

LAPORAN PRAKTIKUM
ALGORITMA PEMROGRAMAN 2
MODUL VII
STRUCT & ARRAY



Oleh :

Geranada Saputra Priambudi

2311102008

IF-11-06

Dosen Pengampu :

Abednego Dwi Septiadi S.Kom., M.Kom.

PROGRAM STUDI S1 INFORMATIKA
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY

I. DASAR TEORI

STRUCT & ARRAY

Dari bentuk tipe data dasar kita dapat menstrukturkan data sesuai dengan kebutuhan:

1. Array adalah jenis tipe data yang merupakan sekumpulan data yang bertipe data sama. Karena itu masing-masing data (atau elemen) dari kumpulan tersebut dapat diakses dengan mengetahui posisinya (indeksnya) dalam kumpulan tersebut.
2. Structure atau record adalah jenis tipe data yang merupakan sekumpulan data yang masing-masing dapat mempunyai tipe yang berbeda. Karena itu masing-masing data (atau field) dari kumpulan tersebut harus diberi nama agar dapat diakses.

Alias (Type)

Pendefinisian tipe baru untuk memudahkan pemrograman. Dengan nama yang lebih ringkas untuk merepresentasikan tipe data yang mungkin kompleks. Tipe data apa saja dapat didefinisikan menjadi tipe baru.

```
type Mahasiswa struct {
    nama string
    ipk real
}
type TabelMahasiswa [NMHS]Mahasiswa
type Cards [28][2]int
```

Struct atau Record

Record berguna untuk mengumpulkan data yang saling terkait dalam satu kelompok, sehingga dengan satu akses semua nilai dapat diperoleh. Masing-masing nilai tersimpan dalam field dari record tersebut. Mirip bahasa C, di Go deklarasi record menggunakan kata kunci struct.

```
// Declaring abc as a record with fields field1, field2, ...
var abc struct {
    field1 type1
    field2 type2 ...
}
```

Array

Array mempunyai ukuran (jumlah elemen) yang tetap (statik) selama eksekusi program, sehingga jumlah elemen array menjadi bagian dari deklarasi variabel dengan tipe array

```
var (
    // array arr mempunyai 73 elemen, masing-masing bertipe
    CircType2
    arr [73]CircType
    // array buf dengan 5 elemen, dengan nilai awal 7, 3, 5, 2, dan
    11.
    buf = [5]byte{7, 3, 5, 2, 11}
```

```
// mhs adalah array dengan 2000 elemen bertipe NewType mhs
[2000]NewType
// rec adalah array dari array, aka matriks, atau array
berdimensi-2
rec [20][40]float64 )
```

Jumlah elemen array dapat diminta dengan fungsi `len` yang tersedia. Sebagai contoh `len(arr)` akan menghasilkan 73 untuk contoh diatas.

Slice (Array dinamik)

Array dalam Go juga dapat mempunyai ukuran yang dinamik. Deklarasinya mirip dengan deklarasi array, tetapi jumlah elemennya dikosongkan.

```
// Prealokasi 10 elemen untuk sl02 dan sejumlah tempat tambahan
var sl02 []int = make([]int, 10, 20)
// Prealokasi 7 elemen untuk sl03 tanpa tempat tambahan
var sl03 []circType = make([]circType, 7)
```

Fungsi builtin `len` dapat digunakan untuk mengetahui ukuran slice. Fungsi lain, `cap`, dapat digunakan untuk mengetahui total tempat yang disediakan untuk slice tersebut.

```
// Cetak jumlah elemen dan tempat yang tersedia untuk sl02
fmt.Println( len(sl02), cap(sl02) )
```

Fungsi builtin `append` dapat digunakan untuk menambahkan elemen ke suatu slice, dan bila perlu memperbesar tempat untuk slice tersebut.

```
/* Append elemen baru, membuat slice baru, dan menyimpan kembali
slice baru ke variabel semula Boleh juga disimpan ke variabel lain,
sehingga variabel semula masih menyimpan slice yang asli.
*/
sl01 = append(sl01, 17)
sl01 = append(sl01, 19, 23)
```

Sebuah slice baru juga dapat terbentuk dengan mengambil slice dari suatu array atau slice yang lain.

```
// Ambil 3 elem
en pertama dari suatu slice atau array
sl04 = arr[:4] // Ambil beberapa elemen terakhir, dimulai dari indeks
5
sl05 = sl01[5:]
// Salin semua dari slice/array aslinya
sl06 = sl05[:]
// Salin element dari indeks 3 sampai, tapi tidak termasuk, 5.
// Jadi dalam contoh hanya 2 elemen sl06[3] dan sl06[4] yang disalin
sl07 = sl06[3:5]
```

Map

Tipe array lain, sebuah array dinamik, dimana indeksnya (disini disebut kunci) tidak harus berbentuk integer. Indeks dapat berasal dari tipe apa saja. Struktur ini disebut `map`.

```
// Deklarasi variabel dct sebagai map bilangan bulat dengan kunci
string
var dct map[string]int

// Deklarasi map lain dct1 dari elemen string dengan kunci juga
string
// Mempunyai nilai awal dct1["john"] = "hi", dct1["anne"] = "darling"
var dct1 = map[string]string{ "john":"hi", "anne":"darling" }

// Deklarasi dan prealokasi tempat untuk map dct2
var dct2 map[float64]int = make(map[float64]int, 10)

// Mengambil nilai yang tersimpan dengan kunci "john"
fmt.Println( dct1["john"] )

// Mengganti nilai yang tersimpan pada kunci "anne", dan
// Membuat entri baru dengan kunci "boy"
dct1["anne"] = "lovely"
dct1["boy"] = "runaround"
// Menghapus entri dengan kunci "john"
delete(dct1, "john")
```

II. GUIDED

1. Source Code:

```
package main

import (
    "fmt"
    "math"
)

// Struktur untuk menyimpan titik dengan koordinat (x,y)
type Titik struct {
    x int
    y int
}

// Struktur untuk menyimpan lingkaran dengan pusat dan radius
type Lingkaran struct {
    pusat Titik
    radius int
}

// Fungsi untuk menghitung jarak titik antara dua titik
func hitungJarak(a, b Titik) float64 {
    return math.Sqrt(float64((a.x-b.x)*(a.x-b.x) + (a.y-
b.y)*(a.y-b.y)))
}

// Fungsi untuk memeriksa apakah titik berada di dalam lingkaran
func titikDiDalamLingkaran(t Titik, l Lingkaran) bool {
    jarak := hitungJarak(t, l.pusat)
    return jarak <= float64(l.radius)
}

func main() {
    // Input untuk lingkaran 1
    var cx1, cy1, r1 int
    fmt.Print("Masukan koordinat pusat dan radius lingkaran 1
(cx1, cy1, r1): ")
    fmt.Scanln(&cx1, &cy1, &r1)
    lingkaran1 := Lingkaran{pusat: Titik{x: cx1, y: cy1}, radius:
r1}

    // Input untuk lingkaran 2
    var cx2, cy2, r2 int
    fmt.Print("Masukan koordinat pusat dan radius lingkaran 2
(cx2, cy2, r2): ")
    fmt.Scanln(&cx2, &cy2, &r2)
    lingkaran2 := Lingkaran{pusat: Titik{x: cx2, y: cy2}, radius:
r2}

    var x, y int
    fmt.Print("Masukan koordinat titik sembarang (x, y): ")
    fmt.Scanln(&x, &y)
```

```

    titik := Titik{x: x, y: y}

    // Pengecekan posisi titik
    diDalam1 := titikDiDalamLingkaran(titik, lingkaran1)
    diDalam2 := titikDiDalamLingkaran(titik, lingkaran2)

    // Menampilkan hasil sesuai kondisi
    if diDalam1 && diDalam2 {
        fmt.Println("Titik di dalam lingkaran 1 dan 2")
    } else if diDalam1 {
        fmt.Println("Titik di dalam lingkaran 1")
    } else if diDalam2 {
        fmt.Println("Titik di dalam lingkaran 2")
    } else {
        fmt.Println("Titik di luar kedua lingkaran")
    }
}

```

Output:

```

PS C:\Users\Asus\Documents\Alpro2> go run "c:\Users\Asus\Documents\Alpro2\Modul7\Guided.go"
Masukan koordinat pusat dan radius lingkaran 1 (cx1, cy1, r1): 4 6 9
Masukan koordinat pusat dan radius lingkaran 2 (cx2, cy2, r2): 4 1 3
Masukan koordinat titik sembarang (x, y): 2 3
Titik di dalam lingkaran 1 dan 2

```

Penjelasan:

Program diatas bertujuan menentukan posisi sebuah titik terhadap dua lingkaran berdasarkan input koordinat pusat dan radius masing-masing lingkaran. Pengguna akan memasukkan data koordinat pusat dan radius untuk dua lingkaran, lalu koordinat titik yang ingin diperiksa. Program menghitung jarak titik tersebut dari pusat kedua lingkaran menggunakan rumus jarak Euclidean, kemudian membandingkannya dengan radius lingkaran. Jika jaraknya kurang dari atau sama dengan radius, titik dianggap berada di dalam lingkaran. Pada contoh input, titik (2, 3) berada di dalam kedua lingkaran karena jaraknya ke pusat kedua lingkaran lebih kecil atau sama dengan radiusnya, sehingga program menampilkan "Titik di dalam lingkaran 1 dan 2." Program ini menggunakan struct untuk mengelompokkan data yang terkait. Struct Titik menyimpan koordinat (x, y), sedangkan struct Lingkaran menyimpan data pusat dan radius lingkaran, sehingga data lebih terorganisir.

III. UNGUIDED

1. 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. 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.

Source Code

```
package main

import (
    "fmt"
    "math"
)

func hitungRataRata(arr []int) float64 {
    total := 0
    for i := 0; i < len(arr); i++ {
        total += arr[i]
    }
    return float64(total) / float64(len(arr))
}

func hitungStandarDeviasi(arr []int, rataRata float64) float64 {
    var jumlahKuadrat float64
    for i := 0; i < len(arr); i++ {
        selisih := float64(arr[i]) - rataRata
        jumlahKuadrat += selisih * selisih
    }
    return math.Sqrt(jumlahKuadrat / float64(len(arr)))
}

func hitungFrekuensi(arr []int, target int) int {
    jumlah := 0
    for i := 0; i < len(arr); i++ {
```

```

        if arr[i] == target {
            jumlah++
        }
    }
    return jumlah
}

func main() {
    var n 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.Printf("Elemen ke-%d: ", i+1)
        fmt.Scan(&arr[i])
    }

    fmt.Println("\nIsi array:", arr)

    fmt.Println("\nElemen dengan indeks ganjil:")
    for i := 1; i < len(arr); i += 2 {
        fmt.Printf("arr[%d] = %d\n", i, arr[i])
    }

    fmt.Println("\nElemen dengan indeks genap:")
    for i := 0; i < len(arr); i += 2 {
        fmt.Printf("arr[%d] = %d\n", i, arr[i])
    }

    var x int
    fmt.Print("\nMasukkan bilangan kelipatan: ")
    fmt.Scan(&x)
    fmt.Printf("Elemen dengan indeks kelipatan %d:\n", x)
    for i := x; i < len(arr); i += x {
        fmt.Printf("arr[%d] = %d\n", i, arr[i])
    }

    var idxHapus int
    fmt.Print("\nMasukkan indeks yang ingin dihapus: ")
    fmt.Scan(&idxHapus)
    if idxHapus >= 0 && idxHapus < len(arr) {
        arr = append(arr[:idxHapus], arr[idxHapus+1:]...)
        fmt.Println("Isi array setelah dihapus:", arr)
    } else {
        fmt.Println("Indeks tidak valid!")
    }

    if len(arr) > 0 {
        rataRata := hitungRataRata(arr)
        fmt.Printf("\nRata-rata array: %.2f\n", rataRata)
    }
}

```



```

        stdDeviasi := hitungStandarDeviasi(arr, rataRata)
        fmt.Printf("Standar deviasi array: %.2f\n", stdDeviasi)
    } else {
        fmt.Println("Array kosong, tidak bisa hitung rata-rata
atau standar deviasi.")
    }

    var target int
    fmt.Print("\nMasukkan bilangan untuk hitung frekuensi: ")
    fmt.Scan(&target)
    frekuensi := hitungFrekuensi(arr, target)
    fmt.Printf("Frekuensi bilangan %d di array: %d\n", target,
frekuensi)
}

```

Output

```

PS C:\Users\Asus\Documents\Alpro2> go run "c:\Users\Asus\Documents\Alpro2\Modul7\Unguided\Unguided1.go"
Masukkan jumlah elemen array: 5
Masukkan elemen array:
Elemen ke-1: 3
Elemen ke-2: 5
Elemen ke-3: 2
Elemen ke-4: 8
Elemen ke-5: 7

Isi array: [3 5 2 8 7]

Elemen dengan indeks ganjil:
arr[1] = 5
arr[3] = 8

Elemen dengan indeks genap:
arr[0] = 3
arr[2] = 2
arr[4] = 7

Masukkan bilangan kelipatan: 2
Elemen dengan indeks kelipatan 2:
arr[2] = 2
arr[4] = 7

Masukkan indeks yang ingin dihapus: 3
Isi array setelah dihapus: [3 5 2 7]

Rata-rata array: 4.25
Standar deviasi array: 1.92

Masukkan bilangan untuk hitung frekuensi: 5
Frekuensi bilangan 5 di array: 1

```

Penjelasan Program

Program di atas menggunakan slice untuk menyimpan sekumpulan bilangan bulat yang dimasukkan oleh pengguna. Dimulai dengan meminta jumlah elemen yang ingin dimasukkan ke dalam array, program kemudian meminta pengguna untuk mengisi nilai setiap elemen. Slice yang dibuat akan berisi elemen-elemen tersebut. Program menampilkan beberapa informasi terkait slice, seperti seluruh isi array,

elemen dengan indeks ganjil, elemen dengan indeks genap, dan elemen dengan indeks kelipatan bilangan yang dimasukkan pengguna. Selain itu, program juga menyediakan fitur untuk menghapus elemen pada indeks tertentu dan menampilkan hasil setelah penghapusan. Program selanjutnya menghitung dan menampilkan rata-rata serta standar deviasi dari elemen-elemen dalam slice jika slice tidak kosong. Terakhir, pengguna dapat memasukkan bilangan tertentu untuk menghitung frekuensinya dalam slice. Output yang ditampilkan termasuk seluruh elemen array, elemen berdasarkan indeks yang diminta, rata-rata, standar deviasi, dan frekuensi bilangan tertentu. Sebagai contoh, jika pengguna memasukkan array [3, 5, 2, 8, 7], program akan menampilkan elemen-elemen sesuai dengan permintaan dan melakukan perhitungan sesuai dengan langkah-langkah yang telah ditentukan.

2. 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 memenangkan pertandingan.

Source Code

```
package main

import "fmt"

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

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)

    for n := 1; ; n++ {
        fmt.Printf("Pertandingan %d : ", n)
        fmt.Scan(&skorA, &skorB)
```

```

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

        var pemenang string
        if skorA > skorB {
            pemenang = klubA
        } else if skorA < skorB {
            pemenang = klubB
        } else {
            pemenang = "Draw"
        }

        hasil = append(hasil, pemenang)
    }

    fmt.Println("\nDaftar klub yang memenangkan pertandingan:")
    for n, h := range hasil {
        fmt.Printf("Hasil %d : %s\n", n+1, h)
    }

    fmt.Println("Pertandingan selesai")
}

```

Output

```

PS C:\Users\Asus\Documents\Alpro2> go run "c:\User
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
Pertandingan 9 : -1 2
Hasil 1 : MU
Hasil 2 : Inter
Hasil 3 : Draw
Hasil 4 : Inter
Hasil 5 : MU
Hasil 6 : MU
Hasil 7 : MU
Hasil 8 : Inter
Pertandingan selesai

```

Penjelasan Program

Program di atas adalah program untuk mencatat hasil pertandingan antara dua klub sepak bola. Program ini menggunakan struct untuk mendefinisikan tipe data

Pertandingan yang berisi informasi tentang nama klub (klubA dan klubB), skor masing-masing klub (skorA dan skorB), serta pemenang pertandingan (pemenang). Pengguna diminta untuk memasukkan nama klub, kemudian program akan mencatat hasil pertandingan dalam sebuah loop yang terus berjalan hingga pengguna memasukkan skor negatif, yang menandakan bahwa pertandingan sudah selesai. Hasil setiap pertandingan disimpan dalam sebuah slice (hasil), yang kemudian dicetak untuk menampilkan siapa yang menang di setiap pertandingan. Slice digunakan karena panjangnya bisa berubah seiring dengan banyaknya pertandingan yang dimasukkan. Output program menunjukkan siapa yang menang di setiap pertandingan, serta mengakhiri dengan mencetak "Pertandingan selesai" setelah skor negatif dimasukkan.

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

Source Code

```
package main

import (
    "fmt"
)

const NMAX = 127

type tabel [NMAX]rune

func isiArray(t *tabel, n *int) {
    var input string
    *n = 0
    for {
        fmt.Scan(&input)
        if input == "." {
            break
        }
        for _, c := range input {
            if *n < NMAX {
                (*t)[*n] = c
                *n++
            }
        }
    }
}

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

```

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

func palindrom(t tabel, n int) bool {
    for i := 0; i < n/2; i++ {
        if t[i] != t[n-i-1] {
            return false
        }
    }
    return true
}

func main() {
    var t tabel
    var n int

    fmt.Print("Teks : ")
    isiArray(&t, &n)

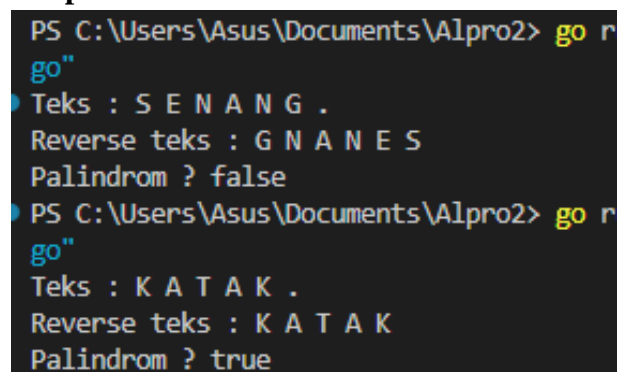
    balikanArray(&t, n)

    fmt.Print("Reverse teks : ")
    cetakArray(t, n)

    fmt.Println("Palindrom ?", palindrom(t, n))
}

```

Output



```

PS C:\Users\Asus\Documents\Alpro2> go run main.go
Teks : S E N A N G .
Reverse teks : G N A N E S
Palindrom ? false
PS C:\Users\Asus\Documents\Alpro2> go run main.go
Teks : K A T A K .
Reverse teks : K A T A K
Palindrom ? true

```

Penjelasan Program

Program ini mengimplementasikan manipulasi array untuk memeriksa apakah teks yang dimasukkan membentuk palindrom, serta membalikkan urutan teks. Pertama, array tabel bertipe rune (yang mampu menampung karakter Unicode) digunakan untuk menyimpan input karakter yang dimasukkan oleh pengguna, yang disalurkan

melalui prosedur isiArray. Setelah itu, urutan karakter pada array dibalik menggunakan prosedur balikanArray. Program kemudian memeriksa apakah array tersebut membentuk palindrom dengan membandingkan karakter dari kiri dan kanan array menggunakan fungsi palindrom. Jika karakter di kedua sisi array tersebut sama, maka teks dianggap palindrom dan program menampilkan hasil true; jika tidak, hasilnya false. Pada contoh pertama, input "S E N A N G" menghasilkan output "G N A N E S" yang tidak membentuk palindrom, sehingga output false. Sedangkan pada input kedua "K A T A K", urutan teks dibalik menjadi "K A T A K" yang tetap sama, menghasilkan output true, yang menunjukkan teks tersebut adalah palindrom.