

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL VII
Struct dan Array**



**Disusun Oleh :
Alfin Adriansyah/ 2311102264
S1IF_11_06**

**Dosen Pengampu :
ABEDNEGO DWI SEPTIADI**

**PROGRAM STUDI S1 TEKNIK INFORMATIKA
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2024**

I. 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. Dari sebuah struct, kita bisa buat variabel baru, yang memiliki atribut sesuai skema struct tersebut. Kita sepakati dalam buku ini, variabel tersebut dipanggil dengan istilah object atau variabel object. Konsep struct di golang mirip dengan konsep class pada OOP, meski sebenarnya memiliki perbedaan. Di sini penulis menggunakan konsep OOP sebagai analogi, untuk mempermudah pembaca untuk memahami pembelajaran di chapter ini. Dengan memanfaatkan struct, penyimpanan data yang sifatnya kolektif menjadi lebih mudah, lebih rapi, dan mudah untuk dikelola. A. Perbedaan teknik rekursif dengan teknik iteratif

Kombinasi keyword type dan struct digunakan untuk deklarasi struct. Di bawah ini merupakan contoh cara penerapannya.

```
type student struct {  
    name string  
    grade int  
}
```

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. Default nilai tiap elemen array pada awalnya tergantung dari tipe datanya. Jika int maka tiap element zero value-nya adalah 0, jika bool maka false, dan seterusnya. Setiap elemen array memiliki indeks berupa angka yang merepresentasikan posisi urutan elemen tersebut. Indeks array dimulai dari 0.

Contoh penerapan array:

```
var names [4]string  
names[0] = "trafalgar"  
names[1] = "d"  
names[2] = "water"  
names[3] = "law"  
  
fmt.Println(names[0], names[1], names[2], names[3])
```


II. GUIDED

1. Buatlah

Sourcecode

```
package main

import (
    "fmt"
    "math"
)

type Titik struct {
    x int
    y int
}

type Lingkaran struct {
    pusat Titik
    radius int
}

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)))
}

func titikDalamLingkaran(t Titik, l Lingkaran) bool {
    jarak := hitungJarak(t, l.pusat)
    return jarak <= float64(l.radius)
}

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

    var cx2, cy2, r2 int
    fmt.Print("Masukkan 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("Masukkan koordinat titik sembarang (x,
y): ")
    fmt.Scanln(&x, &y)
    titik := Titik{x: x, y: y}
```

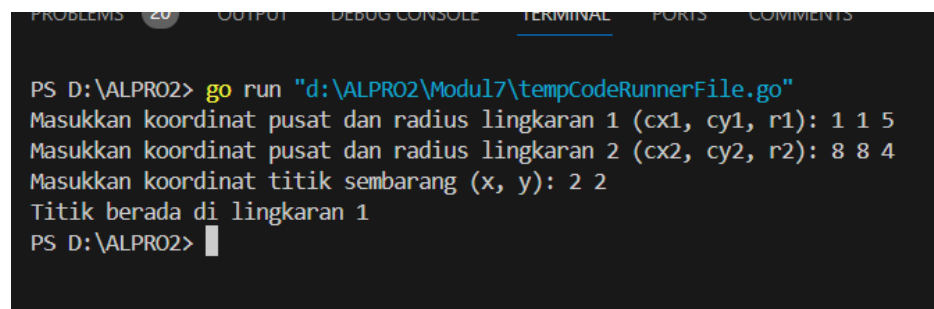
```

        diDalam1 := titikDalamLingkaran(titik, lingkaran)
        diDalam2 := titikDalamLingkaran(titik, lingkaran2)

        if diDalam1 && diDalam2 {
            fmt.Println("Titik berada di kedua lingkaran 1
dan 2")
        } else if diDalam1 {
            fmt.Println("Titik berada di lingkaran 1")
        } else if diDalam2 {
            fmt.Println("Titik berada di lingkaran 2")
        } else {
            fmt.Println("Titik berada di luar kedua
lingkaran")
        }
    }
}

```

Screenshoot Output



```

PROBLEMS 20 OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS
PS D:\ALPRO2> go run "d:\ALPRO2\Modul7\tempCodeRunnerFile.go"
Masukkan koordinat pusat dan radius lingkaran 1 (cx1, cy1, r1): 1 1 5
Masukkan koordinat pusat dan radius lingkaran 2 (cx2, cy2, r2): 8 8 4
Masukkan koordinat titik sembarang (x, y): 2 2
Titik berada di lingkaran 1
PS D:\ALPRO2>

```

Deskripsi Program

Program di atas adalah program Go yang menentukan posisi sebuah titik relatif terhadap dua buah lingkaran. Program meminta input dari pengguna berupa koordinat pusat dan radius untuk dua lingkaran, serta koordinat sebuah titik sembarang. Menggunakan struktur data 'Titik' dan 'Lingkaran', program menghitung jarak antara titik dengan pusat lingkaran menggunakan rumus jarak Euclidean, kemudian menentukan apakah titik tersebut berada di dalam lingkaran pertama, lingkaran kedua, di dalam kedua lingkaran, atau di luar kedua lingkaran, dan akhirnya menampilkan hasil tersebut ke pengguna.

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 indek 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.

Sourcecode

```
package main

import (
    "fmt"
    "math"
)

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

    bilangan := make([]int, jumlahElemen)
    for i := 0; i < jumlahElemen; i++ {
        fmt.Printf("Masukkan bilangan ke-%d: ", i)
        fmt.Scan(&bilangan[i])
    }

    for {
        fmt.Println("\nPilihan Menu:")
        fmt.Println("1. Tampilkan semua isi array")
        fmt.Println("2. Tampilkan elemen dengan indeks ganjil")
        fmt.Println("3. Tampilkan elemen dengan indeks genap")
    }
```

```

        fmt.Println("4. Tampilkan elemen dengan indeks
kelipatan tertentu")
        fmt.Println("5. Hapus elemen pada indeks
tertentu")
        fmt.Println("6. Hitung rata-rata")
        fmt.Println("7. Hitung standar deviasi")
        fmt.Println("8. Hitung frekuensi kemunculan
suatu bilangan")
        fmt.Println("9. Keluar dari program")

        var pilihan int
        fmt.Print("Masukkan pilihan menu (1-9): ")
        fmt.Scan(&pilihan)

        switch pilihan {
        case 1:
            tampilkanArray(bilangan)
        case 2:
            tampilkanIndeksGanjil(bilangan)
        case 3:
            tampilkanIndeksGenap(bilangan)
        case 4:
            var kelipatan int
            fmt.Print("Masukkan nilai kelipatan: ")
            fmt.Scan(&kelipatan)
            tampilkanIndeksKelipatan(bilangan,
kelipatan)
        case 5:
            var indeks int
            fmt.Print("Masukkan indeks yang akan
dihapus: ")
            fmt.Scan(&indeks)
            bilangan = hapusElemen(bilangan, indeks)
        case 6:
            rerata := hitungRataRata(bilangan)
            fmt.Printf("Rata-rata: %.2f\n", rerata)
        case 7:
            simpanganBaku :=
hitungStandarDeviasi(bilangan)
            fmt.Printf("Standar deviasi: %.2f\n",
simpanganBaku)
        case 8:
            var nilai int
            fmt.Print("Masukkan bilangan yang ingin
dihitung frekuensinya: ")
            fmt.Scan(&nilai)
            frekuensi := hitungFrekuensi(bilangan,
nilai)
            fmt.Printf("Frekuensi kemunculan
bilangan %d: %d kali\n", nilai, frekuensi)
        case 9:

```

```

        fmt.Println("Terima kasih, program
selesai.")
        return
    default:
        fmt.Println("Pilihan tidak valid!
Silakan pilih menu 1-9.")
    }
}

func tampilkanArray(arr []int) {
    fmt.Print("Isi array: ")
    for i, nilai := range arr {
        fmt.Printf("[%d]=%d ", i, nilai)
    }
    fmt.Println()
}

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

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

func tampilkanIndeksKelipatan(arr []int, kelipatan int)
{
    fmt.Printf("Elemen dengan indeks kelipatan %d: ",
kelipatan)
    for i := 0; i < len(arr); i++ {
        if i%kelipatan == 0 {
            fmt.Printf("[%d]=%d ", i, arr[i])
        }
    }
    fmt.Println()
}

func hapusElemen(arr []int, indeks int) []int {
    if indeks < 0 || indeks >= len(arr) {
        fmt.Println("Indeks tidak valid!")
        return arr
    }
}

```



```

        hasil := append(arr[:indeks], arr[indeks+1:]...)
        fmt.Println("Elemen berhasil dihapus!")
        return hasil
    }

    func hitungRataRata(arr []int) float64 {
        if len(arr) == 0 {
            return 0
        }
        jumlah := 0
        for _, nilai := range arr {
            jumlah += nilai
        }
        return float64(jumlah) / float64(len(arr))
    }

    func hitungStandarDeviasi(arr []int) float64 {
        if len(arr) == 0 {
            return 0
        }
        rerata := hitungRataRata(arr)

        var jumlahKuadratSelisih float64
        for _, nilai := range arr {
            selisih := float64(nilai) - rerata
            jumlahKuadratSelisih += selisih * selisih
        }

        variansi := jumlahKuadratSelisih / float64(len(arr))
        return math.Sqrt(variansi)
    }

    func hitungFrekuensi(arr []int, nilai int) int {
        frekuensi := 0
        for _, num := range arr {
            if num == nilai {
                frekuensi++
            }
        }
        return frekuensi
    }
}

```

Screenshoot Output

```
PS D:\ALPRO2> go run "d:\ALPRO2\Modul7\unguided1.go"
Masukkan jumlah elemen array (N): 3
Masukkan bilangan ke-0: 1
Masukkan bilangan ke-1: 2
Masukkan bilangan ke-2: 3

Pilihan Menu:
1. Tampilkan semua isi array
2. Tampilkan elemen dengan indeks ganjil
3. Tampilkan elemen dengan indeks genap
4. Tampilkan elemen dengan indeks kelipatan tertentu
5. Hapus elemen pada indeks tertentu
6. Hitung rata-rata
7. Hitung standar deviasi
8. Hitung frekuensi kemunculan suatu bilangan
9. Keluar dari program
Masukkan pilihan menu (1-9): █

Masukkan pilihan menu (1-9): 2
Elemen dengan indeks ganjil: [1]=2

Pilihan Menu:
1. Tampilkan semua isi array
2. Tampilkan elemen dengan indeks ganjil
3. Tampilkan elemen dengan indeks genap
4. Tampilkan elemen dengan indeks kelipatan tertentu
5. Hapus elemen pada indeks tertentu
6. Hitung rata-rata
7. Hitung standar deviasi
8. Hitung frekuensi kemunculan suatu bilangan
9. Keluar dari program
Masukkan pilihan menu (1-9): █

Masukkan pilihan menu (1-9): 4
Masukkan nilai kelipatan: 2
Elemen dengan indeks kelipatan 2: [0]=1 [2]=3

Pilihan Menu:
1. Tampilkan semua isi array
2. Tampilkan elemen dengan indeks ganjil
3. Tampilkan elemen dengan indeks genap
4. Tampilkan elemen dengan indeks kelipatan tertentu
5. Hapus elemen pada indeks tertentu
6. Hitung rata-rata
7. Hitung standar deviasi
8. Hitung frekuensi kemunculan suatu bilangan
9. Keluar dari program
Masukkan pilihan menu (1-9): █

Masukkan pilihan menu (1-9): 6
Rata-rata: 2.00

Pilihan Menu:
1. Tampilkan semua isi array
2. Tampilkan elemen dengan indeks ganjil
3. Tampilkan elemen dengan indeks genap
4. Tampilkan elemen dengan indeks kelipatan tertentu
5. Hapus elemen pada indeks tertentu
6. Hitung rata-rata
7. Hitung standar deviasi
8. Hitung frekuensi kemunculan suatu bilangan
9. Keluar dari program
Masukkan pilihan menu (1-9): █

Masukkan pilihan menu (1-9): 8
Masukkan bilangan yang ingin dihitung frekuensinya: 1
Frekuensi kemunculan bilangan 1: 1 kali

Pilihan Menu:
1. Tampilkan semua isi array
2. Tampilkan elemen dengan indeks ganjil
3. Tampilkan elemen dengan indeks genap
4. Tampilkan elemen dengan indeks kelipatan tertentu
5. Hapus elemen pada indeks tertentu
6. Hitung rata-rata
7. Hitung standar deviasi
8. Hitung frekuensi kemunculan suatu bilangan
9. Keluar dari program
Masukkan pilihan menu (1-9): █

Masukkan indeks yang akan dihapus: 0
Elemen berhasil dihapus!

Pilihan Menu:
1. Tampilkan semua isi array
2. Tampilkan elemen dengan indeks ganjil
3. Tampilkan elemen dengan indeks genap
4. Tampilkan elemen dengan indeks kelipatan tertentu
5. Hapus elemen pada indeks tertentu
6. Hitung rata-rata
7. Hitung standar deviasi
8. Hitung frekuensi kemunculan suatu bilangan
9. Keluar dari program
Masukkan pilihan menu (1-9): 1
Isi array: [0]=2 [1]=3

Pilihan Menu:
1. Tampilkan semua isi array
2. Tampilkan elemen dengan indeks ganjil
3. Tampilkan elemen dengan indeks genap
4. Tampilkan elemen dengan indeks kelipatan tertentu
5. Hapus elemen pada indeks tertentu
6. Hitung rata-rata
7. Hitung standar deviasi
8. Hitung frekuensi kemunculan suatu bilangan
9. Keluar dari program
Masukkan pilihan menu (1-9): █
```

Deskripsi Program

Program ini merupakan aplikasi pengolahan array dalam bahasa Go yang memungkinkan pengguna untuk melakukan berbagai operasi pada sekumpulan bilangan bulat. Program dimulai dengan meminta pengguna memasukkan jumlah elemen array dan nilai-nilainya, kemudian menyediakan menu interaktif dengan 9 pilihan fungsi: menampilkan seluruh isi array, menampilkan elemen dengan indeks ganjil atau genap, menampilkan elemen dengan indeks kelipatan tertentu, menghapus elemen pada indeks tertentu, menghitung rata-rata, menghitung standar deviasi, dan menghitung frekuensi kemunculan suatu bilangan.

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.

Sourcecode

```
package main

import (
    "fmt"
)

func main() {
    var klubA, klubB string

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

    var hasilPertandingan []string
    nomorPertandingan := 1

    for {
        var skorA, skorB int
```

```

        fmt.Printf("Pertandingan %d : ",
        nomorPertandingan)
        fmt.Scan(&skorA, &skorB)

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

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

        hasilPertandingan = append(hasilPertandingan,
        pemenang)
        nomorPertandingan++
    }

    fmt.Println("\nHasil pertandingan:")
    for i, hasil := range hasilPertandingan {
        fmt.Printf("Hasil %d : %s\n", i+1, hasil)
    }

    fmt.Println("Pertandingan selesai")
}

```

Screenshoot Output

```

PS D:\ALPRO2> go run "d:\ALPRO2\Modul7\unguided2.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
Pertandingan 9 : -1 2

Hasil pertandingan:
Hasil 1 : MU
Hasil 2 : Inter
Hasil 3 : Draw
Hasil 4 : Inter
Hasil 5 : MU
Hasil 6 : MU
Hasil 7 : MU
Hasil 8 : Inter

```

Deskripsi Program

Program di atas merupakan aplikasi pencatatan hasil pertandingan sepak bola antara dua klub. Program dimulai dengan meminta input nama kedua klub (Klub A dan Klub B), kemudian secara berulang meminta input skor pertandingan untuk kedua tim tersebut. Setiap kali skor dimasukkan, program akan menentukan pemenang berdasarkan skor tertinggi dan menyimpannya dalam sebuah array, atau mencatat "Draw" jika skornya sama. Program akan terus meminta input skor hingga salah satu atau kedua skor yang dimasukkan bernilai negatif, yang menandakan akhir dari input pertandingan. Setelah input selesai, program akan menampilkan hasil seluruh pertandingan secara berurutan dengan format "Hasil [nomor] : [nama klub pemenang/Draw]" dan diakhiri dengan pesan "Pertandingan selesai".

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.

Sourcecode

```
package main

import (
    "fmt"
)

const MAX int = 127

type tabel [MAX]rune

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

    var kar rune
    *n = 0
    fmt.Print("Masukkan teks (akhiri dengan titik): ")
    fmt.Scanf("%c", &kar)
    for kar != '.' && *n < MAX {
        t[*n] = kar
        *n++
        fmt.Scanf("%c", &kar)
    }
}
```

```

func cetakArray(t *tabel, n int) {

    fmt.Print("Teks: ")
    for i := 0; i < n; i++ {
        fmt.Printf("%c", t[i])
    }
    fmt.Println()
}

func balikkanArray(t *tabel, n int) {

    var temp tabel
    for i := 0; i < n; i++ {
        temp[i] = t[n-1-i]
    }
    for i := 0; i < n; i++ {
        t[i] = temp[i]
    }
}

func palindrom(t *tabel, n int) bool {

    var temp tabel
    for i := 0; i < n; i++ {
        temp[i] = t[i]
    }

    balikkanArray(&temp, n)

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

func main() {
    var tab tabel
    var n int

    isiArray(&tab, &n)

    fmt.Print("Teks: ")
    cetakArray(&tab, n)

    fmt.Print("Reverse teks: ")
    balikkanArray(&tab, n)
    cetakArray(&tab, n)

    balikkanArray(&tab, n)

    fmt.Print("Palindrom: ")
    if palindrom(&tab, n) {
        fmt.Println("True")
    }
}

```

```
        } else {  
            fmt.Println("False")  
        }  
    }  
}
```

Screenshoot Output

```
PS D:\ALPRO2> go run "d:\ALPRO2\Modul7\unguided3.go"  
Masukkan teks (akhiri dengan titik): KASUR RUSAK.  
Teks: Teks: KASUR RUSAK  
Reverse teks: Teks: KASUR RUSAK  
Palindrom: True  
PS D:\ALPRO2> go run "d:\ALPRO2\Modul7\unguided3.go"  
Masukkan teks (akhiri dengan titik): AKU.  
Teks: Teks: AKU  
Reverse teks: Teks: UKA  
Palindrom: False  
PS D:\ALPRO2> 
```

Deskripsi Program

Program di atas bertujuan untuk membaca input teks dari pengguna, menampilkan teks tersebut, membalikkan urutan karakter dalam teks, dan memeriksa apakah teks tersebut adalah palindrom. Program ini menggunakan array dengan tipe data rune untuk menyimpan karakter, serta menyediakan fungsi untuk mengisi array dengan input, mencetak array, membalikkan isi array, dan memeriksa kesamaan antara array asli dan array yang dibalik untuk menentukan apakah teks yang dimasukkan membentuk palindrom. Program diakhiri dengan menampilkan hasil pemeriksaan palindrom, mengeluarkan output "True" jika teks adalah palindrom, atau "False" jika bukan.