

LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2
MODUL: 7
MATERI: STRUCK & ARRAY



DISUSUN OLEH:

NAMA: JESIKA METANIA RAHMA ARIFIN

NIM: 103112400080

KELAS: 12 IF 01

DOSEN:

Dimas Fanny Hebrisianto Permadi S.ST, M.Kom

PROGRAM STUDI S1 TEKNIK INFORMATIKA

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2025/2026

DASAR TEORI

1. STRUCK ATAU RECORD

Stucture memungkinkan pemrograman untuk mengelompokkan beberapa data atau nilai yang memiliki relasi atau keterkaitan tertentu menjadi suatu kesatuan. Masing-masing nilai tersimpan dalam field dari stucture tersebut.

Berbeda dengan bahasa pemrograman lain. kesamaan tipe dari dua variabel berjenis stucture 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 stucture tersebut didefinisikan sebagai sebuah tipe baru, sehingga deklarasi stucture tidak perlu lagi seluruh field-nya ditulis ulang berkali-kali.

CONTOH:

```
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)
}
```

2. ARRAY

Array adalah kumpulan data bertipe sama, yang disimpan dalam sebuah variabel. Array memiliki kapasitas yang nilainya ditentukan pada saat pembuatan, menjadikan elemen/data yang disimpan di array tersebut jumlahnya tidak boleh melebihi yang sudah dialokasikan. Default nilai tiap elemen array pada awalnya tergantung dari tipe datanya. Jika int maka tiap element zero value-nya adalah 0, jika bool maka false, dan seterusnya. Setiap elemen array memiliki indeks berupa angka yang merepresentasikan posisi urutan elemen tersebut. Indeks array dimulai dari 0.

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	}

CONTOH:

```
// Mengganti isi elemen ke-0 dengan nilai dari elemen ke-7
arr[0] = arr[7]
// Mengambil data field x dari elemen ke-i
```

```
currX = arr[i].center.x
// Mengambil elemen terakhir
n := len(arr)
buf := arr[n-1]
```

Slice (Array dinamik)

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

```
1 // declaring chop as an empty slice of float64
2 var chop []float64
3
4 // declaring sl01 as a slice
5 var sl01 = []int{ 11, 2, 3, 5, 7, 13 }
```

Sebuah slice dapat diprealokasi menggunakan fungsi built-in **make**

```
1 // Prealokasi 10 elemen untuk sl02 dan sejumlah tempat tambahan
2 var sl02 []int = make([]int, 10, 20)
3
4 // Prealokasi 7 elemen untuk sl03 tanpa tempat tambahan
5 var sl03 []circType = make([]circType, 7)
```

Fungsi built-in **len** dapat digunakan untuk mengetahui ukuran slice. Fungsi lain, **cap**, dapat digunakan untuk mengetahui total tempat yang disediakan untuk slice tersebut.

```
1 // Cetak jumlah elemen dan tempat yang tersedia untuk sl02
2 fmt.Println( len(sl02), cap(sl02) )
```

Fungsi built-in **append** dapat digunakan untuk menambahkan elemen ke suatu slice, dan bila perlu memperbesar tempat untuk slice tersebut.

```
1 /* Append elemen baru, membuat slice baru, dan menyimpan kembali slice baru
   ke variabel semula. Boleh juga disimpan ke variabel lain, sehingga variabel
   semula masih menyimpan slice yang asli. */
2 sl01 = append(sl01, 17)
3 sl01 = append(sl01, 19, 23)
```

Sebuah slice baru juga dapat terbentuk dengan mengambil slice dari suatu array atau slice yang lain.

```
// Ambil 3 elemen pertama dari suatu slice atau array
sl04 = arr[:4]

// Ambil beberapa elemen terakhir, dimulai dari indeks 5
sl05 = sl01[5:]

// Salin semua dari slice/array aslinya
sl06 = sl05[:]

// Salin element dari indeks 3 sampai, tapi tidak termasuk, 5.
// Jadi dalam contoh hanya 2 elemen sl06[3] dan sl06[4] yang disalin
sl07 = sl06[3:5]
```

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**.

```
1 // Deklarasi variabel dct sebagai map bilangan bulat dengan kunci string
2 var dct map[string]int
3
4 // Deklarasi map lain dct1 dari elemen string dengan kunci juga string
5 // Mempunyai nilai awal dct1["john"] = "hi", dct1["anne"] = "darling"
6 var dct1 = map[string]string{ "john":"hi", "anne":"darling" }
7
8 // Deklarasi dan prealokasi tempat untuk map dct2
9 var dct2 map[float64]int = make(map[float64]int, 10)
10
11 // Mengambil nilai yang tersimpan dengan kunci "john"
12 fmt.Println( dct1["john"] )
13
14 // Mengganti nilai yang tersimpan pada kunci "anne", dan
15 // Membuat entri baru dengan kunci "boy"
16 dct1["anne"] = "lovely"
17 dct1["boy"] = "runaround"
18
19 // Menghapus entri dengan kunci "john"
20 delete(dct1, "john")
```

GUIDED 1

```
package main

import (
    "fmt"
)

func main() {
    // Deklarasi dan inisialisasi array nilai mahasiswa
    nilaiMahasiswa := [5]int{85, 90, 78, 88, 95}

    fmt.Println("Data Nilai Mahasiswa:")
    fmt.Println("=====")

    // Menampilkan nilai per mahasiswa
    for i, nilai := range nilaiMahasiswa {
        fmt.Printf("Mahasiswa %d: %d\n", i+1, nilai)
    }

    // Menghitung rata-rata nilai
    var total int
    for _, nilai := range nilaiMahasiswa {
        total += nilai
    }
    rataRata := float64(total) / float64(len(nilaiMahasiswa))

    fmt.Println("=====")
    fmt.Printf("Rata-rata nilai: %.2f\n", rataRata)

    // Mencari nilai tertinggi dan terendah
    tertinggi := nilaiMahasiswa[0]
    terendah := nilaiMahasiswa[0]
```

```

for _, nilai := range nilaiMahasiswa {
    if nilai > tertinggi {
        tertinggi = nilai
    }
    if nilai < terendah {
        terendah = nilai
    }
}

fmt.Printf("Nilai tertinggi: %d\n", tertinggi)
fmt.Printf("Nilai terendah: %d\n", terendah)

// Contoh array 2 dimensi
fmt.Println("\nContoh Array 2 Dimensi:")
fmt.Println("=====")

// Nilai ujian mahasiswa dalam 2 mata kuliah (Matematika, Bahasa)
nilaiUjian := [3][2]int{
    {80, 85},
    {90, 75},
    {70, 95},
}

// Menampilkan nilai ujian per mahasiswa
fmt.Println("Nilai Ujian Mahasiswa (Matematika, Bahasa):")
for i, nilai := range nilaiUjian {
    fmt.Printf("Mahasiswa %d: Matematika = %d, Bahasa = %d\n", i+1,
nilai[0], nilai[1])
}
}

```

OUTPUT:

```

PROBLEMS (12) OUTPUT DEBUG CONSOLE TERMINAL PORTS
=====
Mahasiswa 1: 85
Mahasiswa 2: 90
Mahasiswa 3: 78
Mahasiswa 4: 88
Mahasiswa 5: 95
=====
Rata-rata nilai: 87.20
Nilai tertinggi: 95
Nilai terendah: 78

Contoh Array 2 Dimensi:
=====
Nilai Ujian Mahasiswa (Matematika, Bahasa):
Mahasiswa 1: Matematika = 80, Bahasa = 85
Mahasiswa 2: Matematika = 90, Bahasa = 75
Mahasiswa 3: Matematika = 70, Bahasa = 95
PS C:\103112400080_MODUL 7>

```

DESKRIPSI PROGRAM:

Program ini adalah program yang di buat pengguna untuk menampilkan dan menghitung nilai rata rata Mahasiswa.

GUIDED 2

```
package main

import (
    "fmt"
    "time"
)

// Struct untuk barang dalam struk belanja
type Item struct {
    Name    string
    Price   float64
    Quantity int
}

// Struct untuk struk belanja
type Receipt struct {
    StoreInfo string
    Date       time.Time
    Items      []Item
    TotalAmount float64
}

// Method untuk menghitung total harga semua item
func (r *Receipt) CalculateTotal() {
    var total float64
    for _, item := range r.Items {
        total += item.Price * float64(item.Quantity)
    }
    r.TotalAmount = total
}

// Method untuk mencetak struk belanja
func (r Receipt) PrintReceipt() {
    fmt.Println("=====")
    fmt.Println(r.StoreInfo)
    fmt.Println("Tanggal:", r.Date.Format("02-01-2006 15:04"))
    fmt.Println("=====")
    fmt.Printf("%-15s %-10s %-8s %-10s\n", "Item", "Harga", "Jumlah", "Total")
    fmt.Println("-----")

    for _, item := range r.Items {
        itemTotal := item.Price * float64(item.Quantity)
        fmt.Printf("%-15s Rp%-9.2f %-8d Rp%-9.2f\n", item.Name,
            item.Price, item.Quantity, itemTotal)
    }

    fmt.Println("=====")
    fmt.Println("=====")
```

```

        fmt.Printf("%-35s Rp%-9.2f\n", "Total Belanja:", r.TotalAmount)
        fmt.Println("=====
=====")
        fmt.Println("Terima kasih telah berbelanja!")
    }

    func main() {
        receipt := Receipt{
            StoreInfo: "Toko Sembako Makmur\nJl. Raya No. 123, Jakarta",
            Date:      time.Now(),
            Items: []Item{
                {Name: "Beras", Price: 12000, Quantity: 5},
                {Name: "Gula", Price: 15000, Quantity: 2},
                {Name: "Minyak", Price: 20000, Quantity: 1},
                {Name: "Telur", Price: 2000, Quantity: 10},
            },
        }

        receipt.CalculateTotal()
        receipt.PrintReceipt()
    }
}

```

OUTPUT:

```

PROBLEMS 12 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\103112400080_MODUL 7> go run "c:\103112400080_MODUL 7\Guided2.go"
=====
Toko Sembako Makmur
Jl. Raya No. 123, Jakarta
Tanggal: 25-04-2025 07:48
=====
Item          Harga      Jumlah    Total
-----
Beras         Rp12000.00    5      Rp60000.00
Gula          Rp15000.00    2      Rp30000.00
Minyak        Rp20000.00    1      Rp20000.00
Telur         Rp2000.00    10     Rp20000.00
=====
Total Belanja:                Rp130000.00
=====
Terima kasih telah berbelanja!
PS C:\103112400080_MODUL 7>

```

DESKRIPSI PROGRAM:

Program ini adalah program yang di buat untuk menghitung total belanjaan pada suatu took sembako.

UNGUIDED 1

Suatu lingkaran didefinisikan dengan koordinat titik pusat (x_1 , y_1) 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".

SOURCECODE

```
// JESIKA METANIA RAHMA ARIFIN
// 10311240080
package main

import "fmt"

func jarakKuadrat(titik1, titik2 [2]int) int {
    return (titik2[0]-titik1[0]) * (titik2[0]-titik1[0]) + (titik2[1]-titik1[1]) *
(titik2[1]-titik1[1])
}

func posisiTitik(titik [2]int, lingkaran1, lingkaran2 [3]int) string {
    jarakKuadratKeLingkaran1 := jarakKuadrat(titik, [2]int{lingkaran1[0],
lingkaran1[1]})
    jarakKuadratKeLingkaran2 := jarakKuadrat(titik, [2]int{lingkaran2[0],
lingkaran2[1]})

    // Kuadrat radius lingkaran
    kuadratRadiusLingkaran1 := lingkaran1[2] * lingkaran1[2]
    kuadratRadiusLingkaran2 := lingkaran2[2] * lingkaran2[2]

    // Tentukan posisi titik di lingkaran 1 dan 2
    diDalamLingkaran1 := jarakKuadratKeLingkaran1 <=
kuadratRadiusLingkaran1
    diDalamLingkaran2 := jarakKuadratKeLingkaran2 <=
kuadratRadiusLingkaran2

    // Menentukan posisi berdasarkan dua lingkaran
    if diDalamLingkaran1 && diDalamLingkaran2 {
        return "Titik di dalam lingkaran 1 dan 2"
    } else if diDalamLingkaran1 {
        return "Titik di dalam lingkaran 1"
    } else if diDalamLingkaran2 {
        return "Titik di dalam lingkaran 2"
    } else {
        return "Titik di luar lingkaran 1 dan 2"
    }
}

func main() {
    // Input untuk lingkaran 1 (koordinat pusat x, y dan radius)
    var lingkaran1 [3]int
    fmt.Scan(&lingkaran1[0], &lingkaran1[1], &lingkaran1[2])
}
```



```

// Input untuk lingkaran 2 (koordinat pusat x, y dan radius)
var lingkaran2 [3]int
fmt.Scan(&lingkaran2[0], &lingkaran2[1], &lingkaran2[2])

// Input untuk titik sembarang (koordinat x, y)
var titik [2]int
fmt.Scan(&titik[0], &titik[1])

// Output hasil posisi titik
fmt.Println(posisiTitik(titik, lingkaran1, lingkaran2))
}

```

OUTPUT:

```

PROBLEMS 12 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\103112400080_MODUL 7> go run "c:\103112400080_MODUL 7\Unguided1.go"
1 1 5
8 8 4
2 2
Titik di dalam lingkaran 1
PS C:\103112400080_MODUL 7> go run "c:\103112400080_MODUL 7\Unguided1.go"
1 2 3
4 5 6
7 8
Titik di dalam lingkaran 2
PS C:\103112400080_MODUL 7> go run "c:\103112400080_MODUL 7\Unguided1.go"
5 10 15
-15 4 20
0 0
Titik di dalam lingkaran 1 dan 2
PS C:\103112400080_MODUL 7> go run "c:\103112400080_MODUL 7\Unguided1.go"

```

DESKRIPSI PROGRAM:

Program ini adalah program yang di buat untuk mengecek suatu titik kordinat untuk melihat dimana tiitk lingkaran berada.

UNGUIDED 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.
- 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

```
// JESIKA METANIA RAHMA ARIFIN
// 103112400080

package main

import (
    "fmt"
    "math"
)

func inputArray(n int) []int {
    arr := make([]int, n)
    for i := 0; i < n; i++ {
        fmt.Printf("Masukkan elemen ke-%d: ", i)
        fmt.Scan(&arr[i])
    }
    return arr
}

func tampilSemua(arr []int) {
    fmt.Println("Isi array:", arr)
}

func tampilIndeksGanjil(arr []int) {
    fmt.Print("Indeks ganjil: ")
    for i := 1; i < len(arr); i += 2 {
        fmt.Print(arr[i], " ")
    }
    fmt.Println()
}

func tampilIndeksGenap(arr []int) {
    fmt.Print("Indeks genap: ")
    for i := 0; i < len(arr); i += 2 {
        fmt.Print(arr[i], " ")
    }
    fmt.Println()
}

func tampilKelipatan(arr []int, x int) {
    fmt.Printf("Indeks kelipatan %d: ", x)
    for i := 0; i < len(arr); i++ {
        if i%x == 0 {
            fmt.Print(arr[i], " ")
        }
    }
    fmt.Println()
}

func hapusIndeks(arr []int, idx int) []int {
```

```

        if idx < 0 || idx >= len(arr) {
            fmt.Println("Indeks tidak valid!")
            return arr
        }
        return append(arr[:idx], arr[idx+1:]...)
    }

func rataRata(arr []int) float64 {
    total := 0
    for _, v := range arr {
        total += v
    }
    return float64(total) / float64(len(arr))
}

func standarDeviasi(arr []int) float64 {
    avg := rataRata(arr)
    var total float64
    for _, v := range arr {
        selisih := float64(v) - avg
        total += selisih * selisih
    }
    return math.Sqrt(total / float64(len(arr)))
}

func frekuensi(arr []int, angka int) int {
    count := 0
    for _, v := range arr {
        if v == angka {
            count++
        }
    }
    return count
}

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

    arr := inputArray(n)
    tampilSemua(arr)
    tampilIndeksGanjil(arr)
    tampilIndeksGenap(arr)

    var x int
    fmt.Print("Masukkan nilai x untuk kelipatan indeks: ")
    fmt.Scan(&x)
    tampilKelipatan(arr, x)

    var hapus int

```

```

    fmt.Print("Masukkan indeks yang ingin dihapus: ")
    fmt.Scan(&hapus)
    arr = hapusIndeks(arr, hapus)
    tampilSemua(arr)

    if len(arr) > 0 {
        fmt.Printf("Rata-rata: %.2f\n", rataRata(arr))
        fmt.Printf("Standar deviasi: %.2f\n", standarDeviasi(arr))
    } else {
        fmt.Println("Array kosong, tidak bisa menghitung rata-rata dan
standar deviasi.")
    }

    var cari int
    fmt.Print("Masukkan bilangan untuk dihitung frekuensinya: ")
    fmt.Scan(&cari)
    fmt.Printf("Frekuensi %d: %d kali\n", cari, frekuensi(arr, cari))
}

```

OUTPUT:

```

PROBLEMS 12 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\103112400080_MODUL 7> go run "c:\103112400080_MODUL 7\Unguided2.go"
Masukkan jumlah elemen array: 3
Masukkan elemen ke-0: 10
Masukkan elemen ke-1: 30
Masukkan elemen ke-2: 70
Isi array: [10 30 70]
Indeks ganjil: 30
Indeks genap: 10 70
Masukkan nilai x untuk kelipatan indeks: 1
Indeks kelipatan 1: 10 30 70
Masukkan indeks yang ingin dihapus: 0
Isi array: [30 70]
Rata-rata: 50.00
Standar deviasi: 20.00
Masukkan bilangan untuk dihitung frekuensinya: 30
Frekuensi 30: 1 kali

```

DESKRIPSI PROGRAM:

Program ini adalah program yang di buat dengan permintaan spesifikasi di atas.

UNGUIDED 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 berlagu. 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.

SOURCECODE

```

//JESIKA METANIA RAHMA ARIFIN
// 103112400080

package main

```

```

import "fmt"

func main() {
    var klubA, klubB string
    fmt.Print("Klub A: ")
    fmt.Scan(&klubA)
    fmt.Print("Klub B: ")
    fmt.Scan(&klubB)

    const jumlahPertandingan = 9

    for i := 1; i <= jumlahPertandingan; i++ {
        var skorA, skorB int
        fmt.Printf("Pertandingan %d: ", i)
        fmt.Scan(&skorA, &skorB)

        if skorA < 0 || skorB < 0 {
            fmt.Printf("Pertandingan %d: Skor tidak valid, dilewati.\n", i)
            continue
        }
        if skorA > skorB {
            fmt.Println(klubA)
        } else if skorB > skorA {
            fmt.Println(klubB)
        } else {
            fmt.Println("Draw")
        }
    }

    fmt.Println("Pertandingan selesai")
}

```

OUTPUT:

```

PS C:\103112400080_MODUL 7> go run "c:\103112400080_MODUL 7\Unguided3.go"
Klub A: MU
Klub B: Inter
Pertandingan 1: 2 0
MU
Pertandingan 2: 1 2
Inter
Pertandingan 3: 2 2
Draw
Pertandingan 4: 0 1
Inter
Pertandingan 5: 3 2
MU
Pertandingan 6: 1 0
MU
Pertandingan 7: 5 2
MU
Pertandingan 8: 3 3
Draw
Pertandingan 9: -1 2
Pertandingan 9: Skor tidak valid, dilewati.
Pertandingan selesai

```

DESKRIPSI PROGRAM:

Program ini adalah program yang di buat untuk melihat kemenangan dari suatu pertandingan.

UNGUIDED 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 */

/*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
}
```

THE ANSWER

```
// JESIKA METANIA RAHMA ARIFIN
// 103112400080

package main

import (
    "fmt"
    "strings"
)

// Fungsi untuk mengecek apakah kata adalah palindrom
func isPalindrom(kata string) bool {
    kata = strings.ToUpper(kata)
    n := len(kata)
    for i := 0; i < n/2; i++ {
        if kata[i] != kata[n-1-i] {
            return false
        }
    }
    return true
}
```

```

    }

    // Fungsi untuk mencetak karakter per huruf dipisah spasi
    func tampilkanTeks(kata string) string {
        chars := []rune(strings.ToUpper(kata))
        var hasil strings.Builder
        for i, c := range chars {
            hasil.WriteRune(c)
            if i != len(chars)-1 {
                hasil.WriteString(" ")
            }
        }
        return hasil.String()
    }

    func main() {
        var kata string
        fmt.Scan(&kata)

        fmt.Println(kata)
        fmt.Println(isPalindrom(kata))
    }
}

```

OUTPUT:

```

PROBLEMS 12 OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\103112400080_MODUL 7> go run "c:\103112400080_MODUL 7\Unguided4.go"
K A T A K
K
true
PS C:\103112400080_MODUL 7> go run "c:\103112400080_MODUL 7\Unguided4.go"
S E N A N G
S
true
PS C:\103112400080_MODUL 7>

```

DESKRIPSI PROGRAM:

Program ini adalah program yang dibuat untuk mengetahui apakah kata yang di masukkan termasuk polindrom atau bukan.

KESIMPULAN

- **Array** cocok digunakan untuk **data yang homogen** dan jumlahnya **tetap**, karena memberikan akses cepat dan efisien.
- **Struct** sangat berguna untuk mengelola **data kompleks** yang terdiri dari beberapa tipe data yang berbeda dalam satu entitas. Ini memberikan struktur yang lebih baik, tetapi dengan sedikit tambahan overhead.

REFERENSI

Dari modul