

**LAPORAN PRAKTIKUM  
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL 7  
STRUCT & ARRAY**



**Universitas  
Telkom**

Oleh:

MUHAMMAD DAFFA AL FAIZ

2311102237

IF-11-02

**S1 TEKNIK INFORMATIKA  
TELKOM UNIVERSITY PURWOKERTO**

**2024**

# I. DASAR TEORI

## 7.1 Tipe Bentuk

Tipe bentuk memungkinkan pemrograman untuk mendefinisikan suatu tipe data baru pada suatu bahasa pemrograman. Tipe bentuk ini dapat dibedakan atas dua jenis, yaitu Alias dan Struct.

### 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>	type <nama alias> <tipe data>
3		
4	algoritma	func main(){
5	...	... }
6		
7		

Sebagai contoh perhatikan program Go berikut beserta hasil eksekusinya!

```
1 package main
2 import "fmt"
3 type bilangan int
4 type pecahan float64
5 func main(){
6     var a,b bilangan
7     var hasil pecahan
8     a = 9
9     b = 5
10    hasil = pecahan(a) / pecahan(b)
11    fmt.Println(hasil)
12 }
```

```
E:\DEV\GO>go build Demo.go
E:\DEV\GO> Demo.exe
1.8
```

### 2) Struct 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.

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

Berbeda dengan bahasa pemrograman yang 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.

```

1 package main
2 import "fmt"
3 type waktu struct {
4     jam, menit, detik int
5 }
6
7 func main(){
8     var wParkir, wPulang, durasi waktu
9     var dParkir, dPulang, lParkir int
10    fmt.Scan(&wParkir.jam, &wParkir.menit, &wParkir.detik)
11    fmt.Scan(&wPulang.jam, &wPulang.menit, &wPulang.detik)
12    dParkir = wParkir.detik + wParkir.menit*60 + wParkir.jam*3600
13    dPulang = wPulang.detik + wPulang.menit*60 + wPulang.jam*3600
14    lParkir = dPulang - dParkir
15    durasi.jam = lParkir / 3600
16    durasi.menit = lParkir % 3600 / 60
17    durasi.detik = lParkir % 3600 % 60
18    fmt.Printf("Lama parkir: %d jam %d menit %d detik",
19        durasi.jam, durasi.menit, durasi.detik)
20 }

```

```

E:\DEV\GO>go build Demo.go
E:\DEV\GO> Demo.exe
7 30 0
10 45 15
Lama parkir: 3 jam 15 menit 15 detik

```

## 7.2 Array

Array mempunyai ukuran jumlah elemen) yang tetap (statls) 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	)

Jumlah elemen array dapat diminta dengan fungsi **len** yang tersedia. Sebagai contoh **len(arr)** akan menghasilkan 73 untuk contoh di atas.

Indeks array dimulai dari **0**, sehingga indeks arr pada contoh adalah **0, 1..**

**len(arr)-1**

Contoh:

```

1 // Mengganti isi elemen ke-0 dengan nilai dari elemen ke-7
2 arr[0] = arr[7]
3
4 // Mengambil data field x dari elemen ke-i
5 currX = arr[i].center.x
6
7 // Mengambil elemen terakhir
8 n := len(arr)
9 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.

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]

```

## 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"

```

---

Halaman 64 | Modul Praktikum Algoritma dan Pemrograman 2

```

18
19 // Menghapus entri dengan kunci "john"
20 delete(dct1, "john")

```

## II. GUIDED

### GUIDED I

#### 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(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)

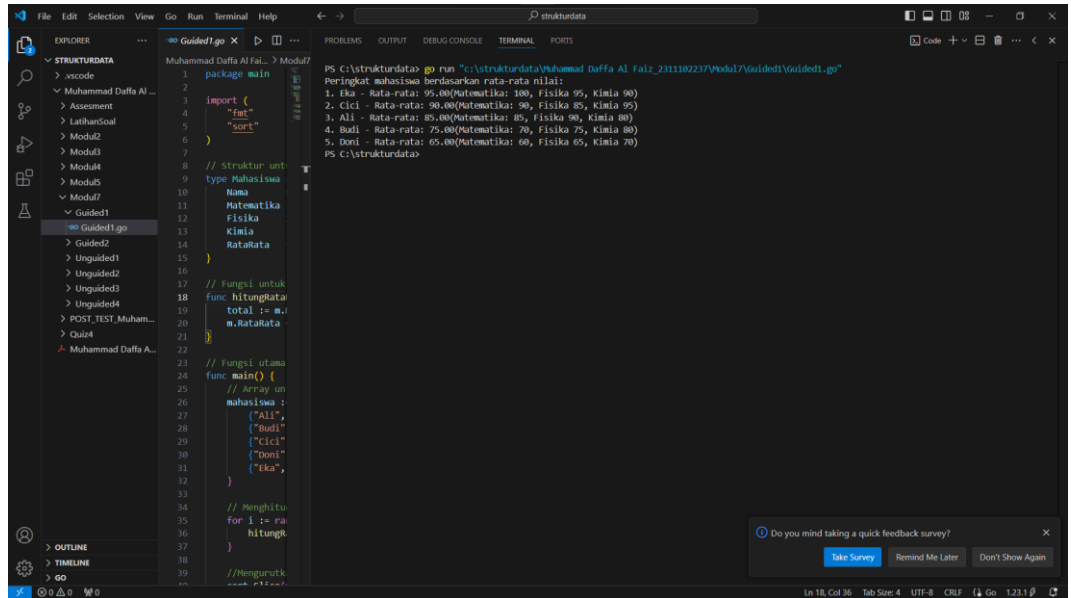
    }

}
```



```
}
```

## Screenshot



## Deskripsi:

Program ini adalah program aplikasi sederhana untuk mengelola dan mengurutka data mahasiswa berdasarkan rata-rata nilai mereka.

## GUIDED II

### 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

The screenshot shows a VS Code editor with a Go file named `Guided2.go` open. The code defines a map of students and performs operations like adding, deleting, and displaying them. The terminal output shows the initial list of students, the deletion of a student with NIM 20231003, and the final list of students after deletion.

```

PS C:\strukturdata> go run "c:\strukturdata\Muhammad Daffa Al Faiz_231102237\Modul2\Guided2\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
PS C:\strukturdata>

```

## Deskripsi:

Program ini adalah program aplikasi sederhana untuk mengelola data mahasiswa menggunakan map.

### **III. UNGUIDED**

#### **UNGUIDED I**

##### **Source Code**

```
package main

import (
    "fmt"
    "math"
)

type Titik struct {
    x, y float64
}

type Lingkaran struct {
```

```
    titik Titik

    radius float64
}

func jarak(a, b Titik) float64 {
    return math.Sqrt(math.Pow(a.x-b.x, 2) + math.Pow(a.y-
b.y, 2))
}

func diDalamLingkaran(t Titik, l Lingkaran) bool {
    return jarak(t, l.titik) <= l.radius
}

func main() {
    var l1, l2 Lingkaran
    var t Titik

    fmt.Print("Masukkan: ")
    fmt.Scan(&l1.titik.x, &l1.titik.y, &l1.radius)

    fmt.Print("Masukkan: ")
    fmt.Scan(&l2.titik.x, &l2.titik.y, &l2.radius)

    fmt.Print("Masukkan: ")
    fmt.Scan(&t.x, &t.y)

    dalamL1 := diDalamLingkaran(t, l1)
    dalamL2 := diDalamLingkaran(t, l2)
```

```

        if dalamL1 && dalamL2 {

            fmt.Println("Titik di dalam lingkaran 1 dan 2")

        } else if dalamL1 {

            fmt.Println("Titik di dalam lingkaran 1")

        } else if dalamL2 {

            fmt.Println("Titik di dalam lingkaran 2")

        } else {

            fmt.Println("Titik di luar lingkaran 1 dan 2")

        }

    }
}

```

## Screenshot

The screenshot shows a Go IDE with the following components:

- EXPLORER:** A tree view on the left showing the project structure. The file `Unguided1.go` is selected.
- EDITOR:** The main window displays the source code of `Unguided1.go`. The code defines a `type Lingkaran` struct with `x` and `y` coordinates, and a `func jarak(a, b)` function to calculate the distance between two points. The `func main()` function uses `fmt.Scan` to read input coordinates and radii, and then uses `if` statements to determine if a point is inside one or both circles.
- TERMINAL:** The bottom right pane shows the output of running the program. It displays the prompts "Masukkan: 1 1 5" and "Masukkan: 8 8 4", followed by the output "Titik di dalam lingkaran 1". Subsequent runs show different inputs and outputs, including "Titik di dalam lingkaran 2" and "Titik di dalam lingkaran 1 dan 2".

## Deskripsi:

Program ini adalah program sebuah aplikasi sederhana untuk menentukan apakah suatu titik berada di dalam salah satu atau kedua lingkaran yang didefinisikan.

## UNGUIDED II

### Source Code

```
package main

import (
    "fmt"
    "math"
)

func main() {
    var N int

    fmt.Print("Masukkan jumlah elemen array: ")
    fmt.Scan(&N)

    arr := make([]int, N)

    fmt.Println("\nMasukkan nilai-nilai array:")
    for i := 0; i < N; i++ {
        fmt.Printf("Elemen ke-%d: ", i+1)
        fmt.Scan(&arr[i])
    }

    fmt.Println("\nSeluruh isi array:")
    for _, v := range arr {
        fmt.Print(v, " ")
    }
}
```

```
fmt.Println()

fmt.Println("\nElemen-elemen array dengan indeks
ganjil:")

for i := 1; i < N; i += 2 {
    fmt.Print(arr[i], " ")
}

fmt.Println()

fmt.Println("\nElemen-elemen array dengan indeks
genap:")

for i := 0; i < N; i += 2 {
    fmt.Print(arr[i], " ")
}

fmt.Println()

var x int

fmt.Print("\nMasukkan nilai x: ")

fmt.Scan(&x)

fmt.Printf("Elemen-elemen array dengan indeks
kelipatan %d: \n", x)

for i := 0; i < N; i += x {
    fmt.Print(arr[i], " ")
}

fmt.Println()

var hapusElemen int

fmt.Print("\nMasukkan indeks yang ingin dihapus: ")

fmt.Scan(&hapusElemen)
```



```
        if hapusElemen >= 0 && hapusElemen < len(arr) {

            arr = append(arr[:hapusElemen],
arr[hapusElemen+1:]...)

            fmt.Println("Array setelah penghapusan:")

            for _, v := range arr {

                fmt.Print(v, " ")

            }

            fmt.Println()

        } else {

            fmt.Println("Indeks tidak valid")

        }

    }

    var sum float64

    for _, v := range arr {

        sum += float64(v)

    }

    rataRata := sum / float64(len(arr))

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

    var sumKuadrat float64

    for _, v := range arr {

        sumKuadrat += float64(v * v)

    }

    varian := (sumKuadrat / float64(len(arr))) -
math.Pow(rataRata, 2)

    standarDeviasi := math.Sqrt(varian)

    fmt.Printf("Standar deviasi: %.2f\n", standarDeviasi)
```

```

var bilangan int

    fmt.Print("\nMasukkan bilangan yang ingin dicari
frekuensinya: ")

    fmt.Scan(&bilangan)

var frekuensi int

for _, v := range arr {
    if v == bilangan {
        frekuensi++
    }
}

fmt.Printf("Frekuensi %d: %d\n", bilangan, frekuensi)
}

```

## Screenshot

```

PS C:\strukturdata> go run "C:\strukturdata\Muhamad Daffa Al Faiz_2311102237\Muhamad7\Unguided2\Unguided2.go"
Masukkan jumlah elemen array: 5
Masukkan nilai-nilai array:
Elemen ke-1: 1
Elemen ke-2: 2
Elemen ke-3: 4
Elemen ke-4: 5
Elemen ke-5: 7
Seluruh isi array:
1 2 4 5 7
Elemen-elemen array dengan indeks ganjil:
2 5
Elemen-elemen array dengan indeks genap:
1 4 7
Masukkan nilai x: 4
Elemen-elemen array dengan indeks kelipatan 4:
1 7
Masukkan indeks yang ingin dihapus: 3
Array setelah penghapusan:
1 2 4 7
Rata-rata: 3.50
Standar deviasi: 2.29
Masukkan bilangan yang ingin dicari frekuensinya: 5
Frekuensi 5: 0
PS C:\strukturdata>

```

## Deskripsi:

Program ini adalah program sebuah aplikasi sederhana yang menginginkan pengguna untuk mengelola sebuah array bilangan bulat.

## UNGUIDED III

### Source Code

```
package main

import "fmt"

type nama struct {
    Nama string
    Skor int
}

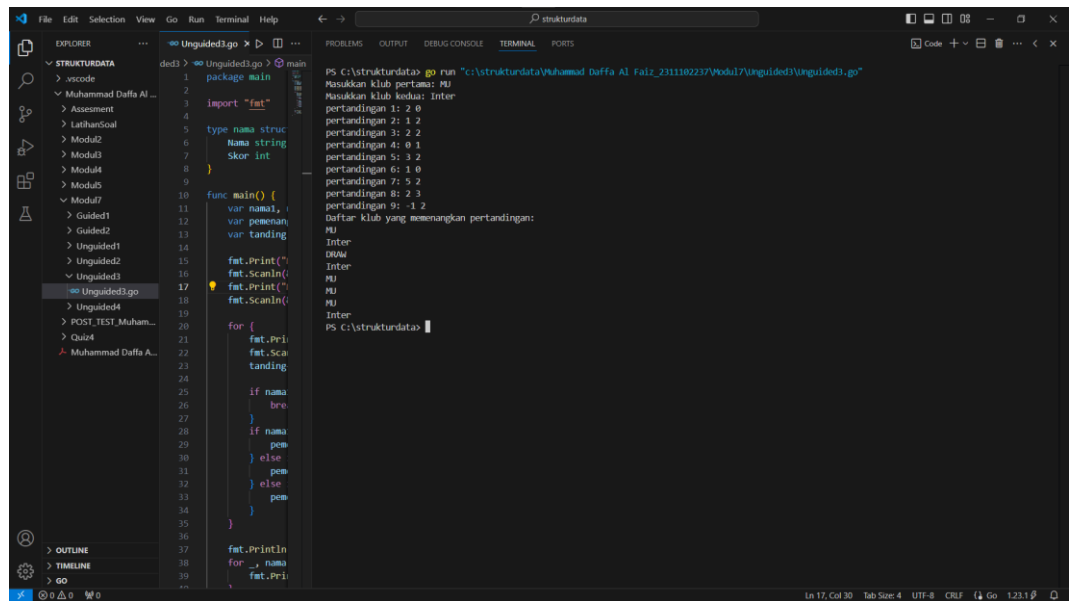
func main() {
    var nama1, nama2 nama
    var pemenang []string
    var tanding int = 1

    fmt.Print("Masukkan klub pertama: ")
    fmt.Scanln(&nama1.Nama)
    fmt.Print("Masukkan klub kedua: ")
    fmt.Scanln(&nama2.Nama)

    for {
        fmt.Printf("pertandingan %v: ", tanding)
        fmt.Scanln(&nama1.Skor, &nama2.Skor)
        tanding++
    }
}
```

```
        if nama1.Skor < 0 || nama2.Skor < 0 {  
            break  
        }  
        if nama1.Skor > nama2.Skor {  
            pemenang = append(pemenang, nama1>Nama)  
        } else if nama2.Skor == nama1.Skor {  
            pemenang = append(pemenang, "DRAW")  
        } else if nama2.Skor > nama1.Skor {  
            pemenang = append(pemenang, nama2>Nama)  
        }  
    }  
  
    fmt.Println("Daftar klub yang memenangkan  
pertandingan:")  
    for _, nama := range pemenang {  
        fmt.Println(nama)  
    }  
}
```

## Screenshot



## Deskripsi:

Program ini adalah program untuk mencatat pemenang dalam serangkaian pertandingan antar dua klub sepak bola.

## UNGUIDED IV

### Source Code

```
package main

import "fmt"

func main() {

    var tab tabel

    var m int

    isiArray(&tab, &m)
```

```

        cetakArrayAndCheckPalindrome(tab, m)
    }

    const NMAX int = 127

    type tabel [NMAX]rune

    func isiArray(t *tabel, n *int) {
        var char rune

        *n = 0

        fmt.Print("Masukkan karakter: ")

        for *n < NMAX {
            fmt.Scanf("%c", &char)

            if char == '.' {
                break
            }

            if char != '\n' {
                t[*n] = char

                *n++
            }
        }
    }

    func balikanArray(t tabel, n int) tabel {
        var hasil tabel
    }

```

```

        for i := 0; i < n; i++ {

            hasil[i] = t[n-1-i]

        }

        return hasil
    }

func palindrom(t tabel, n int) bool {

    for i := 0; i < n/2; i++ {

        if t[i] != t[n-1-i] {

            return false

        }

    }

    return true

}

func cetakArrayAndCheckPalindrome(t tabel, n int) {

    var balik tabel = balikanArray(t, n)

    isPalindrome := palindrom(t, n)

    fmt.Print("Teks\t\t:")

    for i := 0; i < n; i++ {

        fmt.Printf(" %c", t[i])

    }

    fmt.Println()

    fmt.Print("Reverse teks\t:")

    for i := 0; i < n; i++ {

```

```

        fmt.Printf(" %c", balik[i])

    }

    fmt.Println()

    fmt.Print("Palindrom\t? ")

    if isPalindrome {

        fmt.Println("true")

    } else {

        fmt.Println("false")

    }

}

```

## Screenshot

The screenshot shows a Go IDE with the following components:

- EXPLORER:** A file tree on the left showing a project structure with folders like 'STRUKTURDATA', 'Assesment', 'LatharSoal', and several 'Modul' files. The file 'Unguided4.go' is selected.
- EDITOR:** The main window displays the source code of 'Unguided4.go'. The code includes a 'func main()' function that prompts the user for input, checks if it's a palindrome using the 'isPalindrome' function, and prints the result. There is also a 'func balikkanKer...' function for reversing the string.
- TERMINAL:** The bottom right pane shows the output of running the program. It displays the prompt 'Masukkan karakter: S E N A N G', the input 'S E N A N G', and the output 'Palindrom ? false'. A second run shows the input 'K A T A K' and the output 'Palindrom ? true'.

## Deskripsi:

Program ini adalah program untuk memeriksa apakah rangkaian karakter yang dimasukan oleh pengguna merupakan palindrom.