

**LAPORAN PRAKTIKUM  
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL VII  
STRUCT & ARRAY**



**Disusun Oleh :**

**AFRIZAL DWI NUGRAHA / 2311102136**

**S1 IF 11 05**

**Dosen Pengampu :**

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

**PROGRAM STUDI S1 TEKNIK INFORMATIKA**

**FAKULTAS INFORMATIKA**

**TELKOM UNIVERSITY PURWOKERTO**

**2024**

## I. DASAR TEORI

Tipe bentukan memungkinkan pemrograman untuk mendefinisikan suatu tipe data baru pada suatu bahasa pemrograman. Tipe bentukan ini dapat dibedakan atas dua jenis, yaitu Alias dan Struct.

### 1. Alias (Type)

Bahasa pemrograman pada umumnya mengizinkan pemrograman untuk mengubah nama suatu tipe data dengan nama baru yang lebih ringkas dan familiar. Sebagai contoh "integer" dapat dirubah dengan nama alias "bilangan ". Caranya dengan menggunakan kata kunci "type".

|   | Notasi Algoritma              | Notasi dalam bahasa Go        |
|---|-------------------------------|-------------------------------|
| 1 | kamus                         | package main                  |
| 2 | type <nama alias> <tipe data> |                               |
| 3 |                               | type <nama alias> <tipe data> |
| 4 | algoritma                     |                               |
| 5 | ...                           | func main(){                  |
| 6 |                               | ...                           |
| 7 |                               | }                             |

program Go berikut beserta hasil eksekusinya!

```
1 package main
2 import "fmt"
3 type bilangan int
4 type pecahan float64
5 func main(){
6     var a,b bilangan
7     var hasil pecahan
8     a = 9
9     b = 5
10    hasil = pecahan(a) / pecahan(b)
11    fmt.Println(hasil)
12 }
```

```
E:\DEV\GO>go build Demo.go
```

```
E:\DEV\GO> Demo.exe
```

```
1.8
```

### 2. Array

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

|    | Notasi dalam bahasa Go   |
|----|--|
| 1  | var (  |
| 2  | // array arr mempunyai 73 elemen, masing-masing bertipe CircType2      |
| 3  | arr [73]CircType   |
| 4  |  |
| 5  | // array buf dengan 5 elemen, dengan nilai awal 7, 3, 5, 2, dan 11.    |
| 6  | buf = [5]byte{7, 3, 5, 2, 11}  |
| 7  |  |
| 8  | // mhs adalah array dengan 2000 elemen bertipe NewType                 |
| 9  | mhs [2000]NewType  |
| 10 |  |
| 11 | // rec adalah array dari array, yaitu matriks, atau array berdimensi-2 |
| 12 | rec [20][40]float64  |
| 13 | )  |

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

Indeks array dimulai dari 0, sehingga indeks arr pada contoh adalah 0, 1.. len(arr)-1 Contoh:

```
1 // Mengganti isi elemen ke-0 dengan nilai dari elemen ke-7
2 arr[0] = arr[7]
3
4 // Mengambil data field x dari elemen ke-i
5 currX = arr[i].center.x
6
7 // Mengambil elemen terakhir
8 n := len(arr)
9 buf := arr[n-1]
```

## II. GUIDED

### 1. Program Menghitung Lama Parkir

#### 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

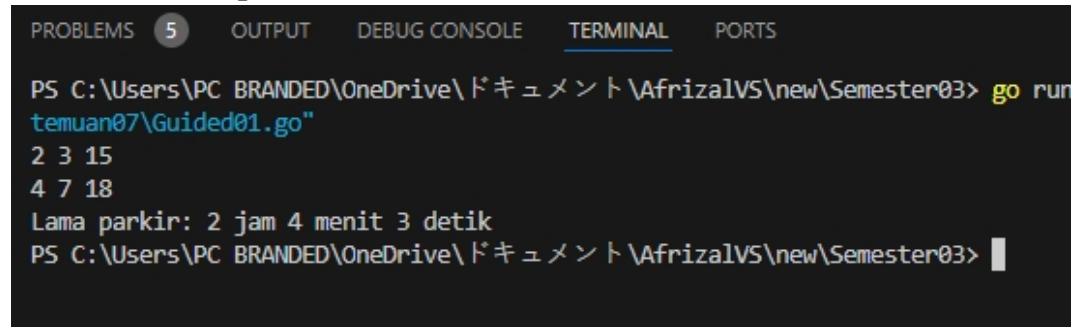
    durasi.jam = lParkir / 3600
    durasi.menit = lParkir % 3600 / 60
```

```

    durasi.detik = IParkir % 3600 % 60
    fmt.Printf("Lama parkir: %d jam %d menit %d detik",
        durasi.jam, durasi.menit, durasi.detik)
}

```

## Screenshoot Output



```

PROBLEMS 5 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\PC BRANDED\OneDrive\ドキュメント\Afriza\VS\new\Semester03> go run
temuan07\Guided01.go
2 3 15
4 7 18
Lama parkir: 2 jam 4 menit 3 detik
PS C:\Users\PC BRANDED\OneDrive\ドキュメント\Afriza\VS\new\Semester03>

```

## Deskripsi Program

Program ini menghitung durasi waktu parkir kendaraan berdasarkan waktu datang (wParkir) dan waktu pulang (wPulang). Struktur data waktu mewakili jam, menit, dan detik.

## 2. Program Slice

### Sourcecode

```

package main

import (
    "fmt"
)

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

func main() {
    // Slice awal untuk daftar teman dengan beberapa data

```

```

daftarTeman := []string{"Andi ", "Budi ", "Cici "}

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

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

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

```

### Screenshoot Output

```

PROBLEMS 7 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\PC BRANDED\OneDrive\ドキュメント\AfrizalVS\new\Semester03> go run
temuan07\Guided02.go
Nama Budi sudah ada dalam daftar.
Daftar Teman: [Andi Budi Cici Dewi Eka]
PS C:\Users\PC BRANDED\OneDrive\ドキュメント\AfrizalVS\new\Semester03>

```

### Deskripsi Program

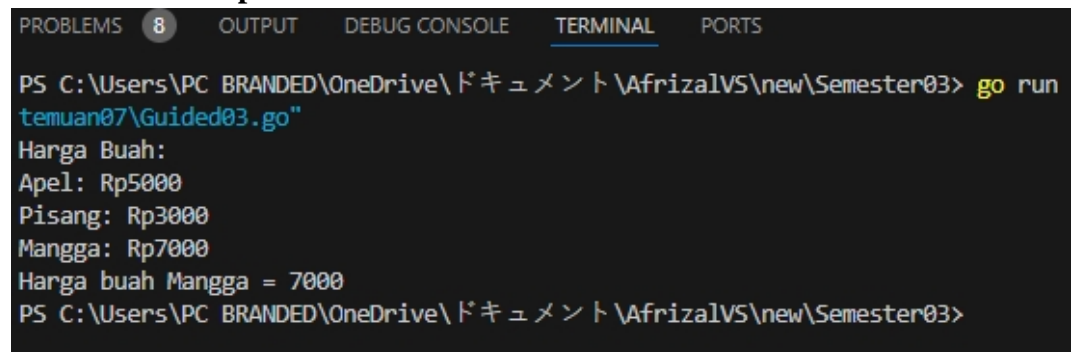
Program adalah untuk menambahkan nama baru ke dalam daftar teman untuk mencegah duplikasi nama. Jika nama yang akan ditambahkan sudah ada di daftar, program akan memberi tahu Anda bahwa nama tersebut sudah ada di daftar.

### 3. Program Map

#### Sourcecode

```
import (  
    "fmt"  
)  
  
func main() {  
    // Membuat map dengan nama buah sebagai kunci dan harga  
    sebagai nilai  
    hargaBuah := map[string]int{  
        "Apel ": 5000,  
        "Pi sang": 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



```
PROBLEMS 8 OUTPUT DEBUG CONSOLE TERMINAL PORTS  
PS C:\Users\PC BRANDED\OneDrive\ドキュメント\AfrizalVS\new\Semester03> go run  
temuan07\Guided03.go  
Harga Buah:  
Apel: Rp5000  
Pisang: Rp3000  
Mangga: Rp7000  
Harga buah Mangga = 7000  
PS C:\Users\PC BRANDED\OneDrive\ドキュメント\AfrizalVS\new\Semester03>
```

#### Deskripsi Program

Program ini dapat menyimpan data harga buah dalam map Go, dengan nama buah digunakan sebagai kunci dan harga buah sebagai nilai. Program ini juga menampilkan daftar harga buah dan memungkinkan Anda menemukan harga buah tertentu berdasarkan namanya.

### III. UNGUIDED

1. 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**".

Fungsi untuk menghitung jarak titik (a,b) dan (c,d) di mana rumus jarak adalah:  $jarak = \sqrt{(a - c)^2 + (b - d)^2}$  dan juga fungsi untuk menentukan posisi sebuah titik sembarang berada di dalam suatu lingkaran atau tidak.

```
function jarak(p, q : titik) -> real
{Mengembalikan jarak antara titik p(x,y) dan titik q(x,y)}

function didalam(c:lingkaran, p:titik) -> boolean
{Mengembalikan true apabila titik p(x,y) berada di dalam lingkaran c yang
memiliki titik pusat (cx,cy) dan radius r}
```

#### Sourcecode

```
package main

import (
    "fmt"
    "math"
)

// Definisi struct
type titik struct {
    x, y int
}

type lingkaran struct {
    pusat titik
    radius int
}
```

```

}

// Fungsi untuk menghitung jarak antar dua titik
func jarak(p, q titik) float64 {
    dx := float64(p.x - q.x)
    dy := float64(p.y - q.y)
    return math.Sqrt(dx*dx + dy*dy)
}

// Fungsi untuk mengecek apakah titik berada di dalam
lingkaran
func di dalam(c lingkaran, p titik) bool {
    return jarak(c.pusat, p) <= float64(c.radius)
}

func main() {
    // Input data lingkaran
    var l1, l2 lingkaran
    fmt.Println("Masukkan data lingkaran 1 (x y radius):")
    fmt.Scan(&l1.pusat.x, &l1.pusat.y, &l1.radius)
    fmt.Println("Masukkan data lingkaran 2 (x y radius):")
    fmt.Scan(&l2.pusat.x, &l2.pusat.y, &l2.radius)

    // Input data titik
    var t titik
    fmt.Println("Masukkan data titik (x y):")
    fmt.Scan(&t.x, &t.y)

    // Mengevaluasi posisi titik
    diLingkaran1 := di dalam(l1, t)
    diLingkaran2 := di dalam(l2, t)

    // Menentukan hasil output
    if diLingkaran1 && diLingkaran2 {
        fmt.Println("Titik di dalam lingkaran 1 dan 2")
    } else if diLingkaran1 {
        fmt.Println("Titik di dalam lingkaran 1")
    } else if diLingkaran2 {
        fmt.Println("Titik di dalam lingkaran 2")
    } else {
        fmt.Println("Titik di luar lingkaran 1 dan 2")
    }
}

```



### Screenshoot Output

```
PS C:\Users\PC BRANDED\OneDrive\ドキュメント\AfriZalVS\new\Semester03> go run
temuan07\Unguided01.go"
Masukkan data lingkaran 1 (x y radius):
1 1 5
Masukkan data lingkaran 2 (x y radius):
8 8 4
Masukkan data titik (x y):
15 20
Titik di luar lingkaran 1 dan 2
PS C:\Users\PC BRANDED\OneDrive\ドキュメント\AfriZalVS\new\Semester03> |
```

### Deskripsi Program

Dengan menggunakan koordinat titik pusat dan radius masing-masing lingkaran, program bertujuan untuk mengetahui posisi sebuah titik terhadap dua lingkaran. Program akan menilai apakah titik berada di dalam salah satu lingkaran, di dalam kedua lingkaran, atau di luar keduanya.

2. Sebuah array digunakan untuk menampung sekumpulan bilangan bulat. Buatlah program yang digunakan untuk mengisi array tersebut sebanyak N elemen nilai. Asumsikan array memiliki kapasitas penyimpanan data sejumlah elemen tertentu. Program dapat menampilkan beberapa informasi berikut:
- Menampilkan keseluruhan isi dari array.
  - Menampilkan elemen-elemen array dengan indeks ganjil saja.
  - Menampilkan elemen-elemen array dengan indeks genap saja (asumsi indeks ke-0 adalah genap).
  - Menampilkan elemen-elemen array dengan indeks kelipatan bilangan x. x bisa diperoleh dari masukan pengguna.
  - Menghapus elemen array pada indeks tertentu, asumsi indeks yang hapus selalu valid. Tampilkan keseluruhan isi dari arraynya, pastikan data yang dihapus tidak tampil
  - Menampilkan rata-rata dari bilangan yang ada di dalam array.
  - Menampilkan standar deviasi atau simpangan baku dari bilangan yang ada di dalam array tersebut.
  - Menampilkan frekuensi dari suatu bilangan tertentu di dalam array yang telah diisi tersebut.

### Sourcecode

```
package main

import (
    "fmt"
    "math"
)

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

    // Masukkan elemen array
    array := make([]int, n)
    fmt.Println("Masukkan elemen array:")
    for i := 0; i < n; i++ {
        fmt.Printf("Elemen ke-%d: ", i)
        fmt.Scan(&array[i])
    }

    // a. Menampilkan keseluruhan isi array
```

```

fmt.Println("Isi array:", array)

// b. Menampilkan elemen dengan indeks ganjil
fmt.Println("Elemen dengan indeks ganjil:")
for i := 1; i < len(array); i += 2 {
    fmt.Printf("Indeks %d: %d\n", i, array[i])
}

// c. Menampilkan elemen dengan indeks genap
fmt.Println("Elemen dengan indeks genap:")
for i := 0; i < len(array); i += 2 {
    fmt.Printf("Indeks %d: %d\n", i, array[i])
}

// d. Menampilkan elemen dengan indeks kelipatan x
var x int
fmt.Print("Masukkan bilangan x untuk kelipatan indeks:
")
fmt.Scan(&x)
fmt.Println("Elemen dengan indeks kelipatan", x, ":")
for i := 0; i < len(array); i++ {
    if i%x == 0 {
        fmt.Printf("Indeks %d: %d\n", i, array[i])
    }
}

// e. Menghapus elemen pada indeks tertentu
var idx int
fmt.Print("Masukkan indeks yang ingin dihapus: ")
fmt.Scan(&idx)
if idx >= 0 && idx < len(array) {
    array = append(array[:idx], array[idx+1:]...)
    fmt.Println("Isi array setelah dihapus:", array)
} else {
    fmt.Println("Indeks tidak valid.")
}

// f. Menghitung rata-rata
sum := 0
for _, val := range array {
    sum += val
}
avg := float64(sum) / float64(len(array))
fmt.Printf("Rata-rata: %.2f\n", avg)

```

```

// g. Menghitung standar deviasi
var varianceSum float64
for _, val := range array {
    varianceSum += math.Pow(float64(val)-avg, 2)
}
stdDev := math.Sqrt(varianceSum / float64(len(array)))
fmt.Printf("Standar Deviasi: %.2f\n", stdDev)

// h. Menghitung frekuensi suatu bilangan
var target int
fmt.Print("Masukkan bilangan untuk menghitung frekuensi: ")
fmt.Scan(&target)
count := 0
for _, val := range array {
    if val == target {
        count++
    }
}
fmt.Printf("Frekuensi bilangan %d: %d kali\n", target, count)
}

```

## Screenshoot Output

```

PROBLEMS 10 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\PC BRANDED\OneDrive\ドキュメント\AfrizalVS\new\Semester03> go run
temuan07\Unguided02.go
Masukkan jumlah elemen array: 4
Masukkan elemen array:
Elemen ke-0: 0
Elemen ke-1: 2
Elemen ke-2: 6
Elemen ke-3: 1
Isi array: [0 2 6 1]
Elemen dengan indeks ganjil:
Indeks 1: 2
Indeks 3: 1
Elemen dengan indeks genap:
Indeks 0: 0
Indeks 2: 6
Masukkan bilangan x untuk kelipatan indeks: 2
Elemen dengan indeks kelipatan 2 :
Indeks 0: 0
Indeks 2: 6
Masukkan indeks yang ingin dihapus: 0
Isi array setelah dihapus: [2 6 1]
Rata-rata: 3.00
Standar Deviasi: 2.16
Masukkan bilangan untuk menghitung frekuensi: 6
Frekuensi bilangan 6: 1 kali
PS C:\Users\PC BRANDED\OneDrive\ドキュメント\AfrizalVS\new\Semester03>

```

### Deskripsi Program

Ini adalah aplikasi berbasis array yang dirancang untuk mengelola data bilangan bulat. Program dapat memasukkan sejumlah elemen ke dalam array dan kemudian melakukan berbagai operasi untuk menampilkan informasi atau mengubah array. Operasi-operasi ini termasuk menampilkan elemen berdasarkan kondisi tertentu, menghitung rata-rata, simpangan baku, dan frekuensi bilangan.

3. 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. 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() {
    // Deklarasi variabel
    var klubA, klubB string
    var skorA, skorB int
    var pemenang []string
    var pertandingan int

    // Input nama klub
    fmt.Print("Klub A: ")
```

```

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

    // Inisialisasi pertandingan
    pertandingan = 1

    fmt.Println("\nMasukkan skor untuk setiap pertandingan
    (skor negatif untuk berhenti):")
    for {
        fmt.Printf("Pertandingan %d:\n", pertandingan)
        fmt.Printf("Skor %s: ", klubA)
        fmt.Scan(&skorA)
        fmt.Printf("Skor %s: ", klubB)
        fmt.Scan(&skorB)

        // Validasi skor negatif untuk menghentikan input
        if skorA < 0 || skorB < 0 {
            fmt.Println("\nPertandingan selesai.")
            break
        }

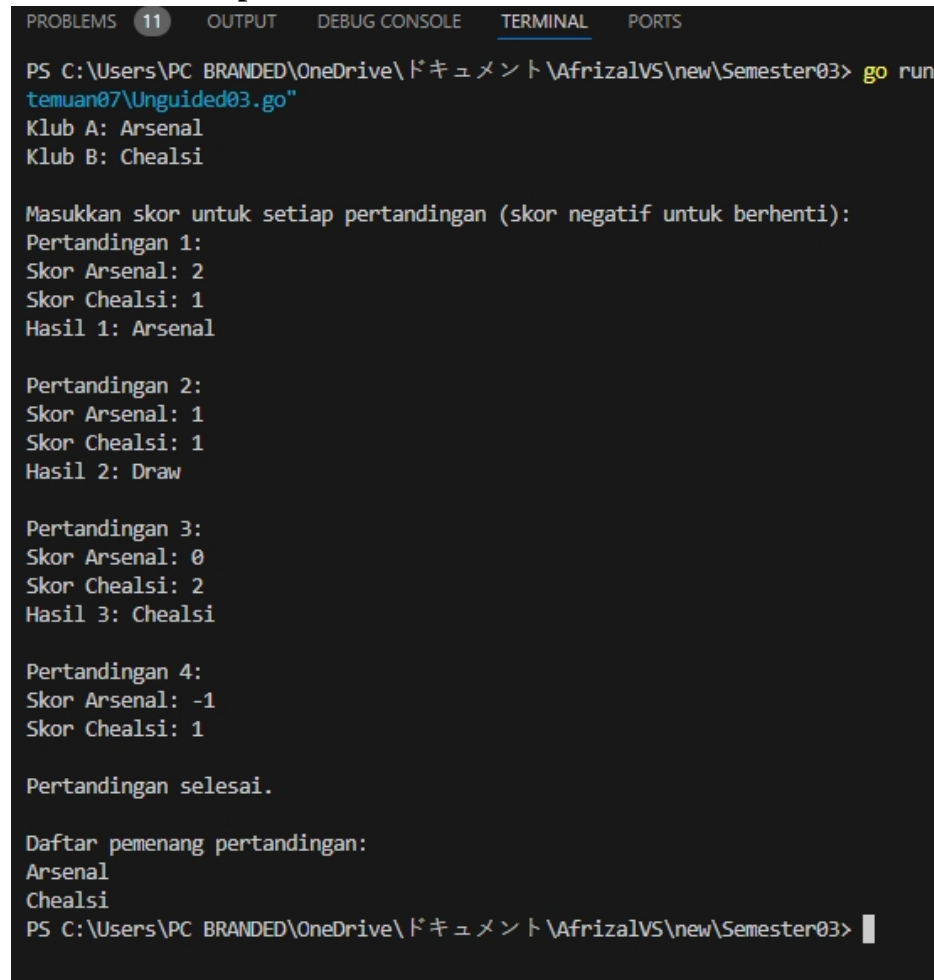
        // Menentukan pemenang
        if skorA > skorB {
            fmt.Printf("Hasil %d: %s\n\n", pertandingan,
            klubA)
            pemenang = append(pemenang, klubA)
        } else if skorB > skorA {
            fmt.Printf("Hasil %d: %s\n\n", pertandingan,
            klubB)
            pemenang = append(pemenang, klubB)
        } else {
            fmt.Printf("Hasil %d: Draw\n\n", pertandingan)
        }

        pertandingan++
    }

    // Menampilkan daftar pemenang
    fmt.Println("\nDaftar pemenang pertandingan:")
    for _, klub := range pemenang {
        fmt.Println(klub)
    }
}

```

## Screenshoot Output

A screenshot of a terminal window showing the execution of a Go program. The terminal has tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, and PORTS. The TERMINAL tab is active. The program prompts for the number of matches, then asks for the names of two clubs (A and B). It then enters a loop where it asks for scores for each match. The program tracks the scores and determines the winner at the end. The output shows four matches: Match 1 (Arsenal 2, Chealsi 1), Match 2 (Arsenal 1, Chealsi 1), Match 3 (Arsenal 0, Chealsi 2), and Match 4 (Arsenal -1, Chealsi 1). The final result shows Arsenal as the winner.

```
PS C:\Users\PC BRANDED\OneDrive\ドキュメント\Afriзал\VS\new\Semester03> go run
temuan07\Unguided03.go
Klub A: Arsenal
Klub B: Chealsi

Masukkan skor untuk setiap pertandingan (skor negatif untuk berhenti):
Pertandingan 1:
Skor Arsenal: 2
Skor Chealsi: 1
Hasil 1: Arsenal

Pertandingan 2:
Skor Arsenal: 1
Skor Chealsi: 1
Hasil 2: Draw

Pertandingan 3:
Skor Arsenal: 0
Skor Chealsi: 2
Hasil 3: Chealsi

Pertandingan 4:
Skor Arsenal: -1
Skor Chealsi: 1

Pertandingan selesai.

Daftar pemenang pertandingan:
Arsenal
Chealsi
PS C:\Users\PC BRANDED\OneDrive\ドキュメント\Afriзал\VS\new\Semester03>
```

## Deskripsi Program

Program yang direncanakan adalah program yang merekap skor berbagai pertandingan sepak bola antara dua klub di satu grup pertandingan. Program meminta pengguna untuk memasukkan nama klub dan skor klub tersebut. Dalam kondisi salah satu skor berupa nilai negatif, program akan menyelesaikan input. Penggunaan iterasi untuk menghentikan input masih belum diketahui, namun di akhir program, program akan menyatakan daftar klub yang menang.

4. Sebuah array digunakan untuk menampung sekumpulan karakter, Anda diminta untuk membuat sebuah 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 isi 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 NMAX int = 127

type tabel struct {
    tab [NMAX]rune
    m    int
}

// Fungsi untuk mengisi array dari input
func isiArray(t *tabel, n *int) {
    var ch rune
    fmt.Println("Masukkan karakter (akhiri dengan '.'): ")

    for {
        // Input karakter
        fmt.Scanf("%c\n", &ch)
        if ch == '.' || *n >= NMAX {
            break
        }
        t.tab[*n] = ch
        *n++
    }
    t.m = *n
}

// Fungsi untuk mencetak isi array
func cetakArray(t tabel, n int) {
    for i := 0; i < n; i++ {
        fmt.Printf("%c", t.tab[i])
    }
    fmt.Println()
}

// Fungsi untuk membalikkan isi array
func balikkanArray(t *tabel, n int) {
    for i, j := 0, n-1; i < j; i, j = i+1, j-1 {
        t.tab[i], t.tab[j] = t.tab[j], t.tab[i]
    }
}
```

```

}

// Fungsi untuk mengecek apakah array membentuk palindrom
func palindrom(t tabel, n int) bool {
    for i := 0; i < n/2; i++ {
        if t.tab[i] != t.tab[n-i-1] {
            return false
        }
    }
    return true
}

func main() {
    var tab tabel
    var n int

    // Mengisi array dengan input dari user
    isiArray(&tab, &n)

    // Menampilkan teks asli
    fmt.Print("Teks: ")
    cetakArray(tab, n)

    // Membalikkan array
    balikkanArray(&tab, n)

    // Menampilkan teks yang telah dibalik
    fmt.Print("Reverse teks: ")
    cetakArray(tab, n)

    // Mengecek apakah palindrom
    if palindrom(tab, n) {
        fmt.Println("Palindrom: true")
    } else {
        fmt.Println("Palindrom: false")
    }
}

```

## Screenshoot Output

```
PROBLEMS 12 OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\PC BRANDED\OneDrive\ドキュメント\AfriзалVS\new\Semester03> go run
temuan07\Unguided04.go
Masukkan karakter (akhiri dengan '.'):
H
I
D
U
P
.
Teks: HIDUP
Reverse teks: PUDIH
Palindrom: false
PS C:\Users\PC BRANDED\OneDrive\ドキュメント\AfriзалVS\new\Semester03> go run
temuan07\Unguided04.go
Masukkan karakter (akhiri dengan '.'):
K
A
P
A
K
.
Teks: KAPAK
Reverse teks: KAPAK
Palindrom: true
PS C:\Users\PC BRANDED\OneDrive\ドキュメント\AfriзалVS\new\Semester03> |
```

## Deskripsi Program

Program ini hanya mengambil serangkaian karakter dari pengguna dan melakukan tiga tugas: mencetak string asli, mencetak string terbalik, dan mencetak status belitan baris itu merupakan palindrom. Karakter yang dimasukkan oleh pengguna hanya ketikan satu persatu. Program juga sederhana dan berisikan logika sederhana untuk membalik string dan untuk memeriksa palindrom.

### **Daftar Pustaka :**

1. Modul 7, STRUCT & ARRAY