inLingkaran2 {
       fmt.Println("Titik di dalam lingkaran 2")
    } else {
       fmt.Println("Titik di luar lingkaran 1 dan 2")
    }
}
```

Output:



Deskripsi Program:

- Program ini digunakan untuk menentukan apakah sebuah titik berada di dalam satu atau dua buah lingkaran berdasarkan posisi dan jari-jari masing-masing lingkaran.
- Struct *Titik* untuk menyimpan koordinat titik dengan x dan y. Struct *Lingkaran* untuk menyimpan pusat (bertipe *Titik*) dan jari-jari *(radius)*
- *func jarak* untuk menghitung jarak Euclidean antara dua titik ppp dan qqq menggunakan rumus jarak = akar (x1−x2)2+(y1−y2)2
- *func didalam* untuk menentukan apakah titik p berada di dalam lingkaran c dan mengembalikan true jika jarak dari pusat ke titik lebih kecil dari radius lingkaran.
- *func main* berfungsi untuk :
  - Input data dua buah lingkaran (koordinat pusat dan radius).
  - Input data sebuah titik sembarang.
  - Membuat objek lingkaran dan titik menggunakan struct.
  - Memanggil fungsi didalam

2. Unguided 2

Source Code:

```
package main

import (
    "fmt"
    "math"
)

func main() {
    var N int
    fmt.Print("Masukkan jumlah elemen array: ")
    fmt.Scan(&N)

    angka := make([]int, N)

    for i := 0; i < N; i++ {
        fmt.Printf("Masukkan elemen ke-%d: ", i)
        fmt.Scan(&angka[i])
    }

    fmt.Println("Isi array:", angka)

    fmt.Print("Indeks ganjil: ")
    for i := 1; i < N; i += 2 {
        fmt.Print(angka[i], " ")
    }
    fmt.Println()

    fmt.Print("Indeks genap: ")
    for i := 0; i < N; i += 2 {
        fmt.Print(angka[i], " ")
```

```
}
fmt.Println()

var x int
fmt.Print("Masukkan bilangan x: ")
fmt.Scan(&x)
fmt.Print("Indeks kelipatan ", x, ": ")
for i := 0; i < N; i++ {
   if i%x == 0 {
      fmt.Print(angka[i], " ")
   }
}
fmt.Println()

var indeksHapus int
fmt.Print("Masukkan indeks yang ingin dihapus: ")
fmt.Scan(&indeksHapus)
angka = append(angka[:indeksHapus], angka[indeksHapus+1:]...)
fmt.Println("Isi array setelah penghapusan:", angka)

var total int
for _, v := range angka {
   total += v
}
rataRata := float64(total) / float64(len(angka))
fmt.Printf("Rata-rata: %.2f\n", rataRata)

var totalKuadrat float64
for _, v := range angka {
   totalKuadrat += math.Pow(float64(v)-rataRata, 2)
}
standarDeviasi := math.Sqrt(totalKuadrat / float64(len(angka)))
fmt.Printf("Standar deviasi: %.2f\n", standarDeviasi)
```

```
    var bilangan int
    fmt.Print("Masukkan bilangan untuk frekuensi: ")
    fmt.Scan(&bilangan)
    frekuensi := 0
    for _, v := range angka {
       if v == bilangan {
          frekuensi++
       }
    }
    fmt.Printf("Frekuensi dari %d: %d\n", bilangan, frekuensi)
}
```

Deskripsi program:

- Program ini digunakan untuk untuk menampung sekumpulan bilangan bulat.
- Fitur dalam program :
  - Menampilkan keseluruhan isi dari array.
  - Menampilkan elemen-elemen array dengan indeks ganjil saja.
  - Menampilkan elemen-elemen array dengan indeks genap saja (asumsi indek ke-0 adalahgenap).
  - Menampilkan elemen-elemen array dengan indeks kelipatan bilangan x. x bisa diperoleh dari masukan pengguna.
  - Menghapus elemen array pada indeks tertentu, asumsi indeks yang hapus selalu valid.
  - Tampilkan keseluruhan isi dari arraynya, pastikan data yang dihapus tidak tampil
  - Menampilkan rata-rata dari bilangan yang ada di dalam array.
  - Menampilkan standar deviasi atau simpangan baku dari bilangan yang ada di dalam array tersebut.
  - Menampilkan frekuensi dari suatu bilangan tertentu di dalam array yang telah diisi tersebut.

3. Unguided 3

Source Code:

```
package main

import (
    "fmt"
)

type Klub struct {
    nama string
}

type Pertandingan struct {
    klubA Klub
    klubB Klub
    skorA int
    skorB int
    pemenang string
}

func main() {
    var namaA, namaB string
    fmt.Print("Klub A: ")
    fmt.Scan(&namaA)
    fmt.Print("Klub B: ")
    fmt.Scan(&namaB)

    klubA := Klub{nama: namaA}
    klubB := Klub{nama: namaB}

    var daftarPertandingan []Pertandingan

    for {
        var skorA, skorB int
        fmt.Printf("Masukkan skor %s dan %s (format: A B): ", klubA.nama, klubB.nama)
```

```
        fmt.Scan(&skorA, &skorB)

        if skorA < 0 || skorB < 0 {
            break
        }

        p := Pertandingan{
            klubA: klubA,
            klubB: klubB,
            skorA: skorA,
            skorB: skorB,
        }

        if skorA > skorB {
            p.pemenang = klubA.nama
            fmt.Printf("Hasil: %s\n", klubA.nama)
        } else if skorB > skorA {
            p.pemenang = klubB.nama
            fmt.Printf("Hasil: %s\n", klubB.nama)
        } else {
            p.pemenang = "Draw"
            fmt.Println("Hasil: Draw")
        }

        daftarPertandingan = append(daftarPertandingan, p)
    }

    fmt.Println("Pertandingan selesai")
    fmt.Println("Daftar klub yang memenangkan pertandingan:")
    for i, p := range daftarPertandingan {
        fmt.Printf("Hasil %d : %s\n", i+1, p.pemenang)
    }
}
```

Deskripsi Program:

- Program ini digunakan untuk menyimpan dan menampilkan nama-nama klub yang memenangkan pertandingan bola pada suatu grup pertandingan.
- Input nama klub yang akan disimpan di struct *Klub*
- Penggunaan looping untuk input skor pertandingan antara Klub A dan Klub B
- Penggunaan percabangan untuk menentukan pemenang :

- Jika skorA > skorB, maka pemenangnya adalah Klub A.
- Jika skorB > skorA, maka pemenangnya adalah Klub B.
- Jika skor sama, maka hasilnya "Draw".

4. Unguided 4

Source Code:

```
package main

import "fmt"

const NMAX int = 127
type tabel [NMAX]rune

func isiArray(t *tabel, n *int) {
   var huruf rune
   *n = 0
   fmt.Print("Teks : ")
   for {
      fmt.Scanf("%c", &huruf)
      if huruf == '.' || *n >= NMAX {
         break
      }
      if huruf != '\n' && huruf != '\r' {
         t[*n] = huruf
         *n++
      }
   }
```

```
}

func cetakArray(t tabel, n int) {
    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 salin tabel

    for i := 0; i < n; i++ {
        salin[i] = t[i]
    }

    balikanArray(&salin, n)

    for i := 0; i < n; i++ {
        if t[i] != salin[i] {
            return false
        }
    }
    return true
}

func main() {
```

```
    var tab tabel
    var m int

    isiArray(&tab, &m)
    fmt.Print("Reverse teks : ")
    var salin tabel

    for i := 0; i < m; i++ {
        salin[i] = tab[i]
    }
    balikanArray(&salin, m)
    cetakArray(salin, m)

    fmt.Print("Palindrom ? ")
    if palindrom(tab, m) {
        fmt.Println("true")
    } else {
        fmt.Println("false")
    }
}
```

Deskripsi Program:

- Program ini digunakan untuk melakukan membalikkan urutan isi array dan memeriksa apakah membentuk palindrom.
- Konstanta :
  - *const NMAX = 127* → batas maksimum karakter.
  - *type tabel [NMAX]rune* → array untuk menyimpan karakter (rune = karakter Unicode).
- *func isiArray* berfungsi untuk :
  - Input karakter satu per satu menggunakan *fmt.Scanf("%c", &huruf)*.
  - Menyimpan ke array jika bukan newline *(\n)* atau carriage return *(\r)*.
  - Berhenti jika:
    - Karakter **.** ditemukan (sebagai penanda akhir input), atau

    - Jumlah karakter mencapai batas maksimum *(NMAX).*
- *func cetakArray* berfungsi untuk menampilkan seluruh isi array sebagai karakter dipisahkan spasi
- *func balikanArray* berfungsi untuk ;
    - Membalik isi array dari depan ke belakang (in-place swap).
    - Digunakan untuk membalik teks.
- *func palindorm* berfungsi untuk :
    - Menyalin array karakter asli ke array baru (salin).
    - Membalik isi array salin.
    - Membandingkan isi salin dengan array asli.
    - Jika seluruh karakter sama → return true (palindrom).
- *func main* berfungsi untuk :
    - Deklarasi array tab dan variabel m (jumlah karakter).
    - Memanggil isiArray() untuk membaca input.
    - Membalik isi array dan mencetaknya sebagai "Reverse teks".
    - Mengecek apakah input palindrom, lalu mencetak hasilnya.

DAFTAR PUSTAKA

novalagung, A.24.Struct (https://dasarpemrogramangolang.novalagung.com/A-struct.html)

novalagung, A.15.Array (https://dasarpemrogramangolang.novalagung.com/A-array.html)