

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL 7
STRUCT & ARRAY**



Disusun Oleh :

Annasya Maulafidatu Zahra / 2311102246

S1-IF-11-05

Dosen Pengampu :

Arif Amrulloh, S.Kom., M.Kom.

PROGRAM STUDI S1 TEKNIK INFORMATIKA

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2024

I. DASAR TEORI

Bahasa pemrograman Go (Golang) memiliki struktur data yang kuat dan efisien, termasuk struct, array, slice, dan map. Masing-masing struktur data ini memiliki karakteristik dan kegunaan yang berbeda, yang memungkinkan pengembang untuk mengelola data dengan cara yang lebih terorganisir dan fleksibel.

1. Struct

Struct adalah tipe data komposit di Go yang digunakan untuk mengelompokkan variabel-variabel (atau properti) dengan tipe data yang berbeda dalam satu entitas. Struct mirip dengan kelas dalam bahasa pemrograman berorientasi objek, meskipun Go tidak mendukung konsep kelas secara langsung.

2. Array

Array adalah kumpulan elemen dengan tipe data yang sama, disimpan dalam satu variabel. Ukuran array ditentukan pada saat deklarasi dan tidak dapat diubah setelahnya.

3. Slice

Slice adalah struktur data yang lebih fleksibel dibandingkan array. Slice adalah representasi dinamis dari array yang memungkinkan penambahan elemen secara dinamis tanpa harus menentukan ukuran sebelumnya.

4. Map

Map adalah struktur data yang menyimpan pasangan key-value. Map sangat berguna ketika kita perlu mengasosiasikan nilai dengan kunci tertentu.

Struct, array, slice, dan map adalah komponen dasar dalam pengembangan aplikasi menggunakan bahasa Go. Memahami cara kerja masing-masing struktur ini sangat penting untuk memanfaatkan potensi penuh dari Go dalam pengelolaan data.

II. GUIDED

1. Guided 1

Soal Studi Case

Siswa kelas IPA di salah satu sekolah menengah atas di Indonesia sedang mengadakan praktikum kimia. Di setiap percobaan akan menggunakan 4 tabung reaksi, yang mana susunan warna cairan di setiap tabung akan menentukan hasil percobaan. Siswa diminta untuk mencatat hasil percobaan tersebut. Percobaan dikatakan berhasil apabila susunan warna zat cair pada gelas 1 hingga gelas 4 secara berurutan adalah 'merah', 'kuning', 'hijau', dan 'ungu' selama 5 kali percobaan berulang.

Buatlah sebuah program yang menerima input berupa warna dari ke 4 gelas reaksi sebanyak 5 kali percobaan. Kemudian program akan menampilkan true apabila urutan warna sesuai dengan informasi yang diberikan pada paragraf sebelumnya, dan false untuk urutan warna yang lainnya.

Sourcecode

```
package main

import "fmt"

type waktu struct {
    jam, menit, detik int
}

func main() {
    var wParkir, wPulang, durasi waktu
    var dParkir, dPulang, lParkir 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 // Konversi ke detik
    dPulang = wPulang.detik + wPulang.menit*60 +
wPulang.jam*3600 // detik
    lParkir = dPulang -
dParkir //detik dari
pulang-datang
```

```

    durasi.jam = lParkir / 3600
    durasi.menit = lParkir % 3600 / 60
    durasi.detik = lParkir % 3600 % 60 //17
    fmt.Printf("Lama Parkir : %d jam %d menit %d detik",
durasi.jam, durasi.menit, durasi.detik)
}

```

Screenshoot Output

```

PS D:\semester 3\Praktikum Alpro 2\modull 6> go run "d
:\semester 3\Praktikum Alpro 2\MODUL 7\guided1.go"
7 30 0
10 45 15
Lama Parkir : 3 jam 15 menit 15 detik
PS D:\semester 3\Praktikum Alpro 2\modull 6>

```

Deskripsi Program

Program ini dirancang untuk menghitung lama waktu parkir kendaraan. Pertama, program meminta pengguna untuk memasukkan waktu kedatangan dan waktu keberangkatan kendaraan dalam format jam, menit, dan detik. Kedua, semua waktu yang dimasukkan dikonversi menjadi detik untuk memudahkan perhitungan. Selanjutnya, program menghitung selisih antara waktu keberangkatan dan kedatangan dalam satuan detik. Terakhir, selisih waktu dalam detik tersebut diubah kembali menjadi format jam, menit, dan detik, kemudian ditampilkan sebagai hasil akhir.

2. Guided 2

Sourcecode

```
//Guided 2 - Map

package main

import (
    "fmt"
)

func main() {
    // Membuat map dengan nama buah sebagai kunci dan harga
    // sebagai nilai
    hargaBuah := map[string]int{
        "Apel": 5000,
        "Pisang": 3000,
        "Mangga": 7000,
    }

    // Menampilkan harga dari setiap buah
    fmt.Println("Harga Buah:")
    for buah, harga := range hargaBuah {
        fmt.Printf("%s: Rp%d\n", buah, harga)
    }

    fmt.Print("Harga buah Mangga = ", hargaBuah["Mangga"])
}

```

Screenshoot Output

```
PS D:\semester 3\Praktikum Alpro 2\modull 6> go run "d
:\semester 3\Praktikum Alpro 2\MODUL 7\guided2map.go"
Harga Buah:
Apel: Rp5000
Pisang: Rp3000
Mangga: Rp7000
Harga buah Mangga = 7000
PS D:\semester 3\Praktikum Alpro 2\modull 6> 

```

Deskripsi Program

Program ini dibuat untuk menyimpan dan menampilkan harga berbagai jenis buah. Program ini menggunakan struktur data map yang

memungkinkan kita untuk menyimpan pasangan nilai berupa nama buah sebagai kunci dan harga buah sebagai nilai. Setelah itu, program akan menampilkan daftar semua buah beserta harganya. Selain itu, program juga dapat digunakan untuk mencari harga buah tertentu, seperti harga buah mangga. Dengan kata lain, program ini berfungsi sebagai sebuah daftar harga buah sederhana yang dapat diakses dan ditampilkan sesuai kebutuhan

3. Guided 3 Sourcecode

```
//Guided 2 - Slice

package main

import (
    "fmt"
)

// Fungsi untuk mengecek apakah nama sudah ada di dalam slice
func sudahAda(daftarTeman []string, nama string) bool {
    for _, teman := range daftarTeman {
        if teman == nama {
            return true
        }
    }
    return false
}

func main() {
    // Slice awal untuk daftar teman dengan beberapa data
    daftarTeman := []string{"Andi", "Budi", "Cici"}

    // Nama-nama baru yang ingin ditambahkan
    namaBaru := []string{"Dewi", "Budi", "Eka"}

    // Menambahkan nama baru hanya jika belum ada di daftar
    for _, nama := range namaBaru {
        if !sudahAda(daftarTeman, nama) {
            daftarTeman = append(daftarTeman, nama)
        } else {
            fmt.Println("Nama", nama, "sudah ada dalam daftar.")
        }
    }
}
```

```

    }

    // Menampilkan daftar teman akhir
    fmt.Println("Daftar Teman:", daftarTeman)
}

```

Screenshoot Output

```

PS D:\semester 3\Praktikum Alpro 2\modul1 6> go run "d
:\semester 3\Praktikum Alpro 2\MODUL 7\guided2slice.go
"
Nama Budi sudah ada dalam daftar.
Daftar Teman: [Andi Budi Cici Dewi Eka]
PS D:\semester 3\Praktikum Alpro 2\modul1 6> 

```

Deskripsi Program

Program ini berfungsi untuk mengelola daftar teman. Program ini dimulai dengan sebuah daftar teman yang sudah ada. Kemudian, program akan mencoba menambahkan nama-nama teman baru ke dalam daftar tersebut. Namun, sebelum menambahkan, program akan terlebih dahulu memeriksa apakah nama yang akan ditambahkan sudah ada atau belum dalam daftar. Jika nama tersebut belum ada, maka nama itu akan ditambahkan ke dalam daftar. Sebaliknya, jika nama tersebut sudah ada, maka program akan memberikan pesan bahwa nama tersebut sudah ada dalam daftar. Setelah semua nama baru diperiksa dan ditambahkan (jika perlu), program akan menampilkan daftar teman yang sudah diperbarui, yang berisi semua nama teman yang unik.

I. UNGUIDED

1. Unguided 1

Soal Studi Case

Suatu lingkaran didefinisikan dengan koordinat titik pusat (cx, cy) dengan radius r . Apabila diberikan dua buah lingkaran, maka tentukan posisi sebuah titik sembarang (x, y) berdasarkan dua lingkaran tersebut. Gunakan tipe bentukan titik untuk menyimpan koordinat, dan tipe bentukan lingkaran

untuk menyimpan titik pusat lingkaran dan radiusnya.

Masukan terdiri dari beberapa tiga baris. Baris pertama dan kedua adalah koordinat titik pusat dan radius dari lingkaran 1 dan lingkaran 2, sedangkan

baris ketiga adalah koordinat titik sembarang.

Keluaran berupa string yang menyatakan posisi titik:

- ☐ "Titik di dalam lingkaran 1 dan 2"
- ☐ "Titik di dalam lingkaran 1"
- ☐ "Titik di dalam lingkaran 2"
- ☐ "Titik di luar lingkaran 1 dan 2"

Sourcecode

```
package main

import (
    "fmt"
    "math"
)

type Point struct {
    x float64
    y float64
}

type Circle struct {
    center Point
    radius float64
}

func (c Circle) contains(p Point) bool {
```



```

        distance := math.Sqrt(math.Pow(p.x-c.center.x, 2) +
math.Pow(p.y-c.center.y, 2))
        return distance < c.radius
    }

func main() {
    var circle1, circle2 Circle
    var point Point

    fmt.Scanf("%f %f %f", &circle1.center.x,
&circle1.center.y, &circle1.radius)

    fmt.Scanf("%f %f %f", &circle2.center.x,
&circle2.center.y, &circle2.radius)

    fmt.Scanf("%f %f", &point.x, &point.y)

    inCircle1 := circle1.contains(point)
    inCircle2 := circle2.contains(point)

    if inCircle1 && inCircle2 {
        fmt.Println("Titik di dalam lingkaran 1 dan 2")
    } else if inCircle1 {
        fmt.Println("Titik di dalam lingkaran 1")
    } else if inCircle2 {
        fmt.Println("Titik di dalam lingkaran 2")
    } else {
        fmt.Println("Titik di luar lingkaran 1 dan 2")
    }
}

```

Screenshoot Output

```

Titik di luar lingkaran 1 dan 2
PS D:\semester 3\Praktikum Alpro 2\modull 6> go run "d:\semester 3\Praktikum Alpro 2\MODUL 7\unguided1.go"
1 5
1 5
1 5
Titik di luar lingkaran 1 dan 2
PS D:\semester 3\Praktikum Alpro 2\modull 6>

```

Deskripsi Program

Program ini dirancang untuk menentukan letak suatu titik terhadap dua buah lingkaran yang berbeda. Pengguna diminta untuk memasukkan koordinat pusat dan jari-jari dari masing-masing lingkaran, serta koordinat dari suatu titik. Program kemudian akan menghitung jarak antara titik tersebut dengan pusat masing-masing lingkaran. Jika jarak antara titik dengan pusat lingkaran lebih kecil dari jari-jari lingkaran, maka titik tersebut berada di dalam lingkaran. Setelah itu, program akan memberikan informasi apakah titik tersebut berada di dalam lingkaran pertama, kedua, keduanya, atau di luar kedua lingkaran.

2. Unguided 2

Soal Studi Case

Setiap hari Pak Andi membawa banyak barang belanjaan dari pasar dengan mengendarai sepeda motor. Barang belanjaan tersebut dibawa dalam kantong terpal di kiri-kanan motor. Sepeda motor tidak akan oleng jika selisih berat barang di kedua kantong sisi tidak lebih dari 9 kg.

Buatlah program Pak Andi yang menerima input dua buah bilangan real positif yang menyatakan berat total masing-masing isi kantong terpal. Program akan terus meminta input bilangan tersebut hingga salah satu kantong terpal berisi 9kg atau lebih.

Lalu modifikasilah program tersebut, program akan menampilkan true jika selisih kedua isi kantong lebih dari atau sama dengan 9 kg. Program berhenti memproses apabila total berat isi kedua kantong melebihi 150kg atau salah satu kantong beratnya negatif. Soal Studi Case

Sebuah array digunakan untuk menampung sekumpulan bilangan bulat.

Buatlah program yang digunakan untuk mengisi array tersebut sebanyak N elemen nilai. Asumsikan array memiliki kapasitas penyimpanan data sejumlah elemen tertentu. Program dapat menampilkan beberapa informasi berikut:

- Menampilkan keseluruhan isi dari array.
- Menampilkan elemen-elemen array dengan indeks ganjil saja.
- Menampilkan elemen-elemen array dengan indeks genap saja (asumsi indeks ke-0 adalah genap).
- Menampilkan elemen-elemen array dengan indeks kelipatan bilangan 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.

- f. Menampilkan rata-rata dari bilangan yang ada di dalam array.
- g. Menampilkan standar deviasi atau simpangan baku dari bilangan yang ada di dalam array tersebut.
- h. Menampilkan frekuensi dari suatu bilangan tertentu di dalam array yang telah diisi tersebut.

Sourcecode

```
package main

import (
    "fmt"
    "math"
)

func displayArray(arr []int) {
    fmt.Println("Isi array:", arr)
}

func displayOddIndices(arr []int) {
    fmt.Print("Elemen dengan indeks ganjil: ")
    for i := 1; i < len(arr); i += 2 {
        fmt.Print(arr[i], " ")
    }
    fmt.Println()
}

func displayEvenIndices(arr []int) {
    fmt.Print("Elemen dengan indeks genap: ")
    for i := 0; i < len(arr); i += 2 {
        fmt.Print(arr[i], " ")
    }
    fmt.Println()
}

func displayMultiples(arr []int, x int) {
    fmt.Print("Elemen dengan indeks kelipatan ", x, ": ")
    for i := 0; i < len(arr); i++ {
        if i%x == 0 {
            fmt.Print(arr[i], " ")
        }
    }
}
```

```

    fmt.Println()
}

func removeElement(arr []int, index int) []int {
    return append(arr[:index], arr[index+1:]...)
}

func calculateAverage(arr []int) float64 {
    sum := 0
    for _, value := range arr {
        sum += value
    }
    return float64(sum) / float64(len(arr))
}

func calculateStandardDeviation(arr []int, average float64)
float64 {
    sum := 0.0
    for _, value := range arr {
        sum += math.Pow(float64(value)-average, 2)
    }
    return math.Sqrt(sum / float64(len(arr)))
}

func frequency(arr []int, number int) int {
    count := 0
    for _, value := range arr {
        if value == number {
            count++
        }
    }
    return count
}

func main() {
    var n, x, index, number int

    fmt.Print("Masukkan jumlah elemen array: ")
    fmt.Scan(&n)

    arr := make([]int, n)

    fmt.Println("Masukkan elemen array:")
    for i := 0; i < n; i++ {
        fmt.Scan(&arr[i])
    }
}

```

```

    }

    displayArray(arr)
    displayOddIndices(arr)
    displayEvenIndices(arr)
    fmt.Print("Masukkan bilangan x untuk kelipatan: ")
    fmt.Scan(&x)
    displayMultiples(arr, x)

    fmt.Print("Masukkan indeks yang ingin dihapus: ")
    fmt.Scan(&index)
    arr = removeElement(arr, index)
    displayArray(arr)

    average := calculateAverage(arr)
    fmt.Printf("Rata-rata: %.2f\n", average)

    stdDev := calculateStandardDeviation(arr, average)
    fmt.Printf("Standar deviasi: %.2f\n", stdDev)

    fmt.Print("Masukkan bilangan untuk menghitung frekuensi: ")
    fmt.Scan(&number)
    freq := frequency(arr, number)
    fmt.Printf("Frekuensi bilangan %d: %d\n", number, freq)
}

```

Screenshoot Output

```

PS D:\semester 3\Praktikum Alpro 2\modul1 > go run "d:\semester 3\Praktikum Alpro 2\MODUL 7\unguided2.go"
Masukkan jumlah elemen array: 3
Masukkan elemen array:
4
5
6
Isi array: [4 5 6]
Elemen dengan indeks ganjil: 5
Elemen dengan indeks genap: 4 6
Masukkan bilangan x untuk kelipatan: 1
Elemen dengan indeks kelipatan 1: 4 5 6
Masukkan indeks yang ingin dihapus: 1
Isi array: [4 6]
Rata-rata: 5.00
Standar deviasi: 1.00
Masukkan bilangan untuk menghitung frekuensi: 5
Frekuensi bilangan 5: 0

```

Deskripsi Program

Program ini dirancang untuk melakukan berbagai operasi pada sebuah array. Pertama, program memungkinkan pengguna untuk memasukkan

jumlah elemen dan nilai-nilai elemen array. Kemudian, program dapat menampilkan seluruh elemen array, elemen dengan indeks ganjil, elemen dengan indeks genap, dan elemen dengan indeks kelipatan suatu bilangan tertentu. Selanjutnya, program dapat menghapus elemen pada indeks tertentu dari array. Selain itu, program juga dapat menghitung rata-rata dan standar deviasi dari elemen-elemen array. Terakhir, program dapat menghitung frekuensi kemunculan suatu bilangan dalam array. Jadi, program ini merupakan alat yang berguna untuk menganalisis dan memanipulasi data dalam bentuk array.

3. Unguided 3

Soal Studi Case

Sebuah program digunakan untuk menyimpan dan menampilkan nama-nama klub yang memenangkan pertandingan bola pada suatu grup pertandingan. Buatlah program yang digunakan untuk merekap skor pertandingan bola 2 buah klub bola yang berlaga.

Pertama-tama program meminta masukan nama-nama klub yang bertanding, kemudian program meminta masukan skor hasil pertandingan kedua klub tersebut. Yang disimpan dalam array adalah nama-nama klub yang menang saja. Proses input skor berhenti ketika skor salah satu atau kedua klub tidak valid (negatif). Di akhir program, tampilkan daftar klub yang menang

Sourcecode

```
package main

import (
    "fmt"
)

func main() {
    var klubA, klubB string
    var skorA, skorB int
    var hasil []string

    fmt.Print("Klub A: ")
    fmt.Scan(&klubA)
    fmt.Print("Klub B: ")
    fmt.Scan(&klubB)

    const maxPertandingan = 9
```

```

i := 1

for i <= maxPertandingan {

    fmt.Printf("Pertandingan %d: ", i)
    _, err := fmt.Scan(&skorA, &skorB)

    if err != nil {
        fmt.Println("Input tidak valid. Silakan masukkan
skor dalam format angka.")
        continue
    }

    if skorA < 0 || skorB < 0 {
        fmt.Println("Skor negatif tidak diperbolehkan.
Pertandingan dihentikan.")
        break
    }

    if skorA > skorB {
        hasil = append(hasil, klubA)
    } else if skorA < skorB {
        hasil = append(hasil, klubB)
    } else {
        hasil = append(hasil, "Draw")
    }

    i++
}

fmt.Println("Pertandingan selesai")
for j := 0; j < len(hasil); j++ {
    fmt.Printf("Hasil %d = %s\n", j+1, hasil[j])
}
}

```

Screenshoot Output

```

PS D:\semester 3\Praktikum Alpro 2\modull 6> go run "d:\semester 3\Praktikum Alpro 2\MODUL 7\unguided3.go"
Klub A: MU
Klub B: Inter
Pertandingan 1: 2 0
Pertandingan 2: 1 2
Pertandingan 3: 2 2
Pertandingan 4: 0 1
Pertandingan 5: 3 2
Pertandingan 6: 1 0
Pertandingan 7: 5 2
Pertandingan 8: 2 3
Skor negatif tidak diperbolehkan. Pertandingan dihentikan.
Pertandingan selesai
Hasil 1 = MU
Hasil 2 = Inter
Hasil 3 = Draw
Hasil 4 = Inter
Hasil 5 = MU
Hasil 6 = MU
Hasil 7 = MU
Hasil 8 = Inter
PS D:\semester 3\Praktikum Alpro 2\modull 6> 

```

Deskripsi Program

Program ini dirancang untuk mensimulasikan sebuah pertandingan sepak bola antara dua klub. Pengguna diminta untuk memasukkan nama kedua klub dan kemudian memasukkan skor untuk setiap pertandingan hingga maksimal 9 pertandingan. Program akan terus meminta input skor hingga pengguna memasukkan skor negatif atau mencapai batas maksimal pertandingan. Setelah semua pertandingan selesai, program akan menampilkan hasil setiap pertandingan, yaitu nama klub yang menang atau "Draw" jika terjadi seri. Program ini juga dilengkapi dengan validasi input untuk memastikan data yang dimasukkan oleh pengguna valid dan sesuai dengan aturan permainan sepak bola.

4. Unguided 4

Soal Studi Case

Sebuah array digunakan untuk menampung sekumpulan karakter, Anda diminta untuk membuat sebuah subprogram untuk melakukan membalikkan urutan isi array dan memeriksa apakah membentuk palindrom.

Sourcecode

```

package main

import (
    "fmt"
    "unicode"
)

```



```

const NMAX int = 127

type tabel [NMAX]rune

func isiArray(t *tabel, n *int) {
    fmt.Println("Masukkan karakter (tuliskan '.' untuk
berhenti):")
    for *n < NMAX {
        var karakter rune
        fmt.Scanf("%c", &karakter)
        if unicode.IsSpace(karakter) {
            continue
        }
        if karakter == '.' {
            break
        }
        t[*n] = unicode.ToLower(karakter)
        (*n)++
    }
}

func cetakArray(t tabel, n int) {
    for indeks := 0; indeks < n; indeks++ {
        fmt.Printf("%c", t[indeks])
    }
    fmt.Println()
}

func balikanArray(awal tabel, terbalik *tabel, n int) {
    for i := 0; i < n; i++ {
        terbalik[i] = awal[n-1-i]
    }
}

func palindrom(awal tabel, terbalik tabel, n int) bool {
    for indeks := 0; indeks < n; indeks++ {
        if awal[indeks] != terbalik[indeks] {
            return false
        }
    }
    return true
}

func main() {

```

```

var tab tabel
var tabkebalik tabel
var jumlahKarakter int = 0

isiArray(&tab, &jumlahKarakter)

fmt.Print("Array: ")
cetakArray(tab, jumlahKarakter)

balikanArray(tab, &tabkebalik, jumlahKarakter)

fmt.Print("Array yang sudah dibalik: ")
cetakArray(tabkebalik, jumlahKarakter)

if palindrom(tab, tabkebalik, jumlahKarakter) {
    fmt.Println("Array adalah palindrome.")
} else {
    fmt.Println("Array bukan palindrome.")
}
}

```

Screenshoot Output

```

Kata ini bukan palindrome.
PS D:\semester 3\Praktikum Alpro 2\modul1 6> go run "d:\semester 3\Praktikum Alpro 2\MODUL 7\unguided4.go"
Masukkan karakter (tuliskan '.' untuk berhenti):
A
N
N
A
S
Y
A
.
Array: annasya
Array yang sudah dibalik: aysanna
Array bukan palindrome.
PS D:\semester 3\Praktikum Alpro 2\modul1 6> 

```

Deskripsi Program

Program ini dirancang untuk memeriksa apakah sebuah kata atau kalimat merupakan palindrome. Program ini meminta pengguna untuk memasukkan karakter hingga pengguna memasukkan titik (.). Karakter-karakter yang dimasukkan kemudian disimpan dalam sebuah array. Setelah itu, program akan membalik urutan karakter dalam array tersebut dan membandingkan apakah array asli dan array yang dibalik identik. Jika identik, maka kata atau kalimat tersebut merupakan palindrome. Selain itu, program juga

menampilkan array asli dan array yang dibalik. Secara sederhana, program ini berfungsi untuk menentukan apakah suatu rangkaian karakter dibaca dari depan maupun dari belakang menghasilkan urutan yang sama.