

**LAPORAN PRAKTIKUM
ALGORITMA PROGRAM 2
MODUL 7
STRUCT & ARRAY**



Oleh:

ADITHANA DHARMA PUTRA

2311102207

IF – 11- 02

**S1 TEKNIK INFORMATIKA
INSTITUT TEKNOLOGI TELKOM 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"
```

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

1. Guided 1

Source Code

```
package main

import "fmt"
import "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
}

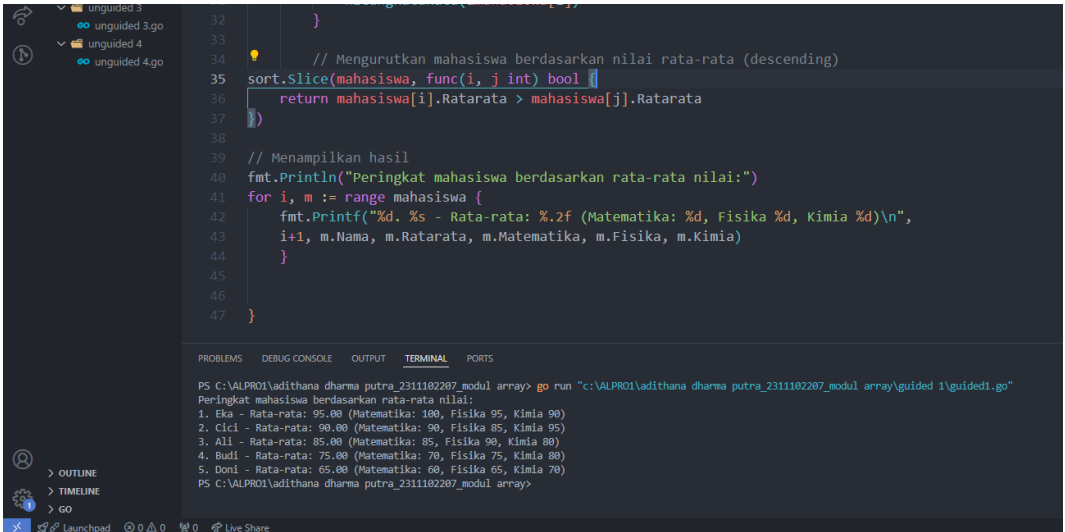
func main() {
    mahasiswa := []Mahasiswa{
        {"Ali", 85, 90, 80, 0},
        {"Budi", 70, 75, 80, 0},
        {"Cici", 90, 85, 95, 0},
        {"Doni", 60, 65, 70, 0},
        {"Eka", 100, 95, 90, 0},
    }
}
```

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

        // 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



```
32 }
33
34 // Mengurutkan mahasiswa berdasarkan nilai rata-rata (descending)
35 sort.Slice(mahasiswa, func(i, j int) bool {
36     return mahasiswa[i].Ratarata > mahasiswa[j].Ratarata
37 })
38
39 // Menampilkan hasil
40 fmt.Println("Peringkat mahasiswa berdasarkan rata-rata nilai:")
41 for i, m := range mahasiswa {
42     fmt.Printf("%d. %s - Rata-rata: %.2f (Matematika: %d, Fisika %d, Kimia %d)\n",
43         i+1, m>Nama, m.Ratarata, m.Matematika, m.Fisika, m.Kimia)
44 }
45
46
47 }
```

PROBLEMS DEBUG CONSOLE OUTPUT **TERMINAL** PORTS

```
PS C:\VALPRO1\adithana dharma putra_2311102207_modul array> go run "c:\VALPRO1\adithana dharma putra_2311102207_modul array\guided 1\guided1.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:\VALPRO1\adithana dharma putra_2311102207_modul array>
```

Deskripsi:

Kode ini mendefinisikan struktur data untuk menyimpan informasi mahasiswa beserta nilai mereka dalam tiga mata pelajaran (Matematika, Fisika, dan Kimia), serta menghitung rata-rata nilai dari ketiga mata pelajaran tersebut. Fungsi `hitungRataRata` digunakan untuk menghitung rata-rata nilai setiap mahasiswa. Dalam fungsi `main`, data beberapa mahasiswa dimasukkan ke dalam slice, kemudian rata-rata nilai mereka dihitung menggunakan fungsi tersebut. Setelah itu, mahasiswa diurutkan berdasarkan nilai rata-rata mereka secara menurun menggunakan `sort.Slice`. Akhirnya, hasil peringkat mahasiswa berdasarkan rata-rata nilai ditampilkan ke layar.

2. Guided 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"

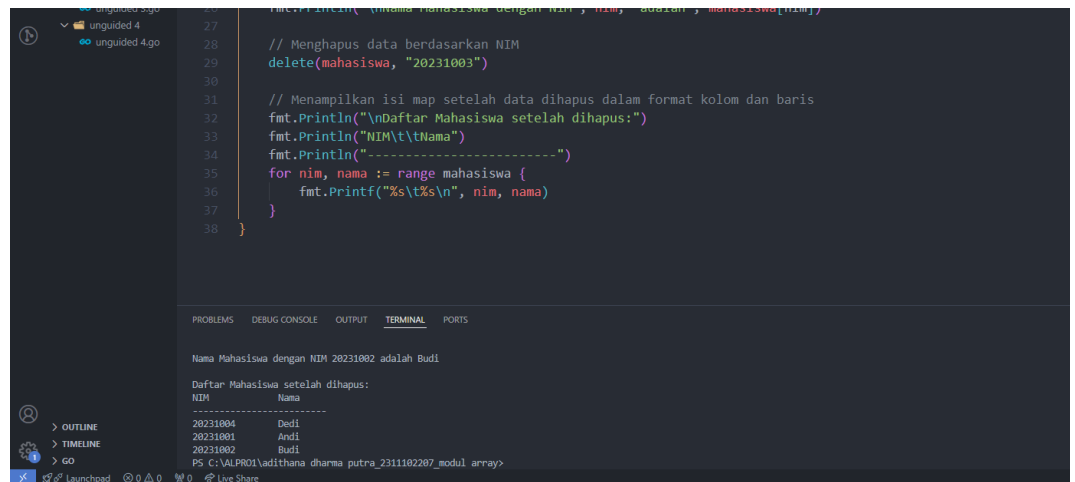
    // Menampilkan seluruh isi map dalam format
    kolom dan baris
    fmt.Println("Daftar Mahasiswa:")
    fmt.Println("NIM\t\tNama")
    fmt.Println("-----")
    for nim, nama := range mahasiswa {
        fmt.Printf("%s\t%s\n", nim, nama)
    }

    // 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



```
27 // Menampilkan isi map mahasiswa dengan NIM, nim, dan nama
28 // Menghapus data berdasarkan NIM
29 delete(mahasiswa, "20231003")
30
31 // Menampilkan isi map setelah data dihapus dalam format kolom dan baris
32 fmt.Println("\nDaftar Mahasiswa setelah dihapus:")
33 fmt.Println("NIM\tNama")
34 fmt.Println("-----")
35 for nim, nama := range mahasiswa {
36     fmt.Printf("%s\t%s\n", nim, nama)
37 }
38 }
```

PROBLEMS DEBUG CONSOLE OUTPUT **TERMINAL** PORTS

Nama Mahasiswa dengan NIM 20231002 adalah Budi

Daftar Mahasiswa setelah dihapus:

NIM	Nama
20231004	Dedi
20231001	Andi
20231002	Budi

PS C:\ALPRO1\adithana_dharma Putra_2311102207_modul array>

Deskripsi:

Kode ini membuat sebuah map untuk menyimpan data mahasiswa dengan NIM sebagai kunci dan nama sebagai nilai. Setelah mendefinisikan beberapa data awal, kode ini menambahkan data baru ke dalam map. Selanjutnya, seluruh isi map ditampilkan dalam format kolom dan baris. Kode ini juga menunjukkan cara mengakses data berdasarkan NIM tertentu dan menghapus data dari map. Setelah penghapusan, isi map kembali ditampilkan untuk menunjukkan perubahan yang terjadi.

II. UNGUIDED

1. Unguided 1

Source Code

```
package main

import "fmt"
import "math"

type data2311102207 struct {
    titikX float64
    titikY float64
    cx      float64
    cy      float64
    r        float64
}

func jarak(titikX, titikY float64, lingkaran1,
lingkaran2 data2311102207) string {
    jarakLingkaran1 :=
math.Sqrt(math.Pow(titikX-lingkaran1.cx, 2) +
math.Pow(titikY-lingkaran1.cy, 2))
    jarakLingkaran2 :=
math.Sqrt(math.Pow(titikX-lingkaran2.cx, 2) +
math.Pow(titikY-lingkaran2.cy, 2))

    if jarakLingkaran1 <= lingkaran1.r &&
jarakLingkaran2 <= lingkaran2.r{
        return "titik didalam lingkaran 1
dan 2"
    } else if jarakLingkaran1 <= lingkaran1.r
{
```

```
        return "titik didalam lingkaran 1"
    } else if jarakLingkaran2 <= lingkaran2.r
    {
        return "titik didalam lingkaran 2"
    } else {
        return "titik diluar lingkaran 1 dan
2"
    }
}

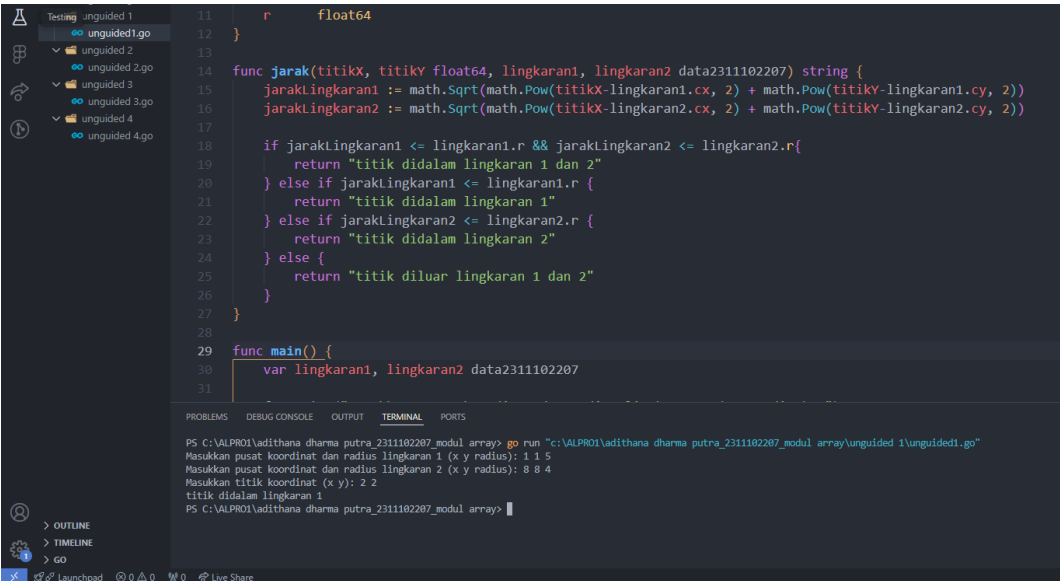
func main() {
    var lingkaran1, lingkaran2 data2311102207

    fmt.Print("Masukkan pusat koordinat dan
radius lingkaran 1 (x y radius): ")
    fmt.Scan(&lingkaran1.cx, &lingkaran1.cy,
&lingkaran1.r)

    fmt.Print("Masukkan pusat koordinat dan
radius lingkaran 2 (x y radius): ")
    fmt.Scan(&lingkaran2.cx, &lingkaran2.cy,
&lingkaran2.r)

    fmt.Print("Masukkan titik koordinat (x
y): ")
    fmt.Scan(&lingkaran1.titikX,
&lingkaran1.titikY)
    hasil := jarak(lingkaran1.titikX,
lingkaran1.titikY, lingkaran1, lingkaran2)
    fmt.Println(hasil)
}
```

Screenshot



```
11  r      float64
12  }
13
14  func jarak(titikX, titikY float64, lingkaran1, lingkaran2 data2311102207) string {
15      jarakLingkaran1 := math.Sqrt(math.Pow(titikX-lingkaran1.cx, 2) + math.Pow(titikY-lingkaran1.cy, 2))
16      jarakLingkaran2 := math.Sqrt(math.Pow(titikX-lingkaran2.cx, 2) + math.Pow(titikY-lingkaran2.cy, 2))
17
18      if jarakLingkaran1 <= lingkaran1.r && jarakLingkaran2 <= lingkaran2.r{
19          return "titik didalam lingkaran 1 dan 2"
20      } else if jarakLingkaran1 <= lingkaran1.r {
21          return "titik didalam lingkaran 1"
22      } else if jarakLingkaran2 <= lingkaran2.r {
23          return "titik didalam lingkaran 2"
24      } else {
25          return "titik diluar lingkaran 1 dan 2"
26      }
27  }
28
29  func main() {
30      var lingkaran1, lingkaran2 data2311102207
31  }
```

PROBLEMS DEBUG CONSOLE OUTPUT **TERMINAL** PORTS

```
PS C:\VALPRO1\adithana dhama putra_2311102207_modul array> go run "c:\VALPRO1\adithana dhama putra_2311102207_modul array\unguided 1\unguided1.go"
Masukkan pusat koordinat dan radius lingkaran 1 (x y radius): 1 1 5
Masukkan pusat koordinat dan radius lingkaran 2 (x y radius): 8 8 4
Masukkan titik koordinat (x y): 2 2
titik didalam lingkaran 1
PS C:\VALPRO1\adithana dhama putra_2311102207_modul array> |
```

Deskripsi

Kode ini mendefinisikan struktur data untuk menyimpan informasi dua lingkaran dan sebuah titik, serta menghitung jarak titik tersebut dari pusat kedua lingkaran. Fungsi jarak digunakan untuk menentukan apakah titik tersebut berada di dalam salah satu atau kedua lingkaran, atau di luar keduanya, berdasarkan jarak dari pusat lingkaran dan radiusnya. Dalam fungsi main, pengguna diminta untuk memasukkan koordinat pusat dan radius dari dua lingkaran, serta koordinat titik yang akan diperiksa. Hasil dari pengecekan posisi titik terhadap lingkaran kemudian ditampilkan ke layar.

2. Unguided 2

Source Code

```
package main

import "fmt"
import "math"

func tampilkanSeluruharray(array207 []int) {
    fmt.Println("Seluruh elemen array:",
array207)
}

func tampilkanIndeksGanjil(array207 []int) {
    fmt.Print("Elemen dengan indeks ganjil: ")
    for i := 1; i < len(array207); i += 2 {
        fmt.Print(array207[i], " ")
    }
    fmt.Println()
}

func tampilkanIndeksGenap(array207 []int) {
    fmt.Print("Elemen dengan indeks genap: ")
    for i := 0; i < len(array207); i += 2 {
        fmt.Print(array207[i], " ")
    }
    fmt.Println()
}

func tampilKandengankelipatan(array207 []int,
kelipatan int) {
```



```

        fmt.Printf("Elemen yang merupakan
kelipatan dari %v :", kelipatan)
        for i := 0; i < len(array207); i +=
kelipatan {
            fmt.Print(array207[i], " ")
        }
        fmt.Println()
    }

func tampilkanFrekuensi(array207 []int, x int)
{
    frekuensi := 0
    for _, angka := range array207 {
        if angka == x {
            frekuensi++
        }
    }
    fmt.Printf("Frekuensi dari %d: %d\n", x,
frekuensi)
}

func hapusElemen(array207 []int, x int) []int
{
    result := []int{}
    for _, elemen := range array207 {
        if elemen != x {
            result = append(result, elemen)
        }
    }
    return result
}

```

```

func hitungRataRata(array207 []int) float64 {
    if len(array207) == 0 {
        return 0
    }
    total := 0
    for _, angka := range array207 {
        total += angka
    }
    return float64(total) /
float64(len(array207))
}

func hitungSimpanganBaku(array207 []int)
float64 {
    if len(array207) == 0 {
        return 0
    }
    rataRata := hitungRataRata(array207)
    var total float64
    for _, angka := range array207 {
        total += math.Pow(float64(angka) -
rataRata, 2)
    }
    return math.Sqrt(total /
float64(len(array207)))
}

func main() {
    var n int

```

```
    fmt.Print("Masukkan jumlah elemen array: ")
    ")

    fmt.Scan(&n)
    array207 := make([]int, n)
    fmt.Print("Masukkan array: ")
    for i := 0; i < n; i++ {
        fmt.Scan(&array207[i])
    }

    tampilkanSeluruharray(array207)
    tampilkanIndeksGanjil(array207)
    tampilkanIndeksGenap(array207)

    var kelipatan int
    fmt.Print("Masukkan kelipatan: ")
    fmt.Scan(&kelipatan)
    tampilkanDengankelipatan(array207,
kelipatan)

    tampilkanFrekuensi(array207, 3)

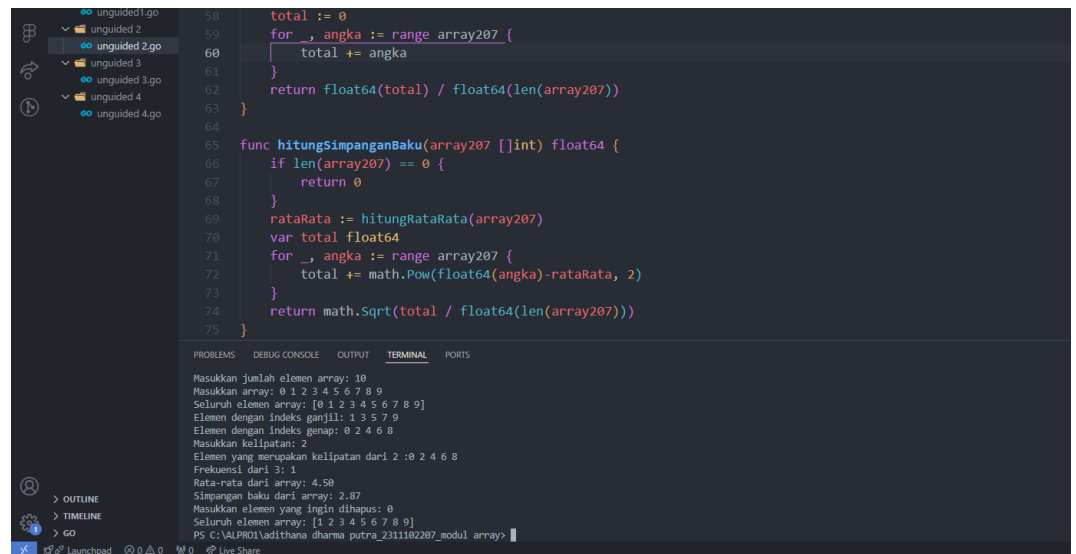
    rataRata := hitungRataRata(array207)
    fmt.Printf("Rata-rata dari array: %.2f\n",
rataRata)

    simpanganBaku :=
hitungSimpanganBaku(array207)
    fmt.Printf("Simpangan baku dari array:
%.2f\n", simpanganBaku)

    var elemenHapus int
```

```
        fmt.Print("Masukkan elemen yang ingin  
dihapus: ")  
        fmt.Scan(&elemenHapus)  
  
        array207 = hapusElemen(array207,  
elemenHapus)  
        tampilkanSeluruharray(array207)  
    }
```

Screenshot



```
58     total := 0
59     for _, angka := range array207 {
60         total += angka
61     }
62     return float64(total) / float64(len(array207))
63 }
64
65 func hitungSimpanganBaku(array207 []int) float64 {
66     if len(array207) == 0 {
67         return 0
68     }
69     rataRata := hitungRataRata(array207)
70     var total float64
71     for _, angka := range array207 {
72         total += math.Pow(float64(angka)-rataRata, 2)
73     }
74     return math.Sqrt(total / float64(len(array207)))
75 }
```

PROBLEMS DEBUG CONSOLE OUTPUT TERMINAL PORTS

Masukkan jumlah elemen array: 10
Masukkan array: 0 1 2 3 4 5 6 7 8 9
Seluruh elemen array: [0 1 2 3 4 5 6 7 8 9]
Elemen dengan indeks ganjil: 1 3 5 7 9
Elemen dengan indeks genap: 0 2 4 6 8
Masukkan kelipatan: 2
Elemen yang merupakan kelipatan dari 2 :0 2 4 6 8
Frekuensi dari 3: 1
Rata-rata dari array: 4.50
Simpangan baku dari array: 2.87
Masukkan elemen yang ingin dihapus: 0
Seluruh elemen array: [1 2 3 4 5 6 7 8 9]
PS C:\ALPRODI\adithana_dharma_putra_2311102207_modul_array>

Deskripsi

Kode ini berfungsi untuk melakukan berbagai operasi pada array integer, seperti menampilkan seluruh elemen, menampilkan elemen dengan indeks ganjil dan genap, menampilkan elemen yang merupakan kelipatan dari suatu angka, menghitung frekuensi kemunculan suatu elemen, menghapus elemen tertentu, serta menghitung rata-rata dan simpangan baku dari elemen-elemen array. Dalam fungsi main, pengguna diminta untuk memasukkan jumlah elemen dan nilai-nilai array, kemudian berbagai fungsi tersebut dipanggil untuk menampilkan hasil operasi yang dilakukan pada array.

3. Unguided 3

Source Code

```
package main

import "fmt"

type Klub207 struct {
    Nama string
    Skor int
}

func main() {
    var Klub2071 Klub207
    var Klub2072 Klub207
    var pemenang []string
    var match int = 1

    fmt.Print("Masukkan nama Klub207 A: ")
    fmt.Scanln(&Klub2071.Nama)
    fmt.Print("Masukkan nama Klub207 B: ")
    fmt.Scanln(&Klub2072.Nama)

    for {
        fmt.Printf("pertandingan %v: ", match)
        fmt.Scanln(&Klub2071.Skor, &Klub2072.Sk
or)

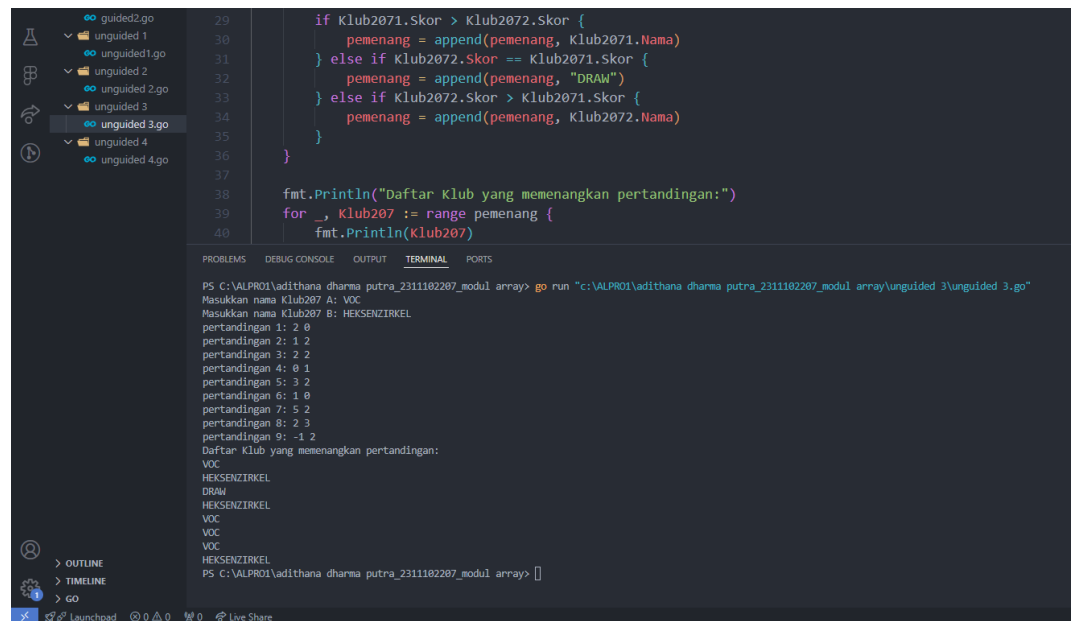
        match++

        if Klub2071.Skor < 0 || Klub2072.Skor
< 0 {
            break
        }
    }
}
```

```
        if Klub2071.Skor > Klub2072.Skor {
            pemenang = append(pemenang,
Klub2071.Nama)
        } else if Klub2072.Skor ==
Klub2071.Skor {
            pemenang = append(pemenang,
"DRAW")
        } else if Klub2072.Skor >
Klub2071.Skor {
            pemenang = append(pemenang,
Klub2072.Nama)
        }
    }

    fmt.Println("Daftar Klub yang memenangkan
pertandingan:")
    for _, Klub207 := range pemenang {
        fmt.Println(Klub207)
    }
}
```

Screenshot



The screenshot shows a Go IDE with a file explorer on the left, a code editor in the center, and a terminal at the bottom. The code editor displays a Go program that simulates a football tournament. It defines a struct for a club, reads input for two clubs and their scores for nine matches, and then prints the winning clubs. The terminal shows the execution of the program, with the user inputting 'VOC' for both clubs and various scores for the matches. The final output lists the winning clubs: VOC, HEKSENZIRKEL, and DRAW.

```
29 if Klub2071.Skor > Klub2072.Skor {
30     pemenang = append(pemenang, Klub2071>Nama)
31 } else if Klub2072.Skor == Klub2071.Skor {
32     pemenang = append(pemenang, "DRAW")
33 } else if Klub2072.Skor > Klub2071.Skor {
34     pemenang = append(pemenang, Klub2072>Nama)
35 }
36
37
38 fmt.Println("Daftar Klub yang memenangkan pertandingan:")
39 for _, Klub207 := range pemenang {
40     fmt.Println(Klub207)
```

PROBLEMS DEBUG CONSOLE OUTPUT TERMINAL PORTS

PS C:\ALPRO1\adithana dharma putra_2311102207_modul array> go run "c:\ALPRO1\adithana dharma putra_2311102207_modul array\unguided 3\unguided 3.go"

Masukkan nama Klub207 A: VOC

Masukkan nama Klub207 B: HEKSENZIRKEL

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

pertandingan 9: -1 2

Daftar Klub yang memenangkan pertandingan:

VOC

HEKSENZIRKEL

DRAW

HEKSENZIRKEL

VOC

VOC

HEKSENZIRKEL

PS C:\ALPRO1\adithana dharma putra_2311102207_modul array>

Deskripsi

Kode ini mendefinisikan struktur data untuk menyimpan informasi dua klub sepak bola dan mencatat skor pertandingan mereka. Dalam fungsi main, pengguna diminta untuk memasukkan nama kedua klub, kemudian skor pertandingan dimasukkan secara berulang hingga salah satu skor negatif dimasukkan, yang menandakan akhir dari input. Berdasarkan skor yang dimasukkan, pemenang setiap pertandingan dicatat dalam slice pemenang. Setelah input selesai, daftar klub yang memenangkan pertandingan ditampilkan ke layar.

4.Unguided 4

Source Code

```
package main
import "fmt"

const NMAX207 int = 127
type tabel [NMAX207]rune

func isiArray(t *tabel, n *int) {
    var karakter rune
    *n = 0
    fmt.Print("Teks\t\t: ")

    for *n < NMAX207 {
        fmt.Scanf("%c", &karakter)
        if karakter == '.' {
            break
        }

        if karakter != '\n' {
            t[*n] = karakter
            *n++
        }
    }
}

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

```

        fmt.Println()
    }

    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) string {
        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

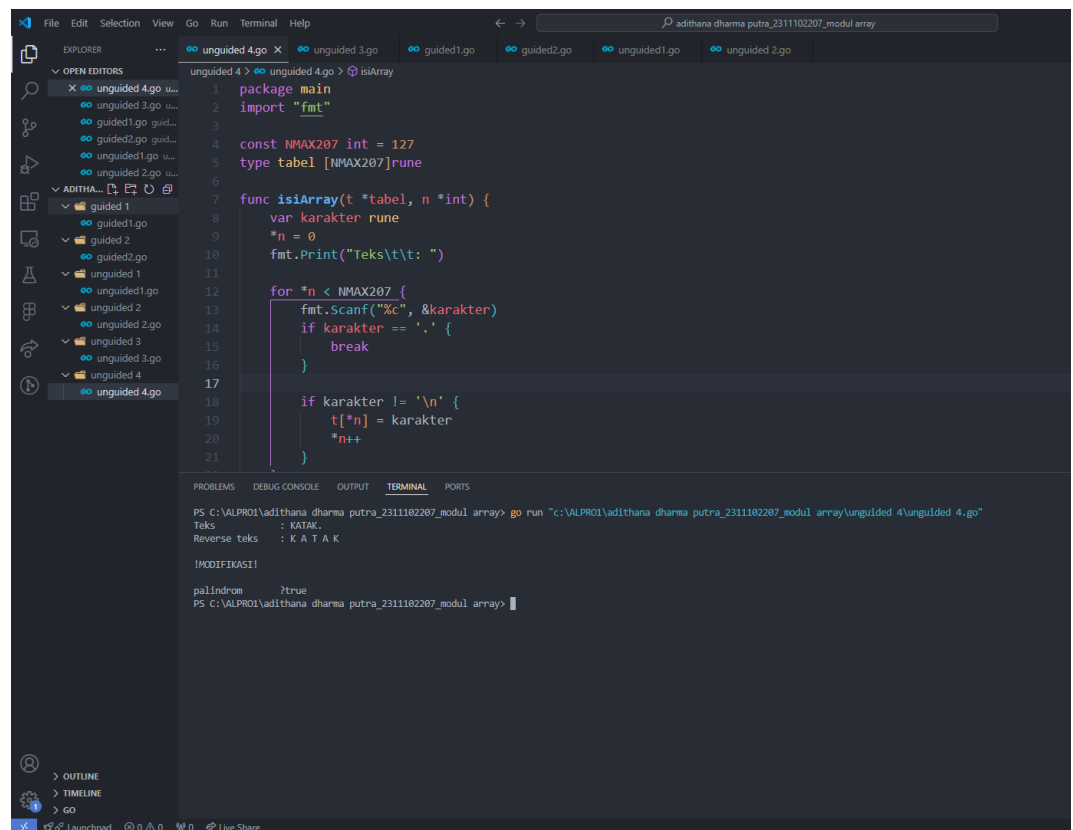
        isiArray(&tab, &m)
        balikanArray(tab, m)
        cetakArray(tab, m)

        fmt.Printf("\n!MODIFIKASI!\n\n")
        cek := palindrom(tab, m)
    }

```

```
fmt.Printf("palindrom\t?%s", cek)
}
```

Screenshot



```
package main
import "fmt"

const NMAX207 int = 127
type tabel [NMAX207]rune

func isiArray(t *tabel, n *int) {
    var karakter rune
    *n = 0
    fmt.Print("Teks\t\t: ")

    for *n < NMAX207 {
        fmt.Scanf("%c", &karakter)
        if karakter == '.' {
            break
        }

        if karakter != '\n' {
            t[*n] = karakter
            *n++
        }
    }
}

func cetakArray(tabel []rune) {
    for i := 0; i < len(tabel); i++ {
        fmt.Print(string(tabel[i]))
    }
}

func balikanArray(tabel []rune) []rune {
    var balikan []rune
    for i := len(tabel) - 1; i >= 0; i-- {
        balikan = append(balikan, tabel[i])
    }
    return balikan
}

func palindrom(tabel []rune) bool {
    return string(tabel) == string(balikanArray(tabel))
}

func main() {
    var tabel [127]rune
    var n int
    isiArray(&tabel, &n)
    cetakArray(tabel)
    balikanArray(tabel)
    palindrom(tabel)
}
```

PS C:\VALPRO1\adithana dhama putra_2311102207_modul array> go run "c:\VALPRO1\adithana dhama putra_2311102207_modul array\unguided 4\unguided 4.go"

Teks : KATAK.
Reverse teks : K A T A K

!MODIFIKASI!

palindrom : true
PS C:\VALPRO1\adithana dhama putra_2311102207_modul array> |

Deskripsi

Kode ini mendefinisikan sebuah array untuk menyimpan karakter-karakter dari sebuah teks yang diinput oleh pengguna, kemudian membalikkan urutan karakter-karakter tersebut dan memeriksa apakah teks tersebut adalah palindrom. Fungsi isiArray digunakan untuk mengisi array dengan karakter-karakter dari input hingga karakter titik (‘.’) ditemukan. Fungsi cetakArray menampilkan teks dalam urutan yang dibalik. Fungsi balikanArray mengembalikan array dengan urutan karakter yang dibalik, dan fungsi palindrom memeriksa apakah teks tersebut adalah palindrom. Dalam fungsi main, array diisi, dibalik, dan

hasilnya ditampilkan, serta dilakukan pengecekan apakah teks tersebut adalah palindrom