

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL VII
STRUCK & ARRAY**



Disusun Oleh :

Muhammad Hamzah Haifan Ma'ruf

2311102091

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

Struct adalah kumpulan data field yang dideklarasikan dengan tipe data. Struct berfungsi sebagai sebuah template untuk kumpulan beberapa data. Jika dibandingkan dengan pemrograman berorientasi objek, struct ini hampir sama seperti sebuah objek atau class. Biasanya, struct digunakan sebagai representasi dari sebuah data dalam aplikasi kita. Contoh data yang bisa dijadikan struct yaitu seperti user, product, dan lain-lain.

Seperti yang sudah kita pahami sebelumnya, struct hanyalah template untuk suatu data. Saat kita membuat struct, kita tidak bisa langsung menggunakannya karena tidak memiliki data apapun. Kita bisa membuat data berdasarkan struct yang sudah kita buat. Hal ini hampir sama seperti saat kita membuat sebuah objek berdasarkan class pada pemrograman berorientasi objek. Dalam pemrograman berorientasi objek, kita biasa membuat sebuah class yang di dalamnya terdapat satu atau lebih method yang memiliki tugasnya masing-masing. Di Golang, kita juga bisa membuat sebuah struct agar seolah-olah memiliki method seperti class dalam pemrograman berorientasi objek.

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.

II. GUIDED

1. GUIDED 1

Studi Case : Program sederhana untuk menghitung lama waktu parkir berdasarkan waktu kedatangan dan waktu pulang.

Sourcecode

```
package main

import "fmt"

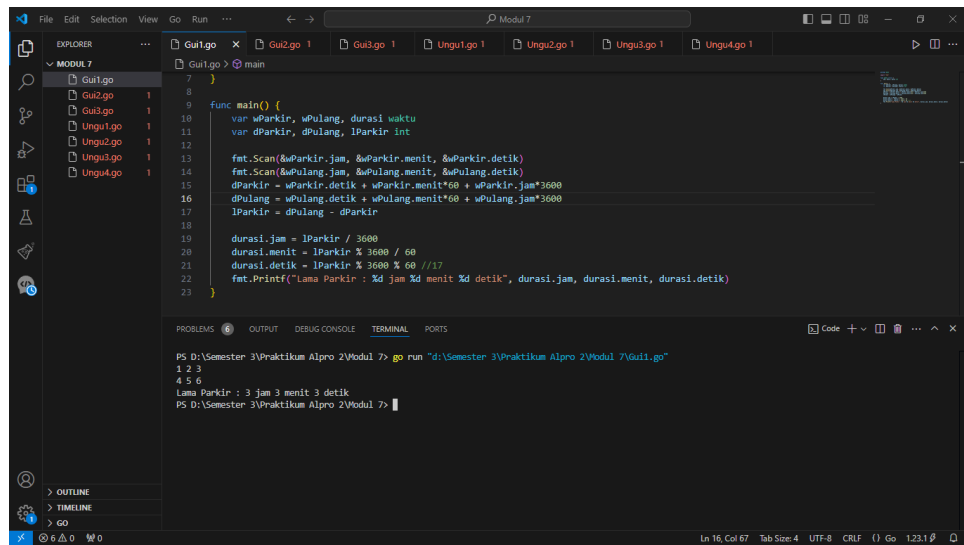
type waktu struct {
    jam, menit, detik int
}

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)
}
```

Screenshoot Output



```
7 }
8
9 func main() {
10     var wParkir, wPulang, durasi waktu
11     var dParkir, dPulang, lParkir int
12
13     fmt.Scan(&wParkir.jam, &wParkir menit, &wParkir detik)
14     fmt.Scan(&dPulang.jam, &dPulang menit, &dPulang detik)
15     dParkir = wParkir.detik + wParkir menit*60 + wParkir.jam*3600
16     dPulang = wPulang.detik + wPulang menit*60 + wPulang.jam*3600
17     lParkir = dPulang - dParkir
18
19     durasi.jam = lParkir / 3600
20     durasi menit = lParkir % 3600 / 60
21     durasi.detik = lParkir % 3600 % 60 //17
22     fmt.Printf("Lama Parkir : %d jam %d menit %d detik", durasi.jam, durasi menit, durasi.detik)
23 }
```

```
PS D:\Semester 3\Praktikum Alpro 2\Modul 7> go run "d:\Semester 3\Praktikum Alpro 2\Modul 7\Gui1.go"
1 2 3
4 5 6
Lama Parkir : 3 jam 3 menit 3 detik
PS D:\Semester 3\Praktikum Alpro 2\Modul 7>
```

Deskripsi Program

Program ini menghitung durasi parkir kendaraan dengan menerima input waktu kedatangan (waktu parkir) dan waktu kepulangan dalam format jam, menit, dan detik. Waktu kedatangan dan kepulangan diubah ke dalam satuan detik masing-masing sebagai `dParkir` dan `dPulang`. Selisih antara keduanya (`lParkir`) menunjukkan durasi parkir dalam detik. Program ini kemudian mengonversi `lParkir` ke dalam jam, menit, dan detik agar hasil lebih mudah dibaca. Hasil akhir ditampilkan dalam format “X jam Y menit Z detik.” Contoh, jika waktu parkir pukul 08:30:00 dan waktu pulang pukul 11:45:15, maka hasil yang ditampilkan adalah “Lama Parkir : 3 jam 15 menit 15 detik.”

2. GUIDED 2

Studi Case : Program sederhana untuk validasi duplikasi nama pada daftar teman.

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() {
    daftarTeman := []string{"Andi", "Budi", "Cici"}
    namaBaru := []string{"Dewi", "Budi", "Eka"}
    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)
}

```

Screenshoot Output

```

1 package main
2
3 import (
4     "fmt"
5 )
6
7 func sudahAda(daftarTeman []string, nama string) bool {
8     for _, teman := range daftarTeman {
9         if teman == nama {
10             return true
11         }
12     }
13     return false
14 }
15
16 func main() {
17     daftarTeman := []string{"Andi", "Budi", "Cici"}
18     namaBaru := []string{"Dewi", "Budi", "Eka"}

```

PS D:\Semester 3\Praktikum Alpro 2\Modul 7> go run "d:\Semester 3\Praktikum Alpro 2\Modul 7\Gui2.go"
 Nama Budi sudah ada dalam daftar.
 Daftar Teman : [Andi Budi Cici Dewi Eka]
 PS D:\Semester 3\Praktikum Alpro 2\Modul 7>

Deskripsi Program

Program ini menghitung durasi parkir kendaraan dengan menerima input waktu kedatangan (waktu parkir) dan waktu kepulangan dalam format jam, menit, dan detik. Waktu kedatangan dan kepulangan diubah ke dalam satuan detik masing-masing sebagai `dParkir` dan `dPulang`. Selisih antara keduanya (`lParkir`) menunjukkan durasi parkir dalam detik. Program ini

kemudian mengonversi `lParkir` ke dalam jam, menit, dan detik agar hasil lebih mudah dibaca. Hasil akhir ditampilkan dalam format “X jam Y menit Z detik.” Contoh, jika waktu parkir pukul 08:30:00 dan waktu pulang pukul 11:45:15, maka hasil yang ditampilkan adalah “Lama Parkir : 3 jam 15 menit 15 detik.”

3. GUIDED 3

Studi Case : Program sederhana untuk menampilkan daftar harga buah.

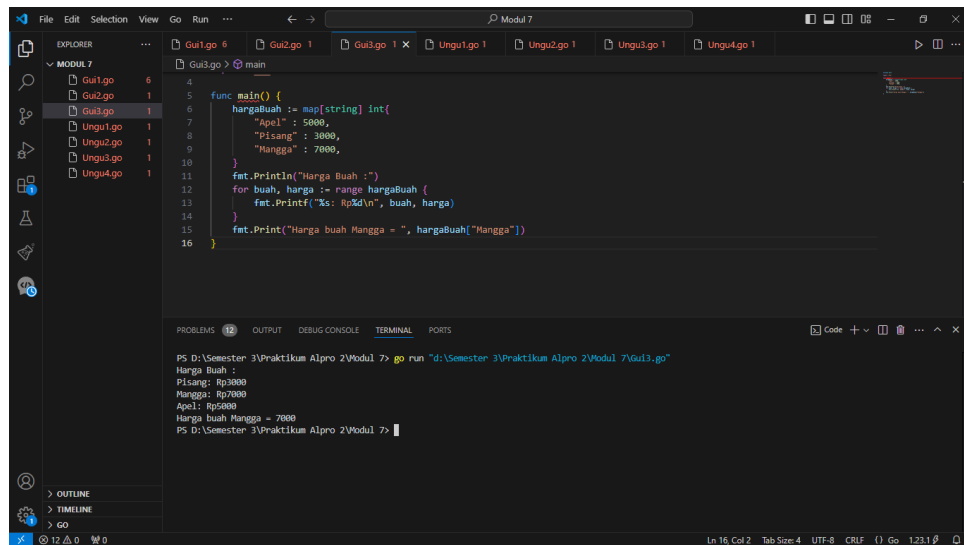
Sourcecode

```
package main

import "fmt"

func main() {
    hargaBuah := map[string] int{
        "Apel" : 5000,
        "Pisang" : 3000,
        "Mangga" : 7000,
    }
    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



```
func main() {
    4
    5     hargaBuah := map[string] int{
    6         "Apel" : 5000,
    7         "Pisang" : 3000,
    8         "Mangga" : 7000,
    9     }
    10
    11     fmt.Println("Harga Buah :")
    12     for buah, harga := range hargaBuah {
    13         fmt.Printf("%s: Rp%d\n", buah, harga)
    14     }
    15     fmt.Printf("Harga buah Mangga = ", hargaBuah["Mangga"])
    16 }
```

```
PS D:\Semester 3\Praktikum Alpro 2\Modul 7> go run "d:\Semester 3\Praktikum Alpro 2\Modul 7\Gui3.go"
Harga Buah :
Pisang: Rp3000
Mangga: Rp7000
Apel: Rp5000
Harga buah Mangga = 7000
PS D:\Semester 3\Praktikum Alpro 2\Modul 7>
```

Deskripsi Program

Program ini bertujuan untuk menyimpan dan menampilkan harga beberapa jenis buah, serta menampilkan harga khusus untuk buah mangga. Program ini menggunakan peta (map) di Go dengan nama hargaBuah, di mana nama buah (string) menjadi kunci, dan harganya (int) menjadi nilai. Terdapat tiga buah yang dimasukkan dalam map ini: Apel, Pisang, dan Mangga dengan harga masing-masing Rp5000, Rp3000, dan Rp7000.

Langkah pertama, program menampilkan semua harga buah yang ada dalam hargaBuah menggunakan loop for dengan range. Untuk setiap buah, program menampilkan nama buah dan harganya. Setelah itu, program secara khusus menampilkan harga untuk buah Mangga dengan mengaksesnya langsung menggunakan kunci "Mangga". Hasilnya adalah daftar harga buah yang tersedia, serta harga khusus untuk buah Mangga.

III. UNGUIDED

1. UNGUIDED 1

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. Asumsi sumbu x dan y dari semua titik dan juga radius direpresentasikan dengan bilangan bulat. Keluaran berupa string yang menyatakan posisi titik "Titik di dalam lingkaran 1 dan 2", "Titik di dalam lingkaran 1", "Titik di dalam lingkaran 2", atau "Titik di luar lingkaran 1 dan 2".

No	Masukan	Keluaran
1	1 1 5 8 8 4 2 2	Titik di dalam lingkaran 1
2	1 2 3 4 5 6 7 8	Titik di dalam lingkaran 2
3	5 10 15 -15 4 20 0 0	Titik di dalam lingkaran 1 dan 2
4	1 1 5 8 8 4 15 20	Titik di luar lingkaran 1 dan 2

Sourcecode

```
package main

import (
    "fmt"
    "math"
)

type Titik struct {
    X int
    Y int
}

type Lingkaran struct {
    Tengah Titik
    Radius int
}

func hitungJarak(t1, t2 Titik) float64 {
    return math.Sqrt(math.Pow(float64(t1.X-t2.X), 2) +
        math.Pow(float64(t1.Y-t2.Y), 2))
}

func tentukanPosisiTitik(lingkaran1, lingkaran2
    Lingkaran, titik Titik) string {
    jarakKeLingkaran1 := hitungJarak(lingkaran1.Tengah,
        titik)
```



```

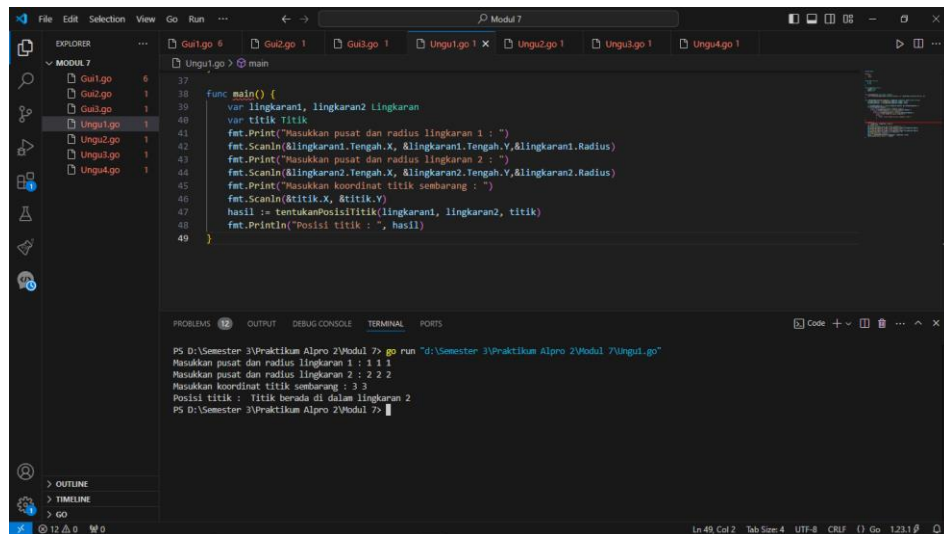
        jarakKeLingkaran2 := hitungJarak(lingkaran2.Tengah,
titik)

        if jarakKeLingkaran1 <= float64(lingkaran1.Radius) &&
jarakKeLingkaran2 <=
float64(lingkaran2.Radius) {
            return "Titik berada di dalam lingkaran 1 dan 2"
        } else if jarakKeLingkaran1 <=
float64(lingkaran1.Radius) {
            return "Titik berada di dalam lingkaran 1"
        } else if jarakKeLingkaran2 <=
float64(lingkaran2.Radius) {
            return "Titik berada di dalam lingkaran
2"
        } else {
            return "Titik berada di luar lingkaran
1 dan 2"
        }
    }

func main() {
    var lingkaran1, lingkaran2 Lingkaran
    var titik Titik
    fmt.Print("Masukkan pusat dan radius lingkaran 1 : ")
    fmt.Scanln(&lingkaran1.Tengah.X,
&lingkaran1.Tengah.Y, &lingkaran1.Radius)
    fmt.Print("Masukkan pusat dan radius lingkaran 2 : ")
    fmt.Scanln(&lingkaran2.Tengah.X,
&lingkaran2.Tengah.Y, &lingkaran2.Radius)
    fmt.Print("Masukkan koordinat titik sembarang : ")
    fmt.Scanln(&titik.X, &titik.Y)
    hasil := tentukanPosisiTitik(lingkaran1, lingkaran2,
titik)
    fmt.Println("Posisi titik : ", hasil)
}

```

Screenshot Output



```
37
38 func main() {
39     var lingkaran1, lingkaran2 Lingkaran
40     var titik Titik
41     fmt.Println("Masukkan pusat dan radius lingkaran 1 : ")
42     fmt.Scanln(&lingkaran1.Tengah.X, &lingkaran1.Tengah.Y, &lingkaran1.Radius)
43     fmt.Println("Masukkan pusat dan radius lingkaran 2 : ")
44     fmt.Scanln(&lingkaran2.Tengah.X, &lingkaran2.Tengah.Y, &lingkaran2.Radius)
45     fmt.Println("Masukkan koordinat titik sembarang : ")
46     fmt.Scanln(&titik.X, &titik.Y)
47     hasil := tentukanPosisiTitik(lingkaran1, lingkaran2, titik)
48     fmt.Println("Posisi titik : ", hasil)
49 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS D:\Semester 3\Praktikum Alpro 2\Modul 7> go run "d:\Semester 3\Praktikum Alpro 2\Modul 7\Ungu1.go"
Masukkan pusat dan radius lingkaran 1 : 1 1
Masukkan pusat dan radius lingkaran 2 : 2 2
Masukkan koordinat titik sembarang : 3 3
Posisi titik : Titik berada di dalam lingkaran 2
PS D:\Semester 3\Praktikum Alpro 2\Modul 7>
```

Deskripsi Program

Program ini menentukan posisi sebuah titik terhadap dua lingkaran berdasarkan jarak antara titik tersebut dan pusat masing-masing lingkaran. Program dimulai dengan mendefinisikan dua struktur, yaitu `Titik` yang menyimpan koordinat `(X, Y)` dan `Lingkaran` yang menyimpan pusat `Titik` serta radius lingkaran. Fungsi `hitungJarak` menghitung jarak antara dua titik menggunakan rumus Euclidean, sementara fungsi `tentukanPosisiTitik` menggunakan `hitungJarak` untuk menentukan posisi titik sembarang berdasarkan jaraknya dari pusat kedua lingkaran. Jika jarak titik ke pusat lingkaran kurang atau sama dengan radius lingkaran, maka titik tersebut berada di dalam lingkaran tersebut. Fungsi ini mengembalikan status posisi titik: apakah berada di dalam lingkaran 1 dan 2, di dalam lingkaran 1 saja, di dalam lingkaran 2 saja, atau di luar kedua lingkaran. Di dalam fungsi `main`, program menerima input koordinat dan radius dari kedua lingkaran serta koordinat titik sembarang. Hasil akhirnya menampilkan status posisi titik yang telah dihitung. Sebagai contoh, jika lingkaran 1 memiliki pusat di (0, 0) dengan radius 5, lingkaran 2 di (8, 0) dengan radius 5, dan titik berada di (3, 0), maka hasil yang ditampilkan adalah "Titik berada di dalam lingkaran 1".

2. UNGUIDED 2

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. 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"
)

func main() {
    elemenArray := make([]int, 10)
    var jumlahElemen int
    fmt.Print("Masukkan jumlah elemen (maksimal 10) : ")
    fmt.Scanln(&jumlahElemen)
    if jumlahElemen > 10 {
        jumlahElemen = 10
        fmt.Println("Jumlah elemen dibatasi maksimal 10. Menggunakan 10 elemen.")
    }
    fmt.Println("Masukkan elemen-elemen array : ")
    for i := 0; i < jumlahElemen; i++ {
        var elemen int
        fmt.Scanln(&elemen)
        elemenArray[i] = elemen
    }
    for {
        fmt.Println("\nMenu pilihan : ")
        fmt.Println("1. Menampilkan seluruh isi array")
        fmt.Println("2. Menampilkan elemen array dengan indeks ganjil")
        fmt.Println("3. Menampilkan elemen array dengan indeks genap")
        fmt.Println("4. Menampilkan elemen array dengan indeks kelipatan bilangan x")
    }
}
```

```

        fmt.Println("5. Menghapus elemen array pada indeks
tertentu")
        fmt.Println("6. Menampilkan rata-rata nilai dalam
array")
        fmt.Println("7. Menampilkan standar deviasi dari
nilai dalam array")
        fmt.Println("8. Menampilkan frekuensi kemunculan
suatu bilangan dalam array")
        fmt.Println("9. Keluar")

        var pilihan int
        fmt.Print("\nMasukkan pilihan : ")
        fmt.Scanln(&pilihan)

        switch pilihan {
            case 1 : tampilkanSeluruhIsiArray(elemenArray,
jumlahElemen)
            case 2 :
                tampilkanElemenIndeksGanjil(elemenArray,
jumlahElemen)
            case 3:
                tampilkanElemenIndeksGenap(elemenArray,
jumlahElemen)
            case 4:
                tampilkanElemenKelipatan(elemenArray,
jumlahElemen)
            case 5:
                jumlahElemen = hapusElemenArray(elemenArray,
jumlahElemen)
            case 6:
                tampilkanRataRata(elemenArray, jumlahElemen)
            case 7:
                tampilkanStandarDeviasi(elemenArray,
jumlahElemen)
            case 8:
                tampilkanFrekuensi(elemenArray, jumlahElemen)
            case 9:
                fmt.Println("Terima kasih! Program selesai.")
                return // Keluar dari program
            default:
                fmt.Println("Pilihan tidak valid")
        }
    }
}

func tampilkanSeluruhIsiArray(array []int, jumlahElemen
int) {
    fmt.Println("Isi array:")
    for i := 0; i < jumlahElemen; i++ {
        fmt.Println(array[i])
    }
}

```

```

func      tampilkanElemenIndeksGanjil(array []int,
jumlahElemen int) {
    fmt.Println("Elemen dengan indeks ganjil:")
    for i := 1; i < jumlahElemen; i += 2 {
        fmt.Println(array[i])
    }
}

func tampilkanElemenIndeksGenap(array []int, jumlahElemen
int) {
    fmt.Println("Elemen dengan indeks genap:")
    for i := 0; i < jumlahElemen; i += 2 {
        fmt.Println(array[i])
    }
}

func tampilkanElemenKelipatan(array []int, jumlahElemen
int) {
    var x int
    fmt.Print("Masukkan bilangan kelipatan: ")
    fmt.Scanln(&x)
    fmt.Println("Elemen dengan indeks kelipatan", x, ":")
    for i := 0; i < jumlahElemen; i++ {
        if i%x == 0 {
            fmt.Println(array[i])
        }
    }
}

func hapusElemenArray(array []int, jumlahElemen int) int
{
    var indeks int
    fmt.Print("Masukkan indeks elemen yang ingin dihapus:
")
    fmt.Scanln(&indeks)
    if indeks < 0 || indeks >= jumlahElemen {
        fmt.Println("Indeks tidak valid")
        return jumlahElemen // Kembalikan jumlahElemen
tanpa perubahan
    }
    for i := indeks; i < jumlahElemen-1; i++ {
        array[i] = array[i+1]
    }
    jumlahElemen--
    fmt.Println("Elemen pada indeks", indeks, "telah
dihapus")
    return jumlahElemen // Kembalikan jumlah elemen
yang baru
}

func tampilkanRataRata(array []int, jumlahElemen int) {

```

```

var total int
for i := 0; i < jumlahElemen; i++ {
    total += array[i]
}
rataRata := float64(total) / float64(jumlahElemen)
fmt.Printf("Rata-rata: %.2f\n", rataRata)
}

func tampilkanStandarDeviasi(array []int, jumlahElemen
int) {
    var total, rataRata float64
    for i := 0; i < jumlahElemen; i++ {
        total += float64(array[i])
    }
    rataRata = total / float64(jumlahElemen)
    var totalDeviasi float64
    for i := 0; i < jumlahElemen; i++ {
        deviasi := float64(array[i]) - rataRata
        totalDeviasi += deviasi * deviasi
    }
    standarDeviasi := totalDeviasi / float64(jumlahElemen)
    standarDeviasi = hitungAkarKuadrat(standarDeviasi)
    fmt.Printf("Standar deviasi: %.2f\n", standarDeviasi)
}

func hitungAkarKuadrat(x float64) float64 {
    if x < 0 {
        return 0
    }
    z := x / 2.0
    for i := 0; i < 10; i++ {
        z = (z + x/z) / 2.0
    }
    return z
}

func tampilkanFrekuensi(array []int, jumlahElemen int) {
    var bilangan int
    fmt.Print("Masukkan bilangan untuk mencari frekuensi:
")
    fmt.Scanln(&bilangan)
    frekuensi := 0
    for i := 0; i < jumlahElemen; i++ {
        if array[i] == bilangan {
            frekuensi++
        }
    }
    fmt.Printf("Frekuensi dari bilangan %d: %d\n",
bilangan, frekuensi)
}

```

Screenshoot Output

```
func tampilanFrekuensi(array []int, jumlahElemen int) {
    var bilangan int
    fmt.Print("Masukkan bilangan untuk mencari frekuensi: ")
    fmt.Scanln(&bilangan)
    frekuensi := 0
    for i := 0; i < jumlahElemen; i++ {
        if array[i] == bilangan {
            frekuensi++
        }
    }
}
```

PS D:\Semester 3\Praktikum Alpro 2\Modul 7> go run "d:\Semester 3\Praktikum Alpro 2\Modul 7\Ungu2.go"

Masukkan jumlah elemen (maksimal 10) : 2

Masukkan elemen-elemen array :

11

22

Menu pilihan :

1. Menampilkan seluruh isi array
2. Menampilkan elemen array dengan indeks ganjil
3. Menampilkan elemen array dengan indeks genap
4. Menampilkan elemen array dengan indeks kelipatan bilangan x
5. Menghapus elemen array pada indeks tertentu
6. Menampilkan rata-rata nilai dalam array
7. Menampilkan standar deviasi dari nilai dalam array
8. Menampilkan frekuensi kemunculan suatu bilangan dalam array
9. Keluar

Masukkan pilihan : 1

Isi array:

11

22

Menu pilihan :

1. Menampilkan seluruh isi array
2. Menampilkan elemen array dengan indeks ganjil

Deskripsi Program

Program ini menyediakan berbagai operasi pada array integer yang berisi maksimal 10 elemen. Pertama, program meminta pengguna memasukkan jumlah elemen array, yang dibatasi maksimal 10. Setelah itu, pengguna mengisi nilai-nilai array sesuai jumlah elemen yang telah ditentukan. Program menyediakan menu dengan berbagai pilihan operasi pada array: menampilkan seluruh isi array, menampilkan elemen dengan indeks ganjil atau genap, menampilkan elemen pada indeks kelipatan bilangan tertentu, menghapus elemen pada indeks tertentu, menghitung rata-rata nilai dalam array, menghitung standar deviasi nilai dalam array, serta menampilkan frekuensi kemunculan suatu bilangan di dalam array.

Setiap pilihan menu memiliki fungsi masing-masing untuk menjalankan operasi terkait. Misalnya, fungsi `tampilkanSeluruhIsiArray` untuk menampilkan semua elemen, sementara `tampilkanElemenIndeksGanjil` dan `tampilkanElemenIndeksGenap` menampilkan elemen-elemen pada indeks ganjil dan genap. Fungsi `tampilkanElemenKelipatan` menampilkan elemen dengan indeks kelipatan bilangan yang dimasukkan pengguna. Program juga mendukung penghapusan elemen menggunakan `hapusElemenArray`, yang menggeser elemen setelah indeks yang dihapus. Selanjutnya, `tampilkanRataRata` menghitung rata-rata, dan `tampilkanStandarDeviasi` menghitung standar deviasi dengan menggunakan rumus matematika dasar. Untuk perhitungan akar kuadrat,

fungsi `hitungAkarKuadrat` digunakan. Terakhir, `tampilkanFrekuensi` menghitung dan menampilkan berapa kali suatu bilangan muncul dalam array. Program berjalan dalam loop sampai pengguna memilih opsi untuk keluar dari menu, yang mengakhiri program.

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 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. Perhatikan sesi interaksi pada contoh berikut ini (teks bergaris bawah adalah input/read)

```
Klub A : MU
Klub B : Inter
Pertandingan 1 : 2 0 // MU = 2 sedangkan Inter = 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
```

Sourcecode

```
package main

import "fmt"

func main() {
    var namaKlubA, namaKlubB string
    var skorKlubA, skorKlubB int
    var daftarHasil []string

    fmt.Print("Masukkan nama klub A : ")
```



```

        fmt.Scanln(&namaKlubA)
        fmt.Print("Masukkan nama klub B : ")
        fmt.Scanln(&namaKlubB)
        for {
            fmt.Printf("Masukkan skor untuk %s dan %s : ",
namaKlubA, namaKlubB)
            _, err := fmt.Scanf("%d %d\n", &skorKlubA,
&skorKlubB)

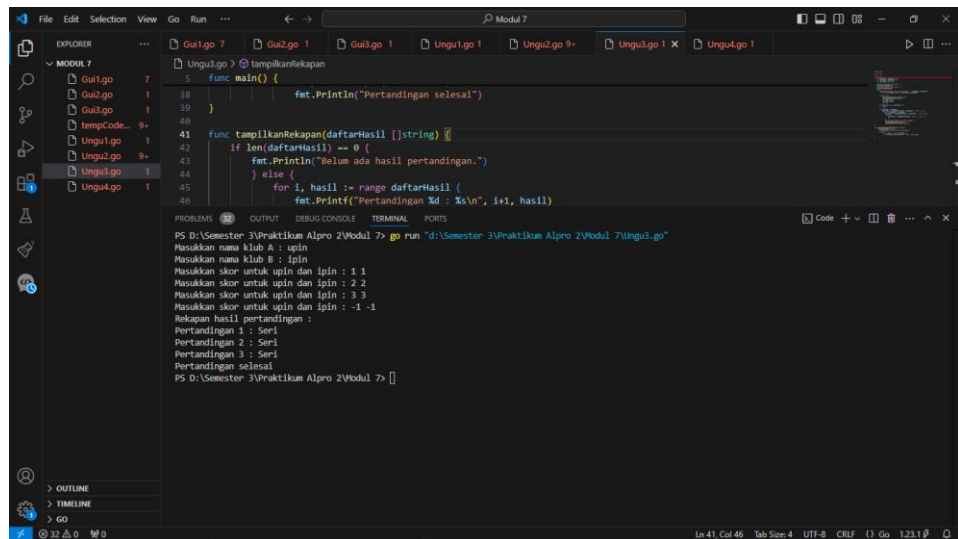
            if err != nil {
                fmt.Println("Inputan tidak valid")
                tampilkanRekapan(daftarHasil)
                var temp string
                fmt.Scanln(&temp)
                continue
            }
            if skorKlubA < 0 || skorKlubB < 0 {
                break
            }
            if skorKlubA > skorKlubB {
                daftarHasil = append(daftarHasil, namaKlubA)
// Klub A menang
            } else if skorKlubA < skorKlubB {
                daftarHasil = append(daftarHasil,
namaKlubB) // Klub B menang
            } else {
                daftarHasil = append(daftarHasil,
"Seri") // Hasil seri
            }
        }
        fmt.Println("Rekapan hasil pertandingan
:")

        tampilkanRekapan(daftarHasil)
        fmt.Println("Pertandingan selesai")
    }

    func tampilkanRekapan(daftarHasil []string) {
        if len(daftarHasil) == 0 {
            fmt.Println("Belum ada hasil pertandingan.")
        } else {
            for i, hasil := range daftarHasil {
                fmt.Printf("Pertandingan %d : %s\n", i+1,
hasil)
            }
        }
    }
}

```

Screenshoot Output



```
func main() {
    fat.Println("Pertandingan selesai")
}

func tampilkanRekapan(daftarHasil []string) {
    if len(daftarHasil) == 0 {
        fat.Println("Belum ada hasil pertandingan.")
    } else {
        for i, hasil := range daftarHasil {
            fat.Printf("Pertandingan %d : %s\n", i+1, hasil)
        }
    }
}
```

PS D:\Semester 3\Praktikum Alpro 2\Modul 7> go run "d:\Semester 3\Praktikum Alpro 2\Modul 7\Ungu3.go"

Masukkan nama klub A : upin
Masukkan nama klub B : ipin
Masukkan skor untuk upin dan ipin : 1 1
Masukkan skor untuk upin dan ipin : 2 2
Masukkan skor untuk upin dan ipin : 3 3
Masukkan skor untuk upin dan ipin : -1 -1
Rekapan hasil pertandingan :
Pertandingan 1 : Seri
Pertandingan 2 : Seri
Pertandingan 3 : Seri
Pertandingan selesai
PS D:\Semester 3\Praktikum Alpro 2\Modul 7>

Deskripsi Program

Program ini dirancang untuk mencatat hasil pertandingan antara dua klub sepak bola, yaitu Klub A dan Klub B. Program dimulai dengan meminta pengguna memasukkan nama kedua klub. Selanjutnya, program memasuki loop di mana pengguna dapat memasukkan skor untuk kedua klub setelah setiap pertandingan. Jika skor Klub A lebih tinggi dari Klub B, hasil pertandingan akan dicatat sebagai kemenangan Klub A. Jika sebaliknya, kemenangan diberikan kepada Klub B. Jika skor kedua klub sama, hasil pertandingan akan dicatat sebagai "Seri".

Setiap hasil pertandingan disimpan dalam array `daftarHasil`. Jika pengguna memasukkan nilai negatif untuk salah satu skor, loop berakhir, menandakan akhir sesi pencatatan. Di akhir program, rekapan hasil pertandingan yang telah dicatat akan ditampilkan. Fungsi `tampilkanRekapan` digunakan untuk mencetak seluruh hasil pertandingan. Jika belum ada pertandingan yang tercatat, pesan "Belum ada hasil pertandingan." akan ditampilkan.

4. UNGUIDED 4

Studi Case : Sebuah array digunakan untuk menampung sekumpulan karakter, Anda diminta untuk membuat subprogram untuk melakukan membalikkan urutan isi array dan memeriksa apakah membentuk palindrom. Lengkapi potongan algoritma berikut ini!

```

package main
import "fmt"
const NMAX int = 127
type tabel [NMAX]rune
    tab : tabel
    m : integer

func isiArray(t *tabel, n *int)
/*I.S. Data tersedia dalam piranti masukan
F.S. Array t berisi sejumlah n karakter yang dimasukkan user,
Proses input selama karakter bukanlah TITIK dan n <= NMAX */

func cetakArray(t tabel, n int)
/*I.S. Terdefinisi array t yang berisi sejumlah n karakter
F.S. n karakter dalam array muncul di layar */

func balikanArray(t *tabel, n int)
/*I.S. Terdefinisi array t yang berisi sejumlah n karakter
F.S. Urutan isi array t terbalik */

func main(){
    var tab tabel
    var m int
    // si array tab dengan memanggil prosedur isiArray

    // Balikian isi array tab dengan memanggil balikanArray

    // Cetak is array tab
}

```

Perhatikan sesi interaksi pada contoh berikut ini (**teks bergaris bawah adalah Input/read**)

```

Teks      : S E N A N G .
Reverse teks : G N A N E S

Teks      : K A T A K .
Reverse teks : K A T A K

```

Modifikasi program tersebut dengan menambahkan fungsi palindrom. Tambahkan instruksi untuk memanggil fungsi tersebut dan menampilkan hasilnya pada program utama.

***Palindrom adalah teks yang dibaca dari awal atau akhir adalah sama, contoh: KATAK, APA, KASUR_RUSAK.**

```

func palindrom(t tabel, n int) bool
/* Mengembalikan true apabila susunan karakter di dalam t membentuk palindrom,
dan false apabila sebaliknya. Petunjuk: Manfaatkan prosedur balikanArray */

```

Perhatikan sesi interaksi pada contoh berikut ini (**teks bergaris bawah adalah Input/read**)

```

Teks      : K A T A K
Palindrom : ? true

Teks      : S E N A N G
Palindrom : ? false

```

Sourcecode

```

package main

import "fmt"

const MaksKarakter int = 127

type arrayKarakter [MaksKarakter]rune

```

```

func inputArrayKarakter(array *arrayKarakter, jumlah int)
{
    var indeks int
    var karakter rune
    fmt.Print("Masukkan karakter : ")
    for indeks = 0; indeks < jumlah; indeks++ {
        fmt.Scanf("%c", &karakter)
        if karakter == '.' {
            break
        }
        array[indeks] = karakter
    }
}

func tampilkanKarakter(array arrayKarakter, jumlah int) {
    for i := 0; i < jumlah; i++ {
        fmt.Printf("%c", array[i])
    }
    fmt.Println()
}

func balikkanUrutan(array *arrayKarakter, jumlah int) {
    for i := 0; i < jumlah/2; i++ {
        sementara := array[i]
        array[i] = array[jumlah-i-1]
        array[jumlah-i-1] = sementara
    }
}

func cekApakahPalindrom(array arrayKarakter, jumlah int)
bool {
    var arraySalinan arrayKarakter
    copy(arraySalinan[:], array[:])
    balikkanUrutan(&arraySalinan, jumlah)
    for i := 0; i < jumlah; i++ {
        if arraySalinan[i] != array[i] {
            return false
        }
    }
    return true
}

func main() {
    var array arrayKarakter
    var jumlahKarakter int
    fmt.Print("Masukkan jumlah karakter : ")
    fmt.Scanln(&jumlahKarakter)
    inputArrayKarakter(&array, jumlahKarakter)
    fmt.Print("Karakter yang dimasukkan adalah : ")
    tampilkanKarakter(array, jumlahKarakter)
    balikkanUrutan(&array, jumlahKarakter)
    fmt.Print("Teks terbalik (reverse) adalah : ")
    tampilkanKarakter(array, jumlahKarakter)
    if cekApakahPalindrom(array, jumlahKarakter) {

```

```

        fmt.Print("Array ini adalah palindrom.")
    } else {
        fmt.Print("Array ini bukan palindrom.")
    }
}

```

Screenshoot Output

The screenshot shows a Go IDE with the following components:

- Explorer:** Shows a project structure with files `Gui1.go`, `Gui2.go`, `Gui3.go`, `Ungu1.go`, `Ungu2.go`, `Ungu3.go`, and `Ungu4.go`.
- Editor:** Displays the source code for `Ungu4.go`. The code includes functions for inputting characters, displaying them, reversing the string, and checking for a palindrome.
- Terminal:** Shows the execution output:


```

PS D:\Semester 3\Praktikum Alpro 2\Modul 7> go run "d:\Semester 3\Praktikum Alpro 2\Modul 7\Ungu4.go"
Masukkan jumlah karakter : 2
Masukkan karakter : ipin
Karakter yang dimasukkan adalah : ip
Teks terbalik (reverse) adalah : pi
Array ini bukan palindrom.
PS D:\Semester 3\Praktikum Alpro 2\Modul 7>

```

Deskripsi Program

Program ini dirancang untuk menerima input berupa sejumlah karakter dari pengguna, kemudian menampilkan karakter dalam urutan normal, menampilkan karakter dalam urutan terbalik, dan memeriksa apakah urutan karakter tersebut membentuk sebuah palindrom (mempunyai urutan yang sama jika dibaca dari depan maupun belakang). Program ini membatasi jumlah karakter maksimal hingga 127. Pada langkah pertama, pengguna diminta memasukkan karakter satu per satu hingga mencapai jumlah yang ditentukan atau hingga karakter titik (`. `) dimasukkan, yang akan mengakhiri input. Setelah itu, fungsi `tampilkanKarakter` akan menampilkan karakter dalam urutan yang sama seperti yang dimasukkan. Selanjutnya, fungsi `balikkanUrutan` akan membalikkan urutan karakter dan menampilkan hasilnya. Kemudian, fungsi `cekApakahPalindrom` akan membuat salinan dari array, membalikkan urutan salinan tersebut, dan membandingkannya dengan array asli untuk menentukan apakah array tersebut adalah palindrom.

IV. KESIMPULAN

Secara keseluruhan, materi mengenai struct dan array yang telah dibahas mencakup penggunaan kedua tipe data ini dalam bahasa pemrograman Go untuk memanipulasi dan menyimpan data secara terstruktur.

1. Struct

`struct` adalah tipe data yang memungkinkan kita untuk mengelompokkan berbagai tipe data yang berbeda dalam satu unit. Sebagai contoh, dalam program yang menggunakan tipe data `Titik` dan `Lingkaran`, struct digunakan untuk menyimpan pasangan koordinat `(X, Y)` pada `Titik`, serta pusat dan radius pada `Lingkaran`. Dengan struct, kita bisa mengorganisir data yang lebih kompleks, seperti data geometris atau entitas dunia nyata (misalnya, koordinat dan radius lingkaran) dalam satu objek.

- Pada penggunaan lain, `struct` juga dapat digunakan untuk mendefinisikan objek yang lebih besar, seperti data karakter dalam array, dengan memberikan nama yang jelas pada setiap elemen data.

2. Array

Array adalah struktur data yang digunakan untuk menyimpan sejumlah elemen dengan tipe yang sama dalam satu variabel. Array memungkinkan kita untuk mengakses dan memanipulasi banyak elemen dengan cara yang terorganisir.

- Dalam contoh program yang diberikan, array digunakan untuk menyimpan karakter-karakter yang dimasukkan oleh pengguna, dan kita dapat melakukan berbagai operasi seperti menampilkan, membalik urutan, dan memeriksa apakah urutan karakter membentuk sebuah palindrom.
- Array juga dapat digunakan untuk menyimpan hasil pertandingan dalam program rekapan hasil pertandingan atau menyimpan elemen-elemen numerik yang digunakan dalam perhitungan statistik.

V. REFERENSI

- [1] Modul VII Praktikum Algoritma dan Pemrograman 2
- [2] #30: Struct - Belajar golang dari dasar. (2022, October 11). Ruang Developer. <https://blog.ruangdeveloper.com/golang-struct/>
- [3] Golang Array - Dasar pemrograman golang. (n.d.). <https://dasarpemrogramangolang.novalagung.com/A-array.html>