

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL VII
STRUCT & ARRAY**



Disusun Oleh :

Egi Umar Ferdhika / 2311102277

11-IF-06

Dosen Pengampu :

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

PROGRAM STUDI S1 TEKNIK INFORMATIKA

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2024

I. DASAR TEORI

Struct dan array adalah dua elemen fundamental dalam bahasa pemrograman Go (Golang) yang berfungsi untuk mengorganisasikan dan mengelola data secara efisien. Struct adalah tipe data komposit yang digunakan untuk mengelompokkan beberapa tipe data ke dalam satu kesatuan logis. Struct sangat bermanfaat untuk merepresentasikan entitas dunia nyata dengan atribut-atribut yang relevan, seperti data pelanggan, barang, atau transaksi. Dalam Go, struct didefinisikan menggunakan kata kunci struct, dan setiap atribut (field) yang ada di dalamnya dapat memiliki tipe data yang berbeda, seperti string, integer, atau bahkan tipe data lainnya. Struct memungkinkan pengembang untuk membangun tipe data kustom sesuai kebutuhan aplikasi, sehingga meningkatkan fleksibilitas dan keterbacaan kode.

Di sisi lain, array adalah struktur data dasar yang digunakan untuk menyimpan sekumpulan elemen dengan tipe data yang sama. Array memiliki ukuran tetap yang dideklarasikan saat pembuatan, sehingga jumlah elemen di dalam array tidak dapat diubah. Setiap elemen dalam array dapat diakses menggunakan indeks, dimulai dari 0. Array sering digunakan untuk menyimpan data yang terorganisasi dengan baik, seperti daftar nilai, data statistik, atau elemen yang jumlahnya diketahui sebelumnya. Meskipun array memiliki keterbatasan dalam ukuran yang tetap, Go menyediakan tipe data slice yang lebih fleksibel, memungkinkan manipulasi data secara dinamis berdasarkan array yang mendasarinya.

Kombinasi antara struct dan array dalam pengelolaan data dapat memberikan efisiensi dan fleksibilitas yang tinggi. Contohnya, sebuah array dapat menyimpan banyak struct yang merepresentasikan data entitas yang seragam, seperti daftar pelanggan atau produk.

II. GUIDED

1. Berisi source code dan output dari kegiatan praktikum yang telah dilaksanakan.

Source Code diberi penjelasan maka akan menjadi nilai ++

GUIDED 1

Sourcecode

```
package main

import (
    "fmt"
    "math"
)

type Titik struct {
    x int
    y int
}

type Lingkaran struct {
    pusat Titik
    radius int
}

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)))
}

func titikDiDalamLingkaran(t Titik, l Lingkaran) bool {
    jarak := hitungJarak(t, l.pusat)
    return jarak <= float64(l.radius)
}

func main() {
    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}

    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}

    var x, y int
    fmt.Print("Masukkan koordinat titik sembarang (x y):
")
}
```

```

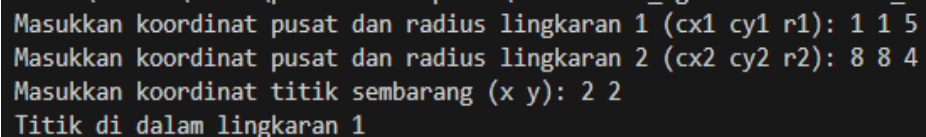
    fmt.Scanln(&x, &y)
    titik := Titik{x: x, y: y}

    diDalamL1 := titikDiDalamLingkaran(titik,
    lingkaran1)
    diDalamL2 := titikDiDalamLingkaran(titik,
    lingkaran2)

    if diDalamL1 && diDalamL2 {
        fmt.Println("Titik di dalam lingkaran 1 dan 2")
    } else if diDalamL1 {
        fmt.Println("Titik di dalam lingkaran 1")
    } else if diDalamL2 {
        fmt.Println("Titik di dalam lingkaran 2")
    } else {
        fmt.Println("Titik berada di luar lingkaran")
    }
}

```

Screenshoot Output



```

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 di dalam lingkaran 1

```

Deskripsi Program

Program di atas merupakan program untuk menentukan apakah suatu titik berada di dalam salah satu atau kedua lingkaran yang ditentukan oleh pengguna. Program menggunakan dua struct, yaitu Titik untuk merepresentasikan koordinat dua dimensi (x, y) dan Lingkaran untuk merepresentasikan lingkaran dengan atribut pusat (sebuah Titik) dan radius. Program dimulai dengan meminta input dari pengguna berupa koordinat pusat dan jari-jari dari dua lingkaran, serta koordinat sebuah titik sembarang. Selanjutnya, program memanfaatkan fungsi hitungJarak, yang menggunakan rumus jarak Euclidean untuk menghitung jarak antara dua titik. Fungsi lain, titikDiDalamLingkaran, memeriksa apakah jarak dari titik sembarang ke pusat lingkaran lebih kecil atau sama dengan radius lingkaran, yang berarti titik berada di dalam lingkaran. Berdasarkan hasil perhitungan, program menampilkan pesan apakah titik tersebut berada di dalam salah satu lingkaran, kedua lingkaran, atau di luar semua lingkaran.

III. UNGUIDED

1. Berisi source code dan output dari kegiatan praktikum yang telah dilaksanakan.

Source Code diberi penjelasan maka akan menjadi nilai ++

UNGUIDED 1

Sourcecode

```
package main

import (
    "fmt"
    "math"
    // Nama : Egi Umar Ferdhika
    // NIM : 2311102277
)

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

    arr := make([]float64, n)
    for i := 0; i < n; i++ {
        fmt.Printf("Masukkan elemen ke-%d: ", i)
        fmt.Scan(&arr[i])
    }

    for {
        fmt.Println("\nMenu:")
        fmt.Println("1. Tampilkan semua isi array")
        fmt.Println("2. Tampilkan elemen dengan indeks ganjil")
        fmt.Println("3. Tampilkan elemen dengan indeks genap")
        fmt.Println("4. Tampilkan elemen dengan indeks kelipatan x")
        fmt.Println("5. Hapus elemen array pada indeks tertentu")
        fmt.Println("6. Tampilkan rata-rata")
        fmt.Println("7. Tampilkan standar deviasi")
        fmt.Println("8. Tampilkan frekuensi bilangan")
        fmt.Println("9. Keluar")

        var choice int
        fmt.Print("Pilih menu: ")
        fmt.Scan(&choice)

        switch choice {
        case 1:
            fmt.Println("Isi array:", arr)
        case 2:
            for i := 1; i < len(arr); i += 2 {
                fmt.Printf("%v ", arr[i])
            }
        }
    }
}
```

```

    }
    fmt.Println()
case 3:
    for i := 0; i < len(arr); i += 2 {
        fmt.Printf("%v ", arr[i])
    }
    fmt.Println()
case 4:
    var x int
    fmt.Print("Masukkan nilai x: ")
    fmt.Scan(&x)
    for i := 0; i < len(arr); i++ {
        if i%x == 0 {
            fmt.Printf("%v ", arr[i])
        }
    }
    fmt.Println()
case 5:
    var idx int
    fmt.Print("Masukkan indeks yang akan
dihapus: ")
    fmt.Scan(&idx)
    if idx >= 0 && idx < len(arr) {
        arr = append(arr[:idx], arr[idx+1:]...)
    }
case 6:
    sum := 0.0
    for _, v := range arr {
        sum += v
    }
    fmt.Printf("Rata-rata: %.2f\n",
sum/float64(len(arr)))
case 7:
    mean := 0.0
    for _, v := range arr {
        mean += v
    }
    mean /= float64(len(arr))

    variance := 0.0
    for _, v := range arr {
        variance += math.Pow(v-mean, 2)
    }
    variance /= float64(len(arr))
    stdDev := math.Sqrt(variance)
    fmt.Printf("Standar deviasi: %.2f\n",
stdDev)
case 8:
    freq := make(map[float64]int)
    for _, v := range arr {
        freq[v]++
    }
    for num, count := range freq {
        fmt.Printf("Bilangan %.2f muncul %d
kali\n", num, count)
    }

```

```

        case 9:
            return
        }
    }
}

```

Screenshoot Output

```

Masukkan jumlah elemen array: 5
Masukkan elemen ke-0: 3
Masukkan elemen ke-1: 4
Masukkan elemen ke-2: 5
Masukkan elemen ke-3: 6
Masukkan elemen ke-4: 7

Menu:
1. Tampilkan semua isi array
2. Tampilkan elemen dengan indeks ganjil
3. Tampilkan elemen dengan indeks genap
4. Tampilkan elemen dengan indeks kelipatan x
5. Hapus elemen array pada indeks tertentu
6. Tampilkan rata-rata
7. Tampilkan standar deviasi
8. Tampilkan frekuensi bilangan
9. Keluar
Pilih menu: 1
Isi array: [3 4 5 6 7]

Menu:
1. Tampilkan semua isi array
2. Tampilkan elemen dengan indeks ganjil
3. Tampilkan elemen dengan indeks genap
4. Tampilkan elemen dengan indeks kelipatan x
5. Hapus elemen array pada indeks tertentu
6. Tampilkan rata-rata
7. Tampilkan standar deviasi
8. Tampilkan frekuensi bilangan
9. Keluar
Pilih menu: 2
4 6

Menu:
1. Tampilkan semua isi array
2. Tampilkan elemen dengan indeks ganjil
3. Tampilkan elemen dengan indeks genap
4. Tampilkan elemen dengan indeks kelipatan x
5. Hapus elemen array pada indeks tertentu
6. Tampilkan rata-rata
7. Tampilkan standar deviasi
8. Tampilkan frekuensi bilangan
9. Keluar
Pilih menu: 3
3 5 7

Menu:
1. Tampilkan semua isi array
2. Tampilkan elemen dengan indeks ganjil
3. Tampilkan elemen dengan indeks genap
4. Tampilkan elemen dengan indeks kelipatan x
5. Hapus elemen array pada indeks tertentu
6. Tampilkan rata-rata
7. Tampilkan standar deviasi
8. Tampilkan frekuensi bilangan
9. Keluar
Pilih menu: 4
Masukkan nilai x: 2
3 5 7

Menu:
1. Tampilkan semua isi array
2. Tampilkan elemen dengan indeks ganjil
3. Tampilkan elemen dengan indeks genap
4. Tampilkan elemen dengan indeks kelipatan x
5. Hapus elemen array pada indeks tertentu
6. Tampilkan rata-rata
7. Tampilkan standar deviasi
8. Tampilkan frekuensi bilangan
9. Keluar
Pilih menu: 6
Rata-rata: 5.00

```

```

Menu:
1. Tampilkan semua isi array
2. Tampilkan elemen dengan indeks ganjil
3. Tampilkan elemen dengan indeks genap
4. Tampilkan elemen dengan indeks kelipatan x
5. Hapus elemen array pada indeks tertentu
6. Tampilkan rata-rata
7. Tampilkan standar deviasi
8. Tampilkan frekuensi bilangan
9. Keluar
Pilih menu: 7
Standar deviasi: 1.41

Menu:
1. Tampilkan semua isi array
2. Tampilkan elemen dengan indeks ganjil
3. Tampilkan elemen dengan indeks genap
4. Tampilkan elemen dengan indeks kelipatan x
5. Hapus elemen array pada indeks tertentu
6. Tampilkan rata-rata
7. Tampilkan standar deviasi
8. Tampilkan frekuensi bilangan
9. Keluar
Pilih menu: 8
Bilangan 4.00 muncul 1 kali
Bilangan 5.00 muncul 1 kali
Bilangan 6.00 muncul 1 kali
Bilangan 7.00 muncul 1 kali
Bilangan 3.00 muncul 1 kali

Menu:
1. Tampilkan semua isi array
2. Tampilkan elemen dengan indeks ganjil
3. Tampilkan elemen dengan indeks genap
4. Tampilkan elemen dengan indeks kelipatan x
5. Hapus elemen array pada indeks tertentu
6. Tampilkan rata-rata
7. Tampilkan standar deviasi
8. Tampilkan frekuensi bilangan
9. Keluar
Pilih menu: 5
Masukkan indeks yang akan dihapus: 2

Menu:
1. Tampilkan semua isi array
2. Tampilkan elemen dengan indeks ganjil
3. Tampilkan elemen dengan indeks genap
4. Tampilkan elemen dengan indeks kelipatan x
5. Hapus elemen array pada indeks tertentu
6. Tampilkan rata-rata
7. Tampilkan standar deviasi
8. Tampilkan frekuensi bilangan
9. Keluar
Pilih menu: 1
Isi array: [3 4 6 7]

```

Deskripsi Program

Program di atas adalah program untuk berbagai operasi pada array, dengan menu interaktif yang memungkinkan pengguna memilih tindakan tertentu. Program ini dimulai dengan meminta pengguna menentukan jumlah elemen array dan mengisi nilai-nilainya. Array diinisialisasi menggunakan `make` untuk mendukung elemen bertipe `float64`. Setelah itu, program memasuki loop interaktif yang menampilkan sembilan pilihan menu dengan berbagai fungsi operasi array, termasuk penampilan elemen, manipulasi, dan analisis statistik.

Penjelasan Menu Operasi:

1. Tampilkan semua isi array: Menampilkan seluruh elemen array dalam bentuk list.
2. Tampilkan elemen dengan indeks ganjil: Menampilkan elemen yang indeksnya bernilai ganjil (dimulai dari indeks 1).
3. Tampilkan elemen dengan indeks genap: Menampilkan elemen dengan indeks genap (dimulai dari indeks 0).
4. Tampilkan elemen dengan indeks kelipatan x: Meminta input nilai x dan menampilkan elemen array dengan indeks yang merupakan kelipatan x.
5. Hapus elemen array pada indeks tertentu: Meminta pengguna memasukkan indeks yang akan dihapus, kemudian memperbarui array dengan elemen tersebut dihilangkan menggunakan operasi `append` pada subarray.
6. Tampilkan rata-rata: Menghitung rata-rata elemen array dengan menjumlahkan semua elemen dan membaginya dengan jumlah elemen.
7. Tampilkan standar deviasi: Menghitung standar deviasi elemen array berdasarkan rumus statistik, yaitu akar dari rata-rata deviasi kuadrat terhadap nilai rata-rata (mean).
8. Tampilkan frekuensi bilangan: Menghitung frekuensi kemunculan setiap bilangan dalam array menggunakan `map` sebagai struktur data pendukung.
9. Keluar: Menghentikan program.

Program ini menggunakan `switch` untuk menangani pilihan menu, memastikan setiap operasi dieksekusi dengan benar berdasarkan input pengguna. Selain itu, program ini menunjukkan pemanfaatan konsep array dinamis, perulangan, dan operasi matematis yang relevan, seperti akar kuadrat (`math.Sqrt`) dan kuadrat (`math.Pow`).

UNGUIDED 2

Sourcecode

```
package main

import (
    "fmt"
    // Nama : Egi Umar Ferdhika
    // NIM : 2311102277
)

type Match struct {
    score1 int
    score2 int
}

func main() {
    var club1, club2 string
    fmt.Print("Masukkan nama klub 1: ")
    fmt.Scan(&club1)
    fmt.Print("Masukkan nama klub 2: ")
    fmt.Scan(&club2)

    matches := make([]Match, 0)
    winners := make([]string, 0)

    for {
        var match Match
        fmt.Printf("\nPertandingan %d:\n",
len(matches)+1)
        fmt.Printf("Skor %s: ", club1)
        fmt.Scan(&match.score1)
        fmt.Printf("Skor %s: ", club2)
        fmt.Scan(&match.score2)

        if match.score1 < 0 || match.score2 < 0 {
            break
        }

        matches = append(matches, match)

        var winner string
        if match.score1 > match.score2 {
            winner = club1
        } else if match.score2 > match.score1 {
            winner = club2
        } else {
            winner = "Draw"
        }
        winners = append(winners, winner)
    }

    fmt.Println("\nHasil pertandingan:")
    for i, match := range matches {
```

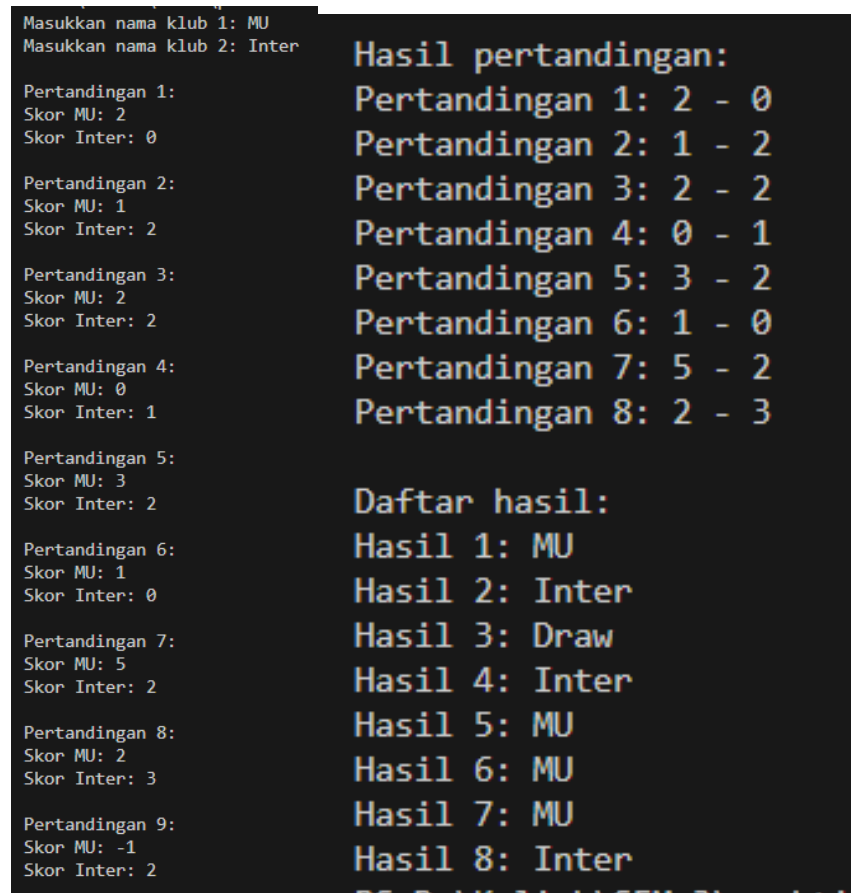
```

        fmt.Printf("Pertandingan %d: %d - %d\n", i+1,
match.score1, match.score2)
    }

    fmt.Println("\nDaftar hasil:")
    for i, winner := range winners {
        fmt.Printf("Hasil %d: %s\n", i+1, winner)
    }
}

```

Screenshot Output



```

Masukkan nama klub 1: MU
Masukkan nama klub 2: Inter

Pertandingan 1:
Skor MU: 2
Skor Inter: 0

Pertandingan 2:
Skor MU: 1
Skor Inter: 2

Pertandingan 3:
Skor MU: 2
Skor Inter: 2

Pertandingan 4:
Skor MU: 0
Skor Inter: 1

Pertandingan 5:
Skor MU: 3
Skor Inter: 2

Pertandingan 6:
Skor MU: 1
Skor Inter: 0

Pertandingan 7:
Skor MU: 5
Skor Inter: 2

Pertandingan 8:
Skor MU: 2
Skor Inter: 3

Pertandingan 9:
Skor MU: -1
Skor Inter: 2

Hasil pertandingan:
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

Daftar hasil:
Hasil 1: MU
Hasil 2: Inter
Hasil 3: Draw
Hasil 4: Inter
Hasil 5: MU
Hasil 6: MU
Hasil 7: MU
Hasil 8: Inter

```

Deskripsi Program

Program di atas adalah aplikasi untuk mencatat hasil pertandingan antara dua klub sepak bola dan menentukan pemenangnya. Program dimulai dengan meminta pengguna memasukkan nama kedua klub. Setiap pertandingan dicatat dalam struct `Match`, yang menyimpan skor dari kedua tim. Array dinamis (slice) digunakan untuk menyimpan data pertandingan dan daftar pemenang. Pengguna diminta untuk memasukkan skor untuk kedua tim secara berulang melalui perulangan

hingga salah satu skor yang dimasukkan negatif, yang menandakan akhir pencatatan. Logika menentukan pemenang dilakukan dengan membandingkan skor kedua tim; jika skor tim pertama lebih besar, maka pemenangnya adalah tim pertama, dan sebaliknya untuk tim kedua. Jika skor sama, hasilnya adalah "Draw". Setelah pencatatan selesai, program menampilkan hasil setiap pertandingan dalam format skor dan daftar pemenang.

UNGUIDED 3

Sourcecode

```
package main

import (
    "fmt"
    // Nama : Egi Umar Ferdhika
    // NIM : 2311102277
)

const NMAX int = 127

// Struct untuk menampung array karakter
type tabel [NMAX]rune

// Fungsi untuk mengisi array
func isiArray(t *tabel, n *int) {
    fmt.Print("Masukkan teks: ")
    var input string
    fmt.Scan(&input)

    *n = 0
    // Mengisi array dengan karakter dari input
    for _, char := range input {
        if char != '.' && *n < NMAX {
            t[*n] = char
            *n++
        }
    }
}

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

// Fungsi untuk membalik array
func balikanArray(t *tabel, n int) {
    reversed := tabel{}
    for i := n - 1; i >= 0; i-- {
        reversed[i] = t[n - 1 - i]
    }
    *t = reversed
}
```

```

        for i := 0; i < n; i++ {
            reversed[i] = t[n-1-i]
        }

        fmt.Print("Reverse teks\t:")
        for i := 0; i < n; i++ {
            fmt.Printf("%c", reversed[i])
            t[i] = reversed[i] // Update array asli dengan
hasil reversed
        }
        fmt.Println()
    }

    // Fungsi untuk mengecek palindrom
    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 main() {
        var tab tabel
        var m int

        // Input dan tampilkan array original
        isiArray(&tab, &m)
        cetakArray(tab, m)

        // Balik array dan tampilkan
        balikanArray(&tab, m)

        // Cek palindrom dan tampilkan hasil
        fmt.Print("Palindrom\t:")
        if palindrom(tab, m) {
            fmt.Println("true")
        } else {
            fmt.Println("false")
        }
    }
}

```

Screenshot Output

```

PS D:\Kuliah\SEM 3\parktikum al
Masukkan teks: KASURRUSAK
Teks           :KASURRUSAK
Reverse teks   :KASURRUSAK
Palindrom      :true
PS D:\Kuliah\SEM 3\parktikum al
Masukkan teks: SUKA
Teks           :SUKA
Reverse teks   :AKUS
Palindrom      :false

```

Deskripsi Program

Program ini adalah implementasi manipulasi teks menggunakan array karakter yang juga memeriksa apakah teks tersebut merupakan palindrom. Program memanfaatkan konstanta NMAX untuk menentukan kapasitas maksimum array, sementara struct tabel digunakan untuk menyimpan karakter teks. Pengguna diminta memasukkan teks, di mana hanya karakter sebelum simbol titik (.) yang akan diproses dan disimpan dalam array. Setelah itu, program menampilkan teks asli, membalikkan urutan karakter, dan memperbarui array dengan hasilnya. Selanjutnya, program memeriksa apakah teks tersebut adalah palindrom, yaitu teks yang sama ketika dibaca dari depan maupun belakang. Pemeriksaan dilakukan dengan membandingkan elemen di awal dan akhir array secara berpasangan hingga mencapai tengah array. Jika semua pasangan karakter cocok, teks dinyatakan sebagai palindrom.