

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

MODUL 7

STRUCK & ARRAY



Disusun Oleh :

Natasya Intan Sukma Jiwanti / 2311102279

S1-IF-11-05

Dosen Pengampu :

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

PROGRAM STUDI S1 TEKNIK INFORMATIKA

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2024

DASAR TEORI

1. Struct

Struct adalah tipe data yang mengelompokkan beberapa field dengan tipe data yang berbeda menjadi satu kesatuan. Mirip dengan class di bahasa OOP lainnya.

Karakteristik struct:

1. Bisa memiliki field dengan tipe data berbeda
2. Bisa memiliki method
3. Mendukung embedding untuk komposisi
4. Field bisa diekspos (huruf kapital) atau private (huruf kecil)

2. Array

Array adalah struktur data yang menyimpan kumpulan elemen dengan tipe data yang sama dan memiliki ukuran tetap.

Karakteristik array:

1. Ukurannya tetap dan harus ditentukan saat deklarasi
2. Indeks dimulai dari 0
3. Semua elemen harus memiliki tipe data yang sama

3. Slice

Struktur data mirip array yang lebih fleksibel dikarenakan ukurannya dapat diatur secara dinamis dan tidak tetap.

Karakteristik slice:

1. Ukurannya dinamis
2. Bisa diperpanjang menggunakan fungsi `append()`

4. Map

Struktur data yang menyimpan pasangan key-value, dimana key diharuskan untuk unik dan value dapat berupa tipe data.

Karakteristik map:

1. Key harus unik dan comparable
2. Value dapat berupa tipe data apapun
3. Ukuran dinamis
4. Tidak berurutan

GUIDED

1. Guided 1

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
    dPulang = wPulang.detik + wPulang.menit*60 +
wPulang.jam*3600
    lParkir = dPulang-
dParkir

    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)
}
```

Screenshot Output

The screenshot shows a terminal window with a light yellow background. At the top, there are tabs for 'PROBLEMS' (with a yellow circle containing '18'), 'OUTPUT', 'DEBUG CONSOLE', 'TERMINAL' (which is selected and underlined), and 'PORTS'. The terminal content shows the execution of a Go program. It starts with the command 'PS D:\Praktikum Alpro\praktikum 5\modul 7> go run "d:\Praktikum Alpro\praktikum 5\modul 7\tempCodeRunnerFile.go"'. The output consists of three lines of numbers: '1 30 43', '6 23 12', and 'Lama Parkir : 4 jam 52 menit 29 detik'. The prompt 'PS D:\Praktikum Alpro\praktikum 5\modul 7>' is shown again at the end, followed by a black cursor block.

```
PROBLEMS 18 OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS D:\Praktikum Alpro\praktikum 5\modul 7> go run "d:\Praktikum Alpro\praktikum 5\modul 7\tempCodeRunnerFile.go"
1 30 43
6 23 12
Lama Parkir : 4 jam 52 menit 29 detik
PS D:\Praktikum Alpro\praktikum 5\modul 7> █
```

Deskripsi Program

Program diatas merupakan program yang digunakan untuk menghitung lamanya penggunaan lahan parkir seseorang. Penggunaan tipe data struct pada program sangat membantu dan meningkatkan efisiensi program. Hal tersebut dikarenakan dalam satu struct kita dapat menyimpan 3 nilai sekaligus, yakni jam, menit dan detik. Pada perhitungan lamanya lahan parkir digunakan program akan meminta inputan kepada user jam datang dan jam pulang dengan format jam menit detik, yang kemudia jam dan menit yang diinputkan akan dikonversi oleh program menjadi detik supaya dapat dihitung selisih nya. Kemudian selisih tersebut akan dikonversikan kembali dengan format jam menit dan detik.

2. Guided 2

Sourcecode

```
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:\Praktikum Alpro\praktikum 5\modul 7> go run "d:\Praktikum Alpro\praktikum 5\modul 7\guided2-map.go"
Harga Buah:
Apel: Rp5000
Pisang: Rp3000
Mangga: Rp7000
Harga buah Mangga = 7000
PS D:\Praktikum Alpro\praktikum 5\modul 7> []
```

Deskripsi Program

Program diatas merupakan program yang dibuat menggunakan map dengan nama harga buah. Key pada map diatas bertipe data string yakni nama buah. Sedangkan, nilai atau value yang terdapat pada map diatas adalah harga dari buah itu sendiri dengan tipe data interger. Untuk mencetak nilai dari hargabuah nya kita akan menggunakan perulangan. Sehingga pada setiap perulangan akan mencetak buah dan harganya. Untuk memanggil value pada map kita juga dapat menggunakan kuncinya.

3. Guided 3

Sourcecode

```
package main
import (
    "fmt"
)
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.")
    }
}
fmt.Println("Daftar Teman:", daftarTeman)
}
```

Screenshot Output

```
PS D:\Praktikum Alpro\praktikum 5\modul 7> go run "d:\Praktikum Alpro\praktikum 5\modul 7\tempCodeRunnerFile.go"
Nama Budi sudah ada dalam daftar.
Daftar Teman: [Andi Budi Cici Dewi Eka]
PS D:\Praktikum Alpro\praktikum 5\modul 7> █
```

Deskripsi Program

Program diatas merupakan program yang menggunakan slice sebagai tipe data dalam penyimpanannya. Pada slice sendiri untuk menambahkan elemen kita menggunakan fungsi append. Pada program diatas juga menggunakan fungsi cari nama yang digunakan untuk mengetahui apakah suatu nama sudah tersimpan pada slice atau belum. Fungsi tersebut akan dipanggil pada program main dalam kondisi percabangan, apabila nama sudah terdapat dalam slice maka program tidak akan menambahkan nama tersebut kembali, sedangkan bila belum ada maka program akan menggunakan append untuk menambahkan nama tersebut kedalam slice.

UNGUIDED

1. Unguided 1

Soal Study 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 titik struct {
    x, y float64
}

type lingkaran struct {
    px, py, r float64
}
```

```

func jarak(t1, t2 titik) float64 {
    return math.Sqrt(math.Pow(t1.x-t2.x, 2) + math.Pow(t1.y-
t2.y, 2))
}

func lokasitik(t titik, l lingkaran) bool {
    return jarak(titik{t.x, t.y}, titik{l.px, l.py}) <= l.r
}

func main() {
    var l1, l2 lingkaran
    var k titik

    fmt.Print("Masukkan Titik Pusat Lingkaran 1 (x y): ")
    fmt.Scan(&l1.px, &l1.py)
    fmt.Print("Masukkan Radius Lingkaran 1: ")
    fmt.Scan(&l1.r)
    fmt.Print("Masukkan Titik Pusat Lingkaran 2 (x y): ")
    fmt.Scan(&l2.px, &l2.py)
    fmt.Print("Masukkan Radius Lingkaran 2: ")
    fmt.Scan(&l2.r)
    fmt.Print("Masukkan Koordinat Titik (x y): ")
    fmt.Scan(&k.x, &k.y)

    if lokasitik(k, l1) && lokasitik(k, l2) {
        fmt.Println("Titik berada di Lingkaran 1 dan 2.")
    } else if lokasitik(k, l1) {
        fmt.Println("Titik berada di Lingkaran 1.")
    } else if lokasitik(k, l2) {
        fmt.Println("Titik berada di Lingkaran 2.")
    } else {
        fmt.Println("Titik berada di luar lingkaran.")
    }
}

```

Screenshoot Output

```

PS D:\Praktikum Alpro\praktikum 5\modul 7> go run "d:\Praktikum Alpro\praktikum 5\modul 7\tempCodeRunnerFile.go"
Masukkan Titik Pusat Lingkaran 1 (x y): 7 10
Masukkan Radius Lingkaran 1: 5
Masukkan Titik Pusat Lingkaran 2 (x y): -7 -9
Masukkan Radius Lingkaran 2: 10
Masukkan Koordinat Titik (x y): 2 -77
Titik berada di luar lingkaran.
PS D:\Praktikum Alpro\praktikum 5\modul 7> █

```


Deskripsi Program

Program diatas merupakan program yang digunakan untuk menghitung jarak titik pusat sebuah lingkaran dengan satu koordinat sembarang. Pada prosesnya, kita akan menggunakan struct titik untuk menyimpan data dari koordinat dan struct lingkaran untuk menyimpan data dari lingkaran seperti titik pusat dan radius. Dengan digunakannya struct ini kita tidak perlu menginisialisasi banyak variabel cukup kita inisialisasi 3 struct, yang akan digunakan untuk lingkaran 1, lingkaran 2 dan koordinat.

2. Unguided 2

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:

- a. Menampilkan keseluruhan isi dari array.
- b. Menampilkan elemen-elemen array dengan indeks ganjil saja.
- c. Menampilkan elemen-elemen array dengan indeks genap saja (asumsi indeks ke-0 adalah genap).
- d. Menampilkan elemen-elemen array dengan indeks kelipatan bilangan x bisa diperoleh dari masukan pengguna.
- e. 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 main() {
    var n int
    fmt.Print("Masukkan ukuran array: ")
    fmt.Scan(&n)
    array := make([]int, n)

    for i := 0; i < n; i++ {
        fmt.Print("Masukkan array indeks ke-%d: ", i)
        fmt.Scan(&array[i])
    }

    pilihanmenu(array)
}

func pilihanmenu(arr []int) {
    var pilihan int
    n := len(arr)

    fmt.Println("-----MENU PENGOLAHAN DATA ARRAY-----")
    fmt.Println("1. Tampilkan seluruh isi")
    fmt.Println("2. Elemen Array ber-indeks ganjil")
    fmt.Println("3. Elemen Array ber-indeks genap")
    fmt.Println("4. Tampilkan elemen array kelipatan x")
    fmt.Println("5. Hapus elemen array indeks tertentu")
    fmt.Println("6. Rata - rata isi array")
    fmt.Println("7. Simpangan baku dari array")
    fmt.Println("8. Tampilkan Frekuensi suatu bilangan")
    fmt.Println("Masukkan pilihan anda: ")
    fmt.Scan(&pilihan)

    switch pilihan {
    case 1:
        for i := 0; i < n; i++ {
            fmt.Print(arr[i], " ")
        }
        fmt.Println()
    }
```

```

case 2:
    for i := 1; i < n; i += 2 {
        fmt.Print(arr[i], " ")
    }
    fmt.Println()
case 3:
    for i := 0; i < n; i += 2 {
        fmt.Print(arr[i], " ")
    }
    fmt.Println()
case 4:
    var x int
    fmt.Print("Masukkan nilai x: ")
    fmt.Scan(&x)
    for i := 0; i < n; i++ {
        if arr[i]%x == 0 {
            fmt.Print(arr[i], " ")
        }
    }
    fmt.Println()
case 5:
    var index int
    fmt.Print("Masukkan indeks yang ingin dihapus: ")
    fmt.Scan(&index)
    if index < 0 || index >= n {
        fmt.Println("Indeks tidak valid!")
        return
    }
    newArr := make([]int, 0, n-1)
    for i := 0; i < n; i++ {
        if i != index {
            newArr = append(newArr, arr[i])
        }
    }
    fmt.Println("Array setelah penghapusan:", newArr)
case 6:
    var jumlah int
    for i := 0; i < n; i++ {
        jumlah += arr[i]
    }
    rata := float64(jumlah) / float64(n)
    fmt.Println("Rata-rata elemen array adalah:", rata)
case 7:
    var selisih, totalselisih float64
    var jumlah int

```

```

        for i := 0; i < n; i++ {
            jumlah += arr[i]
        }
        rata := float64(jumlah) / float64(n)
        for i := 0; i < n; i++ {
            selisih = float64(arr[i]) - rata
            totalselisih += selisih * selisih
        }
        simpanganbaku := math.Sqrt(totalselisih /
float64(n))
        fmt.Println("Simpangan baku pada array adalah:",
simpanganbaku)
        case 8:
            var frekuensi, x int
            fmt.Print("Masukkan nilai x: ")
            fmt.Scan(&x)
            for i := 0; i < n; i++ {
                if arr[i] == x {
                    frekuensi++
                }
            }
            fmt.Printf("Frekuensi bilangan %d pada array adalah
%d\n", x, frekuensi)
        default:
            fmt.Println("Pilihan tidak valid!")
        }
    }
}

```

Screenshoot Output

```

PS D:\Praktikum Alpro\praktikum 5\modul 7> go run "d:\Praktikum Alpro\praktikum 5\modul 7\unguided2.go"
Masukkan ukuran array: 5
Masukkan array indeks ke-0 : 1
Masukkan array indeks ke-1 : 2
Masukkan array indeks ke-2 : 3
Masukkan array indeks ke-3 : 7
Masukkan array indeks ke-4 : 9
-----MENU PENGOLAHAN DATA ARRAY-----
1. Tampilkan seluruh isi
2. Elemen Array ber-indeks ganjil
3. Elemen Array ber-indeks genap
4. Tampilkan elemen array kelipatan x
5. Hapus elemen array indeks tertentu
6. Rata - rata isi array
7. Simpangan baku dari array
8. Tampilkan Frekuensi suatu bilangan
Masukkan pilihan anda:
7
Simpangan baku pada array adalah: 3.0724582991474434
PS D:\Praktikum Alpro\praktikum 5\modul 7> █

```

Deskripsi Program

Program diatas merupakan program yang berisi pembuatan slice yang dinamis dan program manipulasi data pada slice tersebut. Slice yang digunakan pada program tersebut merupakan sebuah slice dimana ukurannya fleksibel, tergantung inputan nilai n yang dimasukkan oleh user. Pada program diatas kita menggunakan switch case dalam pemilihan menu supaya lebih mudah.

3. Unguided 3

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 berlagu.

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)
```

```

    i := 1
    for {
        fmt.Printf("Pertandingan %d: ", i)
        fmt.Scan(&skorA, &skorB)

        if skorA < 0 || skorB < 0 { // Check for termination
condition first
            break
        }

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

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

```

Screenshot Output

```

PS D:\Praktikum Alpro\praktikum 5\modul 7> go run "d:\Praktikum Alpro\praktikum 5\modul 7\tempCodeRunnerFile.go"
Klub A: MU
Klub B: Chelsea
Pertandingan 1: 0 7
Pertandingan 2: 2 1
Pertandingan 3: 3 1
Pertandingan 4: 1 1
Pertandingan 5: -1 0
Hasil 1 = Chelsea
Hasil 2 = MU
Hasil 3 = MU
Hasil 4 = Draw
PS D:\Praktikum Alpro\praktikum 5\modul 7> █

```

Deskripsi Program

Program diatas merupakan program dengan menggunakan slice karena yang disimpan adalah pemenang dari hasil pertandingan yang skornya diinputkan oleh user. Sedangkan program sendiri hanya akan berhenti meminta skor dari user apabila salah satu dari kedua tim yang telah diinputkan sebelumnya bernilai kurang dari 0, atau negatif. Pada proses append nya pun kita menggunakan percabangan, karena yang akan diinputkan pada slice

hanyalah nama dari tim yang menang. Hal itu berarti slice yang kita buat haruslah bertipe data string. Apabila yang memenangkan adalah tim a maka yang akan masuk dan tersimpan pada slice adalah nama dari tim a.

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 input rune
        fmt.Scanf("%c", &input)
        if unicode.IsSpace(input) {
            continue
        }
        if input == '.' {
            break
        }
        t[*n] = unicode.ToLower(input)
        (*n)++
    }
}

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

```

    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 i := 0; i < n; i++ {
        if awal[i] != terbalik[i] {
            return false
        }
    }
    return true
}

func main() {
    var tab tabel
    var tabkebalik tabel
    var n int = 0
    isiArray(&tab, &n)
    fmt.Print("Array: ")
    cetakArray(tab, n)
    balikanArray(tab, &tabkebalik, n)
    fmt.Print("Array yang sudah dibalik: ")
    cetakArray(tabkebalik, n)

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

```


Screenshoot Output

```
PS D:\Praktikum Alpro\praktikum 5\modul 7> go run "d:\Praktikum Alpro\praktikum 5\modul 7\tempCodeRunnerFile.go"
Masukkan karakter (tuliskan '.' untuk berhenti):
N
A
T
A
S
Y
A
.
Array: natasya
Array yang sudah dibalik: aysatan
Array bukan palindrome.
PS D:\Praktikum Alpro\praktikum 5\modul 7> go run "d:\Praktikum Alpro\praktikum 5\modul 7\tempCodeRunnerFile.go"
Masukkan karakter (tuliskan '.' untuk berhenti):
n
a
t
a
n
.
Array: natan
Array yang sudah dibalik: natan
Array adalah palindrome.
PS D:\Praktikum Alpro\praktikum 5\modul 7> █
```

Deskripsi Program

Program diatas merupakan program yang menggunakan tipe data array yang dialiaskan menjadi tabel. Sehingga menyebabkan, program diatas akan menggunakan perulangan baik itu untuk mengisi array, mencetaknya, maupun membaliknya. Array yang digunakan pada program diatas memiliki ukuran maksimal yakni 127 elemen sehingga pada pengisian array program hanya akan berhenti meminta inputan apabila user menginputkan sebanyak 127 elemen atau menginputkan titik. Setelah itu inputan dari user tidak langsung dimasukkan dalam array, namun kita ubah menjadi huruf kecil semua. Barulah kita dapat mencetak dan membalikkan array. Setelah dibalikkan nilai elemen array yang dibalikkan akan dimasukkan pada array terbalik. Untuk membaliknya kita hanya perlu menginisialisasi nilai array terbalik mulai dari nilai akhir array awal. Terakhir, untuk menentukan apakah array tersebut termasuk palindrom (apabila array terbalik = array awal) atau bukan.