

**LAPORAN PRAKTIKUM  
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL 7  
STRUCK & ARRAY**



**Disusun Oleh :**

**ANDIKA NEVIANTORO / 2311102167**

**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. STRUCK

Structure atau struct adalah kumpulan dari beberapa variabel dengan beragam tipe data yang dibungkus dalam satu variabel.

Dalam bahasa pemrograman lain, struct ini bisa disamakan seperti:

- Records kalau di bahasa Pascal;
- Dictionary kalau di bahasa Python;
- Asosiatif Array kalau di bahasa PHP;
- Object kalau di bahasa Javascript.

perhatikan contoh kasus berikut:

Misalnya kita ingin menyimpan data mahasiswa. Kita bisa saja melakukannya seperti ini:

```
char name[] = "Dian";  
char address[] = "Mataram";  
int age = 22;
```

Lalu bagaimana kalau ada lebih dari satu mahasiswa?

Mungkin bisa saja kita buat seperti ini:

```
char name[] = "Dian";  
char address[] = "Mataram";  
int age = 22;  
  
char name2[] = "Bambang";  
char address2[] = "Surabaya";  
int age2 = 23;  
  
char name3[] = "Bimo";  
char address3[] = "Jakarta";  
int age3 = 23;
```

### Latihan Menggunakan Struct

Buatlah program baru dengan nama **contoh\_struct.c** kemudian isi dengan kode berikut.

```
#include <stdio.h>  
  
// membuat struct  
struct Mahasiswa {  
    char *name;  
    char *address;  
    int age;  
};
```

```

int main(){

    // menggunakan struct
    struct Mahasiswa mhs1, mhs2;

    // mengisi nilai ke struct
    mhs1.name = "Dian";
    mhs1.address = "Mataram";
    mhs1.age = 22;

    // mengisi nilai ke struct
    mhs2 = {
        .name = "Bambang",
        .address = "Surabaya",
        .age = 23
    };

    // mencetak isi struct
    printf("## Mahasiswa 1 ##\n");
    printf("Nama: %s\n", mhs1.name);
    printf("Alamat: %s\n", mhs1.address);
    printf("Umur: %d\n", mhs1.age);

    printf("## Mahasiswa 2 ##\n");
    printf("Nama: %s\n", mhs2.name);
    printf("Alamat: %s\n", mhs2.address);
    printf("Umur: %d\n", mhs2.age);

    return 0;
}

```

Hasilnya :

```

c : cb_console_runn — Konsole
File Edit View Bookmarks Settings Help
## Mahasiswa 1 ##
Nama: Dian
Alamat: Mataram
Umur: 22
## Mahasiswa 2 ##
Nama: Bambang
Alamat: Surabaya
Umur: 23

Process returned 9 (0x9)   execution time : 0.002 s
Press ENTER to continue.
c : cb_console_runn

```

## B. ARRAY

Tipe data Array termasuk salah satu istilah yang masih berkaitan dengan bahasa pemrograman dan bagi Anda yang penasaran dengan fungsi Array, bisa simak penjelasannya di bawah ini. Array atau disebut juga dengan larik ditemukan di data Pascal, C, Java dan juga PHP. Tipe data larik ditemukan di dalam dunia pemrograman, seperti di tahap awal pembuatan program.

Data Array adalah tipe data yang memiliki tipe data yang lain. Data Array ini proses penyimpanan data ke variabel jadi lebih mudah dan terutama adalah apabila data tersebut jumlahnya banyak. Array juga termasuk salah satu tipe data yang cukup terstruktur, yaitu terdiri dari beberapa komponen yang memiliki tipe data yang sama.

Contohnya adalah pada satu Array terdiri dari bilangan bulat atau bilangan saja. Tujuan data Array adalah supaya sejumlah item data dengan jenis yang sama ada di satu tempat serta memudahkan saat dilakukan operasi data. Pada struktur data, Array termasuk struktur data linier yang merupakan struktur elemen data yang disusun pada satu dimensi.

### **Jenis – jenis tipe data Array**

#### 1. Array Satu Dimensi

Array satu dimensi merupakan salah satu tipe data yang paling banyak digunakan. Contohnya adalah digunakan untuk membuat daftar sejumlah orang sesuai dengan usianya, maka syarat yang harus dimasukkan dalam menentukan elemennya yaitu usia. Anda bisa memakai rumus kueri untuk menentukannya.

#### 2. Array Dua Dimensi

Array dua dimensi merupakan tipe data yang memakai dua elemen jadi Anda bisa memasukkan dua elemen sekaligus. Umumnya, jenis data dua dimensi ini memiliki fungsi untuk menampilkan sebuah data yang memiliki dua jenis elemen secara berbeda.

Contohnya adalah Anda ingin menampilkan data jumlah siswa dan jumlah bangku di sebuah kelas. Maka, data yang ditampilkan adalah elemen jumlah siswa dan jumlah bangku siswa di sebuah kelas.

#### 3. Array Multi Dimensi

Jenis Array yang terakhir adalah Array multi dimensi atau tipe dengan dimensi yang banyak. Biasanya lebih dari dua dan contohnya adalah Array tiga dimensi.

Begitulah beberapa jenis tipe data Array yang kerap dijumpai dan digunakan dan berikutnya adalah penjelasan mengenai fungsi Array.

### **Keleniham Menggunakan Array**

#### **1. Bisa Mengakses Secara Acak ke Elemen**

Salah satu kelebihan memakai data Array adalah bisa mengakses secara acak ke elemen. Hal seperti ini akan membuat akses elemen sesuai dengan posisi jauh lebih cepat.

#### **2. Mempunyai Cache Locality Lebih Baik**

Penggunaan data Array biasanya mempunyai cache locality yang jauh lebih baik. Hal ini akan membuat kinerja operasi Array lebih cepat dan ngebut. Cache locality merupakan kemampuan prosesor dalam mengakses kumpulan lokasi memori secara berulang dan sama di waktu yang singkat.

#### **3. Mewakili Sejumlah Item**

Kelebihan menggunakan Array berikutnya adalah dapat mewakili sejumlah item data yang memiliki idem yang sama memakai satu nama. Hal ini dapat membantu saat memaksimalkan kode dan dengan menulis kode seperti ini, maka Anda bisa menyimpan banyak nilai pada sebuah Array.

#### **4. Tidak Terjadi Penambahan Alokasi Memori**

Penggunaan Array juga tidak akan menyebabkan penambahan alokasi memori. Hal ini tentu bisa mencegah terjadinya masalah kekurangan memori atau overflow yang kerap terjadi.

#### **5. Waktu Mengakses Lebih Konstan**

Kelebihan yang terakhir adalah waktu untuk mengakses elemen Array jauh lebih konstan, yaitu mempunyai time complexity  $O(1)$ .

### **Kelemahan Menggunakan Tipe Data Array**

#### **1. Tidak Bisa Menambah atau Menghapus Elemen**

Apabila Anda sudah membuatnya tidak bisa menambah atau menghapus elemen lagi. Tipe data Array juga mempunyai ukuran yang tetap dan itulah sebabnya tidak bisa melakukan penambahan

atau penghapusan elemen. Anda juga tidak bisa mengubah ukurannya dengan dinamis.

## 2. Fungsinya Terbatas

Tipe data Array memiliki fungsionalitas yang cukup terbatas dibandingkan data yang lainnya. Array memiliki keunggulan pada tugas-tugas sederhana dan untuk permasalahan yang kompleks tipe data lainnya bisa jadi solusinya.

## 3. Biaya Hapus Item Mahal

Kelemahan tipe data Array berikutnya lagi adalah apabila menghapus item pada Array bisa melibatkan penyalinan di setiap elemen sebelum isi celah yang ditinggalkan pada elemen yang Anda hapus. Nah, biaya untuk menghapus item pada Array ini cukup mahal..

## 4. Penambahan Item Lebih Ribet

Pada saat Anda menambahkan item ke Array, proses penyisipannya harus setiap elemen dipindah dari lokasi yang asli ke slot selanjutnya yang masih tersedia. Biasanya cost yang dibutuhkan untuk pergeseran bisa meningkat secara linier berdasarkan dengan panjang Array.

# II. GUIDED

1. Program di atas digunakan untuk menghitung durasi waktu parkir antara dua waktu yang diberikan.

### 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
        fmt.Printf("Lama Parkir: %d jam %d menit %d detik",
durasi.jam, durasi.menit, durasi.detik)
    }

```

### Screenshoot Output :

```

● PS D:\MATKUL SEMESTER 3\PRAKTIKUM ALPRO 2\MODUL 7> go run
DED\guided1.go
5 15 45
10 25 30
Lama Parkir: 5 jam 9 menit 45 detik
○ PS D:\MATKUL SEMESTER 3\PRAKTIKUM ALPRO 2\MODUL 7> █

```

### Deskripsi Program :

Program di atas digunakan untuk menghitung durasi waktu parkir antara dua waktu yang diberikan dalam format jam, menit, dan detik. Program ini pertama-tama membaca dua waktu, yaitu waktu parkir dan waktu pulang, kemudian mengonversi kedua waktu tersebut menjadi detik. Setelah itu, program menghitung selisih antara waktu pulang dan waktu parkir dalam detik, dan hasilnya diubah kembali menjadi jam, menit, dan detik. Durasi parkir ini kemudian ditampilkan dalam format "jam menit detik". Program ini berguna untuk menentukan lama waktu parkir berdasarkan dua waktu input yang diberikan.

2. Program untuk mengelola dan memperbarui daftar teman dalam sebuah slice.

### Sourcecode

```

//Guided 2 - Slice
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"}

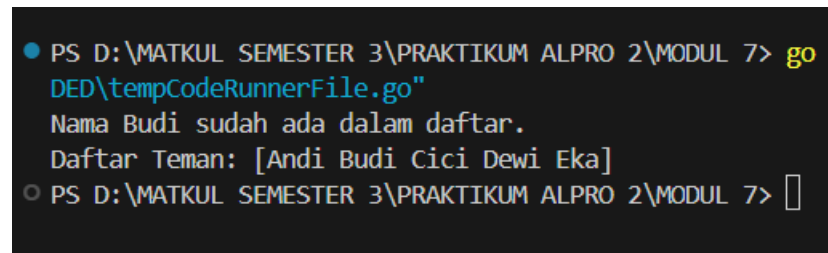
    // Nama-nama baru yang ingin ditambahkan
    namaBaru := []string{"Dewi", "Budi", "Eka"}

    // 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 :



```

● PS D:\MATKUL SEMESTER 3\PRAKTIKUM ALPRO 2\MODUL 7> go r
DED\tempCodeRunnerFile.go
Nama Budi sudah ada dalam daftar.
Daftar Teman: [Andi Budi Cici Dewi Eka]
○ PS D:\MATKUL SEMESTER 3\PRAKTIKUM ALPRO 2\MODUL 7> 

```

#### Deskripsi Program :

Program digunakan untuk memperbarui daftar teman dengan memastikan tidak ada duplikasi nama. Program dimulai dengan mendeklarasikan slice `daftarTeman` yang berisi beberapa nama teman awal, dan slice `namaBaru` yang berisi nama-nama yang ingin ditambahkan. Fungsi `sudahAda` digunakan untuk memeriksa apakah sebuah nama sudah ada di dalam `daftarTeman`. Jika nama baru belum ada, nama tersebut ditambahkan ke dalam `daftarTeman` menggunakan `append`. Jika nama sudah ada, program akan mencetak pesan bahwa nama tersebut sudah ada dalam daftar. Di akhir program, daftar teman yang telah diperbarui akan ditampilkan.

3. Program untuk menyimpan dan menampilkan harga buah menggunakan struktur data map



### Sourcecode

```
//Guided 3 - Map
package main

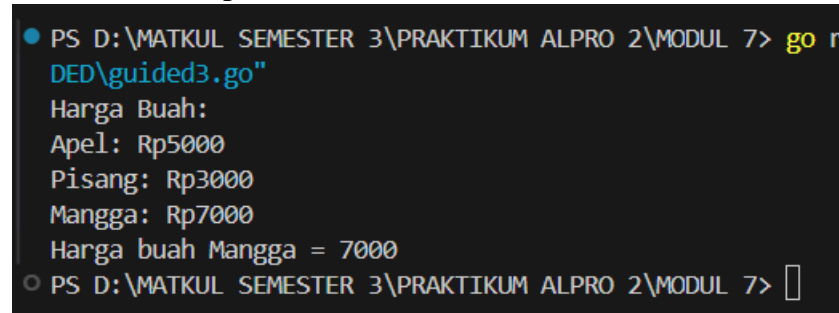
import (
    "fmt"
)

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"])
}
```

### Screenshot Output:



```
PS D:\MATKUL SEMESTER 3\PRAKTIKUM ALPRO 2\MODUL 2> go run DED\guided3.go
Harga Buah:
Apel: Rp5000
Pisang: Rp3000
Mangga: Rp7000
Harga buah Mangga = 7000
PS D:\MATKUL SEMESTER 3\PRAKTIKUM ALPRO 2\MODUL 2> █
```

### Deskripsi Program:

Program di atas digunakan untuk menyimpan dan menampilkan harga buah menggunakan map. Map `hargaBuah` berisi pasangan kunci-nilai, di mana kunci adalah nama buah (seperti "Apel", "Pisang", "Mangga") dan nilai adalah harga masing-masing buah. Program kemudian menampilkan harga setiap buah dengan menggunakan loop `for range` untuk iterasi elemen-elemen dalam map. Selain itu, program juga menampilkan harga buah "Mangga" secara spesifik dengan mengakses nilai map menggunakan kunci "Mangga". Output program menunjukkan daftar harga buah dan harga buah Mangga secara terpisah.

## 2. 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) dimana rumus jarak adalah:  $\text{Jarak} = \sqrt{(a-c)^2 + (b-d)^2}$  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}

Catatan: Lihat paket math dalam lampiran untuk menggunakan fungsi math.Sqrt() untuk menghitung akar kuadrat.

### Sourcecode

```
package main

import (
    "fmt"
    "math"
)

// Tipe bentukan untuk titik
type Titik struct {
    x, y int
}

// Tipe bentukan untuk lingkaran
type Lingkaran struct {
    titik Titik
    radius int
}

// Fungsi untuk menghitung jarak antara dua titik
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))
}

// Fungsi untuk menentukan apakah sebuah titik berada di
// dalam lingkaran
func didalam(c Lingkaran, p Titik) bool {
    return jarak(c.titik, p) <= float64(c.radius)
}

func main() {
    // Input data lingkaran dan titik
    var cx1, cy1, r1 int
    var cx2, cy2, r2 int
    var x, y int

    fmt.Println("Masukkan data lingkaran 1 (cx1 cy1 r1):")
    fmt.Scan(&cx1, &cy1, &r1)

    fmt.Println("Masukkan data lingkaran 2 (cx2 cy2 r2):")
    fmt.Scan(&cx2, &cy2, &r2)

    fmt.Println("Masukkan data titik sembarang (x y):")
    fmt.Scan(&x, &y)
```

```

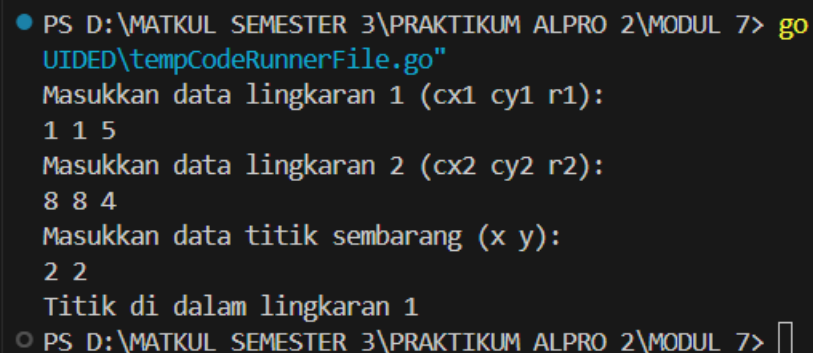
// Membuat objek lingkaran dan titik
lingkaran1 := Lingkaran{Titik{cx1, cy1}, r1}
lingkaran2 := Lingkaran{Titik{cx2, cy2}, r2}
titik := Titik{x, y}

// Menentukan posisi titik
diLingkaran1 := didalam(lingkaran1, titik)
diLingkaran2 := didalam(lingkaran2, titik)

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 D:\MATKUL SEMESTER 3\PRAKTIKUM ALPRO 2\MODUL 7> go
UIDED\tempCodeRunnerFile.go"
Masukkan data lingkaran 1 (cx1 cy1 r1):
1 1 5
Masukkan data lingkaran 2 (cx2 cy2 r2):
8 8 4
Masukkan data titik sembarang (x y):
2 2
Titik di dalam lingkaran 1
PS D:\MATKUL SEMESTER 3\PRAKTIKUM ALPRO 2\MODUL 7> 

```

### Deskripsi Program :

Program di atas digunakan untuk menentukan apakah sebuah titik berada di dalam dua lingkaran yang diberikan. Program ini mendefinisikan dua tipe bentukan, yaitu 'Titik' (untuk menyimpan koordinat x dan y) dan 'Lingkaran' (untuk menyimpan pusat lingkaran dan radius). Fungsi 'jarak' menghitung jarak antara dua titik menggunakan rumus Pythagoras, dan fungsi 'didalam' mengecek apakah titik berada di dalam lingkaran dengan membandingkan jaraknya dengan radius lingkaran. Di dalam fungsi 'main', program menerima input data untuk dua lingkaran dan satu titik, lalu memeriksa apakah titik tersebut berada di dalam salah satu atau kedua lingkaran. Hasilnya akan ditampilkan sesuai dengan posisi titik terhadap lingkaran-lingkaran tersebut.

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 indek ke-0 adalah genap).
  - Menampilkan elemen-elemen array dengan indeks kelipatan bilangan x. x bisa diperoleh dari masukan pengguna.
- Informatics lab
- 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"
)

// Fungsi untuk menampilkan isi array
func tampilkanArray(array []int) {
    fmt.Println("Isi array:", array)
}

// Fungsi untuk menampilkan elemen dengan indeks ganjil
func tampilkanIndeksGanjil(array []int) {
    fmt.Print("Elemen dengan indeks ganjil: ")
    for i := 1; i < len(array); i += 2 {
        fmt.Print(array[i], " ")
    }
    fmt.Println()
}

// Fungsi untuk menampilkan elemen dengan indeks genap
func tampilkanIndeksGenap(array []int) {
    fmt.Print("Elemen dengan indeks genap: ")
    for i := 0; i < len(array); i += 2 {
        fmt.Print(array[i], " ")
    }
    fmt.Println()
}
```

```

}

// Fungsi untuk menampilkan elemen dengan indeks
kelipatan x
func tampilkanKelipatan(array []int, x int) {
    fmt.Printf("Elemen dengan indeks kelipatan %d: ", x)
    for i := 0; i < len(array); i++ {
        if i%x == 0 {
            fmt.Print(array[i], " ")
        }
    }
    fmt.Println()
}

// Fungsi untuk menghapus elemen pada indeks tertentu
func hapusIndeks(array []int, idx int) []int {
    fmt.Printf("Menghapus elemen pada indeks %d\n", idx)
    return append(array[:idx], array[idx+1:]...)
}

// Fungsi untuk menghitung rata-rata elemen array
func rataRata(array []int) float64 {
    total := 0
    for _, val := range array {
        total += val
    }
    return float64(total) / float64(len(array))
}

// Fungsi untuk menghitung standar deviasi
func standarDeviasi(array []int) float64 {
    rata := rataRata(array)
    var total float64
    for _, val := range array {
        total += math.Pow(float64(val)-rata, 2)
    }
    return math.Sqrt(total / float64(len(array)))
}

// Fungsi untuk menghitung frekuensi bilangan tertentu
func frekuensi(array []int, num int) int {
    freq := 0
    for _, val := range array {
        if val == num {
            freq++
        }
    }
    return freq
}

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

    array := make([]int, n)

```

```

    fmt.Println("Masukkan elemen-elemen array:")
    for i := 0; i < n; i++ {
        fmt.Scan(&array[i])
    }

    for {
        fmt.Println("\nMenu:")
        fmt.Println("a. Tampilkan keseluruhan isi array")
        fmt.Println("b. Tampilkan elemen dengan indeks ganjil")
        fmt.Println("c. Tampilkan elemen dengan indeks genap")
        fmt.Println("d. Tampilkan elemen dengan indeks kelipatan x")
        fmt.Println("e. Hapus elemen pada indeks tertentu")
        fmt.Println("f. Tampilkan rata-rata elemen array")
        fmt.Println("g. Tampilkan standar deviasi elemen array")
        fmt.Println("h. Tampilkan frekuensi bilangan tertentu")
        fmt.Println("i. Keluar")
        fmt.Print("Pilih menu: ")

        var pilihan string
        fmt.Scan(&pilihan)

        switch pilihan {
        case "a":
            tampilkanArray(array)
        case "b":
            tampilkanIndeksGanjil(array)
        case "c":
            tampilkanIndeksGenap(array)
        case "d":
            var x int
            fmt.Print("Masukkan nilai x: ")
            fmt.Scan(&x)
            tampilkanKelipatan(array, x)
        case "e":
            var idx int
            fmt.Print("Masukkan indeks yang ingin dihapus: ")
            fmt.Scan(&idx)
            array = hapusIndeks(array, idx)
            tampilkanArray(array)
        case "f":
            fmt.Printf("Rata-rata elemen array: %.2f\n", rataRata(array))
        case "g":
            fmt.Printf("Standar deviasi elemen array: %.2f\n", standarDeviasi(array))
        case "h":
            var num int

```

```
        fmt.Print("Masukkan bilangan yang ingin  
dicari frekuensinya: ")  
        fmt.Scan(&num)  
        fmt.Printf("Frekuensi bilangan %d: %d\n",  
num, frekuensi(array, num))  
        case "i":  
            fmt.Println("Keluar dari program.")  
            return  
        default:  
            fmt.Println("Pilihan tidak valid.")  
        }  
    }  
}
```

**Screenshot Output :**



```
● PS D:\MATKUL SEMESTER 3\PRAKTIKUM ALPRO 2\MODUL 7> g  
UIDED\unguided2.go"
```

```
Masukkan jumlah elemen array: 5
```

```
Masukkan elemen-elemen array:
```

```
10 20 30 40 50
```

```
Menu:
```

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

```
Pilih menu: a
```

```
Isi array: [10 20 30 40 50]
```

```
Menu:
```

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

```
Pilih menu: b
```

```
Elemen dengan indeks ganjil: 20 40
```

```
Menu:
```

- a. Tampilkan keseluruhan isi array
- b. Tampilkan elemen dengan indeks ganjil
- c. Tampilkan elemen dengan indeks genap
- d. Tampilkan elemen dengan indeks kelipatan x
- e. Hapus elemen pada indeks tertentu
- f. Tampilkan rata-rata elemen array

h. Tampilkan frekuensi bilangan tertentu  
i. Keluar

Pilih menu: c

Elemen dengan indeks genap: 10 30 50

Menu:

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

Pilih menu: d

Masukkan nilai x: 2

Elemen dengan indeks kelipatan 2: 10 30 50

Menu:

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

Pilih menu: e

Masukkan indeks yang ingin dihapus: 2

Menghapus elemen pada indeks 2

Isi array: [10 20 40 50]

Menu:

a. Tampilkan keseluruhan isi array  
b. Tampilkan elemen dengan indeks ganjil  
c. Tampilkan elemen dengan indeks genap  
d. Tampilkan elemen dengan indeks kelipatan x

- b. Tampilkan elemen dengan indeks ganjil
- c. Tampilkan elemen dengan indeks genap
- d. Tampilkan elemen dengan indeks kelipatan x
- e. Hapus elemen pada indeks tertentu
- f. Tampilkan rata-rata elemen array
- g. Tampilkan standar deviasi elemen array
- h. Tampilkan frekuensi bilangan tertentu
- i. Keluar

Pilih menu: f

Rata-rata elemen array: 30.00

Menu:

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

Pilih menu: g

Standar deviasi elemen array: 15.81

Menu:

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

Pilih menu: h

Masukkan bilangan yang ingin dicari frekuensinya: 20

Frekuensi bilangan 20: 1

```

Menu:
a. Tampilkan keseluruhan isi array
b. Tampilkan elemen dengan indeks ganjil
c. Tampilkan elemen dengan indeks genap
d. Tampilkan elemen dengan indeks kelipatan x
e. Hapus elemen pada indeks tertentu
f. Tampilkan rata-rata elemen array
g. Tampilkan standar deviasi elemen array
h. Tampilkan frekuensi bilangan tertentu
i. Keluar
Pilih menu: i
Keluar dari program.
PS D:\MATKUL SEMESTER 3\PRAKTIKUM ALPRO 2\MODUL 7>

```

### Deskripsi Program :

Program di atas berfungsi untuk mengelola dan memanipulasi array angka dengan berbagai opsi. Pengguna dapat memasukkan elemen array, kemudian memilih berbagai menu yang tersedia untuk menampilkan isi array, elemen dengan indeks ganjil/genap, elemen dengan indeks kelipatan tertentu, atau menghapus elemen pada indeks tertentu. Selain itu, program ini juga menghitung dan menampilkan rata-rata, standar deviasi, serta frekuensi kemunculan bilangan tertentu dalam array. Setiap opsi menu dijalankan berdasarkan input pengguna, dan program akan terus berjalan hingga pengguna memilih untuk keluar. Program ini memungkinkan manipulasi array secara interaktif dan memberikan informasi statistik dasar terkait array tersebut.

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. 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) ada digambar

### Sourcecode :

```
package main

import (
    "bufio"
    "fmt"
    "os"
    "strings"
)

func main() {
    // Input nama klub
    reader := bufio.NewReader(os.Stdin)
    fmt.Print("Masukkan nama Klub A: ")
    klubA, _ := reader.ReadString('\n')
    klubA = strings.TrimSpace(klubA)

    fmt.Print("Masukkan nama Klub B: ")
    klubB, _ := reader.ReadString('\n')
    klubB = strings.TrimSpace(klubB)

    // Array untuk menyimpan nama klub pemenang
    var pemenang []string

    // Loop untuk memasukkan skor pertandingan
    for i := 1; ; i++ {
        fmt.Printf("Masukkan skor pertandingan %d (%s vs %s): ", i, klubA, klubB)
        var skorA, skorB int
        _, err := fmt.Scan(&skorA, &skorB)

        // Validasi input skor
        if err != nil || skorA < 0 || skorB < 0 {
            fmt.Println("Skor tidak valid! Proses dihentikan.")
            break
        }

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

```

    }

    // Tampilkan klub yang memenangkan pertandingan
    fmt.Println("Pertandingan selesai.")
    if len(pemenang) > 0 {
        fmt.Println("Daftar klub yang memenangkan pertandingan:")
        for _, klub := range pemenang {
            fmt.Println(klub)
        }
    } else {
        fmt.Println("Tidak ada pemenang yang direkap.")
    }
}

```

### Screenshoot Output :

```

● PS D:\MATKUL SEMESTER 3\PRAKTIKUM ALPRO 2\MODUL 7> go run "d:\MATKUL SEMESTER
  UIDED\tempCodeRunnerFile.go"
Masukkan nama Klub A: Manchester City
Masukkan nama Klub B: Manchester United
Masukkan skor pertandingan 1 (Manchester City vs Manchester United): 5 0
Hasil 1: Manchester City
Masukkan skor pertandingan 2 (Manchester City vs Manchester United): 3 1
Hasil 2: Manchester City
Masukkan skor pertandingan 3 (Manchester City vs Manchester United): 2 2
Hasil 3: Draw
Masukkan skor pertandingan 4 (Manchester City vs Manchester United): 2 3
Hasil 4: Manchester United
Masukkan skor pertandingan 5 (Manchester City vs Manchester United): 0 - 1
Skor tidak valid! Proses dihentikan.
Pertandingan selesai.
Daftar klub yang memenangkan pertandingan:
Manchester City
Manchester City
Manchester United
○ PS D:\MATKUL SEMESTER 3\PRAKTIKUM ALPRO 2\MODUL 7> 

```

### Deskripsi Program :

Program di atas digunakan untuk mencatat dan merekap hasil pertandingan antara dua klub sepak bola. Pengguna diminta untuk memasukkan nama dua klub, dan kemudian memasukkan skor pertandingan mereka dalam setiap putaran. Setiap skor yang dimasukkan dibandingkan untuk menentukan pemenangnya, dan hasilnya disimpan dalam sebuah array `pemenang`. Program akan terus meminta input skor hingga skor yang dimasukkan tidak valid (misalnya, skor negatif atau input yang salah). Setelah pertandingan selesai, program menampilkan daftar klub yang memenangkan pertandingan. Jika tidak ada pemenang, program akan memberi tahu bahwa tidak ada pemenang yang direkap.

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, nint)
1.8. Data tersedia dalam piranti masukan
F.S. Array t berisi sejumlah n karakter yang dimasukkan user. Proses
input selama karakter bukanlah TITIK dan ni NMAX/
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 bergarisbawah adalah Input/read)

Teks : S E N A N G . //bergaris bawah

Reverse teks : G N A N E S

Teks : K A T A K. //bergaris bawah

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 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 //bergaris bawah

Palindrom ? true

Teks : S E N A N G //bergaris bawah

Palindrom ? false

### Sourcecode

```
package main

import (
    "bufio"
    "fmt"
    "os"
)

const NMAX int = 127

type tabel [NMAX]rune

// Prosedur untuk mengisi array
func isiArray(t *tabel, n *int) {
    reader := bufio.NewReader(os.Stdin)
    fmt.Println("Masukkan karakter (akhiri dengan TITIK '.'):")

    *n = 0
    for {
        r, _, err := reader.ReadRune() // Membaca satu rune
        if err != nil {
            fmt.Println("Error membaca input:", err)
            return
        }

        if r == '.' || *n >= NMAX { // Berhenti saat TITIK atau mencapai NMAX
            break
        }

        t[*n] = r // Menyimpan karakter ke array
        *n++
    }
}

// Prosedur untuk membalikkan isi array
func balikanArray(t *tabel, n int) {
    for i := 0; i < n/2; i++ {
        t[i], t[n-i-1] = t[n-i-1], t[i]
    }
}
```



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

func main() {
    var tab tabel
    var m int

    // Memanggil prosedur isiArray
    isiArray(&tab, &m)

    // Menampilkan array asli
    fmt.Print("Teks: ")
    for i := 0; i < m; i++ {
        fmt.Printf("%c ", tab[i])
    }
    fmt.Println()

    // Mengecek apakah array membentuk palindrom
    isPalindrom := palindrom(tab, m)
    fmt.Printf("Palindrom? %v\n", isPalindrom)

    // Membalikkan isi array
    balikanArray(&tab, m)

    // Menampilkan array setelah dibalik
    fmt.Print("Reverse teks: ")
    for i := 0; i < m; i++ {
        fmt.Printf("%c ", tab[i])
    }
    fmt.Println()
}
```

### Screenshot output:

```
● PS D:\MATKUL SEMESTER 3\PRAKTIKUM ALPRO 2\MODUL 7> go run
  UIDED\tempCodeRunnerFile.go
Masukkan karakter (akhiri dengan TITIK '.'):
S E N A N G .
Teks: S E N A N G
Palindrom? false
Reverse teks: G N A N E S
○ PS D:\MATKUL SEMESTER 3\PRAKTIKUM ALPRO 2\MODUL 7> █
```

**Deskripsi Program:**

Program di atas meminta pengguna untuk memasukkan karakter satu per satu hingga mencapai tanda titik (.) atau jumlah karakter mencapai batas maksimum yang ditentukan (NMAX). Karakter-karakter tersebut disimpan dalam array `tabel`. Setelah itu, program menampilkan teks yang dimasukkan, memeriksa apakah teks tersebut merupakan palindrom (teks yang sama jika dibaca dari depan atau belakang), dan menampilkan hasil pemeriksaannya. Program juga membalikkan urutan karakter dalam array dan menampilkan teks yang telah dibalik. Dengan demikian, program ini berfungsi untuk memproses teks dengan cara memeriksa palindrom dan membalikkan urutan karakter.

**KESIMPULAN**

Kesimpulan mengenai praktikum struct dan array adalah sebagai berikut:

Struct adalah tipe data yang memungkinkan penggabungan beberapa variabel dengan tipe data yang berbeda dalam satu kesatuan. Struktur data ini berguna ketika kita ingin menyimpan informasi terkait yang berbeda, seperti data mahasiswa yang terdiri dari nama, alamat, dan umur. Di berbagai bahasa pemrograman, struct memiliki padanan seperti record di Pascal, dictionary di Python, atau objek di JavaScript. Dengan menggunakan struct, data dapat dikelompokkan dengan lebih terstruktur dan mudah diakses.

Array adalah tipe data yang menyimpan kumpulan elemen dengan tipe data yang sama dalam satu variabel. Array memiliki beberapa jenis, seperti array satu dimensi, dua dimensi, dan multi dimensi. Keunggulan utama array adalah kemampuan untuk mengakses elemen dengan cepat dan efisien, serta kemudahan dalam menyimpan banyak data dengan tipe yang sama. Namun, array memiliki kelemahan, seperti ukuran yang tetap (tidak bisa diubah secara dinamis), serta kesulitan dalam menambah atau menghapus elemen setelah array dibuat.

Secara umum, struct lebih cocok untuk menyimpan data yang terdiri dari berbagai tipe, sedangkan array lebih efisien untuk menyimpan dan mengelola data dengan tipe yang seragam. Keduanya memiliki keunggulan dan kelemahan yang perlu dipertimbangkan dalam penggunaannya sesuai dengan kebutuhan aplikasi yang dikembangkan.

## **REFERENSI**

[1] M. ahmad, Belajar Pemrograman C#13: Mengenal Tipe Data Struct. Petanikode.

<https://www.petanikode.com/c-struct/>

[2] Coding Studio Team, Fungsi Array: Pengertian dan Jenisnya. 2023.

<https://codingstudio.id/blog/fungsi-array-pengertian-jenis/>