

LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2
MODUL VII
STRUCT & ARRAY



Oleh:

ANISSA FAUZIA ISYANTI

2311102219

S1IF-11-02

S1 TEKNIK INFORMATIKA
INSTITUT TEKNOLOGI TELKOM PURWOKERTO
2024

I. DASAR TEORI

A. Tipe Bentukkan

Tipe bentkan dibedakan menjadi 2 jenis:

1. Alias (Type)

Bahasa pemrograman pada umumnya mengizinkan pemrograman untuk mengubah nama suatu tipe data dengan nama baru yang lebih ringkas dan familiar. Sebagai contoh **“integer”** dapat dirubah dengan nama alias **“bilangan”**. Caranya dengan menggunakan kata kunci **“type”**.

	Notasi Algoritma	Notasi dalam bahasa Go
1	kamus	package main
2	type <nama alias> <tipe data>	
3		type <nama alias> <tipe data>
4	algoritma	
5	...	func main(){
6		... }
7		

2. Struct atau Record

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

	Notasi Algoritma	Notasi dalam bahasa Go
1	kamus	package main
2	type <nama struct> <	type <nama struct> struct {
3	<field 1> <tipe data>	<field 1> <tipe data>
4	<field 2> <tipe data>	<field 2> <tipe data>
5	<field 3> <tipe data>	<field 3> <tipe data>
6	>	}
7		

Berbeda dengan bahasa pemrograman lain, kesamaan tipe dari dua variabel berjenis structure bukan karena namanya tetapi karena strukturnya. Dua variabel dengan nama-nama field dan tipe field yang sama (dan dalam urutan yang sama) dianggap mempunyai tipe yang sama. Tentunya akan lebih memudahkan jika structure tersebut didefinisikan sebagai sebuah tipe baru, sehingga deklarasi structure tidak perlu lagi seluruh field-nya ditulis ulang berkali-kali.

B. Array

Array mempunyai ukuran (jumlah elemen) yang tetap (statis) selama eksekusi program, sehingga jumlah elemen array menjadi bagian dari deklarasi variabel dengan tipe array.

	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)

1. Slice (Array dinamik)

Array dalam Go juga dapat mempunyai ukuran yang dinamik. 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.

```

1 // Ambil 3 elemen pertama dari suatu slice atau array
2 sl04 = arr[:4]
3
4 // Ambil beberapa elemen terakhir, dimulai dari indeks 5
5 sl05 = sl01[5:]
6
7 // Salin semua dari slice/array aslinya
8 sl06 = sl05[:]
9
10 // Salin element dari indeks 3 sampai, tapi tidak termasuk, 5.
11 // Jadi dalam contoh hanya 2 elemen sl06[3] dan sl06[4] yang disalin
12 sl07 = sl06[3:5]

```

2. Map

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

```

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")

```

II. GUIDED

1. Source Code

```
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 (descend)
    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\n",
            (Matematika: %d, Fisika: %d, Kimia: %d)\n",
                i+1,            m>Nama,            m.RataRata,
                m.Matematika, m.Fisika, m.Kimia)

    }
}

```

Screenshot Program

```

C:\Users\LENOVO\Documents\Alpro 2\Anissa Fauzia Isyanti_2311102219_Modul7>go run
7\Guided\guided1.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)

```

Program menghitung rata-rata nilai dari tiga mata pelajaran: Matematika, Fisika, dan Kimia. Program ini dimulai dengan mendefinisikan struktur Mahasiswa yang menyimpan nama mahasiswa dan nilai pelajaran serta rata-rata nilainya. Program menginisialisasi array yang berisi data lima mahasiswa beserta nilai mereka. Kemudian, fungsi hitungRataRata digunakan untuk menghitung rata-rata nilai setiap mahasiswa. Setelah itu, data mahasiswa diurutkan berdasarkan nilai rata-rata secara menurun menggunakan fungsi sort.Slice. Diakhir program akan dicetak peringkat mahasiswa berdasarkan nilai rata-rata.

2. Source Code

```

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)
}

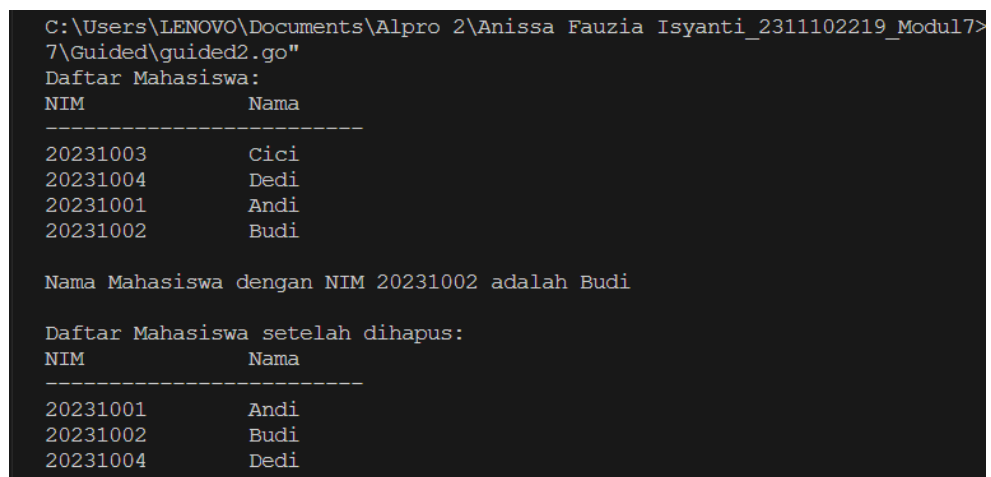
// Mengakses data berdasarkan NIM
nim := "20231002"
fmt.Println("\nNama Mahasiswa dengan NIM", nim,
"adalah", mahasiswa[nim])

// 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)
}
}

```

Screenshot Program



```

C:\Users\LENOVO\Documents\Alpro 2\Anissa Fauzia Isyanti_2311102219_Modul7>
7\Guided\guided2.go"
Daftar Mahasiswa:
NIM      Nama
-----
20231003  Cici
20231004  Dedi
20231001  Andi
20231002  Budi

Nama Mahasiswa dengan NIM 20231002 adalah Budi

Daftar Mahasiswa setelah dihapus:
NIM      Nama
-----
20231001  Andi
20231002  Budi
20231004  Dedi

```

Program ini untuk mengelola data mahasiswa menggunakan struktur data map. Map ini menyimpan NIM (Nomor Induk Mahasiswa) sebagai kunci

dan nama mahasiswa sebagai nilai. Program dimulai dengan inisialisasi map berisi tiga mahasiswa, kemudian menambahkan satu mahasiswa baru. Setelah itu, program menampilkan seluruh isi map dalam format tabel yang terstruktur. Program juga dapat mengakses nama mahasiswa berdasarkan NIM tertentu dan menghapus entri mahasiswa dari map.

III. UNGUIDED

1. Source Code

```
package main

import (
    "fmt"
    "math"
)

func main() {
    var cx1, cy1, r1, cx2, cy2, r2, x, y int
    fmt.Scan(&cx1, &cy1, &r1, &cx2, &cy2, &r2, &x,
    &y)

    dalamLingkaran1 := titikdalamlingkaran(cx1, cy1,
    r1, x, y)
    dalamLingkaran2 := titikdalamlingkaran(cx2, cy2,
    r2, x, y)

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

func hitungJarak_219(cx1, cy1, cx2, cy2 int) int {
    return int(math.Sqrt(float64((cx1-cx2)*(cx1-cx2)
    + (cy1-cy2)*(cy1-cy2))))
}

func titikdalamlingkaran(tengahX, tengahY, jariJari,
    x, y int) bool {
    return hitungJarak_219(tengahX, tengahY, x, y) <
    jariJari
}
```

Screenshot Program

```
C:\Users\LENOVO\Documents\Alpro 2\Anissa Fauzia Isyanti_23111022
19_Modul7>go run "c:\Users\LENOVO\Documents\Alpro 2\Anissa Fauzia Is
1 1 5
8 8 4
2 2
Titik di dalam lingkaran 1

C:\Users\LENOVO\Documents\Alpro 2\Anissa Fauzia Isyanti_2311102219_M
ded\Unguided1.go"
5 10 15
-15 4 20
0 0
Titik di dalam lingkaran 1 dan 2
```

Program ini digunakan untuk menentukan apakah sebuah titik berada di dalam salah satu atau kedua lingkaran yang koordinat pusat dan jari-arinya dimasukkan oleh pengguna. Di awal program, pengguna diminta memasukkan koordinat titik pusat (cx1, cy1, cx2, cy2 dan radius (r1 & r2) dari dua lingkaran, serta koordinat dari titik sembarang yang akan diperiksa. Program akan menghitung jarak antara titik yang diberikan dengan titik pusat lingkaran menggunakan fungsi `hitungJarak_219`, program akan memeriksa apakah jarak tersebut lebih kecil dari radius lingkaran dengan fungsi `titikdalamlingkaran`. Jika jarak lebih kecil, maka titik berada dalam lingkaran. Program menggunakan struktur if-else untuk mencetak pesan yang sesuai untuk menunjukkan posisi titik .

2. Source Code

```
package main

import (
    "fmt"
    "math"
)

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

    //Inisialisasi array
    array := make([]int, N)

    //Mengisi array dengan nilai dari pengguna
```

```

for i := 0; i < N; i++ {
    fmt.Printf("Masukkan elemen ke-%d: ", i)
    fmt.Scan(&array[i])
}

//Menampilkan keseluruhan isi dari array
fmt.Println("\nIsi array:")
fmt.Println(array)

//Menampilkan elemen dengan indeks ganjil
fmt.Println("\nElemen dengan indeks ganjil:")
for i := 1; i < N; i += 2 {
    fmt.Print(array[i], " ")
}

//Menampilkan elemen dengan indeks genap
fmt.Println("\n\nElemen dengan indeks genap:")
for i := 0; i < N; i += 2 {
    fmt.Print(array[i], " ")
}

//Menampilkan elemen dengan indeks kelipatan
bilangan x
var x int
fmt.Print("\n\nMasukkan bilangan x: ")
fmt.Scan(&x)
fmt.Println("Elemen dengan indeks kelipatan", x,
":")
for i := 0; i < N; i++ {
    if i%x == 0 {
        fmt.Print(array[i], " ")
    }
}

//Menghapus elemen pada indeks tertentu
var indeksHapus int
fmt.Print("\n\nMasukkan indeks yang akan
dihapus: ")
fmt.Scan(&indeksHapus)
array = append(array[:indeksHapus],
array[indeksHapus+1:]...)
fmt.Println("Array setelah dihapus elemen pada
indeks", indeksHapus, ":")
fmt.Println(array)

//Menghitung dan menampilkan rata-rata
var total int
for _, value := range array {
    total += value
}
rataRata := float64(total) / float64(len(array))

```

```

        fmt.Printf("\nRata-rata: %.2f\n", rataRata)

        //Menghitung dan menampilkan standar deviasi
        var variance float64
        for _, value := range array {
            variance += math.Pow(float64(value) -
rataRata, 2)
        }
        variance /= float64(len(array))
        standarDeviasi := math.Sqrt(variance)
        fmt.Printf("Standar deviasi: %.2f\n",
standarDeviasi)

        //Menampilkan frekuensi dari suatu bilangan
        tertentu
        var bilangan int
        fmt.Print("\nMasukkan bilangan yang frekuensinya
ingin dihitung: ")
        fmt.Scan(&bilangan)
        frekuensi := 0
        for _, value := range array {
            if value == bilangan {
                frekuensi++
            }
        }
        fmt.Printf("Frekuensi bilangan %d: %d\n",
bilangan, frekuensi)
    }
}

```

Screenshot Program

```
C:\Users\LENOVO\Documents\Alpro 2\Anissa Fauzia Isyanti_2311102219
19_Modul7>go run "c:\Users\LENOVO\Documents\Alpro 2\Anissa FauziIs
a Isyanti_2311102219_Modul7\Ungided\Unguided2.go"
Masukkan jumlah elemen array: 12
Masukkan elemen ke-0: 15
Masukkan elemen ke-1: 33
Masukkan elemen ke-2: 20
Masukkan elemen ke-3: 22
Masukkan elemen ke-4: 28
Masukkan elemen ke-5: 15
Masukkan elemen ke-6: 13
Masukkan elemen ke-7: 30
Masukkan elemen ke-8: 31
Masukkan elemen ke-9: 12
Masukkan elemen ke-10: 21
Masukkan elemen ke-11: 17

Isi array:
[15 33 20 22 28 15 13 30 31 12 21 17]

Elemen dengan indeks ganjil:
33 22 15 30 12 17
Masukkan elemen ke-7: 30
Masukkan elemen ke-8: 31
Masukkan elemen ke-9: 12
Masukkan elemen ke-10: 21
Masukkan elemen ke-11: 17

Isi array:
[15 33 20 22 28 15 13 30 31 12 21 17]

Elemen dengan indeks ganjil:
33 22 15 30 12 17
Masukkan elemen ke-11: 17

Isi array:
[15 33 20 22 28 15 13 30 31 12 21 17]

Elemen dengan indeks ganjil:
33 22 15 30 12 17

Elemen dengan indeks ganjil:
33 22 15 30 12 17

Elemen dengan indeks genap:
15 20 28 13 31 21

Masukkan bilangan x: 4
15 20 28 13 31 21

Masukkan bilangan x: 4
Elemen dengan indeks kelipatan 4 :
15 28 31

Masukkan indeks yang akan dihapus: 4
Array setelah dihapus elemen pada indeks 4 :
[15 33 20 22 15 13 30 31 12 21 17]

Rata-rata: 20.82
Standar deviasi: 7.13

Masukkan bilangan yang frekuensinya ingin dihitung: 15
Frekuensi bilangan 15: 2
```

Program ini mengimplementasikan serangkaian operasi pada. Pertama, program meminta pengguna menentukan jumlah elemen dan memasukkan

nilainya satu per satu. Program menampilkan isi array secara penuh, elemen dengan indeks ganjil, indeks genap, dan elemen pada indeks kelipatan dari suatu bilangan x yang dimasukkan pengguna, menghapus elemen array pada indeks tertentu, menghitung rata-rata, menghitung standar deviasi atau simpangan baku, serta menampilkan frekuensi kemunculan bilangan di dalam array.

3. Source Code

```
package main

import (
    "fmt"
)

func main() {
    var klubA, klubB string
    var skorA, skorB int
    var pemenang []string // Array untuk menyimpan
    hasil pemenang setiap pertandingan

    // Input nama klub yang bertanding
    fmt.Print("Klub A : ")
    fmt.Scanln(&klubA)
    fmt.Print("Klub B : ")
    fmt.Scanln(&klubB)

    //Input skor untuk setiap pertandingan
    pertandingan := 1
    for {
        fmt.Printf("Pertandingan      %d      :      ",
    pertandingan)
        fmt.Scan(&skorA, &skorB)
```

```

        //Memeriksa validitas skor, jika negatif
maka berhenti
        if skorA < 0 || skorB < 0 {
            break
        }

        //Menyimpan hasil pertandingan dalam array
`pemenang`
        if skorA > skorB {
            pemenang = append(pemenang, klubA)
        } else if skorB > skorA {
            pemenang = append(pemenang, klubB)
        } else {
            pemenang = append(pemenang, "Draw")
        }

        pertandingan++
    }

    //Menampilkan daftar hasil pertandingan setelah
semua input selesai
    fmt.Println("\nHasil Pertandingan:")
    for i, hasil := range pemenang {
        if hasil != "Draw" {
            fmt.Printf("Hasil  %d  :  %s\n", i+1,
hasil)
        } else {
            fmt.Printf("Hasil %d : Draw\n", i+1)
        }
    }

    //Menampilkan keterangan bahwa pertandingan
sudah selesai
    fmt.Println("Pertandingan selesai")

```

```
}
```

Screenshot Program

```
C:\Users\LENOVO\Documents\Alpro 2\Anissa Fauzia Isyanti_231110
ers\LENOVO\Documents\Alpro 2\Anissa Fauzia Isyanti_231110
go"
Klub A : MU
Klub B : Inter
Pertandingan 1 : 2 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 Pertandingan:
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
```

Program ini merekap hasil pertandingan antara dua klub sepak bola yang ditentukan pengguna. Pertama, pengguna memasukkan nama kedua klub, lalu memasukkan skor untuk setiap pertandingan yang dilakukan secara berulang. Jika skor salah satu klub bernilai negatif, proses input skor dihentikan. Berdasarkan skor yang dimasukkan, program menentukan pemenang untuk setiap pertandingan dan menyimpannya dalam array. Jika skor kedua klub sama, hasilnya ditandai sebagai "Draw". Setelah semua data terkumpul, program menampilkan hasil setiap pertandingan dan mencetak pemenangnya atau jika berakhir seri. Di akhir, program menampilkan keterangan "Pertandingan selesai" untuk menandakan akhir proses.

4. Source Code

```
package main

import (
    "fmt"
)

const NMAX int = 127

type tabel [NMAX]rune

//Fungsi untuk mengisi array dengan karakter hingga
titik ditemukan atau mencapai batas NMAX
func isiArray(t *tabel, n *int) {
    var input rune
    *n = 0

    for *n < NMAX {
        fmt.Scanf("%c", &input)
        if input == '.' {
            break
        }
        t[*n] = input
        *n++
    }
}

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

```

//Fungsi 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 memeriksa apakah array merupakan
palindrom
func palindrom(t tabel, n int) bool {
    var reversed tabel
    copy(reversed[:], t[:n])
    balikanArray(&reversed, n)

    for i := 0; i < n; i++ {
        if t[i] != reversed[i] {
            return false
        }
    }
    return true
}

func main() {
    var tab tabel
    var m int

    fmt.Print("Teks: ")
    isiArray(&tab, &m)

    fmt.Print("Teks: ")
    cetakArray(tab, m)

    isPalindrome := palindrom(tab, m)

```

```
        fmt.Printf("palindrom? %t\n", isPalindrome)
    }
```

Screenshot Program

```
C:\Users\LENOVO\Documents\Alpro 2\Anissa Fauzia Isyanti_2311102219
Teks: SENANG.
Teks: SENANG
Reverse teks: GNANES
```

Setelah Modifikasi

```
C:\Users\LENOVO\Documents\Alpro 2\Anissa Fauzia Isyanti_2311102219
ded\Unguided4MOD.go"
Teks: KATAK.
Teks: KATAK
palindrom? true
```

Program ini merupakan program untuk mengecek palindrom kata. Program ini menerima input karakter dari pengguna, membalik urutan karakter tersebut, dan memeriksa apakah karakter yang dimasukkan adalah palindrom (sama jika dibaca dari depan dan belakang). Program mengisi array dengan karakter hingga titik (.) ditemukan, kemudian mencetak karakter yang dimasukkan. Setelah itu, program membalik urutan karakter dan memeriksa apakah urutan tersebut membentuk palindrom dengan membandingkan array asli dan yang dibalik.