

**LAPORAN PRAKTIKUM
ALGORITMA PEMROGRAMAN 2**

**MODUL VII
STRUCK & ARRAY**



Oleh:

Mansyuroh

NIM:

2311102234

**S1 TEKNIK INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2024**

I. DASAR TEORI

A. 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
kamus type <nama alias> <tipe data> algoritma	package main type <nama alias> <tipe data> func main(){

2) Struct atau Record

Structure 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
kamus type struct < 1> <nama <tipe data> <field 2> <tipe data> <field 3> <tipe data> <field	package main type <nama struct> struct { <field 1> <tipe data> <field 2> <tipe data> < field 3> <tipe data>

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.

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.

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.

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.

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 berdasarkan nilai rata-rata
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
    })

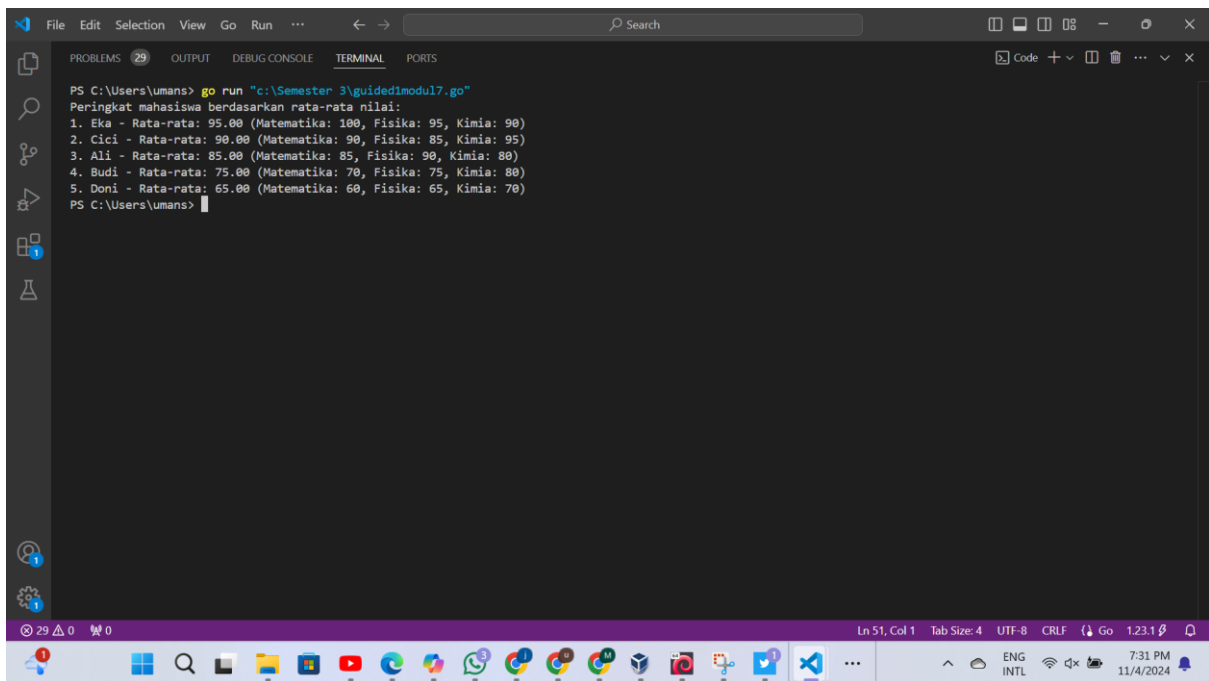
    // Menampilkan hasil
    fmt.Println("Peringkat mahasiswa berdasarkan rata-rata
nilai:")
}
```

```

        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 program



```

PS C:\Users\umans> go run "c:\Semester 3\guidedimodul7.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\umans>

```

Deskripsi program Program ini untuk menghitung rata-rata nilai mahasiswa dari tiga mata pelajaran (Matematika, Fisika, Kimia), mengurutkan mereka dari nilai rata-rata tertinggi ke terendah, lalu menampilkan peringkat beserta nilai tiap mata pelajaran.

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"

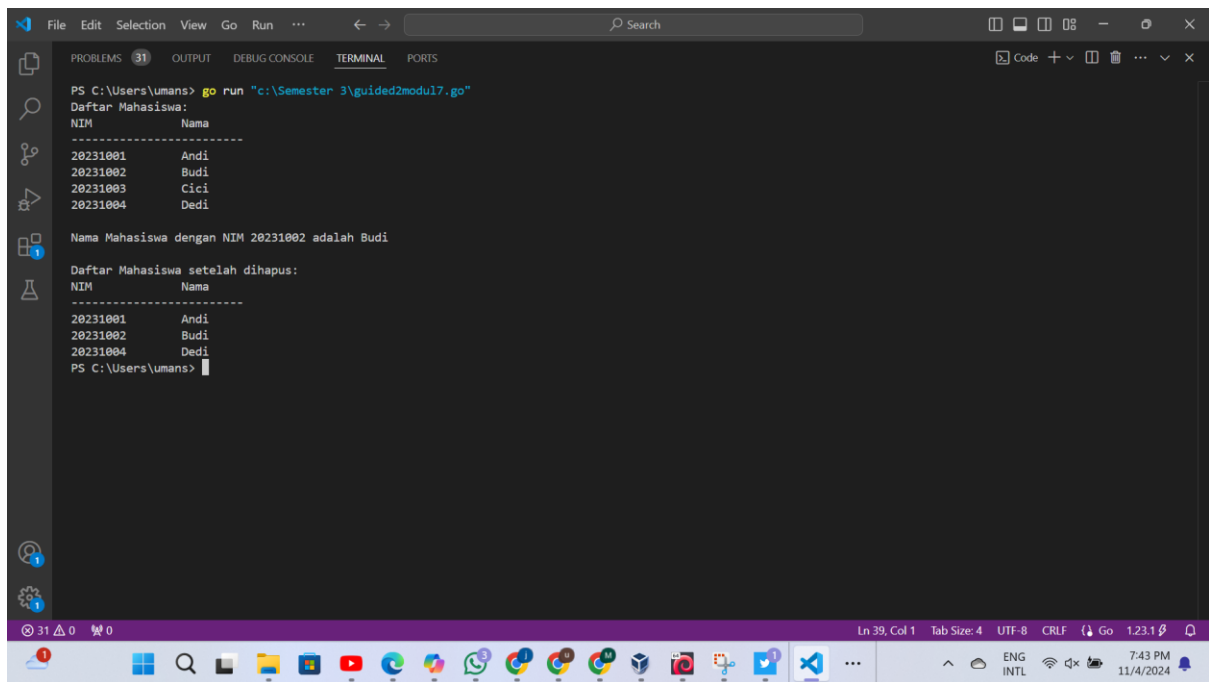
    // Menampilkan seluruh isi map dalam format kolom dan
    // baris
    fmt.Println("Daftar Mahasiswa:")
    fmt.Println("NIM\t\tNama")
    fmt.Println("-----")
    for nim, nama := range mahasiswa {
        fmt.Printf("%s\t%s\n", nim, nama)
    }

    // 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



```
PS C:\Users\umans> go run "c:\Semester 3\guided2modul7.go"
Daftar Mahasiswa:
NIM      Nama
-----
20231001 Andi
20231002 Budi
20231003 Cici
20231004 Dedi

Nama Mahasiswa dengan NIM 20231002 adalah Budi

Daftar Mahasiswa setelah dihapus:
NIM      Nama
-----
20231001 Andi
20231002 Budi
20231004 Dedi
PS C:\Users\umans>
```

Deskripsi program

Program ini untuk mengelola data mahasiswa menggunakan map: menambah, menampilkan, mengakses berdasarkan NIM, dan menghapus data, lalu menampilkan hasil akhirnya dalam format tabel.

III. UNGUIDED

1. Source code

```
package main

import (
    "fmt"
    "math"
)

type Titik234 struct {
```

```
    x, y int
}

type Lingkaran234 struct {
    pusat Titik234
    radius int
}

func jarak234(p, q Titik234) float64 {
    return math.Sqrt(float64((p.x-q.x)*(p.x-q.x) + (p.y-q.y)*(p.y-q.y)))
}

func diDalam234(c Lingkaran234, p Titik234) bool {
    return jarak234(c.pusat, p) <= float64(c.radius)
}

func main() {
    var cx1234, cy1234, r1234, cx2234, cy2234, r2234, x234, y234 int

    fmt.Println("Masukkan koordinat pusat dan radius lingkaran 1:")
    fmt.Scan(&cx1234, &cy1234, &r1234)
    lingkaran1234 := Lingkaran234{Titik234{cx1234, cy1234}, r1234}

    fmt.Println("Masukkan koordinat pusat dan radius lingkaran 2:")
    fmt.Scan(&cx2234, &cy2234, &r2234)
    lingkaran2234 := Lingkaran234{Titik234{cx2234, cy2234}, r2234}

    fmt.Println("Masukkan koordinat titik sembarang:")
    fmt.Scan(&x234, &y234)
    titik234 := Titik234{x234, y234}

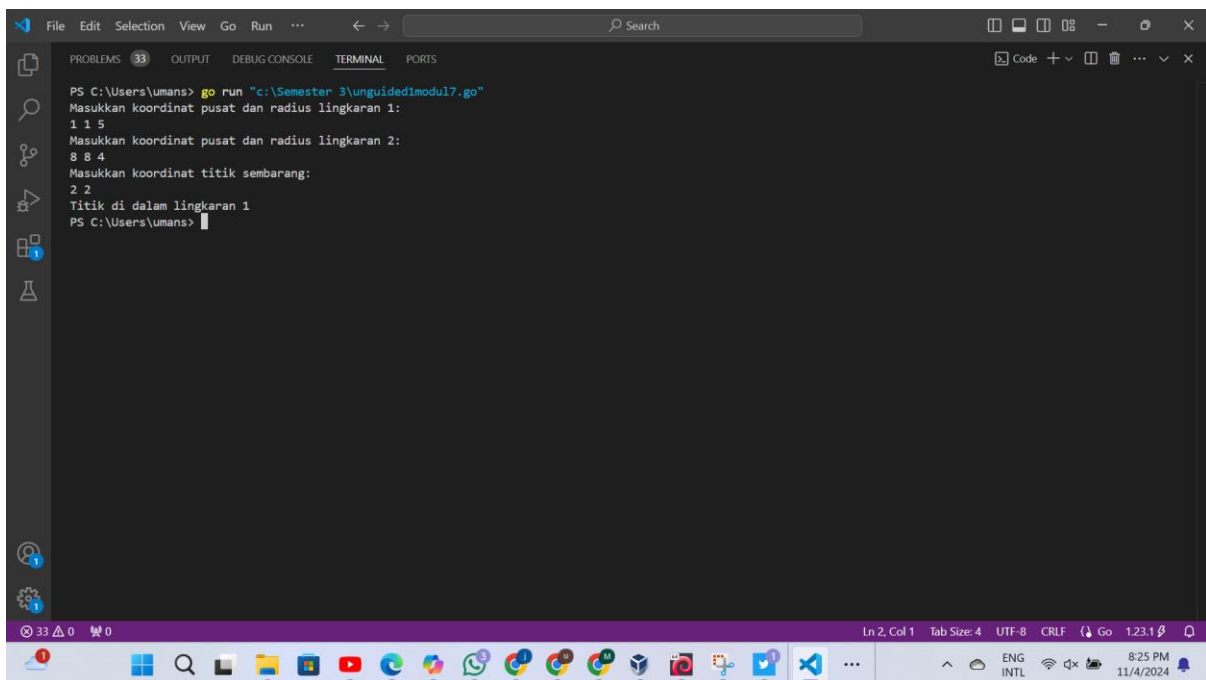
    dalamLingkaran1234 := diDalam234(lingkaran1234, titik234)
```



```
dalamLingkaran2234 := diDalam234(lingkaran2234, titik234)

if dalamLingkaran1234 && dalamLingkaran2234 {
    fmt.Println("Titik di dalam lingkaran 1 dan 2")
} else if dalamLingkaran1234 {
    fmt.Println("Titik di dalam lingkaran 1")
} else if dalamLingkaran2234 {
    fmt.Println("Titik di dalam lingkaran 2")
} else {
    fmt.Println("Titik di luar lingkaran 1 dan 2")
}
}
```

Screenshot program



The screenshot shows a Windows terminal window with the Go IDE interface. The terminal displays the following output:

```
PS C:\Users\umans> go run "c:\Semester 3\unguided1modul7.go"
Masukkan koordinat pusat dan radius lingkaran 1:
1 1 5
Masukkan koordinat pusat dan radius lingkaran 2:
8 8 4
Masukkan koordinat titik sembarang:
2 2
Titik di dalam lingkaran 1
PS C:\Users\umans>
```

The terminal window has a dark theme and a sidebar on the left with icons for Explorer, Search, Source Control, Run and Debug, and Testing. The bottom status bar shows the file path, line and column numbers, tab size, encoding, line endings, and the current time and date.

Deskripsi program

Program ini menentukan posisi suatu titik terhadap dua lingkaran. Setelah memasukkan koordinat pusat dan radius kedua lingkaran serta koordinat titik sembarang, program mengecek apakah titik berada di dalam salah satu, kedua, atau di luar kedua lingkaran, lalu menampilkan hasilnya.

2. Source code

```
package main

import (
    "fmt"
    "math"
)

func faktorBilangan234(bilangan234, i234 int) {
    if i234 > bilangan234 {
        return
    }
    if bilangan234%i234 == 0 {
        fmt.Print(i234, " ")
    }
    faktorBilangan234(bilangan234, i234+1)
}

func main() {
    var N234 int
    fmt.Print("Masukkan jumlah elemen array: ")
    fmt.Scan(&N234)
```

```
array234 := make([]int, N234)

for i234 := 0; i234 < N234; i234++ {
    fmt.Printf("Masukkan elemen ke-%d: ", i234)
    fmt.Scan(&array234[i234])
}

fmt.Println("\nIsi array:")
fmt.Println(array234)

fmt.Println("\nElemen dengan indeks ganjil:")
for i234 := 1; i234 < N234; i234 += 2 {
    fmt.Print(array234[i234], " ")
}

fmt.Println("\n\nElemen dengan indeks genap:")
for i234 := 0; i234 < N234; i234 += 2 {
    fmt.Print(array234[i234], " ")
}

var x234 int
fmt.Print("\n\nMasukkan bilangan x: ")
fmt.Scan(&x234)
fmt.Println("Elemen dengan indeks kelipatan", x234, ":")
for i234 := 0; i234 < N234; i234++ {
    if i234%x234 == 0 {
        fmt.Print(array234[i234], " ")
    }
}
```

```

    }

}

var indeksHapus234 int

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

fmt.Scan(&indeksHapus234)

if indeksHapus234 >= 0 && indeksHapus234 < len(array234) {
    array234 = append(array234[:indeksHapus234],
array234[indeksHapus234+1:]...)

    fmt.Println("Array setelah dihapus elemen pada indeks",
indeksHapus234, ":")

    fmt.Println(array234)

} else {

    fmt.Println("Indeks tidak valid.")

}

var total234 int

for _, value234 := range array234 {
    total234 += value234
}

if len(array234) > 0 {
    rataRata234 := float64(total234) / float64(len(array234))

    fmt.Printf("\nRata-rata: %.2f\n", rataRata234)

    var variance234 float64

    for _, value234 := range array234 {
        variance234 += math.Pow(float64(value234)-rataRata234,
2)

    }

    variance234 /= float64(len(array234))

```

```

        standarDeviasi234 := math.Sqrt(variance234)

        fmt.Printf("Standar deviasi: %.2f\n", standarDeviasi234)
    } else {

        fmt.Println("Array kosong, tidak dapat menghitung rata-rata
dan standar deviasi.")
    }

    var bilangan234 int

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

    fmt.Scan(&bilangan234)

    frekuensi234 := 0

    for _, value234 := range array234 {
        if value234 == bilangan234 {
            frekuensi234++
        }
    }

    fmt.Printf("Frekuensi bilangan %d: %d\n", bilangan234,
frekuensi234)

    fmt.Print("\nMasukkan bilangan untuk mencari faktornya: ")

    var inputBilangan234 int

    fmt.Scan(&inputBilangan234)

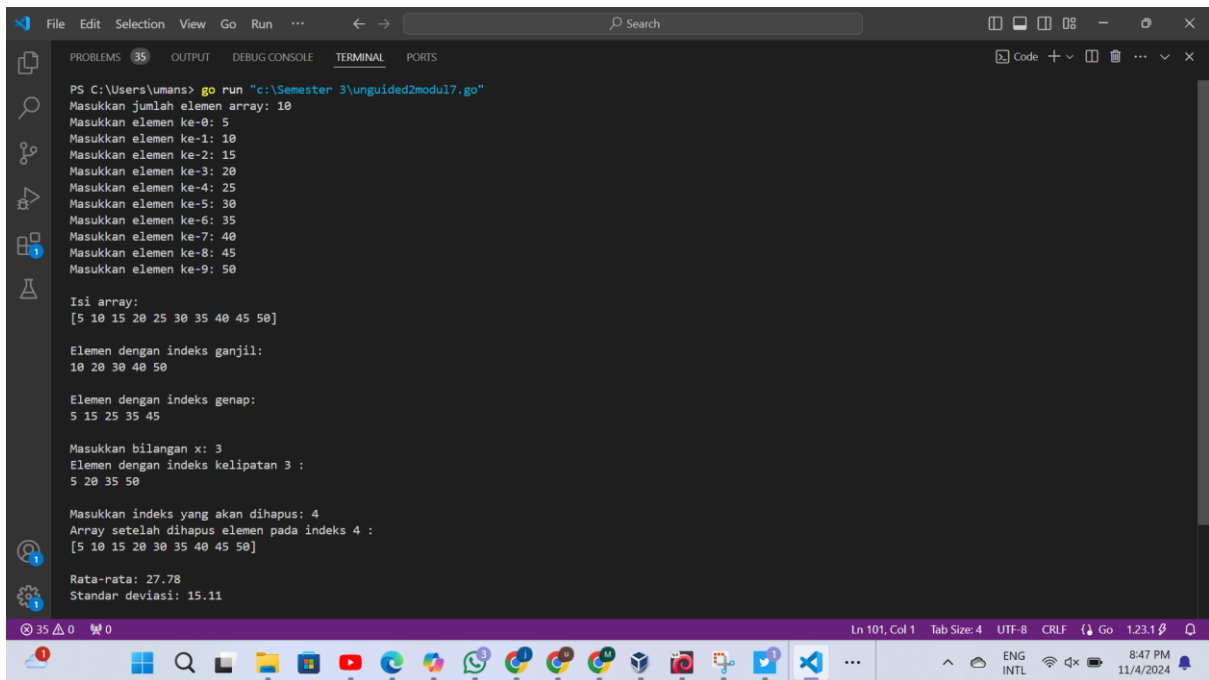
    fmt.Println("Faktor dari", inputBilangan234, "adalah:")

    faktorBilangan234(inputBilangan234, 1)

    fmt.Println()
}

```

Screenshot program



```
PS C:\Users\umans> go run "c:\Semester 3\unguided2modul7.go"
Masukkan jumlah elemen array: 10
Masukkan elemen ke-0: 5
Masukkan elemen ke-1: 10
Masukkan elemen ke-2: 15
Masukkan elemen ke-3: 20
Masukkan elemen ke-4: 25
Masukkan elemen ke-5: 30
Masukkan elemen ke-6: 35
Masukkan elemen ke-7: 40
Masukkan elemen ke-8: 45
Masukkan elemen ke-9: 50

Isi array:
[5 10 15 20 25 30 35 40 45 50]

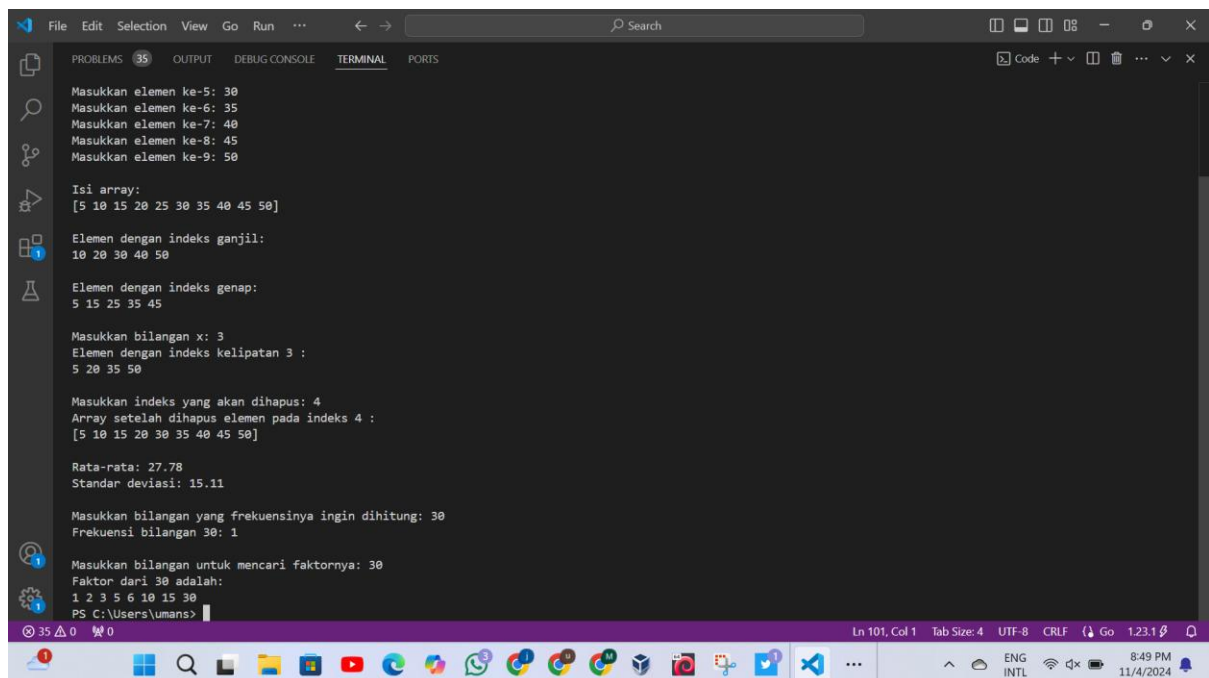
Elemen dengan indeks ganjil:
10 20 30 40 50

Elemen dengan indeks genap:
5 15 25 35 45

Masukkan bilangan x: 3
Elemen dengan indeks kelipatan 3 :
5 20 35 50

Masukkan indeks yang akan dihapus: 4
Array setelah dihapus elemen pada indeks 4 :
[5 10 15 20 30 35 40 45 50]

Rata-rata: 27.78
Standar deviasi: 15.11
```



```
Masukkan elemen ke-5: 30
Masukkan elemen ke-6: 35
Masukkan elemen ke-7: 40
Masukkan elemen ke-8: 45
Masukkan elemen ke-9: 50

Isi array:
[5 10 15 20 25 30 35 40 45 50]

Elemen dengan indeks ganjil:
10 20 30 40 50

Elemen dengan indeks genap:
5 15 25 35 45

Masukkan bilangan x: 3
Elemen dengan indeks kelipatan 3 :
5 20 35 50

Masukkan indeks yang akan dihapus: 4
Array setelah dihapus elemen pada indeks 4 :
[5 10 15 20 30 35 40 45 50]

Rata-rata: 27.78
Standar deviasi: 15.11

Masukkan bilangan yang frekuensinya ingin dihitung: 30
Frekuensi bilangan 30: 1

Masukkan bilangan untuk mencari faktornya: 30
Faktor dari 30 adalah:
1 2 3 5 6 10 15 30
PS C:\Users\umans>
```

Deskripsi program

Program ini adalah aplikasi konsol dalam bahasa Go yang memungkinkan pengguna untuk:

1. Menginput sejumlah elemen untuk array
2. Menampilkan isi array

3. Menampilkan elemen dengan indeks ganjil dan genap
4. Menampilkan elemen berdasarkan indeks kelipatan bilangan tertentu
5. Menghapus elemen pada indeks yang ditentukan
6. Menghitung dan menampilkan rata-rata dan standar dari elemen array
7. Menghitung frekuensi kemunculan bilangan tertentu dalam array
8. Menampilkan factor dari bilangan yang dimasukkan.

3. Source code

```
package main

import (
    "bufio"
    "fmt"
    "os"
    "strconv"
)

func main() {
    // Membuat scanner untuk membaca input pengguna
    scanner := bufio.NewScanner(os.Stdin)

    // Meminta nama-nama klub
    fmt.Print("Masukkan nama Klub A: ")
    scanner.Scan()
    klubA := scanner.Text()

    fmt.Print("Masukkan nama Klub B: ")
    scanner.Scan()
    klubB := scanner.Text()

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

    // Loop untuk input skor pertandingan
    for pertandingan := 1; ; pertandingan++ {
        // Input skor Klub A
        fmt.Printf("Masukkan skor %s pada pertandingan %d:", klubA, pertandingan)
        scanner.Scan()
        skorA, errA := strconv.Atoi(scanner.Text())

        // Input skor Klub B
        fmt.Printf("Masukkan skor %s pada pertandingan %d:", klubB, pertandingan)
        scanner.Scan()
        skorB, errB := strconv.Atoi(scanner.Text())

        // Cek jika skor negatif atau input tidak valid
```

```

        if errA != nil || errB != nil || skorA < 0 ||
skorB < 0 {
            fmt.Println("Pertandingan selesai")
            break
        }

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

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

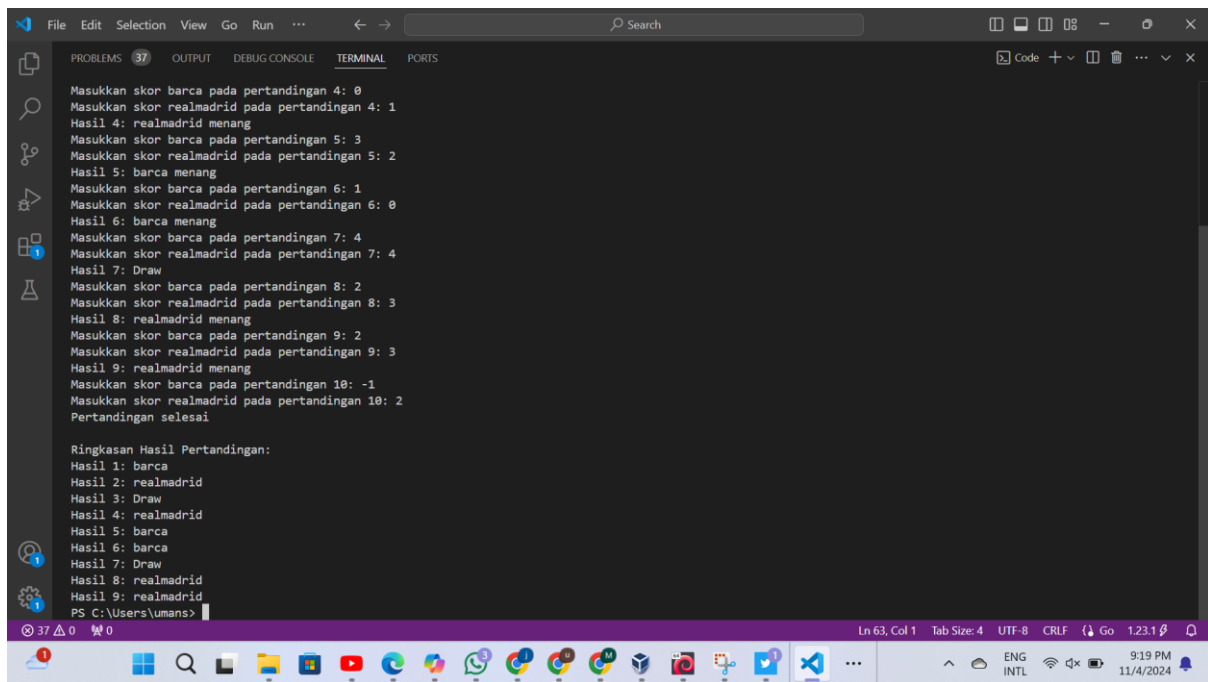
```

Screenshot program

```

PS C:\Users\umans> go run "c:\Semester 3\unguided3modul7.go"
Masukkan nama Klub A: barca
Masukkan nama Klub B: realmadrid
Masukkan skor barca pada pertandingan 1: 2
Masukkan skor realmadrid pada pertandingan 1: 0
Hasil 1: barca menang
Masukkan skor barca pada pertandingan 2: 1
Masukkan skor realmadrid pada pertandingan 2: 3
Hasil 2: realmadrid menang
Masukkan skor barca pada pertandingan 3: 2
Masukkan skor realmadrid pada pertandingan 3: 2
Hasil 3: Draw
Masukkan skor barca pada pertandingan 4: 0
Masukkan skor realmadrid pada pertandingan 4: 1
Hasil 4: realmadrid menang
Masukkan skor barca pada pertandingan 5: 3
Masukkan skor realmadrid pada pertandingan 5: 2
Hasil 5: barca menang
Masukkan skor barca pada pertandingan 6: 1
Masukkan skor realmadrid pada pertandingan 6: 0
Hasil 6: barca menang
Masukkan skor barca pada pertandingan 7: 4
Masukkan skor realmadrid pada pertandingan 7: 4
Hasil 7: Draw
Masukkan skor barca pada pertandingan 8: 2
Masukkan skor realmadrid pada pertandingan 8: 3
Hasil 8: realmadrid menang
Masukkan skor barca pada pertandingan 9: 2
Masukkan skor realmadrid pada pertandingan 9: 3
Hasil 9: realmadrid menang
Masukkan skor barca pada pertandingan 10: -1
Masukkan skor realmadrid pada pertandingan 10: 2
Pertandingan selesai

```

```
File Edit Selection View Go Run ... Search
PROBLEMS 37 OUTPUT DEBUG CONSOLE TERMINAL PORTS
Masukkan skor barca pada pertandingan 4: 0
Masukkan skor realmadrid pada pertandingan 4: 1
Hasil 4: realmadrid menang
Masukkan skor barca pada pertandingan 5: 3
Masukkan skor realmadrid pada pertandingan 5: 2
Hasil 5: barca menang
Masukkan skor barca pada pertandingan 6: 1
Masukkan skor realmadrid pada pertandingan 6: 0
Hasil 6: barca menang
Masukkan skor barca pada pertandingan 7: 4
Masukkan skor realmadrid pada pertandingan 7: 4
Hasil 7: Draw
Masukkan skor barca pada pertandingan 8: 2
Masukkan skor realmadrid pada pertandingan 8: 3
Hasil 8: realmadrid menang
Masukkan skor barca pada pertandingan 9: 2
Masukkan skor realmadrid pada pertandingan 9: 3
Hasil 9: realmadrid menang
Masukkan skor barca pada pertandingan 10: -1
Masukkan skor realmadrid pada pertandingan 10: 2
Pertandingan selesai

Ringkasan Hasil Pertandingan:
Hasil 1: barca
Hasil 2: realmadrid
Hasil 3: Draw
Hasil 4: realmadrid
Hasil 5: barca
Hasil 6: barca
Hasil 7: Draw
Hasil 8: realmadrid
Hasil 9: realmadrid
PS C:\Users\umans>
```

Deskripsi Program

Program ini adalah untuk mencatat hasil pertandingan antara dua klub sepak bola. Pengguna memasukkan nama klub, lalu skor setiap pertandingan. Program menentukan pemenang atau hasil imbang berdasarkan skor yang dimasukkan dan menampilkan ringkasan hasil pertandingan setelah selesai. Input dihentikan jika pengguna memasukkan skor negatif atau nilai yang tidak valid.

4. Source code

```
package main

import (
    "fmt"
)

const NMAX int = 127

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

func isiArray(t *tabel, n *int) {
```

```

var input string
fmt.Print("Masukkan teks: ")
fmt.Scanf("%s", &input)

t.m = len(input)
*n = t.m
for i := 0; i < *n && i < NMAX; i++ {
    t.tab[i] = rune(input[i])
}
}

func cetakArray(t tabel, n int) {
    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++ {
        j := n - i - 1
        t.tab[i], t.tab[j] = t.tab[j], t.tab[i]
    }
}

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 main() {
    var tab tabel
    var m int

    isiArray(&tab, &m)

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

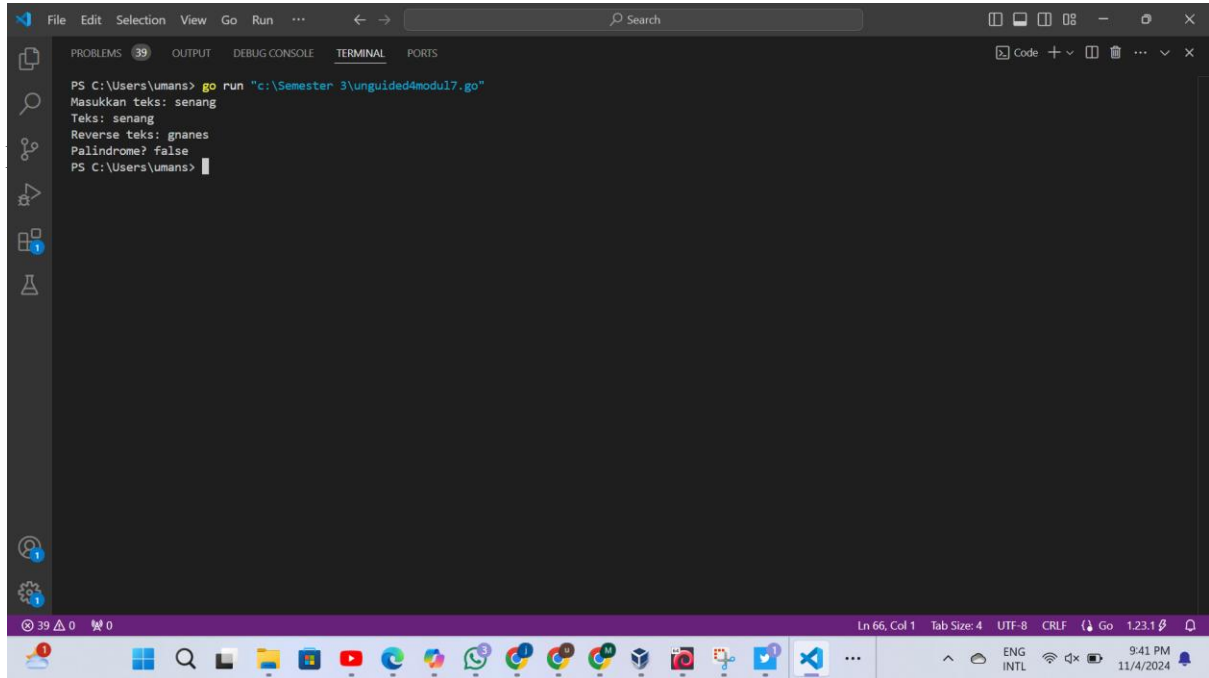
    balikanArray(&tab, m)
    fmt.Print("Reverse teks: ")
    cetakArray(tab, m)

    isPalindrome := palindrom(tab, m)
    fmt.Print("Palindrome? ")
    fmt.Println(isPalindrome)
}

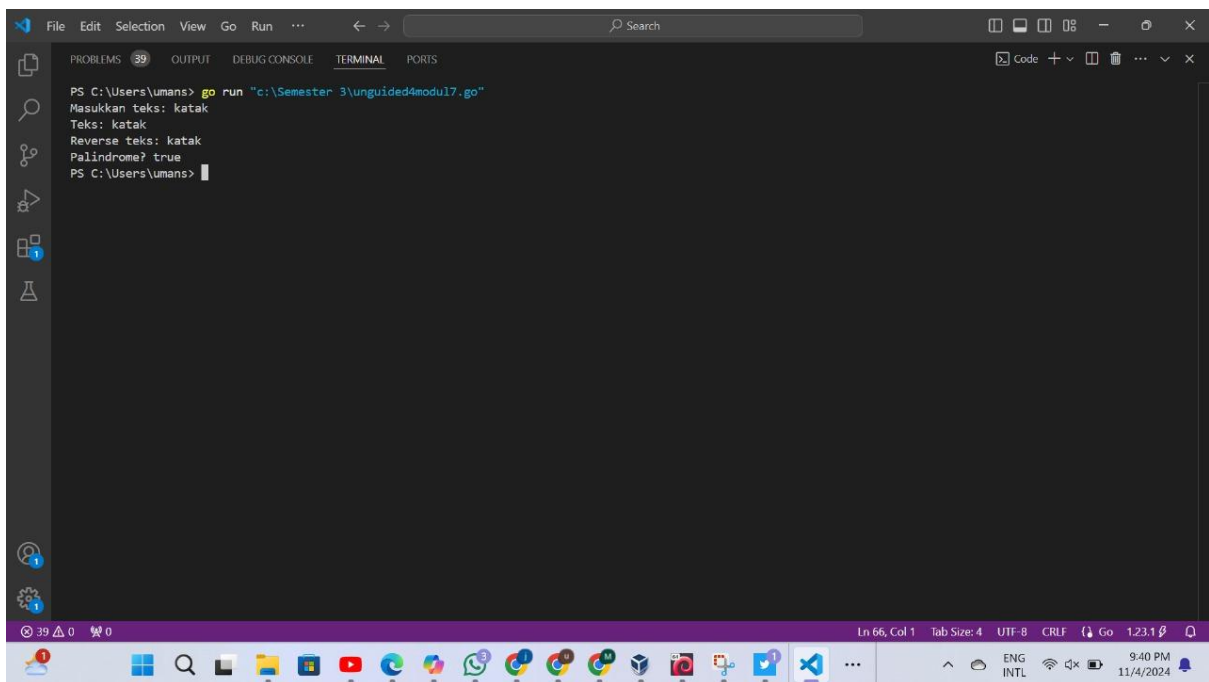
```

```
}
```

Screenshot program



```
PS C:\Users\umans> go run "c:\Semester 3\unguided4modul7.go"
Masukkan teks: senang
Teks: senang
Reverse teks: gnanes
Palindrome? false
PS C:\Users\umans>
```



```
PS C:\Users\umans> go run "c:\Semester 3\unguided4modul7.go"
Masukkan teks: katak
Teks: katak
Reverse teks: katak
Palindrome? true
PS C:\Users\umans>
```

Deskripsi program

Program ini membaca teks dari pengguna, menampilkan teks asli, membalik urutannya, dan mengecek apakah teks tersebut adalah palindrom (sama dibaca dari depan dan belakang). Jika teks adalah palindrom, program akan menampilkan true; jika tidak, false.