

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL VII
Struct & Array**



Disusun Oleh :

Nadhif Atha Zaki / 2311102007

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 tipe data yang memungkinkan pengelompokan elemen-elemen data yang berbeda dalam satu unit. Setiap elemen dalam struct disebut field atau atribut, yang bisa memiliki tipe data yang berbeda-beda. Struct sering digunakan untuk menggambarkan objek dunia nyata dalam pemrograman. Misalnya, kita bisa membuat struct untuk menggambarkan informasi tentang buku yang berisi judul, pengarang, dan tahun terbit, yang masing-masing memiliki tipe data yang berbeda (misalnya string untuk judul dan pengarang, integer untuk tahun terbit). Struct memungkinkan kita untuk mengorganisir dan menyimpan data yang saling berkaitan dalam satu tempat yang terstruktur.
- Array adalah struktur data yang digunakan untuk menyimpan sejumlah elemen dengan tipe data yang sama dalam satu variabel. Elemen-elemen dalam array disusun secara berurutan dan diakses menggunakan indeks. Array memiliki ukuran yang tetap, artinya kita perlu menentukan jumlah elemen yang akan disimpan di dalamnya saat array dideklarasikan. Dengan menggunakan array, kita bisa mengelola data dalam jumlah besar dengan cara yang efisien, karena array menyediakan akses langsung ke elemen berdasarkan indeks. Misalnya, jika kita ingin menyimpan sepuluh nilai integer, kita bisa menggunakan array dengan sepuluh elemen integer. Array berguna dalam kasus di mana kita membutuhkan struktur data yang sederhana dan statis dengan elemen-elemen yang homogen (semua elemen memiliki tipe yang sama).

Kedua struktur data ini sangat berguna untuk mengorganisir dan memanipulasi data dalam berbagai jenis aplikasi, dengan struct lebih cocok untuk data yang beragam dan array lebih cocok untuk data yang seragam.

II. GUIDED

1.

Sourcecode

```
package main

import "fmt"

// Mendefinisikan tipe data struct bernama 'waktu' yang memiliki atribut jam,
menit, dan detik bertipe int
type waktu struct {
    jam, menit, detik int
}

func main() {
    // Mendeklarasikan variabel wParkir, wPulang, dan durasi dengan tipe data
waktu
    var wParkir, wPulang, durasi waktu

    // Mendeklarasikan variabel dParkir, dPulang, dan lParkir sebagai variabel
integer
    // dParkir dan dPulang akan menyimpan waktu dalam detik, sedangkan lParkir
akan menyimpan lama parkir dalam detik
    var dParkir, dPulang, lParkir int

    // Mengambil input dari pengguna untuk waktu parkir dalam bentuk jam, menit,
dan detik
    fmt.Scan(&wParkir.jam, &wParkir.menit, &wParkir.detik)

    // Mengambil input dari pengguna untuk waktu pulang dalam bentuk jam,
menit, dan detik
    fmt.Scan(&wPulang.jam, &wPulang.menit, &wPulang.detik)

    // Mengonversi waktu parkir (wParkir) menjadi detik, dan menyimpannya ke
variabel dParkir
    dParkir = wParkir.detik + wParkir.menit*60 + wParkir.jam*3600

    // Mengonversi waktu pulang (wPulang) menjadi detik, dan menyimpannya ke
variabel dPulang
    dPulang = wPulang.detik + wPulang.menit*60 + wPulang.jam*3600

    // Menghitung lama parkir dalam detik dengan mengurangi dPulang dengan
dParkir
    lParkir = dPulang - dParkir
}
```

```
// Mengonversi total durasi parkir dari detik ke jam, menit, dan detik
durasi.jam = lParkir / 3600 // Menghitung jam dengan membagi total detik
dengan 3600
durasi.menit = lParkir % 3600 / 60 // Menghitung menit dari sisa detik setelah
diambil jam
durasi.detik = lParkir % 3600 % 60 // Menghitung detik dari sisa setelah
diambil jam dan menit

// Menampilkan lama parkir dalam format "jam menit detik"
fmt.Printf("Lama Parkir : %d jam %d menit %d detik", durasi.jam, durasi.menit,
durasi.detik)
}
```

Screenshoot Output

```
PS D:\Nathhh\matkul\smst 3\Praktikum Alpro 2\Modul-VII Struct and Array> go run
and Array\guided1\guided1.go"
2 15 20
3 16 30
Lama Parkir : 1 jam 1 menit 10 detik
PS D:\Nathhh\matkul\smst 3\Praktikum Alpro 2\Modul-VII Struct and Array> █
```

Deskripsi Program

Program ini menghitung durasi parkir berdasarkan waktu parkir dan waktu pulang yang dimasukkan pengguna dalam format jam, menit, dan detik. Waktu parkir dan waktu pulang dikonversi ke detik agar mudah dihitung. Selanjutnya, program menghitung selisih antara waktu pulang dan waktu parkir untuk mendapatkan lama parkir dalam detik. Durasi parkir ini kemudian dikonversi kembali ke dalam format jam, menit, dan detik. Hasilnya ditampilkan dalam format yang mudah dipahami oleh pengguna.

2.

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

```

PS D:\Nathhh\matkul\smst 3\Praktikum Alpro 2\Modul-VII Struct and Array> go run
and Array\guided2\guided2.go
Nama Budi sudah ada dalam daftar.
Daftar Teman: [Andi Budi Cici Dewi Eka]
PS D:\Nathhh\matkul\smst 3\Praktikum Alpro 2\Modul-VII Struct and Array>

```

Deskripsi Program

Program ini mengelola daftar teman dalam bentuk slice. Fungsi sudahAda digunakan untuk memeriksa apakah nama yang ingin ditambahkan sudah ada dalam daftar teman. Di dalam main, program memulai dengan daftar teman yang sudah ada, lalu menambahkan nama baru ke dalam daftar jika nama tersebut belum ada. Jika nama baru sudah ada dalam daftar, program akan menampilkan pesan bahwa nama tersebut sudah ada. Hasil akhirnya adalah daftar teman yang diperbarui.

3.

Sourcecode

```
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

```
PS D:\Nathhh\matkul\smst 3\Praktikum Alpro 2\Modul-VII Struct and Array> go run "d
and Array\guided3\guided3.go"
Harga Buah:
Mangga: Rp7000
Apel: Rp5000
Pisang: Rp3000
Harga buah Mangga = 7000
PS D:\Nathhh\matkul\smst 3\Praktikum Alpro 2\Modul-VII Struct and Array>
```

Deskripsi Program

Program ini membuat sebuah map yang menyimpan data harga buah, di mana nama buah dijadikan sebagai kunci dan harga buah sebagai nilai. Program ini kemudian menampilkan harga setiap buah menggunakan perulangan for range, dan menampilkan harga buah Mangga secara spesifik dengan cara mengakses nilai berdasarkan kunci "Mangga". Hasil akhirnya adalah daftar harga buah dan harga spesifik untuk Mangga.

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".

Contoh

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

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.

ian 65 | Modul Praktikum Algoritma dan Pemrograman 2

```
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"
)

// Struktur data untuk merepresentasikan titik dengan koordinat X dan Y
type Titik struct {
    X int
    Y int
}

// Struktur data untuk merepresentasikan lingkaran dengan pusat (Tengah) dan radius
type Lingkaran struct {
    Tengah Titik
    Radius int
}

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

// Fungsi untuk memeriksa posisi titik relatif terhadap dua lingkaran
func tentukanPosisiTitik(lingkaran1, lingkaran2 Lingkaran, titik Titik) string {
    // Menghitung jarak titik dari pusat lingkaran pertama
    jarakKeLingkaran1 := hitungJarak(lingkaran1.Tengah, titik)

    // Menghitung jarak titik dari pusat lingkaran kedua
    jarakKeLingkaran2 := hitungJarak(lingkaran2.Tengah, titik)

    // Menentukan apakah titik berada di dalam kedua lingkaran, salah satu
    // lingkaran, atau di luar keduanya
    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() {
    // Deklarasi variabel untuk dua lingkaran dan satu titik
    var lingkaran1, lingkaran2 Lingkaran
    var titik Titik

    // Meminta input untuk pusat dan radius lingkaran pertama
    fmt.Println("Masukkan pusat dan radius lingkaran 1:")
    fmt.Scanln(&lingkaran1.Tengah.X, &lingkaran1.Tengah.Y,
    &lingkaran1.Radius)

    // Meminta input untuk pusat dan radius lingkaran kedua
    fmt.Println("Masukkan pusat dan radius lingkaran 2:")
    fmt.Scanln(&lingkaran2.Tengah.X, &lingkaran2.Tengah.Y,
    &lingkaran2.Radius)

    // Meminta input untuk koordinat titik
    fmt.Println("Masukkan koordinat titik sembarang:")
    fmt.Scanln(&titik.X, &titik.Y)

    // Menentukan posisi titik terhadap kedua lingkaran dan menampilkan hasilnya
    hasil := tentukanPosisiTitik(lingkaran1, lingkaran2, titik)
    fmt.Println("Posisi titik:", hasil)
}

```

Screenshoot Output

```

PS D:\Nathhh\matkul\smst 3\Praktikum Alpro 2\Modul-VII Struct and Array> go run "d:\Nathhh\matkul\smst 3\Praktikum Alpro 2\Modul-VII Struct and Array\unguided1\unguided1.go"
Masukkan pusat dan radius lingkaran 1:
2 3 4
Masukkan pusat dan radius lingkaran 2:
4 5 2
Masukkan koordinat titik sembarang:
3 2
Posisi titik: Titik berada di dalam lingkaran 1
PS D:\Nathhh\matkul\smst 3\Praktikum Alpro 2\Modul-VII Struct and Array>

```

Deskripsi Program

Program ini digunakan untuk menentukan posisi sebuah titik relatif terhadap dua lingkaran. Pertama, program meminta input untuk pusat dan radius dari dua lingkaran, serta koordinat titik yang akan diperiksa. Program kemudian menghitung jarak antara titik dan pusat kedua lingkaran menggunakan rumus jarak Euclidean. Berdasarkan perbandingan jarak titik dengan radius kedua lingkaran, program menentukan apakah titik berada di dalam satu atau kedua lingkaran, atau di luar keduanya. Hasil posisi titik akan ditampilkan sesuai dengan kondisi yang terpenuhi.

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:

- 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() {
    // Membuat array dengan kapasitas 10 untuk menampung elemen-elemen
    elemenArray := make([]int, 10)

    // Meminta input dari pengguna untuk jumlah elemen yang akan dimasukkan ke
    // dalam array
    var jumlahElemen int
    fmt.Print("Masukkan jumlah elemen (maksimal 10): ")
    fmt.Scanln(&jumlahElemen)

    // Memastikan jumlah elemen tidak melebihi kapasitas array
    if jumlahElemen > 10 {
        jumlahElemen = 10
        fmt.Println("Jumlah elemen dibatasi maksimal 10. Menggunakan 10
elemen.")
    }

    // Meminta pengguna untuk memasukkan nilai elemen-elemen array
    fmt.Println("Masukkan elemen-elemen array:")
    for i := 0; i < jumlahElemen; i++ {
        var elemen int
        fmt.Scanln(&elemen)
        elemenArray[i] = elemen
    }

    for {
        // Menampilkan menu pilihan operasi yang dapat dilakukan pada array
        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")

        // Meminta input dari pengguna untuk memilih salah satu menu
        var pilihan int
        fmt.Print("\nMasukkan pilihan: ")
        fmt.Scanln(&pilihan)
    }
}

```

```

// Menangani pilihan yang dimasukkan pengguna
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) //
    Perbarui jumlah elemen setelah penghapusan
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")
}
}

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

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

// Fungsi untuk menampilkan elemen array dengan indeks genap

```

```

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

// Fungsi untuk menampilkan elemen array dengan indeks kelipatan bilangan x
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])
        }
    }
}

// Fungsi untuk menghapus elemen array pada indeks tertentu
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
    }
    // Menggeser elemen-elemen setelah elemen yang dihapus untuk menghapusnya
    for i := indeks; i < jumlahElemen-1; i++ {
        array[i] = array[i+1]
    }
    jumlahElemen-- // Kurangi jumlah elemen
    fmt.Println("Elemen pada indeks", indeks, "telah dihapus")
    return jumlahElemen // Kembalikan jumlah elemen yang baru
}

// Fungsi untuk menampilkan rata-rata nilai dalam array
func tampilkanRataRata(array []int, jumlahElemen int) {
    var total int
    // Menghitung total dari semua elemen dalam array
    for i := 0; i < jumlahElemen; i++ {
        total += array[i]
    }
}

```

```

// Menghitung rata-rata
rataRata := float64(total) / float64(jumlahElemen)
fmt.Printf("Rata-rata: %.2f\n", rataRata)
}

// Fungsi untuk menghitung dan menampilkan standar deviasi dari elemen array
func tampilkanStandarDeviasi(array []int, jumlahElemen int) {
    var total, rataRata float64
    // Menghitung total nilai dalam array
    for i := 0; i < jumlahElemen; i++ {
        total += float64(array[i])
    }
    // Menghitung rata-rata
    rataRata = total / float64(jumlahElemen)

    var totalDeviasi float64
    // Menghitung deviasi setiap elemen terhadap rata-rata secara manual
    for i := 0; i < jumlahElemen; i++ {
        deviasi := float64(array[i]) - rataRata
        totalDeviasi += deviasi * deviasi
    }
    // Menghitung standar deviasi secara manual
    standarDeviasi := totalDeviasi / float64(jumlahElemen)
    standarDeviasi = hitungAkarKuadrat(standarDeviasi)
    fmt.Printf("Standar deviasi: %.2f\n", standarDeviasi)
}

// Fungsi untuk menghitung akar kuadrat secara manual (seperti math.Sqrt)
func hitungAkarKuadrat(x float64) float64 {
    // Metode estimasi akar kuadrat menggunakan metode Newton-Raphson
    if x < 0 {
        return 0
    }
    z := x / 2.0
    for i := 0; i < 10; i++ {
        z = (z + x/z) / 2.0
    }
    return z
}

// Fungsi untuk menampilkan frekuensi kemunculan suatu bilangan dalam array
func tampilkanFrekuensi(array []int, jumlahElemen int) {
    var bilangan int
    // Meminta input untuk bilangan yang frekuensinya ingin dihitung
    fmt.Print("Masukkan bilangan untuk mencari frekuensi: ")

```

```

    fmt.Scanln(&bilangan)
    frekuensi := 0
    // Menghitung frekuensi kemunculan bilangan dalam array
    for i := 0; i < jumlahElemen; i++ {
        if array[i] == bilangan {
            frekuensi++
        }
    }
    fmt.Printf("Frekuensi dari bilangan %d: %d\n", bilangan, frekuensi)
}

```

Screenshoot Output

```

PS D:\Wathhh\matkul\smst 3\Praktikum Alpro 2\Modul-VII Struct and Array> go run "d:\Wathhh\matkul\smst 3\Prakt
Masukkan jumlah elemen (maksimal 10): 5
Masukkan elemen-elemen array:
3
2
4
3
2

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:
3
2
4
3
2

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: █

```

Deskripsi Program

Program ini memungkinkan pengguna untuk memasukkan hingga 10 elemen ke dalam array dan memilih berbagai operasi pada array melalui menu. Pengguna dapat menampilkan elemen berdasarkan indeks ganjil/genap atau kelipatan bilangan tertentu. Selain itu, program menyediakan fitur untuk menghapus elemen, menghitung rata-rata, dan standar deviasi dari array. Pengguna juga dapat mencari frekuensi kemunculan suatu bilangan dalam array. Program terus berjalan hingga pengguna memilih opsi untuk keluar.

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)

```
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

// Meminta input nama klub
fmt.Print("Masukkan nama klub A: ")
fmt.Scanln(&namaKlubA)
fmt.Print("Masukkan nama klub B: ")
fmt.Scanln(&namaKlubB)

// Loop untuk menerima input skor
for {
    fmt.Printf("Masukkan skor untuk %s dan %s: ", namaKlubA, namaKlubB)
    _, err := fmt.Scanf("%d %d\n", &skorKlubA, &skorKlubB)

    // Periksa jika input tidak valid
    if err != nil {
        fmt.Println("Input tidak valid. Harap masukkan dua angka.")
        // Menampilkan rekapan hasil pertandingan sejauh ini
        tampilkanRekapan(daftarHasil)
        // Mengosongkan buffer input agar input berikutnya tidak terpengaruh
        var temp string
        fmt.Scanln(&temp)
        continue
    }

    // Keluar dari loop jika ada skor negatif
    if skorKlubA < 0 || skorKlubB < 0 {
        break
    }

    // Tentukan hasil pertandingan
    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
    }
}

// Menampilkan hasil akhir
fmt.Println("Rekapan hasil pertandingan:")
tampilkanRekapan(daftarHasil)
fmt.Println("Pertandingan selesai")
}

```

```
// Fungsi untuk menampilkan rekap hasil pertandingan
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)
        }
    }
}
```

Screenshot Output

```
Masukkan nama klub A: nath
Masukkan nama klub B: jago
Masukkan skor untuk nath dan jago: 4 5
Masukkan skor untuk nath dan jago: 3 5
Masukkan skor untuk nath dan jago: 3 3
Masukkan skor untuk nath dan jago: 2 4
Masukkan skor untuk nath dan jago: -1 4
Rekapan hasil pertandingan:
Pertandingan 1: jago
Pertandingan 2: jago
Pertandingan 3: Seri
Pertandingan 4: jago
Pertandingan selesai
PS D:\Nathhh\matkul\smst 3\Praktikum Alpro 2\Modul-VII Struct and Array>
```

Deskripsi Program

Program ini meminta input nama dua klub dan skor pertandingan antara keduanya. Program akan terus menerima skor pertandingan hingga ada skor negatif yang dimasukkan, yang menandakan akhir pertandingan. Setiap kali skor valid dimasukkan, program akan menentukan pemenang atau jika pertandingan berakhir seri, dan menambahkannya ke daftar hasil. Jika terjadi kesalahan input, program akan menampilkan rekap hasil pertandingan yang sudah dimasukkan. Setelah semua pertandingan selesai, program akan menampilkan rekap hasil pertandingan yang lengkap.

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 */
```

an 67 | Modul Praktikum Algoritma dan Pemrograman 2

```
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
    // isi array tab dengan memanggil prosedur isiArray

    // Balikkan 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 I 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 I A K
Palindrom ? true

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

Sourcecode

```
package main
```

```

package main

import "fmt"

const MaksKarakter int = 127 // Konstanta batas maksimum jumlah karakter
dalam array

// Tipe data 'arrayKarakter' adalah array yang bisa menyimpan hingga 127
karakter bertipe rune (Unicode)
type arrayKarakter [MaksKarakter]rune

// Fungsi untuk mengisi array dengan karakter-karakter yang dimasukkan oleh
pengguna
func inputArrayKarakter(array *arrayKarakter, jumlah int) {
    /* I.S. Data karakter tersedia dalam input
       F.S. Array array terisi sejumlah jumlah karakter yang dimasukkan pengguna,
       Proses berhenti jika karakter titik('.') dimasukkan atau jumlah <=
       MaksKarakter */
    var indeks int
    var karakter rune
    fmt.Println("Masukkan karakter: ")
    for indeks = 0; indeks < jumlah; indeks++ {
        // Memasukkan karakter satu per satu
        fmt.Scanf("%c", &karakter)
        // Jika karakter yang dimasukkan adalah titik, berhenti
        if karakter == '.' {
            break
        }
        // Menyimpan karakter ke dalam array
        array[indeks] = karakter
    }
}

// Fungsi untuk menampilkan seluruh isi array ke layar
func tampilkanKarakter(array arrayKarakter, jumlah int) {
    /* I.S. Terdefinisi array yang berisi sejumlah jumlah karakter
       F.S. n karakter dalam array ditampilkan di layar */
    for i := 0; i < jumlah; i++ {
        // Mencetak karakter satu per satu
        fmt.Printf("%c", array[i])
    }
    // Menambahkan baris baru setelah mencetak karakter
    fmt.Println()
}

```

```

// Fungsi untuk membalikkan urutan karakter dalam array
func balikkanUrutan(array *arrayKarakter, jumlah int) {
    /* I.S. Terdefinisi array yang berisi sejumlah jumlah karakter
       F.S. Urutan isi array dibalik */
    for i := 0; i < jumlah/2; i++ {
        // Menyimpan nilai sementara untuk pertukaran
        sementara := array[i]
        // Melakukan pertukaran posisi karakter
        array[i] = array[jumlah-i-1]
        array[jumlah-i-1] = sementara
    }
}

// Fungsi untuk memeriksa apakah array membentuk palindrom
func cekApakahPalindrom(array arrayKarakter, jumlah int) bool {
    /* Mengembalikan true jika susunan karakter di dalam array membentuk
       palindrom,
       dan false jika tidak. Petunjuk: Gunakan fungsi balikkanUrutan */
    var arraySalinan arrayKarakter
    // Menyalin array ke arraySalinan
    copy(arraySalinan[:], array[:])
    // Membalikkan arraySalinan
    balikkanUrutan(&arraySalinan, jumlah)

    // Memeriksa apakah array dan arraySalinan sama
    for i := 0; i < jumlah; i++ {
        if arraySalinan[i] != array[i] {
            // Jika ada perbedaan, berarti bukan palindrom
            return false
        }
    }
    // Jika semua elemen sama, berarti palindrom
    return true
}

func main() {
    // Deklarasi array untuk menyimpan karakter dan jumlah karakter yang
    // dimasukkan
    var array arrayKarakter
    var jumlahKarakter int
    // Meminta pengguna untuk memasukkan jumlah karakter
    fmt.Println("Masukkan jumlah karakter: ")
    fmt.Scanln(&jumlahKarakter)
    // Mengisi array dengan karakter-karakter yang dimasukkan
    inputArrayKarakter(&array, jumlahKarakter)
}

```

```

// Menampilkan karakter-karakter yang dimasukkan
fmt.Println("Karakter yang dimasukkan adalah:")
tampilkanKarakter(array, jumlahKarakter)

// Membalikkan urutan array dan menampilkan reverse text
balikkanUrutan(&array, jumlahKarakter)
fmt.Println("Teks terbalik (reverse) adalah:")
tampilkanKarakter(array, jumlahKarakter)

// Memeriksa apakah array membentuk palindrom
if cekApakahPalindrom(array, jumlahKarakter) {
    // Jika ya, tampilkan pesan palindrom
    fmt.Println("Array ini adalah palindrom.")
} else {
    // Jika tidak, tampilkan pesan bukan palindrom
    fmt.Println("Array ini bukan palindrom.")
}
}

```

Screenshoot Output

```

PS D:\Nathhh\matkul\smst 3\Praktikum Alpro 2\Modul-VII Struct and Array> go run .\Modul-VII Struct and Array\unguided4\unguided4.go
Masukkan jumlah karakter:
7
Masukkan karakter:
kambing
Karakter yang dimasukkan adalah:
kambing
Teks terbalik (reverse) adalah:
gnibmak
Array ini bukan palindrom.
PS D:\Nathhh\matkul\smst 3\Praktikum Alpro 2\Modul-VII Struct and Array> 

```

Deskripsi Program

Program ini meminta pengguna untuk memasukkan sejumlah karakter dan menyimpannya dalam sebuah array dengan tipe data rune (Unicode). Pengguna dapat memasukkan karakter hingga mencapai batas maksimum atau sampai memasukkan titik ('.'), yang akan menghentikan input. Setelah itu, program menampilkan karakter yang dimasukkan, membalikkan urutan karakter dalam array, dan menampilkan hasilnya. Program juga memeriksa apakah urutan karakter dalam array membentuk palindrom, yaitu apakah urutannya sama jika dibaca dari belakang. Hasil pemeriksaan ini akan ditampilkan kepada pengguna.