

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL 7
STRUCK & ARRAY**



Disusun Oleh :

Afif Rijal Azzami / 2311102235

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

A. Struct

Structure memungkinkan pemrograman untuk mengelompokkan beberapa data atau nilai yang memiliki relasi atau keterkaitan tertentu menjadi suatu kesatuan. Berbeda dengan bahasa pemrograman lain, kesamaan tipe dari dua variabel berjenis structure bukan karena namanya tetapi karena strukturnya. Dua variabel dengan nama-nama field dan tipe field yang sama (dan dalam urutan yang sama) dianggap mempunyai tipe yang sama. Tentunya akan lebih memudahkan jika structure tersebut didefinisikan sebagai sebuah tipe baru, sehingga deklarasi structure tidak perlu lagi seluruh field-nya ditulis ulang berkali-kali.

B. Array

Array adalah kumpulan-kumpulan variabel yang menyimpan data dengan tipe yang sama atau data-data yang tersusun secara linear dimana di dalamnya terdapat elemen dengan tipe yang sama. Indeks dalam array menyatakan elemen yang disimpan dan panjang atau length menyatakan total elemen yang tersimpan.

1. Slice

Array dalam Go juga dapat mempunyai ukuran yang dinamik. (Tidak digunakan di kelas Algoritma Pemrograman). Deklarasinya mirip dengan deklarasi array, tetapi jumlah elemennya dikosongkan.

2. Map

Tipe array lain, sebuah array dinamik. Indeksnya (di sini disebut kunci) tidak harus berbentuk integer. Indeks dapat berasal dari tipe apa saja. Struktur ini disebut map.

II. GUIDED 1

Sourcecode

```
package main

import "fmt"

type waktu struct {
    jam, menit, detik int
}

func main() {
    var wParkir, wPulang, durasi waktu
    var dParkir, dPulang, lParkir int

    fmt.Println("Masukan Waktu kedatangan:")

    fmt.Scan(&wParkir.jam, &wParkir.menit, &wParkir.detik)

    fmt.Println("Masukan Waktu pulang:")

    fmt.Scan(&wPulang.jam, &wPulang.menit, &wPulang.detik)

    dParkir = wParkir.detik + wParkir.menit*60 +
wParkir.jam*3600 // Konversi ke detik

    dPulang = wPulang.detik + wPulang.menit*60 +
wPulang.jam*3600 // detik

    lParkir = dPulang - dParkir //detik
dari pulang-datang

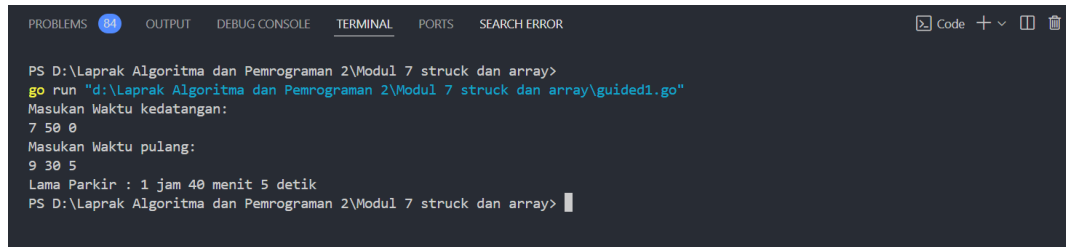
    durasi.jam = lParkir / 3600

    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



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SEARCH ERROR
PS D:\Laprak Algoritma dan Pemrograman 2\Modul 7 struck dan array>
go run "d:\Laprak Algoritma dan Pemrograman 2\Modul 7 struck dan array\guided1.go"
Masukan Waktu kedatangan:
7 50 0
Masukan Waktu pulang:
9 30 5
Lama Parkir : 1 jam 40 menit 5 detik
PS D:\Laprak Algoritma dan Pemrograman 2\Modul 7 struck dan array>
```

Soal Studi Case

Program sederhana untuk menghitung lama waktu parkir berdasarkan waktu kedatangan dan waktu pulang

Deskripsi Program

Sintaks tersebut adalah program untuk menghitung lama waktu parkir dengan mendeklarasikan struck yang dinamai waktu yang berisikan jam, menit, dan detik yang bertipe integer, pertama kita mendeklarasikan variabel wparkit, wpulang, dan wdurasi yang bertipe data waktu, jadi masing-masing variabel tersebut dapat menyimpan waktu dalam format jam, menit, dan detik. Kemudian untuk menghitung lama waktu kita perlu mengonversikan waktu kedatangan dan kepulangan ke detik agar bisa dioperasikan dengan rumus $wpulang - wparkit$ yang hasilnya disimpan di lparkir. Kemudian lparkir dikonversikan dari bentuk detik ke jam, menit.

III. GUIDED 2

```
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.")

    }

}

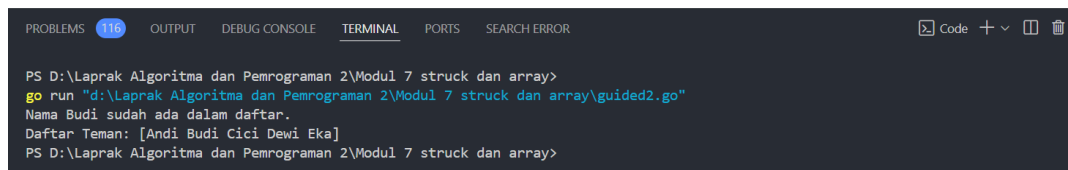
// Menampilkan daftar teman akhir

fmt.Println("Daftar Teman:", daftarTeman)

}

```

Screenshot Output



```

PS D:\Laprak Algoritma dan Pemrograman 2\Modul 7 struck dan array>
go run "d:\Laprak Algoritma dan Pemrograman 2\Modul 7 struck dan array\guided2.go"
Nama Budi sudah ada dalam daftar.
Daftar Teman: [Andi Budi Cici Dewi Eka]
PS D:\Laprak Algoritma dan Pemrograman 2\Modul 7 struck dan array>

```

Soal Studi Case

Mengelola daftar teman pada slice agar tidak duplikasi

Deskripsi Program

Sintaks tersebut adalah program untuk mengelola daftar teman dengan cara menambahkan nama-nama baru ke dalam daftar jika nama tersebut belum ada pada daftar teman dan melakukan validasi agar nama yang sudah ada tidak terduplikasi. Pertama kita mendeklarasikan fungsi yang dinamai sudahAda dengan parameter yang berisikan slice dan nama betipe string, fungsi tersebut akan melakukan iterasi menggunakan for terhadap setiap elemen teman dalam daftarTeman, Jika ditemukan elemen teman yang sama dengan nama, fungsi mengembalikan true, dan Jika selesai iterasi dan tidak ditemukan nama yang cocok, fungsi mengembalikan false. Selanjutnya kita mendeklarasikan slice yang dinami daftarTeman yang awalnya diisi dengan nama Andi, Budi, dan Cici. Selanjutnya kita melakukan pengecekan jika nama belum ada di slice daftarTeman, nama tersebut ditambahkan ke daftarTeman menggunakan fungsi append, dan jika nama sudah ada, program mencetak “nama tersebut sudah ada dalam daftar”.

IV. GUIDED 3

```
package main

import (
    "fmt"
)

func main() {

    // Membuat map dengan nama buah sebagai kunci dan harga
    sebagai nilai

    hargaBuah := map[string]int{

        "Apel": 5000,

        "Pisang": 3000,

        "Mangga": 7000,

    }

    // Menampilkan harga dari setiap buah

    fmt.Println("Harga Buah:")

    for buah, harga := range hargaBuah {

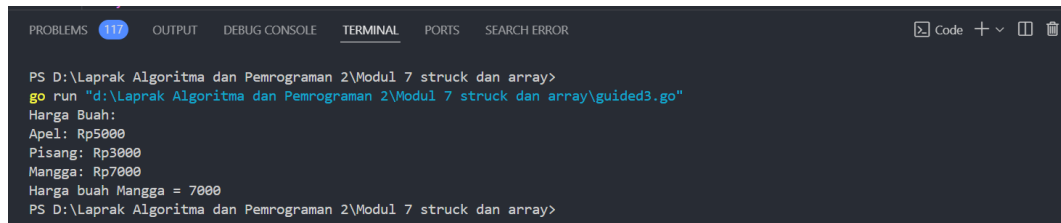
        fmt.Printf("%s: Rp%d\n", buah, harga)

    }

    fmt.Print("Harga buah Mangga = ", hargaBuah["Mangga"])

}
```

Screenshoot Output



```
PROBLEMS 117 OUTPUT DEBUG CONSOLE TERMINAL PORTS SEARCH ERROR
PS D:\Laprak Algoritma dan Pemrograman 2\Modul 7 struck dan array>
go run "d:\Laprak Algoritma dan Pemrograman 2\Modul 7 struck dan array\guided3.go"
Harga Buah:
Apel: Rp5000
Pisang: Rp3000
Mangga: Rp7000
Harga buah Mangga = 7000
PS D:\Laprak Algoritma dan Pemrograman 2\Modul 7 struck dan array>
```

Soal Studi Case

Implementasi map untuk mencetak harga buah

Deskripsi Program

Sintaks tersebut adalah program dengan menggunakan tipe data map di untuk menyimpan pasangan key-value, di mana nama buah berfungsi sebagai key, dan harga buah sebagai value. Pertama kita mendeklarasikan map dengan nama hargaBuah, dengan key-nya yaitu nama buah bertipe string dan value-nya harga buah bertipe integer. Kemudian untuk menampilkan map tersebut kita menggunakan iterasi.

I. UNGUIDED 1

Sourcecode

```
package main

import (
    "fmt"
    "math"
)

type Titik struct {
    x, y int
}

type Lingkaran struct {
    titikPusat Titik
    radius      int
}

func jarak(p, q Titik) float64 {
    return math.Sqrt(math.Pow(float64(p.x-q.x), 2) +
math.Pow(float64(p.y-q.y), 2))
}

func diDalamLingkaran(lingkaran Lingkaran, titik Titik) bool {
    return jarak(lingkaran.titikPusat, titik) <=
float64(lingkaran.radius)
}

func main() {
    var lingkaran1, lingkaran2 Lingkaran
    var titik Titik
```

```
fmt.Print("Masukkan titik pusat dan radius lingkaran 1 (cx cy  
r): ")

    fmt.Scan(&lingkaran1.titikPusat.x, &lingkaran1.titikPusat.y,  
&lingkaran1.radius)

    fmt.Print("Masukkan titik pusat dan radius lingkaran 2 (cx  
cy r): ")

    fmt.Scan(&lingkaran2.titikPusat.x, &lingkaran2.titikPusat.y,  
&lingkaran2.radius)

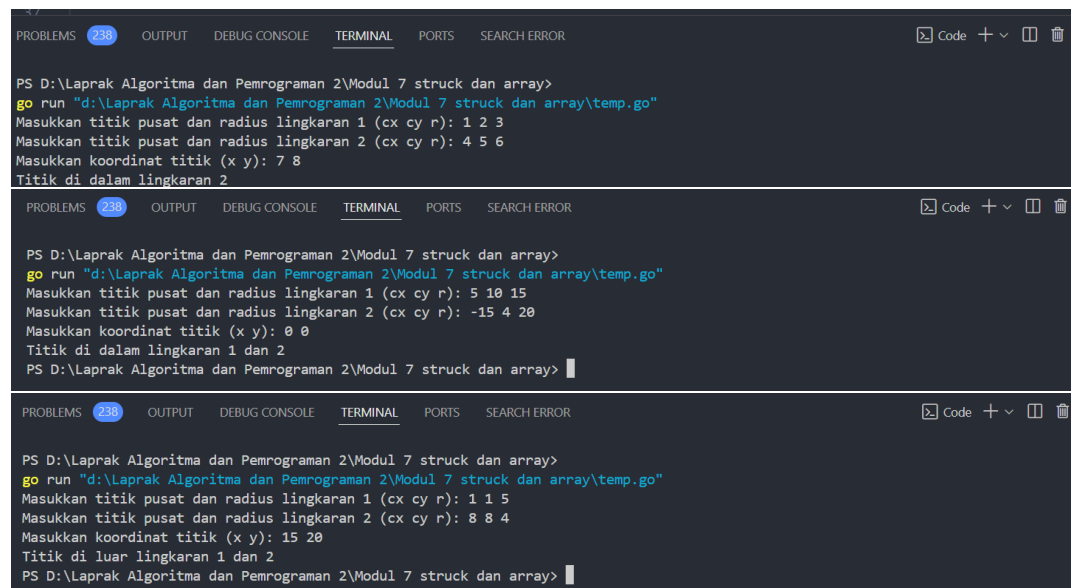
    fmt.Print("Masukkan koordinat titik (x y): ")

    fmt.Scan(&titik.x, &titik.y)

    dalamLingkaran1 := diDalamLingkaran(lingkaran1, titik)
    dalamLingkaran2 := diDalamLingkaran(lingkaran2, titik)

    if dalamLingkaran1 && dalamLingkaran2 {
        fmt.Println("Titik di dalam lingkaran 1 dan 2")
    } else if dalamLingkaran1 {
        fmt.Println("Titik di dalam lingkaran 1")
    } else if dalamLingkaran2 {
        fmt.Println("Titik di dalam lingkaran 2")
    } else {
        fmt.Println("Titik di luar lingkaran 1 dan 2")
    }
}
```

Screenshoot Output



```
PS D:\Laprak Algoritma dan Pemrograman 2\Modul 7 struck dan array>
go run "d:\Laprak Algoritma dan Pemrograman 2\Modul 7 struck dan array\temp.go"
Masukkan titik pusat dan radius lingkaran 1 (cx cy r): 1 2 3
Masukkan titik pusat dan radius lingkaran 2 (cx cy r): 4 5 6
Masukkan koordinat titik (x y): 7 8
Titik di dalam lingkaran 2

PS D:\Laprak Algoritma dan Pemrograman 2\Modul 7 struck dan array>
go run "d:\Laprak Algoritma dan Pemrograman 2\Modul 7 struck dan array\temp.go"
Masukkan titik pusat dan radius lingkaran 1 (cx cy r): 5 10 15
Masukkan titik pusat dan radius lingkaran 2 (cx cy r): -15 4 20
Masukkan koordinat titik (x y): 0 0
Titik di dalam lingkaran 1 dan 2
PS D:\Laprak Algoritma dan Pemrograman 2\Modul 7 struck dan array>

PS D:\Laprak Algoritma dan Pemrograman 2\Modul 7 struck dan array>
go run "d:\Laprak Algoritma dan Pemrograman 2\Modul 7 struck dan array\temp.go"
Masukkan titik pusat dan radius lingkaran 1 (cx cy r): 1 1 5
Masukkan titik pusat dan radius lingkaran 2 (cx cy r): 8 8 4
Masukkan koordinat titik (x y): 15 20
Titik di luar lingkaran 1 dan 2
PS D:\Laprak Algoritma dan Pemrograman 2\Modul 7 struck dan array>
```

Deskripsi Program

Sintaks tersebut adalah program untuk mengecek apakah sebuah titik berada di dalam, di luar, atau di dalam dua lingkaran. Pertama kita membuat fungsi yang dinamai Jarak untuk menghitung jarak titik dengan rumus akar dari $(x_2 - x_1)^2 + (y_2 - y_1)^2$. Kemudian kita membuat fungsi yang dinamai diDalamlingkaran untuk memeriksa apakah jarak dari suatu titik ke pusat lingkaran lebih kecil atau sama dengan radius lingkaran. Jika ya, titik berada di dalam lingkaran.

II. UNGUIDED 2

```
package main

import (
    "fmt"
    "math"
)

func main() {
    var n235 int

    fmt.Print("Masukkan jumlah elemen array: ")

    fmt.Scan(&n235)

    array235 := make([]int, n235)

    fmt.Println("Masukkan elemen array:")

    for i235 := 0; i235 < n235; i235++ {
        fmt.Printf("Elemen ke-%d: ", i235)

        fmt.Scan(&array235[i235])
    }

    for {
        fmt.Println("\n=== MENU ===")

        fmt.Println("a. Tampilkan keseluruhan isi array")

        fmt.Println("b. Tampilkan elemen array dengan indeks ganjil")

        fmt.Println("c. Tampilkan elemen array dengan indeks genap")

        fmt.Println("d. Tampilkan elemen array dengan indeks kelipatan x")

        fmt.Println("e. Hapus elemen array pada indeks tertentu")
    }
}
```

```
fmt.Println("e. Hapus elemen array pada indeks tertentu")

    fmt.Println("f. Tampilkan rata-rata dari bilangan dalam
array")

    fmt.Println("g. Tampilkan standar deviasi dari bilangan
dalam array")

    fmt.Println("h. Tampilkan frekuensi dari bilangan
tertentu")

    fmt.Println("i. Keluar")

    fmt.Print("Pilih opsi: ")

    var opsi235 string

    fmt.Scan(&opsi235)

    switch opsi235 {

    case "a":

        fmt.Println("Isi array:", array235)

    case "b":

        fmt.Println("Elemen dengan indeks ganjil:")

        for i235 := 1; i235 < n235; i235 += 2 {

            fmt.Printf("Index %d: %d\n", i235, array235[i235])

        }

    case "c":

        fmt.Println("Elemen dengan indeks genap:")

        for i235 := 0; i235 < n235; i235 += 2 {

            fmt.Printf("Index %d: %d\n", i235, array235[i235])

        }

    }
```

```

case "d":

    var x235 int

    fmt.Print("Masukkan nilai x: ")

    fmt.Scan(&x235)

    fmt.Println("Elemen dengan indeks kelipatan", x235,
":")

    for i235 := 0; i235 < n235; i235++ {

        if i235%x235 == 0 {

            fmt.Printf("Index %d: %d\n", i235,
array235[i235])

        }

    }

case "e":

    var index235 int

    fmt.Print("Masukkan indeks yang ingin dihapus: ")

    fmt.Scan(&index235)

    if index235 >= 0 && index235 < n235 {

        array235 = append(array235[:index235],
array235[index235+1:]...)

        n235--

        fmt.Println("Array setelah penghapusan:", array235)

    } else {

        fmt.Println("Indeks tidak valid!")

    }

```

```
case "f":

    sum235 := 0

    for _, val235 := range array235 {
        sum235 += val235
    }

    avg235 := float64(sum235) / float64(len(array235))

    fmt.Printf("Rata-rata: %.2f\n", avg235)


case "g":

    sum235 := 0

    for _, val235 := range array235 {
        sum235 += val235
    }

    avg235 := float64(sum235) / float64(len(array235))

    var variance235 float64

    for _, val235 := range array235 {
        variance235 += math.Pow(float64(val235)-avg235, 2)
    }

    variance235 /= float64(len(array235))

    stdDev235 := math.Sqrt(variance235)

    fmt.Printf("Standar deviasi: %.2f\n", stdDev235)
```

```
case "h":

    var num235 int

    fmt.Print("Masukkan bilangan yang ingin dicari  
frekuensinya: ")

    fmt.Scan(&num235)

    count235 := 0

    for _, val235 := range array235 {

        if val235 == num235 {

            count235++

        }

    }

    fmt.Printf("Frekuensi bilangan %d: %d kali\n", num235,  
count235)


case "i":

    fmt.Println("Program selesai.")

    return


default:

    fmt.Println("Opsi tidak valid!")

}

}
```


Screenshoot Output

```
Elemen ke-1: 10
Elemen ke-2: 13
Elemen ke-3: 25
Elemen ke-4: 45

=== MENU ===
a. Tampilkan keseluruhan isi array
b. Tampilkan elemen array dengan indeks ganjil
c. Tampilkan elemen array dengan indeks genap
d. Tampilkan elemen array dengan indeks kelipatan x
e. Hapus elemen array pada indeks tertentu
f. Tampilkan rata-rata dari bilangan dalam array
g. Tampilkan standar deviasi dari bilangan dalam array
h. Tampilkan frekuensi dari bilangan tertentu
i. Keluar
```

```
i. Keluar
Pilih opsi: a
Isi array: [12 10 13 25 45]
```

```
Pilih opsi: f
Rata-rata: 21.00
```

```
Pilih opsi: b
Elemen dengan indeks ganjil:
Index 1: 10
Index 3: 25
```

```
Pilih opsi: c
Elemen dengan indeks genap:
Index 0: 12
Index 2: 13
Index 4: 45
```

Deskripsi Program

Sintaks tersebut adalah program menu untuk mengoprasikan elemen pada array. Karna terdapat banyak menu saya akan menjelaskan beberapa yng ada di program menu tersebut. Pertama program akan meminta user untuk memasukan jumlah elemen pada array dan menginput isi array. Untuk menu a yaitu untuk menampilkan isi array, dan menu b untuk menampilkan indeks array yang ganjil dengan cara iterasi yang di mana $i = 1$ adalah indeks ganjil pertama, setiap iterasi, nilai i bertambah 2 untuk melompat ke indeks ganjil berikutnya, perulangan berjalan selama $i < n$, di mana n adalah jumlah elemen dalam array. Menu c untuk menampilkan indeks array yang genap dengan cara iterasi yang di mana $i = 0$, dimulai dari indeks 0 karena 0 adalah indeks genap pertama, setiap iterasi nilai i bertambah 2 untuk melompat ke indeks genap berikutnya, Perulangan berjalan selama $i < n$, di mana n adalah jumlah

elemen dalam array. Menu f untuk menampilkan rata-rata dari isi array dengan rumus hasil penjumlahan isi dalam array dibagi dengan jumlah panjang dari array.

III. UNGUDED 3

Sourcecode

```
package main

import "fmt"

func main() {

    var klubA235, klubB235 string

    var skorA235, skorB235 int

    var pemenang235 []string

    var pertandingan235 int = 1


    fmt.Print("Klub A: ")

    fmt.Scan(&klubA235)

    fmt.Print("Klub B: ")

    fmt.Scan(&klubB235)


    fmt.Println("\nMasukkan skor untuk setiap pertandingan. Masukkan skor negatif untuk menghentikan.")

    for {

        fmt.Printf("Pertandingan %d - Skor %s: ",
pertandingan235, klubA235)
```

```
fmt.Scan(&skorA235)

    fmt.Printf("Pertandingan %d - Skor %s: ",
pertandingan235, klubB235)

    fmt.Scan(&skorB235)


    if skorA235 < 0 || skorB235 < 0 {

        fmt.Println("Pertandingan selesai.")

        break

    }


    if skorA235 > skorB235 {

        pemenang235 = append(pemenang235, klubA235)

    } else if skorB235 > skorA235 {

        pemenang235 = append(pemenang235, klubB235)

    } else {

        pemenang235 = append(pemenang235, "Draw")

    }


    pertandingan235++

}


fmt.Println("\nHasil pertandingan:")

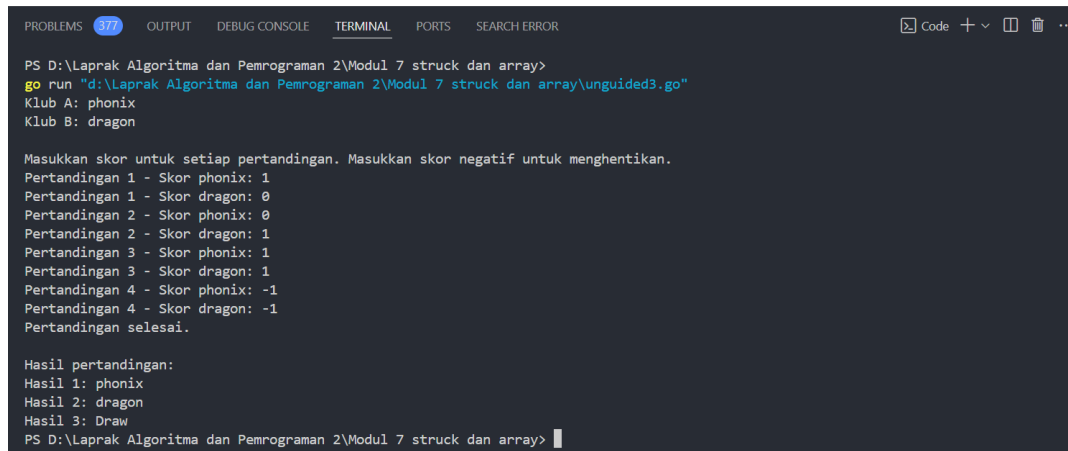
for i235, hasil235 := range pemenang235 {

    fmt.Printf("Hasil %d: %s\n", i235+1, hasil235)

}

}
```

Screenshoot Output



```
PROBLEMS 377 OUTPUT DEBUG CONSOLE TERMINAL PORTS SEARCH ERROR
PS D:\Laprak Algoritma dan Pemrograman 2\Modul 7 struck dan array>
go run "d:\Laprak Algoritma dan Pemrograman 2\Modul 7 struck dan array\unguided3.go"
Klub A: phoenix
Klub B: dragon

Masukkan skor untuk setiap pertandingan. Masukkan skor negatif untuk menghentikan.
Pertandingan 1 - Skor phoenix: 1
Pertandingan 1 - Skor dragon: 0
Pertandingan 2 - Skor phoenix: 0
Pertandingan 2 - Skor dragon: 1
Pertandingan 3 - Skor phoenix: 1
Pertandingan 3 - Skor dragon: 1
Pertandingan 4 - Skor phoenix: -1
Pertandingan 4 - Skor dragon: -1
Pertandingan selesai.

Hasil pertandingan:
Hasil 1: phoenix
Hasil 2: dragon
Hasil 3: Draw
PS D:\Laprak Algoritma dan Pemrograman 2\Modul 7 struck dan array>
```

Deskripsi Program

Sintaks tersebut adalah program untuk mencatat hasil pertandingan antara dua klub. Untuk mencari pemenang kita menggunakan kengondisian if, if pertama jika skor A > skor B maka klub A akan ditambah kedalam slice pemenang, if kedua jika skor B > skor A maka klub B akan ditambah kedalam slice pemenang, dan pada elsenya yaitu jika hasil kedua skor antara klub sama maka draw, iterasi akan dihentikan jika salah satu skor bernilai negatif.

IV. UNGUDED 4

Sourcecode

```
package main

import (
    "fmt"
)

// Fungsi rekursif untuk mencari faktor dari N
func cariFaktor(n, i int) {
    if i > n {
        return
    }
    if n%i == 0 {
        fmt.Print(i, " ")
    }
    cariFaktor(n, i+1)
}

func main() {
    var N int
    fmt.Print("Masukkan bilangan: ")
    fmt.Scan(&N)
    fmt.Print("Faktor dari ", N, " adalah: ")
    cariFaktor(N, 1)
    fmt.Println()
}
```

Screenshoot Output

```
PROBLEMS 431 OUTPUT DEBUG CONSOLE TERMINAL PORTS SEARCH ERROR
PS D:\Laprak Algoritma dan Pemrograman 2\Modul 7 struck dan array>
go run "d:\Laprak Algoritma dan Pemrograman 2\Modul 7 struck dan array\temp.go"
Masukkan teks, akhiri dengan karakter titik ('.'):
bogor.
Teks: bogor
Reverse teks: rogob
Palindrom? false
PS D:\Laprak Algoritma dan Pemrograman 2\Modul 7 struck dan array>
```

```
PROBLEMS 431 OUTPUT DEBUG CONSOLE TERMINAL PORTS SEARCH ERROR
PS D:\Laprak Algoritma dan Pemrograman 2\Modul 7 struck dan array>
go run "d:\Laprak Algoritma dan Pemrograman 2\Modul 7 struck dan array\temp.go"
Masukkan teks, akhiri dengan karakter titik ('.'):
tamat.
Teks: tamat
Reverse teks: tamat
Palindrom? true
PS D:\Laprak Algoritma dan Pemrograman 2\Modul 7 struck dan array>
```

Deskripsi Program

Sintaks tersebut adalah program untuk melakukan pengecekan apakah kata yang di-inputkan user true atau false. Untuk membalikan kata kita menggunakan fungsi `balikanArray`, cara kerjanya yaitu elemen pertama ditukar dengan elemen terakhir, elemen kedua ditukar dengan elemen kedua terakhir, dan seterusnya, Proses berhenti setelah setengah panjang array $n/2$, karena seluruh elemen sudah tertukar. Untuk mengecek apakah kata tersebut palindrom atau tidak dengan fungsi `palindrom`, cara kerjanya yaitu ,membandingkan elemen pertama dengan elemen terakhir, elemen kedua dengan elemen kedua terakhir, dan seterusnya, jika ada elemen yang tidak sama, fungsi langsung mengembalikan false, Jika semua elemen sama hingga setengah panjang array, maka teks adalah palindrom dan fungsi mengembalikan true.