

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL VII
STRUCK & ARRAY**



Disusun Oleh :

Deshan Rafif Alfarisi / 2311102326

S1-IF-11-06

Dosen Pengampu :

Abednego Dwi Septiadi S.Kom., M.Kom.

PROGRAM STUDI S1 TEKNIK INFORMATIKA

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2024

I. DASAR TEORI

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>	
3		type <nama alias> <tipe data>
4	algoritma	
5		func main(){
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. Strum 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.

	Notas1 A1gor1tma	Notas1 dalam bahasa Go
1	kamus	package main
2	type mama struct»«	type mama 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 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 st ruct (
4     jam, menit, detik int
5 }
6
7 func main(){
8     var wPark1r, 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

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.

2.

	Notas1 dalam bahasa Go
J	var (
2	// array arr mempunyai 73 elemen, masing-masing bertipe CircType2
3	arr [73]CircType
4	
5	// array buf dengan 5 elemen, dengan n11a1 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	
17	// 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 // declaring s101 as a slice
4 var s101 = []int{ 11, 2, 3, 5, 7, 13 }
5
```

Sebuah slice dapat diprealokasi menggunakan fungsi built-in make

```
1 // Prealokasi 10 elemen untuk s102 dan sejumlah tempat tambahan
2 var s102 []int = make([]int, 10, 20)
3
4 // Prealokasi 7 elemen untuk s103 tanpa tempat tambahan
5 var s103 []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 s102
2 fmt.Println( len(s102), cap(s102) )
```

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 ask:l. */
2 s101 = append(s101, 17)
3 s101 = append(s101, 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 s104 = arr[:4]
3
4 // Ambil beberapa elemen terakhir, dimulai dari indeks 5
5 s105 = s101[5:]
6
7 // Salin semua dari slice/array aslinya
8 s106 = s105[:]
9
10 // Salin element dari indeks 3 sampai, tapi tidak termasuk, 5.
11 // Jadi dalam contoh hanya 2 elemen s106[3] dan s106[4] yang disalin
12 s107 = s106[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 dct7 dari elemen string dengan kunci juga string
5 // Mempunyai nilai awal dct7["john"] = "hi", dct7["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 // Flenghapus entri dengan kunci: "j ohn"
20 defet e(dct1, "j ohn")
```

II. GUIDED

1. Suatu lingkaran didefinisikan dengan koordinat titik pusat (cx, cy) dengan radius r . Apabila diberikan dua buah lingkaran, maka tentukan posisi sebuah titik sembarang (z, 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

```
package main

import (
    "fmt"
    "math"
)

// Struktur untuk menyimpan titik dengan koordinat (x, y)
type Titik struct {
    x int
    y int
}

// Struktur untuk menyimpan lingkaran dengan pusat dan radius
type Lingkaran struct {
    pusat Titik
    radius int
}

// Fungsi untuk menghitung jarak antara dua titik
func hitungJarak(a, b Titik) float64 {
```

```

        return math.Sqrt(float64((a.x-b.x)*(a.x-b.x) + (a.y-
b.y)*(a.y-b.y)))
    }

    // Fungsi untuk memeriksa apakah titik berada di dalam
    lingkaran
    func titikDalamLingkaran(t Titik, l Lingkaran) bool {
        jarak := hitungJarak(t, l.pusat)
        return jarak <= float64(l.radius)
    }

    func main() {
        // Input untuk lingkaran 1
        var cx1, cy1, r1 int
        fmt.Print("Masukkan koordinat pusat dan radius lingkaran
1 (cx1 cy1 r1): ")
        fmt.Scanln(&cx1, &cy1, &r1)
        lingkaran1 := Lingkaran{pusat: Titik{x: cx1, y: cy1},
radius: r1}

        // Input untuk lingkaran 2
        var cx2, cy2, r2 int
        fmt.Print("Masukkan koordinat pusat dan radius lingkaran
2 (cx2 cy2 r2): ")
        fmt.Scanln(&cx2, &cy2, &r2)
        lingkaran2 := Lingkaran{pusat: Titik{x: cx2, y: cy2},
radius: r2}

        // Input untuk titik sembarang
        var x, y int
        fmt.Print("Masukkan koordinat titik sembarang (x y): ")
        fmt.Scanln(&x, &y)
        titik := Titik{x: x, y: y}

        // Pengecekan posisi titik
        diDalam1 := titikDalamLingkaran(titik, lingkaran1)
        diDalam2 := titikDalamLingkaran(titik, lingkaran2)

        // Menampilkan hasil sesuai kondisi
        if diDalam1 && diDalam2 {
            fmt.Println("Titik berada di dalam lingkaran 1 dan
2")
        } else if diDalam1 {
            fmt.Println("Titik berada di dalam lingkaran 1")
        } else if diDalam2 {

```



```
        fmt.Println("Titik berada di dalam lingkaran 2")
    } else {
        fmt.Println("Titik berada di luar kedua lingkaran")
    }
}
```

Screenshot Output

```
Frekuensi bilangan 4 di dalam array: 0
PS C:\Users\Lenovo\Documents\file kuliah\Laprak Alpro 2\modul 7> go run "c:\Users\Lenovo\Documents\file kuliah\Laprak Alpro 2\modul 7\tempCodeRunnerFile.go"
Masukkan koordinat pusat dan radius lingkaran 1 (cx1 cy1 r1): 1 1 5
Masukkan koordinat pusat dan radius lingkaran 2 (cx2 cy2 r2): 8 8 4
Masukkan koordinat titik sembarang (x y): 2 2
Titik berada di dalam lingkaran 1
PS C:\Users\Lenovo\Documents\file kuliah\Laprak Alpro 2\modul 7>
```

Deskripsi Program

Program ini untuk memeriksa apakah sebuah titik berada di dalam satu atau kedua lingkaran yang diberikan pengguna. Program ini meminta pengguna untuk memasukkan koordinat pusat dan jari-jari dari dua lingkaran, serta koordinat dari sebuah titik. Kemudian, program menghitung jarak antara titik tersebut dengan pusat masing-masing lingkaran. Jika jarak lebih kecil atau sama dengan jari-jari lingkaran, maka titik tersebut dianggap berada di dalam lingkaran. Terakhir, program akan menampilkan pesan yang menginformasikan apakah titik tersebut berada di dalam lingkaran pertama, lingkaran kedua, keduanya, atau tidak satupun.

III. UNGUIDED

1. 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 indek 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"
    "math"
)

// Fungsi untuk menghitung rata-rata dari array
func hitungRataRata(arr []int) float64 {
    total := 0
    for _, v := range arr {
        total += v
    }
    return float64(total) / float64(len(arr))
}

// Fungsi untuk menghitung standar deviasi dari array
func hitungStandarDeviasi(arr []int, rataRata float64)
float64 {
    total := 0.0
```

```

    for _, v := range arr {
        total += math.Pow(float64(v)-rataRata, 2)
    }
    return math.Sqrt(total / float64(len(arr)))
}

// Fungsi untuk menghapus elemen pada indeks tertentu
func hapusElemen(arr []int, index int) []int {
    return append(arr[:index], arr[index+1:]...)
}

// Fungsi untuk menghitung frekuensi bilangan dalam array
func hitungFrekuensi(arr []int, bilangan int) int {
    count := 0
    for _, v := range arr {
        if v == bilangan {
            count++
        }
    }
    return count
}

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

    // Membuat array dengan kapasitas tertentu
    arr := make([]int, n)

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

    // a. Menampilkan keseluruhan isi dari array
    fmt.Println("\nKeseluruhan isi array:", arr)

    // b. Menampilkan elemen-elemen array dengan indeks ganjil
    fmt.Println("Elemen dengan indeks ganjil: ")
    for i := 1; i < len(arr); i += 2 {

```

```

        fmt.Print(arr[i], " ")
    }
    fmt.Println()

    // c. Menampilkan elemen-elemen array dengan indeks
    genap
    fmt.Print("Elemen dengan indeks genap: ")
    for i := 0; i < len(arr); i += 2 {
        fmt.Print(arr[i], " ")
    }
    fmt.Println()

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

    // e. Menghapus elemen array pada indeks tertentu
    var index int
    fmt.Print("\nMasukkan indeks untuk menghapus elemen: ")
    fmt.Scan(&index)
    if index >= 0 && index < len(arr) {
        arr = hapusElemen(arr, index)
        fmt.Println("Isi array setelah elemen dihapus:",
arr)
    } else {
        fmt.Println("Indeks tidak valid!")
    }

    // f. Menampilkan rata-rata dari bilangan yang ada di
    dalam array
    rataRata := hitungRataRata(arr)
    fmt.Printf("\nRata-rata dari array: %.2f\n", rataRata)

    // g. Menampilkan standar deviasi dari bilangan yang ada
    di dalam array
    stdDeviasi := hitungStandarDeviasi(arr, rataRata)

```

```

    fmt.Printf("Standar deviasi dari array: %.2f\n",
stdDeviasi)

    // h. Menampilkan frekuensi dari suatu bilangan tertentu
    di dalam array
    var bilangan int
    fmt.Print("\nMasukkan bilangan untuk mencari
frekuensinya: ")
    fmt.Scan(&bilangan)
    frekuensi := hitungFrekuensi(arr, bilangan)
    fmt.Printf("Frekuensi bilangan %d di dalam array: %d\n",
bilangan, frekuensi)
}

```

Screenshoot Output

```

o\Documents\file kuliah\Laprak Alpro 2\modul 7\unguided1.go"
Masukkan jumlah elemen array: 4
Masukkan elemen array:
Elemen ke-0: 4
Elemen ke-1: 3
Elemen ke-2: 2
Elemen ke-3: 1

Keseluruhan isi array: [4 3 2 1]
Elemen dengan indeks ganjil: 3 1
Elemen dengan indeks genap: 4 2

Masukkan bilangan x untuk kelipatan indeks: 1
Elemen dengan indeks kelipatan 1: 4 3 2 1

Masukkan indeks untuk menghapus elemen: 1
Isi array setelah elemen dihapus: [4 2 1]

Rata-rata dari array: 2.33
Standar deviasi dari array: 1.25

Masukkan bilangan untuk mencari frekuensinya: 1
Frekuensi bilangan 1 di dalam array: 1
PS C:\Users\Lenovo\Documents\file kuliah\Laprak Alpro 2\modul 7>

```

Deskripsi Program

Program ini melakukan berbagai operasi pada sebuah array bilangan bulat. Program ini memungkinkan pengguna untuk memasukkan elemen-elemen array, menampilkan elemen-elemen tertentu berdasarkan indeks, menghapus elemen pada indeks tertentu, menghitung rata-rata dan standar deviasi dari seluruh elemen, serta menghitung frekuensi kemunculan suatu bilangan dalam array. Program ini juga mencakup validasi input untuk memastikan pengguna memasukkan nilai yang valid.

2. 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 ber laga.

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

```
package main

import (
    "fmt"
)

func main() {
    // Meminta pengguna memasukkan nama kedua klub
    var klubA, klubB string
    fmt.Print("Klub A: ")
    fmt.Scanln(&klubA)
    fmt.Print("Klub B: ")
    fmt.Scanln(&klubB)

    // Array untuk menyimpan hasil pemenang
    var pemenang []string

    // Memulai loop untuk memasukkan skor pertandingan
    pertandingan := 1
    for {
        var skorA, skorB int
        fmt.Printf("Pertandingan %d : ", pertandingan)
        fmt.Scan(&skorA, &skorB)

        // Jika salah satu skor negatif, hentikan input
        if skorA < 0 || skorB < 0 {
            fmt.Println("Pertandingan selesai")
        }
    }
}
```

```

        break
    }

    // Menentukan pemenang atau hasil draw
    if skorA > skorB {
        fmt.Printf("// %s menang\n", klubA)
        pemenang = append(pemenang, klubA)
    } else if skorB > skorA {
        fmt.Printf("// %s menang\n", klubB)
        pemenang = append(pemenang, klubB)
    } else {
        fmt.Println("// Draw")
        pemenang = append(pemenang, "Draw")
    }

    // Naikkan nomor pertandingan
    pertandingan++
}

// Menampilkan hasil akhir
fmt.Println("\nHasil pertandingan:")
for i, hasil := range pemenang {
    fmt.Printf("Hasil %d: %s\n", i+1, hasil)
}
}

```

Screenshot Output

```

Pertandingan
5 : 3 2
// MU menang
Pertandingan
6 : 1 0
// MU menang
Pertandingan
7 : 5 2
// MU menang
Pertandingan
8 : 2 3
// Inter men
ang
Pertandingan 9 : -1 2
Pertandingan selesai

```

```

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

```

PS C:\Users\Lenovo\Documents\file kuliah\Laprak Alpro 2\modul 7> █

Deskripsi Program

Program mencatat hasil pertandingan antara dua klub. Program ini akan terus meminta pengguna memasukkan skor untuk setiap pertandingan hingga pengguna memasukkan skor negatif. Setelah itu, program akan menampilkan daftar pemenang atau hasil seri untuk setiap pertandingan yang telah dimasukkan. Program ini juga menghitung jumlah kemenangan masing-masing tim.

3. 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 Is1Array(t *tabel, n *int)
/*Z.S. Data tersedia dalam parameter masukan
F.S. Array t berisi sejumlah n karakter yang dimasukkan user,
Proses Input selama karakter bukanlah TITIK dan n <= NMAX */
```

```
func cetakArray(t tabel, n int)
/*Z.S. Terdefinisi array t yang berisi sejumlah n karakter
F.S. n karakter data array /runca di layar */

func balikArray(t *tabel, n int)
/*Z.S. Terdefinisi array t yang berisi sejumlah n karakter
F.S. Urutan array t terbalik */

func main(){
    var tab tabel
    var n :int
    // s& array tab dengan menggunakan prosedur Is1Array

    // Balikan array tab dengan menggunakan balikArray

    // Cetak array tab
```

Perhatikan sesi interaksi pada contoh berikut ini (teks bergaris bawah adalah input/read)


```
Teks      :? E N A N G .
          :G N A N E S
Rekerse teks . K U T O K
```

Modifikasi program tersebut dengan menambahkan fungsi palindrom.
Tambahkan instruksi untuk memanggil fungsi tersebut dan menampilkan hasilnya pada program utama.

"**Palindrom** adalah **teks yang** dlbaca dari awal atau akhir adalah sama, contoh:
KATAK, APA,

KASUR_RUSAK.

```
func palindrom(t tabel, n int) bool
/* ienge/rba11kan true apabila susunan karakter d1 dalam t nemöentt/k pa11ndro/r,
```

Perhatikan sesi interaksi pada contoh berikut ini (teks bergaris bawah adalah Input/read)

```
Teks      :K A ? A K
Palindrom  ?true

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

Sourcecode

```
package main

import (
    "fmt"
)

const NMAX int = 127

type tabel [NMAX]rune

// Fungsi untuk mengisi array dengan karakter
func isiArray(t *tabel, n *int) {
    var ch rune
    fmt.Println("Masukkan teks (akhiri dengan '.'): ")
    for *n = 0; *n < NMAX; *n++ {
```

```

        fmt.Scanf("%c", &ch)
        if ch == '.' { // Berhenti saat menemukan titik
            break
        }
        t[*n] = ch
    }
}

// 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 balikkanArray(t *tabel, n int) {
    for i := 0; i < n/2; i++ {
        t[i], t[n-1-i] = t[n-1-i], t[i]
    }
}

// Fungsi untuk memeriksa apakah array membentuk palindrome
func palindrome(t tabel, n int) bool {
    for i := 0; i < n/2; i++ {
        if t[i] != t[n-1-i] {
            return false
        }
    }
    return true
}

func main() {
    var tab tabel
    var n int

    // Memanggil fungsi isiArray untuk mengisi data
    isiArray(&tab, &n)

    // Mencetak isi array asli
    fmt.Print("Teks: ")
    cetakArray(tab, n)

    // Membalikkan isi array

```

```

    balikkanArray(&tab, n)
    fmt.Print("Reverse teks: ")
    cetakArray(tab, n)

    // Memeriksa apakah teks merupakan palindrome
    if palindrome(tab, n) {
        fmt.Println("Palindrome: true")
    } else {
        fmt.Println("Palindrome: false")
    }
}

```

Screenshoot Output

```

Masukkan teks (akhiri dengan '.'):
SENANG.
Teks: SENANG
Reverse teks: GNANES
Palindrome: false
PS C:\Users\Lenovo\Documents\file kuliah\Laprak Alpro 2\modul 7>

```

Deskripsi Program

Program ini untuk memeriksa apakah suatu teks merupakan palindrome. Program ini meminta pengguna untuk memasukkan sebuah teks, kemudian membalik teks tersebut, dan selanjutnya memeriksa apakah teks asli dan teks yang dibalikkan adalah sama. Jika sama, maka teks tersebut adalah palindrome. Program ini juga menampilkan teks asli, teks yang dibalikkan, dan hasil pengecekan palindrome.