

LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2
MODUL 7
STRUCK & ARRAY



Oleh :

FAHRUR RIZQI

2311102059

S1 IF-11-02

S1 TEKNIK INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2024

I. DASAR TEORI

1. Tipe Bentuk (Structs and Aliases)

1.1. Tipe Alias

Tipe alias memungkinkan programmer untuk mendefinisikan nama baru untuk tipe data yang sudah ada, sehingga lebih mudah dan lebih intuitif untuk digunakan. Misalnya, dalam bahasa pemrograman Go, kita dapat menggunakan kata kunci `type` untuk membuat alias. Contoh:

1. `type` bilangan `int`
2. `type` pecahan `float64`

Dengan cara ini, kita dapat menggunakan bilangan dan pecahan sebagai tipe data baru yang lebih mudah diingat dan digunakan dalam kode.

1.2. Struct

Struct adalah tipe bentuk yang memungkinkan programmer untuk mengelompokkan beberapa data yang memiliki relasi menjadi satu kesatuan. Setiap data dalam struct disimpan dalam field yang berbeda. Contoh definisi struct dalam Go adalah sebagai berikut:

1. `type` waktu struct {
2. jam `int`
3. menit `int`
4. detik `int`
5. }

Dalam contoh ini, waktu adalah struct yang memiliki tiga field: jam, menit, dan detik, semuanya bertipe `int`.

1.3. Kesamaan Tipe

Dalam Go, dua variabel dengan struktur yang sama (nama field dan tipe field yang sama) dianggap memiliki tipe yang sama, meskipun nama struct-nya berbeda. Ini memudahkan penggunaan struct dalam program.

2. Array

2.1. Definisi Array

Array adalah kumpulan elemen dengan tipe yang sama, di mana ukuran (jumlah elemen) ditentukan saat deklarasi dan bersifat tetap selama eksekusi program. Contoh deklarasi array dalam Go:

```
1. var arr [73]CircType
```

Di sini, arr adalah array dengan 73 elemen yang bertipe CircType.

2.2. Mengakses Elemen Array

Indeks array dimulai dari 0, sehingga elemen pertama dapat diakses dengan arr[0], dan elemen terakhir dengan arr[len(arr)-1]. Fungsi len dapat digunakan untuk mendapatkan jumlah elemen dalam array.

2.3. Slice

Slice adalah array dinamik yang dapat berubah ukuran. Slice dapat dideklarasikan tanpa menentukan jumlah elemen, dan dapat diperluas menggunakan fungsi built-in append.

Contoh deklarasi slice:

```
1. var s []int
2. s = append(s, 1, 2, 3)
```

Fungsi len dan cap dapat digunakan untuk mengetahui ukuran dan kapasitas slice.

2.4. Map

Map adalah tipe data yang menyimpan pasangan kunci-nilai, di mana kunci tidak harus berupa integer. Contoh deklarasi map dalam Go:

```
1. var dct map[string]int
2. dct = make(map[string]int)
3. dct["john"] = 1
```

Map memungkinkan akses cepat ke nilai berdasarkan kunci yang diberikan.

Kesimpulan

Modul ini memberikan pemahaman dasar tentang tipe bentukan dan array dalam pemrograman, khususnya dalam bahasa Go. Tipe bentukan seperti alias dan struct membantu dalam mendefinisikan tipe data yang lebih kompleks, sedangkan array dan slice memungkinkan pengelolaan koleksi data dengan cara yang efisien. Dengan memahami konsep-konsep ini, programmer dapat menulis kode yang lebih terstruktur dan mudah dipahami.

II. GUIDED

I. Guided 1

Sourcode

```
package main

import (
    "fmt"
    "sort"
)

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

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

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},
    }
}
```

```

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

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

    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)
    }
}

```

Output :

```

PS C:\Users\ASUS\Documents\File Fahrur\SEMESTER 3\PRAKTIKUM ALPRO 2\MODUL 7> go run
ahrur\SEMESTER 3\PRAKTIKUM ALPRO 2\MODUL 7\tempCodeRunnerFile.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\ASUS\Documents\File Fahrur\SEMESTER 3\PRAKTIKUM ALPRO 2\MODUL 7>

```

Deskripsi Program

Program ini mendefinisikan sebuah program dalam bahasa Go untuk menghitung rata-rata nilai dari beberapa mahasiswa, kemudian mengurutkan mahasiswa berdasarkan nilai rata-rata tersebut secara menurun.

Hasilnya adalah daftar mahasiswa yang terurut berdasarkan rata-rata nilai dari yang tertinggi ke terendah.

II. Guided 2

Sourcode

```
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 :

```

PS C:\Users\ASUS\Documents\File Fahrur\SEMESTER 3\PRAKTIKUM ALPRO 2\MODUL 7> go
ahrur\SEMESTER 3\PRAKTIKUM ALPRO 2\MODUL 7\tempCodeRunnerFile.go"
Daftar Mahasiswa:
NIM          Nama
-----
20231002     Budi
20231003     Cici
20231004     Dedi
20231001     Andi

Nama Mahasiswa dengan NIM 20231002 adalah Budi

Daftar Mahasiswa setelah dihapus:
NIM          Nama
-----
20231004     Dedi
20231001     Andi
20231002     Budi
PS C:\Users\ASUS\Documents\File Fahrur\SEMESTER 3\PRAKTIKUM ALPRO 2\MODUL 7>

```

Deskripsi Program

Kode tersebut membuat program dalam bahasa Go yang mengelola data mahasiswa menggunakan struktur **map**. Berikut penjelasan singkatnya:

1. Membuat **map** `mahasiswa` dengan NIM sebagai kunci dan nama mahasiswa sebagai nilai.

2. Menambahkan data baru dengan NIM "20231004" dan nama "Dedi" ke dalam **map**.
3. Menampilkan seluruh data mahasiswa dalam format tabel dengan kolom NIM dan Nama.
4. Mengakses data berdasarkan NIM tertentu ("20231002") dan menampilkan nama mahasiswa terkait.
5. Menghapus data mahasiswa dengan NIM "20231003".
6. Menampilkan data mahasiswa setelah penghapusan dalam format tabel.

Program ini menggunakan **map** untuk penyimpanan dinamis, sehingga data dapat ditambah dan dihapus dengan mudah.

III. UNGUIDED

a. Unguided 1

Sourcode

```
package main

import (
    "fmt"
    "math"
)

type titik struct {
    x, y int
}

type lingkaran struct {
    cx, cy, r int
}

func jarak(p titik, c lingkaran) float64 {
    //fahrur059
    return math.Sqrt(float64((p.x-c.cx)*(p.x-c.cx) + (p.y-c.cy)*(p.y-c.cy)))
}

func posisiTitik(p titik, l1, l2 lingkaran) string {
    jarak1 := jarak(p, l1)
    jarak2 := jarak(p, l2)

    dalam1 := jarak1 < float64(l1.r)
    dalam2 := jarak2 < float64(l2.r)

    if dalam1 && dalam2 {
```

```

        return "Titik di dalam lingkaran 1 dan 2"
    } else if dalam1 {
        return "Titik di dalam lingkaran 1"
    } else if dalam2 {
        return "Titik di dalam lingkaran 2"
    } else {
        return "Titik di luar lingkaran 1 dan 2"
    }
}

func main() {
    var l1, l2 lingkaran
    var p titik

    fmt.Print("Masukkan koordinat pusat dan radius lingkaran 1 (cx cy r): ")
    fmt.Scan(&l1.cx, &l1.cy, &l1.r)

    fmt.Print("Masukkan koordinat pusat dan radius lingkaran 2 (cx cy r): ")
    fmt.Scan(&l2.cx, &l2.cy, &l2.r)

    fmt.Print("Masukkan koordinat titik sembarang (x y): ")
    //fahrur059
    fmt.Scan(&p.x, &p.y)
    fmt.Print("\n")

    hasil := posisiTitik(p, l1, l2)
    fmt.Println(hasil)
}

```

Output :

```
PS C:\Users\ASUS\Documents\File Fahrur\SEMESTER 3\PRAKTIKUM ALPRO 2\MODUL 7> go run
ahrur\SEMESTER 3\PRAKTIKUM ALPRO 2\MODUL 7\tempCodeRunnerFile.go"
Masukkan koordinat pusat dan radius lingkaran 1 (cx cy r): 1 1 5
Masukkan koordinat pusat dan radius lingkaran 2 (cx cy r): 8 8 4
Masukkan koordinat titik sembarang (x y): 2 2

Titik di dalam lingkaran 1
PS C:\Users\ASUS\Documents\File Fahrur\SEMESTER 3\PRAKTIKUM ALPRO 2\MODUL 7> |
```

Deskripsi Program

Program ini berfungsi untuk menentukan posisi suatu titik terhadap dua lingkaran.

Berikut langkah-langkahnya:

1. Program mendefinisikan dua struktur data: `titik` (berisi koordinat x dan y) dan `lingkaran` (berisi koordinat pusat dan radius).
2. Fungsi `jarak` menghitung jarak antara titik dengan pusat lingkaran menggunakan rumus jarak Euclidean.
3. Fungsi `posisiTitik` mengecek apakah titik tersebut berada di dalam lingkaran pertama, lingkaran kedua, atau keduanya dengan membandingkan jarak ke pusat lingkaran dengan radius masing-masing lingkaran.
4. Program meminta input pengguna untuk koordinat pusat dan radius kedua lingkaran, serta koordinat titik yang akan dicek.
5. Program kemudian mencetak posisi titik berdasarkan hasil perbandingan jarak.

Hasilnya, program akan menampilkan apakah titik berada di dalam lingkaran 1, lingkaran 2, di kedua lingkaran, atau di luar keduanya.

b. Unguided 2

Sourcode

```
package main

import (
    "fmt"
    "math"
)

func rataRata(arr []int) float64 {
    total := 0
    //fahzur059
    for _, v := range arr {
        total += v
    }
    return float64(total) / float64(len(arr))
}

func standarDeviasi(arr []int, mean float64) float64 {
    var total float64
    //fahzur059
    for _, v := range arr {
        total += math.Pow(float64(v)-mean, 2)
    }
    return math.Sqrt(total / float64(len(arr)))
}

func frekuensi(arr []int, bilangan int) int {
    count := 0
    for _, v := range arr {
        //fahzur059
```

```

        if v == bilangan {
            count++
        }
    }
    return count
}

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

    arr := make([]int, N)

    fmt.Println("Masukkan elemen array:")
    for i := 0; i < N; i++ {
        //fahrur059
        fmt.Printf("Elemen ke-%d: ", i+1)
        fmt.Scan(&arr[i])
    }

    fmt.Println("Isi array:", arr)

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

    fmt.Print("Elemen dengan indeks genap: ")
    for i := 0; i < N; i += 2 {

```

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

var x int
fmt.Print("Masukkan bilangan x untuk kelipatan: ")
fmt.Scan(&x)

fmt.Print("Elemen dengan indeks kelipatan ", x, ": ")
for i := 0; i < N; i++ {
    //fahrur059
    if i%x == 0 {
        fmt.Print(arr[i], " ")
    }
}
fmt.Println()

var indeksHapus int
fmt.Print("Masukkan indeks yang ingin dihapus: ")
fmt.Scan(&indeksHapus)

if indeksHapus >= 0 && indeksHapus < N {
    arr = append(arr[:indeksHapus], arr[indeksHapus+1:]...)
    fmt.Println("Array setelah penghapusan:", arr)
} else {
    fmt.Println("Indeks tidak valid.")
}

mean := rataRata(arr)
fmt.Printf("Rata-rata: %.2f\n", mean)
```

```

stdDev := standarDeviasi(arr, mean)

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

var bilangan int

fmt.Print("Masukkan bilangan untuk menghitung frekuensi: ")

fmt.Scan(&bilangan)

freq := frekuensi(arr, bilangan)

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

}

```

Output :

```

PS C:\Users\ASUS\Documents\File Fahrur\SEMESTER 3\PRAKTIKUM ALPRO 2\MODUL 7> go run
ahrur\SEMESTER 3\PRAKTIKUM ALPRO 2\MODUL 7\tempCodeRunnerFile.go"
Masukkan jumlah elemen array: 5
Masukkan elemen array:
Elemen ke-1: 2
Elemen ke-2: 4
Elemen ke-3: 6
Elemen ke-4: 8
Elemen ke-5: 9
Isi array: [2 4 6 8 9]
Elemen dengan indeks ganjil: 4 8
Elemen dengan indeks genap: 2 6 9
Masukkan bilangan x untuk kelipatan: 4
Elemen dengan indeks kelipatan 4: 2 9
Masukkan indeks yang ingin dihapus: 2
Array setelah penghapusan: [2 4 8 9]
Rata-rata: 5.75
Standar deviasi: 2.86
Masukkan bilangan untuk menghitung frekuensi: 4
Frekuensi bilangan 4: 1
PS C:\Users\ASUS\Documents\File Fahrur\SEMESTER 3\PRAKTIKUM ALPRO 2\MODUL 7>

```

Deskripsi Program

Program ini menghitung statistik dasar dan melakukan operasi pada elemen-elemen array. Berikut adalah fungsionalitasnya:

1. **Input Array:** Pengguna diminta memasukkan jumlah elemen dan nilai dari elemen-elemen array.
2. **Menampilkan Isi Array:** Program menampilkan seluruh elemen, elemen dengan

indeks ganjil, dan elemen dengan indeks genap.

3. **Kelipatan Indeks:** Program meminta input angka x dari pengguna dan menampilkan elemen pada indeks yang merupakan kelipatan dari x .

4. **Hapus Elemen:** Program meminta indeks untuk dihapus, kemudian menampilkan array yang diperbarui.

5. **Statistik:**

- **Rata-rata:** Menghitung rata-rata elemen array.

- **Standar Deviasi:** Menghitung standar deviasi berdasarkan rata-rata.

6. **Frekuensi:** Program menghitung dan menampilkan frekuensi kemunculan suatu bilangan dalam array.

Program ini mencetak hasil setiap operasi secara langsung.

c. Unguided 3

Sourcode

```
package main

import (
    "fmt"
)

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

    fmt.Print("Masukkan nama Klub A: ")
    fmt.Scan(&klubA)
    //fahrur059
    fmt.Print("Masukkan nama Klub B: ")
    fmt.Scan(&klubB)
```

```
fmt.Print("\n")

for {
    fmt.Printf("Masukkan skor %s dan %s : ", klubA, klubB)
    fmt.Scan(&skorA, &skorB)

    if skorA < 0 || skorB < 0 {
        fmt.Println("Program dihentikan.")
        break
    }

    if skorA > skorB {
        hasil = append(hasil, klubA)
        fmt.Printf("Hasil: %s menang\n", klubA)
    } else if skorB > skorA {
        //fahrur059
        hasil = append(hasil, klubB)
        fmt.Printf("Hasil: %s menang\n", klubB)
    } else {
        fmt.Println("Hasil: Draw")
    }
}

fmt.Println("\nDaftar klub yang memenangkan pertandingan:")
for _, klub := range hasil {
    fmt.Println(klub)
}
}
```

Output :

```
PS C:\Users\ASUS\Documents\File Fahrur\SEMESTER 3\PRAKTIKUM ALPRO 2\MODUL 7> go run
fahrur\SEMESTER 3\PRAKTIKUM ALPRO 2\MODUL 7\tempCodeRunnerFile.go"
Masukkan nama Klub A: MU
Masukkan nama Klub B: Inter

Masukkan skor MU dan Inter : 2 0
Hasil: MU menang
Masukkan skor MU dan Inter : 1 2
Hasil: Inter menang
Masukkan skor MU dan Inter : 2 2
Hasil: Draw
Masukkan skor MU dan Inter : 3 5
Hasil: Inter menang
Masukkan skor MU dan Inter : 4 1
Hasil: MU menang
Masukkan skor MU dan Inter : 2 3
Hasil: Inter menang
Masukkan skor MU dan Inter : -1 2
Program dihentikan.

Daftar klub yang memenangkan pertandingan:
MU
Inter
Inter
MU
Inter
PS C:\Users\ASUS\Documents\File Fahrur\SEMESTER 3\PRAKTIKUM ALPRO 2\MODUL 7>
```

Deskripsi Program

Program ini digunakan untuk mencatat hasil pertandingan antara dua klub sepak bola.

Berikut fungsionalitasnya:

1. **Input Nama Klub:** Pengguna memasukkan nama untuk Klub A dan Klub B.
2. **Input Skor:** Pengguna menginput skor kedua klub berulang kali. Jika salah satu skor bernilai negatif, program berhenti.
3. **Penentuan Pemenang:**
 - Jika skor Klub A lebih tinggi, Klub A dianggap menang.
 - Jika skor Klub B lebih tinggi, Klub B dianggap menang.
 - Jika skor sama, pertandingan dinyatakan seri (draw).
4. **Pencatatan Pemenang:** Setiap klub yang menang dicatat dalam daftar.
5. **Menampilkan Hasil:** Program menampilkan daftar klub yang pernah memenangkan pertandingan sepanjang sesi input.

Program berakhir dengan menampilkan daftar pemenang setiap pertandingan sebelum dihentikan.

d. Unguided 4 A

Sourcode

```
package main

import (
    "fmt"
)

const NMAX int = 127

type tabel struct {
    tab [NMAX]rune
    m int
}

func isiArray(t *tabel, n *int) {
    var char rune
    *n = 0
    fmt.Print("Teks : ")
    for {
        fmt.Scanf("%c", &char)
        if char == '.' || *n >= NMAX {
            break
        }
        t.tab[*n] = char
        *n++
    }
}
```

```

    t.m = *n
}

func cetakArray(t tabel, n int) {
    fmt.Print("Reverse teks :")
    //fahrur059
    for i := 0; i < n; i++ {
        fmt.Printf("%c", t.tab[i])
    }
    fmt.Println()
}

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

func main() {
    var tab tabel
    var n int

    isiArray(&tab, &n)

    balikanArray(&tab, n)

    cetakArray(tab, n)
}

```

Output :

```
PS C:\Users\ASUS\Documents\File Fahrur\SEMESTER 3\PRAKTIKUM ALPRO 2\MODUL 7> go run
ahrur\SEMESTER 3\PRAKTIKUM ALPRO 2\MODUL 7\tempCodeRunnerFile.go"
Teks          : S E N A N G .
Reverse teks  : G N A N E S
PS C:\Users\ASUS\Documents\File Fahrur\SEMESTER 3\PRAKTIKUM ALPRO 2\MODUL 7> █
```

Deskripsi Program

Program ini membalik teks yang dimasukkan oleh pengguna dan mencetak hasilnya dalam urutan terbalik.

Fungsionalitas program:

1. **Pengisian Array:** Prosedur `isiArray` meminta input teks dari pengguna satu karakter per satuan hingga karakter titik (.) atau batas maksimum (127 karakter). Teks disimpan dalam array `tab`.
2. **Pembalikan Array:** Prosedur `balikanArray` membalik urutan karakter dalam array `tab`.
3. **Pencetakan Array:** Prosedur `cetakArray` mencetak teks yang telah dibalik.

Hasil akhirnya adalah teks input pengguna ditampilkan dalam urutan terbalik setelah dibalik oleh prosedur `balikanArray`.

e. Unguided 4 B

Sourcode

```
package main

import (
    "fmt"
)

const NMAX int = 127

type tabel struct {
```

```

    tab [NMAX]rune
    m int
}

func isiArray(t *tabel, n *int) {
    var char rune
    *n = 0
    //fahrur059
    fmt.Print("Teks      : ")
    for {
        fmt.Scanf("%c", &char)
        if char == '\n' || *n >= NMAX {
            break
        }
        t.tab[*n] = char
        *n++
    }
    t.m = *n
}

func palindrom(t tabel, n int) bool {
    for i := 0; i < n/2; i++ {
        if t.tab[i] != t.tab[n-i-1] {
            return false
        }
    }
    return true
}

func cetakArray(t tabel, n int) {
    fmt.Print("Reverse teks : ")

```

```

//fahrur059

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

func main() {
    var tab tabel
    var n int

    isiArray(&tab, &n)

    isPalindrom := palindrom(tab, n)

    fmt.Print("Palindrom  : ")
    if isPalindrom {
        fmt.Println("true")
    } else {
        fmt.Println("false")
    }
}

```

Output :

```

PS C:\Users\ASUS\Documents\File Fahrur\SEMESTER 3\PRAKTIKUM ALPRO 2\MODUL 7> go run
ahrur\SEMESTER 3\PRAKTIKUM ALPRO 2\MODUL 7\tempCodeRunnerFile.go"
Teks      : S E N A N G .
Palindrom : false
PS C:\Users\ASUS\Documents\File Fahrur\SEMESTER 3\PRAKTIKUM ALPRO 2\MODUL 7>

```

Deskripsi Program

Program ini mengecek apakah teks yang dimasukkan pengguna adalah palindrom.

Berikut fungsionalitasnya:

1. **Pengisian Array:** Prosedur `isiArray` meminta input teks dari pengguna satu karakter per satuan hingga tombol Enter ditekan atau batas maksimum (127 karakter) tercapai. Teks disimpan dalam array `tab`.
2. **Pengecekan Palindrom:** Fungsi `palindrom` membandingkan karakter di awal dan akhir array, berlanjut ke tengah, untuk memastikan apakah teks sama jika dibaca terbalik. Fungsi mengembalikan nilai `true` jika teks adalah palindrom, dan `false` jika tidak.
3. **Pencetakan Hasil:** Program menampilkan apakah teks adalah palindrom dengan menampilkan `true` atau `false`.

Program ini mengakhiri dengan hasil palindrom yang menunjukkan apakah teks tersebut adalah palindrom atau bukan.