

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL 7
STRUCT DAN ARRAY**



Disusun Oleh :

Haposan Felix Marcel Siregar/ 2311102210

Dosen Pengampu :

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

PROGRAM STUDI S1 TEKNIK INFORMATIKA

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2024

I. DASAR TEORI

Dunia pemrograman memiliki berbagai cara untuk menyimpan dan mengorganisir data, dan di antara cara-cara tersebut, terdapat dua struktur data fundamental yang sering digunakan, yaitu struktur (struct) dan array. Keduanya memiliki peran penting dalam manajemen data, dengan karakteristik serta kegunaan masing-masing.

Pengertian Struktur (Struct)

Struktur adalah kumpulan variabel yang dikelompokkan dalam satu kesatuan dan diidentifikasi dengan sebuah nama. Struktur memungkinkan pengembang untuk mengorganisasikan data yang berhubungan dalam format yang lebih tersusun. Setiap variabel dalam struktur bisa memiliki tipe data yang berbeda, sehingga memberikan fleksibilitas dalam pengelompokan informasi. Misalnya, dalam suatu aplikasi data personel, kita bisa membuat struktur yang mencakup nama (string), umur (integer), dan tinggi badan (float). Dengan mengelompokkan informasi ini ke dalam struktur, pemrogram dapat memudahkan pengelolaan dan manipulasi data terkait individu.

Pengertian Array

Sementara itu, array adalah koleksi data yang terdiri dari elemen-elemen yang memiliki tipe yang sama dan tersimpan dalam urutan kontigu dalam memori komputer. Array ini digunakan untuk menyimpan sejumlah elemen yang diketahui jumlahnya di awal. Contohnya, ketika kita mendeklarasikan sebuah array dengan `int A[5];`, kita menyatakan bahwa A adalah sebuah array berukuran 5 yang berisi bilangan bulat. Keuntungan dari array adalah kemudahan dalam melakukan akses dan manajemen data, karena kita dapat langsung mengakses elemen berdasarkan indeksinya tanpa harus melalui elemen lainnya. Hal ini menjadikan array sangat efisien untuk pemrosesan data yang berulang, terutama dalam algoritma yang membutuhkan pengulangan atau iterasi.

Kesimpulan

Baik struktur maupun array adalah komponen penting dalam pemrograman yang membantu dalam pengelolaan data. Struktur memberikan cara yang lebih fleksibel untuk menyimpan informasi yang berbeda jenis dalam satu kesatuan, sedangkan array menawarkan cara yang efisien untuk menyimpan dan mengakses kumpulan data homogen. Memahami kedua konsep ini memberikan fondasi yang kuat bagi siapa pun yang ingin mendalami lebih lanjut tentang pemrograman dan pengelolaan data dalam berbagai aplikasi. Dengan memanfaatkan struktur dan array secara efektif, pemrogram dapat menciptakan aplikasi yang lebih rapi dan terorganisir.

II. GUIDED

1. Soal Studi Case

Suatu lingkaran didefinisikan dengan koordinat titik pusat (cx, cy) dengan radius r. apabila diberikan dua buah lingkaran, maka tentukan posisi sebuah titik sembarang (x,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 didalam lingkaran 1”, “Titik didalam lingkaran 2” atau “Titik diluar 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 titik 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 titikDiDalamLingkaran(t Titik, lLingkaran) bool { jarak := hitungJarak(t, l.pusat)
return jarak <= float64(l.radius)
}

func identitas (){
fmt.Println("Program Menghitung Jarak Antar Titik Dengan Lingkaran")
fmt.Println("=====")
fmt.Println("Nama: Haposan Siregar")
fmt.Println("NIM: 2311102210")
fmt.Println("=====")
}

func main(){
    identitas()
    // Input untuk lingkaran 1
    var cx1, cy1, r1 int
    fmt.Println("Masukan 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.Println("Masukan 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.Println("Masukan koordinat titik sembarang (x, y): ")
    fmt.Scanln(&x, &y)
    titik := Titik{x: x, y: y}

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

```

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

Screenshot Output

```

40 func main() {
41     // Masukan koordinat pusat dan radius lingkaran 1 (cx1, cy1, r1)
42     fmt.Scanln(&cx1, &cy1, &r1)
43     lingkaran1 := Lingkaran{pusat: Titik{x: cx1, y: cy1}, radius: r1}
44
45     // Input untuk lingkaran 2
46     var cx2, cy2, r2 int
47     fmt.Print("Masukan koordinat pusat dan radius lingkaran 2 (cx2, cy2, r2): ")
48     fmt.Scanln(&cx2, &cy2, &r2)
49     lingkaran2 := Lingkaran{pusat: Titik{x: cx2, y: cy2}, radius: r2}
50
51     var x, y int
52
53     // Masukan koordinat titik sembarang (x, y)
54     fmt.Scanln(&x, &y)
55
56     // Menampilkan hasil sesuai kondisi
57     if diDalam1 && diDalam2 {
58         fmt.Println("Titik di dalam lingkaran 1 dan 2")
59     } else if diDalam1 {
60         fmt.Println("Titik di dalam lingkaran 1")
61     } else if diDalam2 {
62         fmt.Println("Titik di dalam lingkaran 2")
63     } else {
64         fmt.Println("Titik di luar kedua lingkaran")
65     }
66 }

```

```

PS C:\Golang> go run "c:\Golang\unguided63.go"
Program Menghitung Jarak Antar Titik Dengan Lingkaran
=====
Nama: Haposan Siregar
NIM: 2311102210
=====
Masukan koordinat pusat dan radius lingkaran 1 (cx1, cy1, r1): 6 8 14
Masukan koordinat pusat dan radius lingkaran 2 (cx2, cy2, r2): 2 4 5
Masukan koordinat titik sembarang (x, y): 12 8
Titik di dalam lingkaran 1
PS C:\Golang>

```

Deskripsi Program

Program ini dirancang dengan menggunakan struct untuk menyimpan beberapa variabel. Tujuan dari program ini adalah untuk menentukan posisi suatu titik, apakah berada di dalam lingkaran 1, lingkaran 2, kedua lingkaran, atau tidak berada di keduanya. Cara kerja program ini adalah sebagai berikut: pertama, program akan meminta pengguna untuk memasukkan koordinat pusat serta radius dari lingkaran 1 dan 2. Selanjutnya, pengguna diminta untuk memasukkan koordinat titik acak. Setelah itu, program akan melakukan perhitungan menggunakan rumus yang telah disiapkan dan menampilkan hasil sesuai dengan perhitungan tersebut, termasuk kalimat yang menunjukkan posisi titik tersebut.

III. UNGUIDED

1. Soal Studi Case

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 tertentu.

Source code

```
package main

import (
    "fmt"
    "math"
)

func identitas () {
    fmt.Println("=====")
    fmt.Println("Nama: Haposan Siregar")
    fmt.Println("NIM: 231102210")
    fmt.Println("=====")
}

func main() {
    identitas ()
    var bilangan [10]int

    for i := 0; i < 10; i++ {
        bilangan[i] = i
    }

    count := 10
    // Jawaban A
    fmt.Print("A.Array : ")
    for i := 0; i < count; i++ {
        fmt.Printf("%d ", bilangan[i])
    }
}
```

```
}
```

```
// Jawab B
```

```
fmt.Print("\\nB.Ganjil : ")
```

```
for i := 0; i < count; i++ {
```

```
if bilangan[i] % 2 == 1 { fmt.Printf("%d ", bilangan[i])
```

```
}
```

```
}
```

```
// Jawab C
```

```
fmt.Print("\\nC.Genap : ")
```

```
for i := 0; i < count; i++ {
```

```
if bilangan[i] == 0 { fmt.Printf("%d ", bilangan[i])
```

```
} else if bilangan[i] % 2 == 0 { fmt.Printf("%d ", bilangan[i])
```

```
}
```

```
}
```

```
// Jawab D
```

```
fmt.Print("\\nD.Kelipatan : ")
```

```
var kelipatan int
```

```
fmt.Scan(&kelipatan)
```

```
for i := 0; i < count; i++ {
```

```
if bilangan[i] % kelipatan == 0 {
```

```
    fmt.Printf("%d ", bilangan[i])
```

```
}
```

```
}
```

```
// Jawab E
```

```
var slice []int = make([]int, len(bilangan))
```

```
copy(slice, bilangan[:])
```

```
fmt.Print("\\nE. Indeks : ")
```

```
var bilHapus int
```

```
fmt.Scan(&bilHapus)
```

```
// Pengecekan indeks valid
```

```
if bilHapus >= 0 && bilHapus < len(slice) {
```

```
// Hapus elemen pada indeks
```

```
slice = append(slice[:bilHapus], slice[bilHapus+1:]...)
```

```

fmt.Println("Array setelah penghapusan:", slice)
} else {
fmt.Println("Indeks tidak valid")
}

// Jawab F
fmt.Println("\nF.")
var resultAvg int
for i := 0; i < count; i++ { resultAvg += bilangan[i]
}
resultAvg = resultAvg / count
fmt.Printf("Rata rata : %d", resultAvg)

// Jawab G
fmt.Println("\nG.")
var variance float64
for _, val := range bilangan {
variance += math.Pow(float64(val - resultAvg), 2)
}
variance /= float64(count - 1)
stdDev := math.Sqrt(variance)
fmt.Printf("%.2f\n", stdDev)

// Jawab H
fmt.Println("\nH.")
var sliceForH []int = make([]int, len(bilangan))
copy(sliceForH, bilangan[:])
frekuensi := hitungFrekuensiDenganArray(sliceForH)

// Cetak hasil
for i := 0; i <= 100; i++ { if frekuensi[i] > 0 {
    fmt.Printf("Angka %d muncul sebanyak %d kali\n", i, frekuensi[i])
}
}

func hitungFrekuensiDenganArray(data []int) []int {
// Asumsikan nilai maksimum dalam data adalah 100 (sesuaikan jika perlu)
maxVal := 100
freq := make([]int, maxVal+1) // Index 0 sampai maxVal

```



```

for _, val := range data { freq[val]++
}

return freq
}

```

Screenshot Output

```

Angka 8 muncul sebanyak 1 kali
Angka 9 muncul sebanyak 1 kali
PS C:\Golang> go run "C:\Golang\unguided63.go"
=====
Nama: Hapusan Siregar
NIM: 231102210
=====
A.Array : 0 1 2 3 4 5 6 7 8 9
B.Ganjil : 1 3 5 7 9
C.Genap : 0 2 4 6 8
D.Kelipatan : 8
0 8
E. Indeks : 4
Array setelah penghapusan: [0 1 2 3 5 6 7 8 9]
F.Rata rata : 4
G.3.07

H.Angka 0 muncul sebanyak 1 kali
Angka 1 muncul sebanyak 1 kali
Angka 2 muncul sebanyak 1 kali
Angka 3 muncul sebanyak 1 kali
Angka 4 muncul sebanyak 1 kali
Angka 5 muncul sebanyak 1 kali
Angka 6 muncul sebanyak 1 kali
Angka 7 muncul sebanyak 1 kali
Angka 8 muncul sebanyak 1 kali
Angka 9 muncul sebanyak 1 kali
PS C:\Golang>

```

Deskripsi Program

Program ini ditulis menggunakan bahasa pemrograman Go (Golang) dan berfungsi untuk mengelola serta menganalisis data yang tersimpan dalam bentuk array. program menampilkan semua elemen dari array tersebut. Kemudian, program melanjutkan dengan mencari dan mencetak bilangan ganjil yang terdapat dalam array, diikuti oleh pencarian dan pencetakan bilangan genap, termasuk angka 0. Tak hanya itu, program ini juga meminta pengguna untuk memasukkan sebuah bilangan untuk mencari dan mencetak semua elemen dalam array yang merupakan kelipatan dari bilangan tersebut. Selain itu, program membuat slice dengan menyalin elemen dari array dan meminta pengguna untuk memasukkan indeks dari elemen yang ingin dihapus. Jika indeks yang dimasukkan valid, elemen tersebut akan dihapus, dan slice yang telah diperbarui akan ditampilkan. Selanjutnya, program menghitung dan mencetak rata-rata dari elemen-elemen dalam array, kemudian melanjutkan dengan menghitung dan mencetak deviasi standar berdasarkan rata-rata yang telah dihitung sebelumnya. Sebagai bagian terakhir dari proses, program menghitung frekuensi kemunculan setiap angka dari 0 hingga 100, kemudian mencetak frekuensi tersebut jika angka itu muncul lebih dari sekali. Dengan begitu, program ini memberikan contoh yang komprehensif tentang pengolahan data dalam array, sekaligus membantu pengguna untuk memahami berbagai konsep dasar dalam pemrograman seperti pengulangan, kontrol alur, dan penggunaan fungsi. Program ini juga memiliki potensi untuk

dimodifikasi lebih lanjut, sehingga dapat menangani data yang lebih kompleks atau memenuhi kebutuhan analisis data tertentu.

2. Soal Studi Case

Buatlah sebuah program untuk menentukan siapa pemenang dari histori pertandingan dua klub, Pertama program akan meminta untuk memasukan kedua klub yang dipertandingkan dan akan dimulai untuk memasukan jumlah gol yang tercipta, mulai dari pertandingan 1, gol untuk tim 1 dan selanjutnya gol untuk tim 2 begitu terus sampai berhenti. Program akan berhenti ketika gol tim 1 atau tim 2 memiliki nilai minus(-).

Sourcecode

```
package main

import "fmt"
type Klub struct {
    skor1 int
    skor2 int
    hasil string
}

func identitas() {
    fmt.Println("=====")
    fmt.Println("Nama: Haposan Siregar")
    fmt.Println("NIM: 231102210")
    fmt.Println("=====")
}

func main() {
    identitas() // Menampilkan identitas

    var klub1, klub2 string
    var jumlahMatch int
    var match [100]Klub
    fmt.Print("Masukkan Klub 1: ")
    fmt.Scanln(&klub1)
    fmt.Print("Masukkan Klub 2: ")
    fmt.Scanln(&klub2)
```

```

for i := 0; i++ {
    fmt.Printf("Skor Pertandingan %d : ", i+1)
    fmt.Printf("\n %s : ", klub1)
    fmt.Scanln(&match[i].skor1)
    fmt.Printf("%s : ", klub2)
    fmt.Scanln(&match[i].skor2)

    if match[i].skor1 < 0 || match[i].skor2 < 0 {
        fmt.Print("Pertandingan Selesai")
        break
    }

    if match[i].skor1 > match[i].skor2 {
        match[i].hasil = klub1
    } else if match[i].skor1 == match[i].skor2 {
        match[i].hasil = "Draw"
    } else {
        match[i].hasil = klub2
    }

    jumlahMatch++
}

fmt.Println("\nHasil pertandingan:")
for i := 0; i < jumlahMatch; i++ {
    fmt.Printf("Hasil pertandingan %d: %s\n", i+1, match[i].hasil)
}
}

```

The image shows a code editor with a Go program for a football match simulation. The code is as follows:

```
16
17
18 Codeium: Refactor | Explain | Generate GoDoc | X
19 func main() {
20     identitas() // Menampilkan identitas
21
22     var klub1, klub2 string
23     var jumlahMatch int
24     var match [100]Klub
25     fmt.Println("Masukkan Klub 1:")
26     fmt.Scanln(&klub1)
27     fmt.Println("Masukkan Klub 2:")
28     fmt.Scanln(&klub2)
29
30     for i := 0; i++ {
31         fmt.Println("Skor Pertandingan %d:", i+1)
32         fmt.Println("1m %s:", klub1)
33         fmt.Scanln(&match[i].skor1)
34         fmt.Println("2m %s:", klub2)
35         fmt.Scanln(&match[i].skor2)
36     }
37
38     // Menampilkan skor pertandingan
39     for i := 0; i++ {
40         fmt.Println("Skor Pertandingan %d:", i+1)
41         fmt.Println("Klub 1: %s", klub1)
42         fmt.Println("Klub 2: %s", klub2)
43         fmt.Println("Skor 1: %d", match[i].skor1)
44         fmt.Println("Skor 2: %d", match[i].skor2)
45     }
46 }
47
48 // Menampilkan identitas
49 func identitas() {
50     fmt.Println("Nama: Rapasan Siregar")
51     fmt.Println("NIM: 2311102210")
52 }
53
54 // Struktur data Klub
55 type Klub struct {
56     skor1 int
57     skor2 int
58 }
```

The terminal output shows the execution of the program:

```
PS C:\Golang> go run "c:\Golang\unguided63.go"
-----
Nama: Rapasan Siregar
NIM: 2311102210
-----
Masukkan Klub 1: Madrid
Masukkan Klub 2: Barca
Skor Pertandingan 1 :
Madrid : 10
Barca : 0
Skor Pertandingan 2 :
Madrid : 9
Barca : 1
Skor Pertandingan 3 :
Madrid : -2
Barca : -9
Pertandingan Selesai
Hasil pertandingan:
Hasil pertandingan 1: Madrid
Hasil pertandingan 2: Madrid
PS C:\Golang>
```

Program ini berfungsi sebagai simulasi pertandingan sepak bola antara dua klub, di mana pengguna memiliki kesempatan untuk memasukkan skor untuk setiap pertandingan yang berlangsung. Proses dimulai dengan pengguna diminta untuk mengisi nama kedua klub yang akan bertanding. Selanjutnya, setelah nama klub dimasukkan, pengguna diminta untuk mencatat skor dari setiap pertandingan antara kedua klub tersebut.

Setelah semua pertandingan selesai dicatat, program akan menampilkan hasil dari setiap pertandingan, termasuk siapa pemenangnya atau jika hasilnya seri. Dalam implementasinya, program menggunakan array untuk menyimpan skor dan hasil dari setiap pertandingan, serta memanfaatkan perulangan agar pengguna dapat terus memasukkan skor hingga memutuskan untuk berhenti.

3. Soal Studi Case

Buatlah program dari kode yang sudah disiapkan dan melengkapi potongan potongan kode yang sudah disiapkan, tujuan dari ini adalah untuk melakukan pengecekan kalimat yang diinputkan polindrom atau bukan.

Sourcecode

```
package main

import (
    "bufio"
    "fmt"
    "os"
    "strings"
)

const NMAX int = 127
type tabel struct {
    tab [NMAX]rune
    m int
}

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++ {
        t.tab[i], t.tab[n-i-1] = t.tab[n-i-1], t.tab[i]
    }
}

func isiArray(t *tabel) {
    reader := bufio.NewReader(os.Stdin)
    fmt.Print("Masukkan teks: ")
    input, _ := reader.ReadString('\n')
```

```

input = strings.TrimSpace(input)

t.m = len(input)
for i, c := range input { t.tab[i] = c
}
}

func identitas () {
    fmt.Println ("=====")
    fmt.Println ("Nama: Haposan Siregar")
    fmt.Println ("NIM: 2311102210")
    fmt.Println ("=====")
}

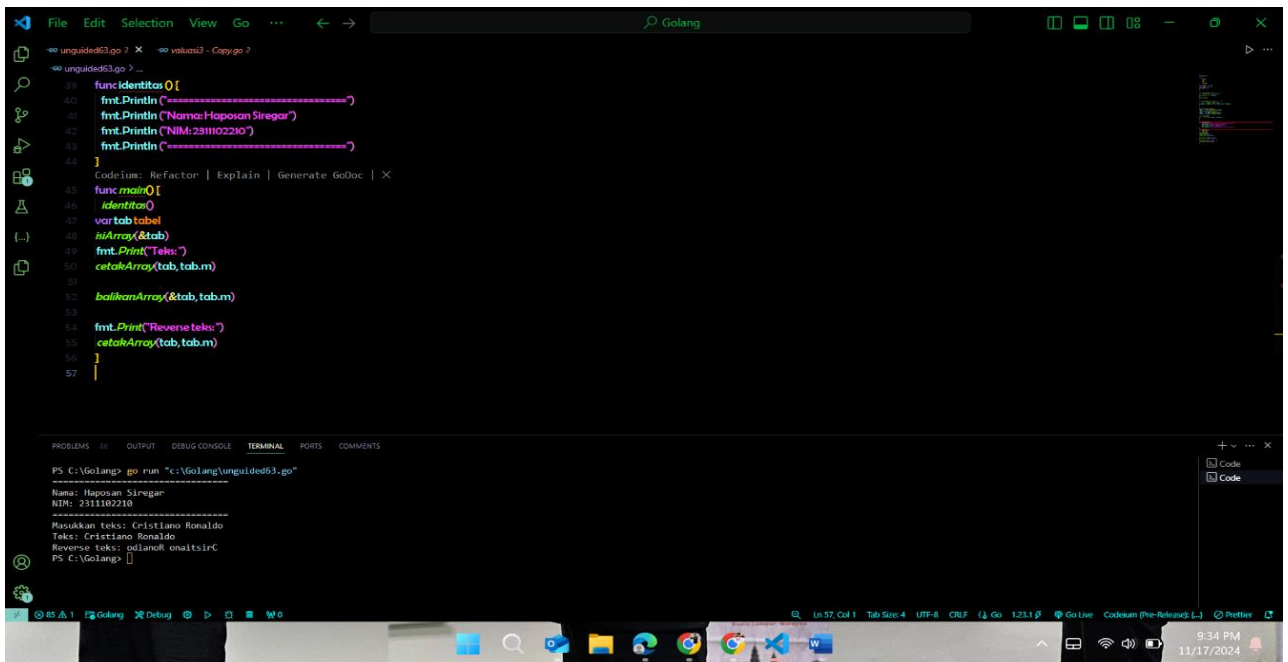
func main() {
    identitas()
    var tab tabel
    isiArray(&tab)
    fmt.Print("Teks: ")
    cetakArray(tab, tab.m)

    balikanArray(&tab, tab.m)

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

```

Screenshoot Program



The screenshot shows a Go program in a dark-themed editor. The code defines a `func identitas()` function that prints a name and NIM, and a `func main()` function that calls `identitas()`, prompts for text, and calls `cetakArray()` and `balikanArray()`. The terminal at the bottom shows the execution output.

```
19 func identitas() {  
20     fmt.Println("=====")  
21     fmt.Println("Nama: Haposen Siregar")  
22     fmt.Println("NIM: 2311102210")  
23     fmt.Println("=====")  
24 }  
25  
26 func main() {  
27     identitas()  
28     var tab tabel  
29     isiArray(&tab)  
30     fmt.Println("Teks:")  
31     cetakArray(tab, tab.m)  
32  
33     balikanArray(&tab, tab.m)  
34  
35     fmt.Println("Reverse teks")  
36     cetakArray(tab, tab.m)  
37 }
```

Terminal Output:

```
PS C:\Golang> go run "c:\Golang\unguided63-go"  
-----  
Nama: Haposen Siregar  
NIM: 2311102210  
-----  
Masukkan teks: Cristiano Ronaldo  
Teks: Cristiano Ronaldo  
Reverse teks: olanoR onailsiR  
PS C:\Golang>
```

Deskripsi Program

Program ini bertujuan untuk menerima input teks dari pengguna, menampilkan teks itu, kemudian membalik urutan karakter dalam teks dan menampilkan kembali. Berikut adalah penjelasan detail mengenai fungsionalitasnya:

1. **Fungsi isiArray:** Fungsi ini berfungsi untuk mengambil teks dari pengguna melalui buffer. Karakter-karakter dari teks tersebut disimpan dalam array dan panjang teks dicatat dalam variabel `m`.
2. **Fungsi cetakArray:** Tugas dari fungsi ini adalah menampilkan elemen-elemen dari array yang telah diisi berdasarkan panjang yang ditentukan (`n`).
3. **Fungsi balikanArray:** Fungsi ini membalik urutan karakter dalam array dengan metode menukar elemen pertama dengan yang terakhir, elemen kedua dengan yang kedua dari belakang, dan seterusnya.

Pada langkah awal, program membaca teks yang dimasukkan oleh pengguna dan menyimpannya dalam array. Setelah itu, program menampilkan teks asli yang diketikan. Kemudian, teks dibalik menggunakan fungsi `balikanArray` dan pada akhirnya, program menampilkan teks yang sudah dibalik tersebut.