

**LAPORAN PRAKTIKUM  
ALGORITMA PEMROGRAMAN 2**

**MODUL 7**

**STRUCK & ARRAY**



Oleh:

Arvan Murbiyanto

2311102074

IF-11-02

**S1 TEKNIK INFORMATIKA  
INSTITUT TEKNOLOGI TELKOM PURWOKERTO**

**2024**

## I. DASAR TEORI

### (\*)Pengertian Array

Array adalah struktur data yang digunakan untuk menyimpan kumpulan elemen dengan tipe data yang sama secara berurutan dalam memori. Setiap elemen dalam array memiliki posisi unik, yang dikenal sebagai indeks, dimulai dari 0. Dengan indeks ini, kita dapat mengakses elemen-elemen array secara cepat. Array memiliki beberapa karakteristik utama yang membedakannya dari struktur data lainnya:

1. **Homogenitas:** Setiap elemen dalam array harus memiliki tipe data yang sama, menjaga konsistensi data.
2. **Ukuran Tetap:** Ukuran array ditentukan saat pembuatan dan tidak dapat diubah selama program berjalan.
3. **Akses Acak:** Array memungkinkan akses langsung ke elemen melalui indeks, sehingga pengambilan atau modifikasi data dapat dilakukan dengan cepat.
4. **Banyak Dimensi:** Array dapat berbentuk satu dimensi (vektor) atau lebih, seperti array dua dimensi (matriks), yang mendukung penyimpanan data yang lebih kompleks.

### (\*)Pengertian Struct

Struct, atau struktur, adalah tipe data yang memungkinkan pengelompokan variabel dari berbagai tipe data dalam satu kesatuan logis. Struct sangat berguna dalam mengelola data kompleks yang memiliki beberapa tipe data berbeda namun saling terkait, seperti data mahasiswa yang terdiri dari nama, umur, dan nilai. Variabel dalam struct disebut sebagai *field* atau elemen. Struct memiliki karakteristik sebagai berikut:

1. **Beragam Tipe Data:** Struct dapat mengandung berbagai tipe data dalam satu unit, seperti integer, float, dan string, sehingga cocok untuk merepresentasikan data yang kompleks.
2. **Organisasi Data yang Terstruktur:** Struct menyatukan data terkait ke dalam satu kesatuan yang terorganisir, membuatnya lebih mudah dipahami dan diakses.
3. **Deklarasi yang Fleksibel:** Struct dapat dideklarasikan dengan berbagai cara, baik dengan mendefinisikan dan menginisialisasi field secara langsung, atau menyusunnya bertahap sesuai kebutuhan program.

## II. GUIDED

### Guided 1

```
package main

import (
    "fmt"
    "sort"
)

// Struktur untuk menampung data mahasiswa
type Mahasiswa struct {
    Nama      string
    Matematika int
    Fisika    int
    Kimia     int
    RataRata  float64
}

// Fungsi untuk menghitung rata-rata nilai tiap mahasiswa
func hitungRataRata(m *Mahasiswa) {
    total := m.Matematika + m.Fisika + m.Kimia
    m.RataRata = float64(total) / 3.0
}

// Fungsi utama untuk mengelola dan mengurutkan data mahasiswa
func main() {
    //Array untuk menampung data mahasiswa
    mahasiswa := []Mahasiswa{
        {"Ali", 85, 90, 80, 0},
        {"Budi", 70, 75, 80, 0},
        {"Cici", 90, 85, 95, 0},
        {"Doni", 60, 65, 70, 0},
        {"Eka", 100, 95, 90, 0},
    }

    //Menghitung rata-rata nilai tiap mahasiswa
    for i := range mahasiswa {
        hitungRataRata(&mahasiswa[i])
    }
    //Mengurutkan mahasiswa berdasarkan nilai rata-rata (descending)
    sort.Slice(mahasiswa, func(i, j int) bool {
        return mahasiswa[i].RataRata > mahasiswa[j].RataRata
    })

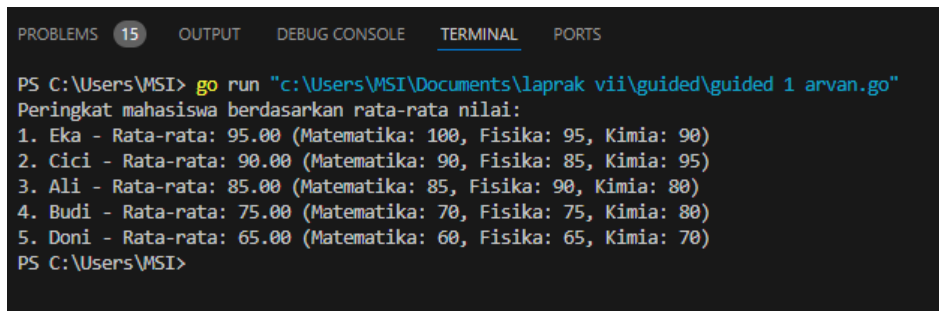
    //Menampilkan hasil
    fmt.Println("Peringkat mahasiswa berdasarkan rata-rata nilai: ")
}
```

```

    for i, m := range mahasiswa {
        fmt.Printf("%d. %s - Rata-Rata: %.2f (Matematika: %d, Fisika: %d, Kimia: %d)\n",
            i+1, m>Nama, m.RataRata, m.Matematika, m.Fisika, m.Kimia)
    }
}

```

Screenshot output



```

PROBLEMS 15 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\MSI> go run "c:\Users\MSI\Documents\laprak vii\guided\guided 1 arvan.go"
Peringkat mahasiswa berdasarkan rata-rata nilai:
1. Eka - Rata-rata: 95.00 (Matematika: 100, Fisika: 95, Kimia: 90)
2. Cici - Rata-rata: 90.00 (Matematika: 90, Fisika: 85, Kimia: 95)
3. Ali - Rata-rata: 85.00 (Matematika: 85, Fisika: 90, Kimia: 80)
4. Budi - Rata-rata: 75.00 (Matematika: 70, Fisika: 75, Kimia: 80)
5. Doni - Rata-rata: 65.00 (Matematika: 60, Fisika: 65, Kimia: 70)
PS C:\Users\MSI>

```

Penjelasan :

Program di atas mengelola dan mengurutkan data mahasiswa berdasarkan nilai rata-rata. Program dimulai dengan mendeklarasikan tipe data `Mahasiswa` yang memiliki atribut nama, nilai matematika, fisika, kimia, dan rata-rata. Fungsi `hitungRataRata` digunakan untuk menghitung rata-rata nilai matematika, fisika, dan kimia tiap mahasiswa. Dalam fungsi `main`, beberapa data mahasiswa dimasukkan ke dalam array, kemudian program menghitung rata-rata nilai tiap mahasiswa. Selanjutnya, data mahasiswa diurutkan berdasarkan rata-rata (dari tertinggi ke terendah) menggunakan fungsi `sort.Slice`. Akhirnya, program menampilkan daftar mahasiswa yang sudah diurutkan sesuai nilai rata-ratanya.

## Guided 2

```
package main

import "fmt"

func main() {
    // Membuat map dengan NIM sebagai kunci dan Nama sebagai nilai
    mahasiswa := map[string]string{
        "20231001": "Andi",
        "20231002": "Budi",
        "20231003": "Cici",
    }

    // Menambahkan data baru ke map
    mahasiswa["20231004"] = "Dedi"

    // Menampilkan seluruh isi map dalam format kolom dan baris
    fmt.Println("Daftar Mahasiswa:")
    fmt.Println("NIM\t\tNama")
    fmt.Println(".....")
    for nim, nama := range mahasiswa {
        fmt.Printf("%s\t%s\n", nim, nama)
    }

    // Mengakses data berdasarkan NIM
    nim := "20231002"
    fmt.Println("\nNama Mahasiswa dengan NIM", nim, "adalah",
mahasiswa[nim])

    // Menghapus data berdasarkan NIM
    delete(mahasiswa, "20231003")

    // Menampilkan isi map setelah data dihapus dalam format kolom dan baris
    fmt.Println("\nDaftar Mahasiswa setelah dihapus:")
    fmt.Println("NIM\t\tNama")
    fmt.Println(".....")
    for nim, nama := range mahasiswa {
        fmt.Printf("%s\t%s\n", nim, nama)
    }
}
```

Output :



The screenshot shows a Go IDE with a terminal window displaying the output of a program. The terminal output is as follows:

```
20231004 Dedi
20231001 Andi
20231002 Budi
20231003 Cici

Nama Mahasiswa dengan NIM 20231002 adalah Budi

Daftar Mahasiswa setelah dihapus:
NIM      Nama
-----
20231001 Andi
20231002 Budi
20231004 Dedi
PS C:\Users\MSI>
```

Penjelasan :

Program di atas mengelola data mahasiswa menggunakan struktur `map`. Program dimulai dengan membuat `map` bernama `mahasiswa` dengan NIM (Nomor Induk Mahasiswa) sebagai kunci dan nama mahasiswa sebagai nilainya. Program kemudian menambahkan data baru ke `map`, menampilkan seluruh data mahasiswa dalam format tabel, dan mengakses data tertentu berdasarkan NIM. Selanjutnya, program menghapus salah satu data mahasiswa berdasarkan NIM tertentu dan menampilkan kembali daftar mahasiswa setelah penghapusan. Program ini menggunakan beberapa fungsi dari package `fmt` untuk mencetak hasil dalam format yang rapi.

### III. UNGUIDED

#### Unguided 1

```
package main

import (
    "fmt"
    "sort"
)

type Mahasiswa struct {
    Nama      string
    Matematika int
    Fisika    int
    Kimia     int
    RataRata  float64
}

func hitungRataRata(m *Mahasiswa) {
    m.RataRata = float64(m.Matematika + m.Fisika +
m.Kimia) / 3.0
}

func hitungRataRataSemua(mahasiswa []Mahasiswa)
{
    for i := range mahasiswa {
        hitungRataRata(&mahasiswa[i])
    }
}

func urutkanBerdasarkanRataRata(mahasiswa
[]Mahasiswa) {
    sort.Slice(mahasiswa, func(i, j int) bool {
        return mahasiswa[i].RataRata >
mahasiswa[j].RataRata
    })
}

func main() {
    mahasiswa := []Mahasiswa{
        {"Ali", 85, 90, 80, 0},
        {"Budi", 70, 75, 80, 0},
        {"Cici", 90, 85, 95, 0},
        {"Doni", 60, 65, 70, 0},
        {"Eka", 100, 95, 90, 0},
    }
}
```

```

hitungRataRataSemua(mahasiswa)
urutkanBerdasarkanRataRata(mahasiswa)

fmt.Println("Peringkat mahasiswa berdasarkan rata-rata nilai:")
for i, m := range mahasiswa {
    fmt.Printf("%d. %s - Rata-rata: %.2f (Matematika: %d,
        Fisika: %d, Kimia: %d)\n",
            i+1, m>Nama, m.RataRata, m.Matematika, m.Fisika,
            m.Kimia)
}
}

```

Screenshot output

```

PROBLEMS 17 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\MSI> go run "c:\Users\MSI\Documents\laprak vii\unguided\unguided 1 arvan.go"
Masukkan titik pusat (x, y) dan radius lingkaran 1: 1 1 5
Masukkan titik pusat (x, y) dan radius lingkaran 2: 8 8 4
Masukkan koordinat titik sembarang (x, y): 2 2
Titik berada di dalam lingkaran 1
PS C:\Users\MSI>

```

### Penjelasan

Program di atas menentukan posisi sebuah titik sembarang berdasarkan dua lingkaran. Program dimulai dengan mendefinisikan struktur `titik` untuk koordinat `(x, y)` dan `lingkaran` untuk menyimpan pusat lingkaran serta radiusnya. Fungsi `jarak` digunakan untuk menghitung jarak antara dua titik, sementara fungsi `didalam` mengecek apakah sebuah titik berada di dalam lingkaran tertentu. Program menerima input untuk koordinat pusat dan radius kedua lingkaran serta koordinat titik sembarang, lalu mengevaluasi posisi titik tersebut apakah berada di dalam lingkaran 1, lingkaran 2, keduanya, atau di luar keduanya, dan menampilkan hasilnya.



## Unguided 2

```
package main

import (
    "fmt"
    "math"
)

func main() {
    var n, x, delIndex, searchValue 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.Printf("Elemen ke-%d: ", i+1)
        fmt.Scan(&array[i])
    }

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

    fmt.Print("b. Elemen dengan indeks ganjil: ")
    for i := 1; i < n; i += 2 {
        fmt.Print(array[i], " ")
    }
    fmt.Println()

    fmt.Print("c. Elemen dengan indeks genap: ")
    for i := 0; i < n; i += 2 {
        fmt.Print(array[i], " ")
    }
    fmt.Println()

    fmt.Print("Masukkan nilai x untuk menampilkan elemen dengan indeks kelipatan x: ")
    fmt.Scan(&x)
    fmt.Printf("d. Elemen dengan indeks kelipatan %d: ", x)
```

```

for i := x; i < n; i += x {
    fmt.Print(array[i], " ")
}
fmt.Println()

fmt.Print("Masukkan indeks elemen yang ingin dihapus: ")
fmt.Scan(&delIndex)
if delIndex >= 0 && delIndex < n {
    array = append(array[:delIndex], array[delIndex+1:]...)
    fmt.Println("e. Isi array setelah penghapusan:", array)
} else {
    fmt.Println("Indeks tidak valid.")
}

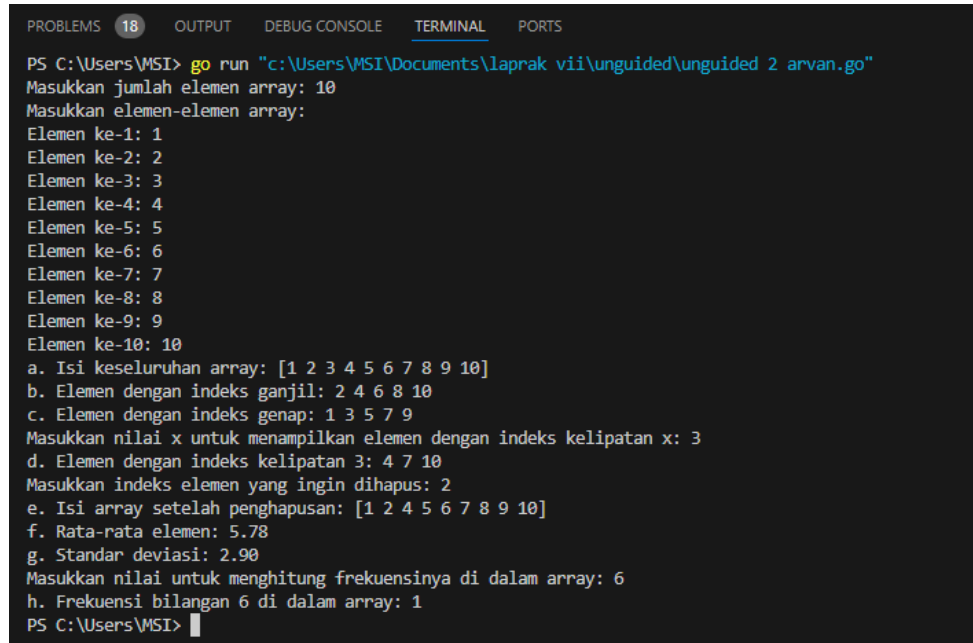
sum := 0
for _, num := range array {
    sum += num
}
average := float64(sum) / float64(len(array))
fmt.Printf("f. Rata-rata elemen: %.2f\n", average)

var deviationSum float64
for _, num := range array {
    deviationSum += math.Pow(float64(num)-average, 2)
}
standardDeviation := math.Sqrt(deviationSum /
float64(len(array)))
fmt.Printf("g. Standar deviasi: %.2f\n", standardDeviation)

fmt.Print("Masukkan nilai untuk menghitung frekuensinya di
dalam array: ")
fmt.Scan(&searchValue)
frequency := 0
for _, num := range array {
    if num == searchValue {
        frequency++
    }
}
fmt.Printf("h. Frekuensi bilangan %d di dalam array: %d\n",
searchValue, frequency)
}

```

## Screenshot output

A screenshot of a terminal window showing the execution of a Go program. The terminal has tabs for PROBLEMS (18), OUTPUT, DEBUG CONSOLE, TERMINAL, and PORTS. The output shows the program running in a PowerShell prompt, where the user enters 'go run' followed by the file path. The program prompts for the number of array elements (10) and then for each element (1 through 10). It then displays various array operations: the full array, elements at odd indices, elements at even indices, elements at a multiple of 3, the array after deleting an element at index 2, the average (5.78), standard deviation (2.90), and the frequency of the number 6 (1).

```
PS C:\Users\MSI> go run "c:\Users\MSI\Documents\laprak vii\unguided\unguided 2 arvan.go"
Masukkan jumlah elemen array: 10
Masukkan elemen-elemen array:
Elemen ke-1: 1
Elemen ke-2: 2
Elemen ke-3: 3
Elemen ke-4: 4
Elemen ke-5: 5
Elemen ke-6: 6
Elemen ke-7: 7
Elemen ke-8: 8
Elemen ke-9: 9
Elemen ke-10: 10
a. Isi keseluruhan array: [1 2 3 4 5 6 7 8 9 10]
b. Elemen dengan indeks ganjil: 2 4 6 8 10
c. Elemen dengan indeks genap: 1 3 5 7 9
Masukkan nilai x untuk menampilkan elemen dengan indeks kelipatan x: 3
d. Elemen dengan indeks kelipatan 3: 4 7 10
Masukkan indeks elemen yang ingin dihapus: 2
e. Isi array setelah penghapusan: [1 2 4 5 6 7 8 9 10]
f. Rata-rata elemen: 5.78
g. Standar deviasi: 2.90
Masukkan nilai untuk menghitung frekuensinya di dalam array: 6
h. Frekuensi bilangan 6 di dalam array: 1
PS C:\Users\MSI>
```

## Penjelasan

Program di atas mengelola array bilangan bulat dengan berbagai fungsi. Pertama, program meminta pengguna untuk memasukkan jumlah elemen yang diinginkan untuk array. Setelah itu, pengguna diminta untuk mengisi nilai untuk setiap elemen array. Program kemudian menampilkan seluruh isi array, elemen dengan indeks ganjil, elemen dengan indeks genap, dan elemen pada indeks kelipatan bilangan yang dimasukkan pengguna. Selain itu, program juga memungkinkan pengguna untuk menghapus elemen pada indeks tertentu dan menampilkan isi array setelah penghapusan. Program menghitung rata-rata dan standar deviasi dari elemen-elemen dalam array, serta menghitung frekuensi kemunculan bilangan tertentu yang diminta pengguna dalam array.

### Unguided 3

```
package main

import (
    "fmt"
)

func main() {
    var klubA, klubB string
    var skorA, skorB int
    var hasil []string
    pertandingan := 1

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

    for {
        fmt.Printf("Pertandingan %d: ", pertandingan)
        fmt.Scan(&skorA, &skorB)

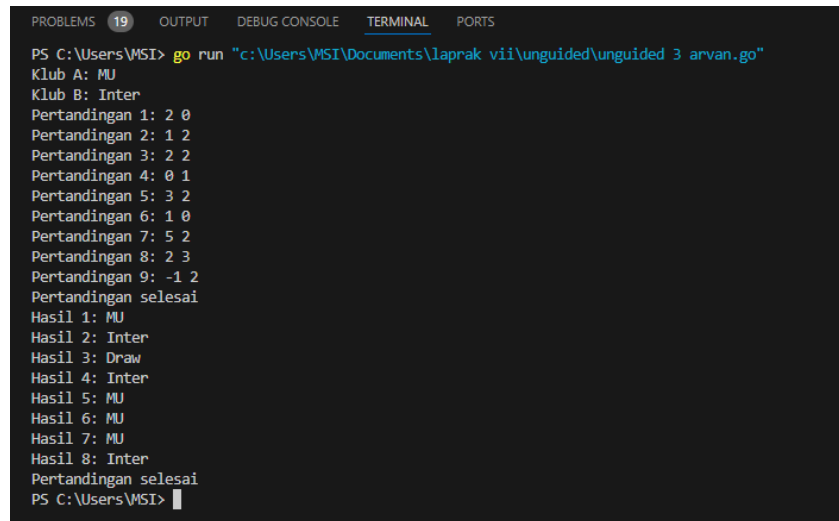
        if skorA < 0 || skorB < 0 {
            fmt.Println("Pertandingan selesai")
            break
        }

        if skorA > skorB {
            hasil = append(hasil, fmt.Sprintf("Hasil %d: %s",
pertandingan, klubA))
        } else if skorB > skorA {
            hasil = append(hasil, fmt.Sprintf("Hasil %d: %s",
pertandingan, klubB))
        } else {
            hasil = append(hasil, fmt.Sprintf("Hasil %d: Draw",
pertandingan))
        }

        pertandingan++
    }

    for _, result := range hasil {
        fmt.Println(result)
    }
    fmt.Println("Pertandingan selesai")
}
```

## Screenshot output

A screenshot of a Visual Studio Code terminal window showing the execution of a Go program. The terminal has tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, and PORTS. The TERMINAL tab is active, displaying the command 'go run "c:\Users\MSI\Documents\laprak vii\unguided\unguided 3 arvan.go"' and its output. The output shows the program asking for 'Klub A' (MU) and 'Klub B' (Inter), then recording 9 matches with scores. The results are: Match 1: 2 0, Match 2: 1 2, Match 3: 2 2, Match 4: 0 1, Match 5: 3 2, Match 6: 1 0, Match 7: 5 2, Match 8: 2 3, Match 9: -1 2. The program then prints the results for each match: 'Hasil 1: MU', 'Hasil 2: Inter', 'Hasil 3: Draw', 'Hasil 4: Inter', 'Hasil 5: MU', 'Hasil 6: MU', 'Hasil 7: MU', 'Hasil 8: Inter', and finally 'Pertandingan selesai'.

## Penjelasan

Program di atas merekam hasil pertandingan antara dua klub sepak bola. Pengguna diminta untuk memasukkan nama kedua klub, setelah itu program meminta input skor untuk setiap pertandingan dalam format angka. Skor akan diinput dalam sebuah loop hingga salah satu atau kedua skor bernilai negatif, yang menandakan akhir dari input pertandingan. Hasil dari setiap pertandingan disimpan dalam slice `hasil`, di mana klub yang menang dicatat, atau "Draw" jika hasilnya imbang. Setelah loop berakhir, program mencetak semua hasil pertandingan yang telah direkam dan menampilkan pesan "Pertandingan selesai."

## Unguided 4

```
package main

import "fmt"

const NMAX int = 127

type tabel [NMAX]rune

func isiArray(t *tabel, n *int) {
    var char rune
    fmt.Println("Masukkan karakter (akhiri dengan titik.): ")
    for *n = 0; *n < NMAX; *n++ {
        fmt.Scanf("%c", &char)
        if char == '.' {
            break
        }
        t[*n] = char
    }
}

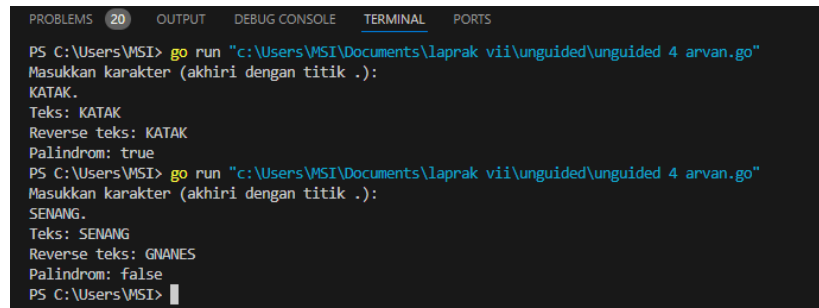
func cetakArray(t tabel, n int) {
    for i := 0; i < n; i++ {
        fmt.Printf("%c", t[i])
    }
    fmt.Println()
}

func balikanArray(t *tabel, n int) {
    for i, j := 0, n-1; i < j; i, j = i+1, j-1 {
        t[i], t[j] = t[j], t[i]
    }
}

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  
  
    isiArray(&tab, &m)  
  
    fmt.Print("Teks: ")  
    cetakArray(tab, m)  
  
    balikanArray(&tab, m)  
    fmt.Print("Reverse teks: ")  
    cetakArray(tab, m)  
  
    if palindrom(tab, m) {  
        fmt.Println("Palindrom: true")  
    } else {  
        fmt.Println("Palindrom: false")  
    }  
}
```

## Screenshot output



```
PROBLEMS 20 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\MSI> go run "c:\Users\MSI\Documents\laprak vii\unguided\unguided 4 arvan.go"
Masukkan karakter (akhiri dengan titik.):
KATAK.
Teks: KATAK
Reverse teks: KATAK
Palindrom: true
PS C:\Users\MSI> go run "c:\Users\MSI\Documents\laprak vii\unguided\unguided 4 arvan.go"
Masukkan karakter (akhiri dengan titik.):
SENANG.
Teks: SENANG
Reverse teks: GNANES
Palindrom: false
PS C:\Users\MSI> 
```

## Penjelasan

Program di atas mengelola array karakter (`tabel`) dan memiliki beberapa fungsi untuk mengisi array, mencetaknya, membalik urutan elemen, dan memeriksa apakah array tersebut adalah palindrom. Pertama, pengguna diminta untuk memasukkan karakter satu per satu, yang akan disimpan dalam array `tabel` hingga maksimal `NMAX` karakter atau sampai pengguna memasukkan titik (`.`) untuk mengakhiri input. Program kemudian mencetak teks asli, membalikinya, dan mencetak teks hasil balikkannya. Terakhir, program memeriksa apakah teks yang dimasukkan adalah palindrom (teks yang sama jika dibaca dari depan atau belakang) dan mencetak hasilnya sebagai `true` atau `false`.