

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL VII
STRUCT & ARRAY**



Oleh :

NAMA : Felix Pedrosa Valentino

NIM : 103112400056

KELAS : IF – 12 – 01

**S1 TEKNIK INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO**

2025

I. DASAR TEORI

Tipe Bentukan

Tipe bentukan memungkinkan pemrograman untuk mendefinisikan tipe data baru dalam suatu bahasa pemrograman. Tipe bentukan ini dibedakan menjadi dua kategori, yaitu Alias dan Struct.

1) Alias (Type)

Bahasa pemrograman umumnya memungkinkan kita untuk menetapkan nama baru yang lebih ringkas dan akrab untuk suatu tipe data. Misalnya, tipe "integer" dapat diberi nama alias "bilangan". Proses ini dilakukan dengan menggunakan kata kunci "type".

2) Struct atau Record

Struct merupakan sekumpulan definisi variabel (atau properti) serta fungsi (atau metode) yang dibuat dan dibungkus menjadi sebuah tipe data baru dengan nama tertentu.

Struct atau structure memungkinkan kita untuk mengelompokkan beberapa data atau nilai yang memiliki hubungan tertentu ke dalam satu kesatuan. Masing-masing nilai akan disimpan dalam field dari structure tersebut.

Berbeda dengan bahasa pemrograman lainnya, kesamaan tipe antara dua variabel yang merupakan struktur tidak ditentukan oleh nama, melainkan oleh strukturnya. Dua variabel yang memiliki nama field dan tipe field yang sama, serta berada dalam urutan yang sama, dianggap memiliki tipe yang serupa. Tentu saja, ini akan sangat memudahkan jika struktur tersebut didefinisikan sebagai tipe baru, sehingga kita tidak perlu menuliskan seluruh field-nya berulang kali saat melakukan deklarasi.

Array

Array adalah sekumpulan data dengan tipe yang sama yang disimpan dalam sebuah variabel. Saat dibuat, array memiliki kapasitas

yang sudah ditentukan, sehingga jumlah elemen atau data yang dapat disimpan di dalamnya tidak boleh melebihi kapasitas yang telah ditentukan sebelumnya.

Nilai default dari setiap elemen array pada awalnya bergantung pada tipe datanya. Misalnya, jika tipe data adalah integer (int), maka nilai awal setiap elemen adalah 0; jika tipe data adalah boolean (bool), maka nilai awalnya adalah false, dan seterusnya. Setiap elemen dalam array memiliki indeks yang berupa angka, yang menggambarkan posisi urutan elemen tersebut. Perlu diingat bahwa indeks array dimulai dari angka 0.

Slice (Array Dinamis)

Slice merupakan elemen referensi dari sebuah array. Kita dapat membuat slice baru atau menghasilkan slice dari manipulasi array atau slice lainnya. Karena slice adalah tipe data yang berbasis referensi, perubahan yang dilakukan pada elemen di dalam sebuah slice akan berpengaruh pada slice lain yang memiliki alamat memori yang sama.

Dalam bahasa Go, array juga dapat memiliki ukuran dinamis. Meskipun tidak digunakan dalam kelas Algoritma Pemrograman, deklarasi slice mirip dengan deklarasi array, namun jumlah elemennya dibiarkan kosong.

Kita dapat menggunakan fungsi bawaan `len` untuk mengetahui ukuran slice. Selain itu, fungsi `cap` dapat digunakan untuk mengetahui total kapasitas yang disediakan untuk slice tersebut.

Fungsi bawaan `append` memungkinkan kita untuk menambahkan elemen ke dalam slice dan, jika diperlukan, memperbesar kapasitas slice tersebut. Selain itu, sebuah slice baru juga dapat dibuat dengan mengambil bagian dari array atau slice lain.

II. GUIDED

1. Guided 1

Source Code :

```
// Felix Pedrosa V

package main

import (
    "fmt"
    "time"
)

// Struct untuk barang dalam struk belanja
type Item struct {
    Name    string
    Price   float64
    Quantity int
}

// Struct untuk struk belanja
type Receipt struct {
    StoreInfo string
    Date      time.Time
    Items     []Item
    TotalAmount float64
}

// Method untuk menghitung total harga semua item
func (r *Receipt) CalculateTotal() {
    var total float64
    for _, item := range r.Items {
        total += item.Price * float64(item.Quantity)
    }
    r.TotalAmount = total
}

// Method untuk mencetak struk belanja
func (r Receipt) PrintReceipt() {
```

```

    fmt.Println("=====
")
    fmt.Println(r.StoreInfo)
    fmt.Println("Tanggal:", r.Date.Format("02-01-2006 15:04"))
    fmt.Println("=====
")
    fmt.Printf("%-15s %-10s %-8s %-10s\n", "Item", "Harga", "Jumlah",
"Total")
    fmt.Println("-----")

    for _, item := range r.Items {
        itemTotal := item.Price * float64(item.Quantity)
        fmt.Printf("%-15s Rp%-9.2f %-8d Rp%-9.2f\n", item.Name,
item.Price, item.Quantity, itemTotal)
    }

    fmt.Println("=====
")
    fmt.Printf("%-35s Rp%-9.2f\n", "Total Belanja:", r.TotalAmount)
    fmt.Println("=====
")
    fmt.Println("Terima kasih telah berbelanja!")
}

func main() {
    receipt := Receipt{
        StoreInfo: "Toko Sembako Makmur\nJl. Raya No. 123, Jakarta",
        Date:      time.Now(),
        Items: []Item{
            {Name: "Beras", Price: 12000, Quantity: 5},
            {Name: "Gula", Price: 15000, Quantity: 2},
            {Name: "Minyak", Price: 20000, Quantity: 1},
            {Name: "Telur", Price: 2000, Quantity: 10},
        },
    }

    receipt.CalculateTotal()
    receipt.PrintReceipt()
}

```

Output :

```
PS D:\KULIAH\SEMESTER 2\Algoritma & Pemrograman 2\CODING - GOLANG - Alpro 2> go run "d:\KULIAH\SEMESTER 2\Algoritma & Pemrograman 2\CODING - GOLANG - Alpro 2\alpro2_week8_modul17\103112400056_Guided1(struct)\103112400056_Guided1.go"
=====
Toko Sembako Makmur
Jl. Raya No. 123, Jakarta
Tanggal: 20-04-2025 17:17
=====
Item          Harga      Jumlah    Total
-----
Beras         Rp12000.00  5         Rp60000.00
Gula          Rp15000.00  2         Rp30000.00
Minyak        Rp20000.00  1         Rp20000.00
Telur         Rp2000.00   10        Rp20000.00
=====
Total Belanja:                Rp130000.00
=====
Terima kasih telah berbelanja!
```

Penjelasan Program :

Program diatas ditulis menggunakan bahasa Go dan bertujuan untuk mencetak struk belanja. Terdapat dua struktur utama dalam aplikasi ini: yang pertama adalah Item, yang merepresentasikan barang dengan atribut nama, harga, dan jumlah. Yang kedua adalah Receipt, yang menyimpan informasi mengenai toko, tanggal transaksi, daftar item, serta total belanja.

Dalam program ini, terdapat dua metode utama. Metode pertama, CalculateTotal, berfungsi untuk menghitung total harga semua item yang dibeli. Metode kedua, PrintReceipt, digunakan untuk mencetak struk belanja dengan format yang rapi.

Pada fungsi main, program ini membuat objek Receipt yang berisi informasi tentang toko serta daftar item yang dibeli. Setelah itu, total belanja dihitung dan struk tersebut dicetak.

2. Guided 2

Source Code :

```
// Felix Pedrosa V

package main

import (
    "fmt"
)

func main() {
    // Deklarasi dan inisialisasi array nilai mahasiswa
```

```

nilaiMahasiswa := [5]int{85, 90, 78, 88, 95}

fmt.Println("Data Nilai Mahasiswa:")
fmt.Println("=====")

// Menampilkan nilai per mahasiswa
for i, nilai := range nilaiMahasiswa {
    fmt.Printf("Mahasiswa %d: %d\n", i+1, nilai)
}

// Menghitung rata-rata nilai
var total int
for _, nilai := range nilaiMahasiswa {
    total += nilai
}
rataRata := float64(total) / float64(len(nilaiMahasiswa))

fmt.Println("=====")
fmt.Printf("Rata-rata nilai: %.2f\n", rataRata)

// Mencari nilai tertinggi dan terendah
tertinggi := nilaiMahasiswa[0]
terendah := nilaiMahasiswa[0]

for _, nilai := range nilaiMahasiswa {
    if nilai > tertinggi {
        tertinggi = nilai
    }
    if nilai < terendah {
        terendah = nilai
    }
}

fmt.Printf("Nilai tertinggi: %d\n", tertinggi)
fmt.Printf("Nilai terendah: %d\n", terendah)

// Contoh array 2 dimensi
fmt.Println("\nContoh Array 2 Dimensi:")
fmt.Println("=====")

```

```
// Nilai ujian mahasiswa dalam 2 mata kuliah (Matematika, Bahasa)
nilaiUjian := [3][2]int{
    {80, 85},
    {90, 75},
    {70, 95},
}

// Menampilkan nilai ujian per mahasiswa
fmt.Println("Nilai Ujian Mahasiswa (Matematika, Bahasa):")
for i, nilai := range nilaiUjian {
    fmt.Printf("Mahasiswa %d: Matematika = %d, Bahasa = %d\n", i+1,
nilai[0], nilai[1])
}
}
```

Output :

```
PS D:\KULIAH\SEMESTER 2\Algoritma & Pemrograman 2\CODING - GOLANG - Alpro 2> go run "d:\KULIAH\SEMESTER 2\Algoritma & Pemrograman 2\CODING -
GOLANG - Alpro 2\alpro2_week8_modul17\103112400056_guided2(array)\103112400056_guided2.go"
Data Nilai Mahasiswa:
=====
Mahasiswa 1: 85
Mahasiswa 2: 90
Mahasiswa 3: 78
Mahasiswa 4: 88
Mahasiswa 5: 95
=====
Rata-rata nilai: 87.20
Nilai tertinggi: 95
Nilai terendah: 78

Contoh Array 2 Dimensi:
=====
Nilai Ujian Mahasiswa (Matematika, Bahasa):
Mahasiswa 1: Matematika = 80, Bahasa = 85
Mahasiswa 2: Matematika = 90, Bahasa = 75
Mahasiswa 3: Matematika = 70, Bahasa = 95
```

Penjelasan Program :

Program diatas ditulis menggunakan bahasa Go yang bertujuan untuk mengelola dan menganalisis nilai mahasiswa. Di dalam aplikasi ini, terdapat sebuah array satu dimensi yang menyimpan nilai dari lima mahasiswa, yang kemudian ditampilkan secara berurutan. Program ini juga dapat menghitung rata-rata nilai serta menentukan nilai tertinggi dan terendah di antara nilai-nilai yang ada.

Selain itu, terdapat pula contoh penggunaan array dua dimensi untuk menyimpan nilai ujian mahasiswa di dua mata kuliah, yaitu Matematika dan Bahasa. Nilai ujian ditampilkan dengan format yang jelas, memudahkan pengguna untuk melihat performa akademik setiap mahasiswa.

III. UNGUIDED

1. UnGuided 1

Source Code :

```
// Felix Pedrosa V

package main

import (
    "fmt"
    "math"
)

// Fungsi untuk menghitung jarak antara dua titik
func hitungJarak(x1, y1, x2, y2 float64) float64 {
    return math.Sqrt(math.Pow(x1-x2, 2) + math.Pow(y1-y2, 2))
}

// Fungsi untuk memeriksa apakah titik berada di dalam lingkaran
func titikDalamLingkaran(centerX, centerY, radius, pointX, pointY
float64) bool {
    return hitungJarak(centerX, centerY, pointX, pointY) <= radius
}

func main() {
    var centerX1, centerY1, radius1 float64
    var centerX2, centerY2, radius2 float64
    var pointX, pointY float64

    // Input untuk lingkaran pertama
    fmt.Scan(&centerX1, &centerY1, &radius1)
    // Input untuk lingkaran kedua
    fmt.Scan(&centerX2, &centerY2, &radius2)
    // Input untuk titik
    fmt.Scan(&pointX, &pointY)

    // Memeriksa apakah titik berada di dalam lingkaran
    isInCircle1 := titikDalamLingkaran(centerX1, centerY1,
```

```

radius1, pointX, pointY)
    isInCircle2 := titikDalamLingkaran(centerX2, centerY2,
radius2, pointX, pointY)

// Menentukan dan mencetak hasil
if isInCircle1 && isInCircle2 {
    fmt.Println("Titik di dalam lingkaran 1 dan 2")
} else if isInCircle1 {
    fmt.Println("Titik di dalam lingkaran 1")
} else if isInCircle2 {
    fmt.Println("Titik di dalam lingkaran 2")
} else {
    fmt.Println("Titik di luar lingkaran 1 dan 2")
}
}

```

Output :

```

PS D:\KULIAH\SEMESTER 2\Algoritma & Pemrograman 2\CODING - GOLANG - Alpro 2> go run "d:\KULIAH\SEMESTER 2\Algoritma & Pemrograman 2\CODING -
GOLANG - Alpro 2\alpro2_week8_modul7\103112400056_Unguided1\103112400056_Unguided1.go"
1 1 5
8 8 4
2 2
Titik di dalam lingkaran 1
PS D:\KULIAH\SEMESTER 2\Algoritma & Pemrograman 2\CODING - GOLANG - Alpro 2> go run "d:\KULIAH\SEMESTER 2\Algoritma & Pemrograman 2\CODING -
GOLANG - Alpro 2\alpro2_week8_modul7\103112400056_Unguided1\103112400056_Unguided1.go"
1 1 5
8 8 4
15 20
Titik di luar lingkaran 1 dan 2

```

Penjelasan Program :

Program diatas ditulis menggunakan bahasa Go yang bertujuan untuk menentukan posisi sebuah titik relatif terhadap dua lingkaran yang telah ditentukan. Dengan memanfaatkan fungsi hitungJarak, program ini menghitung jarak antara pusat lingkaran dan titik yang ingin diperiksa. Selanjutnya, fungsi titikDalamLingkaran akan memverifikasi apakah titik tersebut berada di dalam lingkaran berdasarkan jarak yang telah dihitung dan radius masing-masing lingkaran. Di dalam fungsi utama, pengguna akan diminta untuk memasukkan koordinat pusat serta radius dari kedua lingkaran, disertai dengan koordinat titik yang ingin dianalisis. Setelah melakukan pengecekan, program ini akan menampilkan hasil yang menunjukkan apakah titik tersebut berada di dalam salah satu atau kedua lingkaran, atau justru berada di luar keduanya.

2. UnGuided 2

Source Code :

```
// Felix Pedrosa V

package main

import (
    "fmt"
    "math"
)

func main() {
    var jumlahElemen int
    fmt.Print("Masukkan jumlah elemen array: ")
    fmt.Scan(&jumlahElemen)
    dataArray := make([]int, jumlahElemen)

    // Input elemen array
    for i := 0; i < jumlahElemen; i++ {
        fmt.Printf("Masukkan elemen ke-%d: ", i)
        fmt.Scan(&dataArray[i])
    }

    // a. Menampilkan semua elemen array
    fmt.Println("\na. Menampilkan semua elemen array:")
    fmt.Println(dataArray)

    // b. Menampilkan elemen dengan indeks ganjil
    fmt.Println("b. Elemen dengan indeks ganjil:")
    for i := 1; i < len(dataArray); i += 2 {
        fmt.Printf("Indeks %d: %d\n", i, dataArray[i])
    }

    // c. Menampilkan elemen dengan indeks genap
    fmt.Println("c. Elemen dengan indeks genap:")
    for i := 0; i < len(dataArray); i += 2 {
        fmt.Printf("Indeks %d: %d\n", i, dataArray[i])
    }

    // d. Menampilkan elemen dengan indeks kelipatan x
    var kelipatan int
```

```

    fmt.Print("d. Masukkan nilai x (untuk mencari indeks kelipatan
x): ")
    fmt.Scan(&kelipatan)
    fmt.Printf("  Elemen dengan indeks kelipatan %d:\n",
kelipatan)
    for i := 0; i < len(dataArray); i++ {
        if i%kelipatan == 0 {
            fmt.Printf("Indeks %d: %d\n", i, dataArray[i])
        }
    }

// e. Menghapus elemen pada indeks tertentu
var indeksUntukHapus int
fmt.Print("e. Masukkan indeks yang ingin dihapus: ")
fmt.Scan(&indeksUntukHapus)
if indeksUntukHapus >= 0 && indeksUntukHapus <
len(dataArray) {
    dataArray = append(dataArray[:indeksUntukHapus],
dataArray[indeksUntukHapus+1:]...)
    fmt.Println("  Isi array setelah elemen dihapus:")
    fmt.Println(dataArray)
} else {
    fmt.Println("  Indeks tidak valid!")
}

// f. Menampilkan rata-rata
totalNilai := 0
for _, nilai := range dataArray {
    totalNilai += nilai
}
rataRata := float64(totalNilai) / float64(len(dataArray))
fmt.Printf("f. Rata-rata dari elemen array: %.2f\n", rataRata)

// g. Menampilkan simpangan baku (standar deviasi)
var totalSelisihKuadrat float64
for _, nilai := range dataArray {
    selisih := float64(nilai) - rataRata
    totalSelisihKuadrat += selisih * selisih
}
simpanganBaku := math.Sqrt(totalSelisihKuadrat /

```

```

float64(len(dataArray)))
    fmt.Printf("g. Simpangan baku dari elemen array: %.2f\n",
simpanganBaku)

    // h. Menampilkan frekuensi kemunculan bilangan tertentu
    var bilanganDicari int
    fmt.Print("h. Masukkan bilangan yang ingin dicari
frekuensinya: ")
    fmt.Scan(&bilanganDicari)
    frekuensiKemunculan := 0
    for _, nilai := range dataArray {
        if nilai == bilanganDicari {
            frekuensiKemunculan++
        }
    }
    fmt.Printf(" Frekuensi kemunculan angka %d: %d kali\n",
bilanganDicari, frekuensiKemunculan)
}

```

Output :

```

PS D:\KULIAH\SEMESTER 2\Algoritma & Pemrograman 2\CODING - GOLANG - Alpro 2> go run "d:\KULIAH\SEMESTER 2\Algoritma & Pemrograman 2\CODING - GOLANG - Alpro 2\alpro2_week8_modul17\10311240056_Unguided2\10311240056_Unguided2.go"
Masukkan jumlah elemen array: 4
Masukkan elemen ke-0: 2
Masukkan elemen ke-1: 4
Masukkan elemen ke-2: 6
Masukkan elemen ke-3: 8

a. Menampilkan semua elemen array:
[2 4 6 8]
b. Elemen dengan indeks ganjil:
Indeks 1: 4
Indeks 3: 8
c. Elemen dengan indeks genap:
Indeks 0: 2
Indeks 2: 6
d. Masukkan nilai x (untuk mencari indeks kelipatan x): 2
Elemen dengan indeks kelipatan 2:
Indeks 0: 2
Indeks 2: 6
e. Masukkan indeks yang ingin dihapus: 2
Isi array setelah elemen dihapus:
[2 4 8]
f. Rata-rata dari elemen array: 4.67
g. Simpangan baku dari elemen array: 2.49
h. Masukkan bilangan yang ingin dicari frekuensinya: 4
Frekuensi kemunculan angka 4: 1 kali

```

Penjelasan Program :

Program diatas ditulis menggunakan bahasa Go yang bertujuan untuk mengelola dan menganalisis sekumpulan bilangan bulat yang disimpan dalam sebuah array. Pengguna diminta untuk memasukkan jumlah elemen yang ingin dimasukkan ke dalam array, diikuti dengan nilai-nilai dari elemen-elemen tersebut. Setelah itu, program menawarkan berbagai fungsi analisis, seperti menampilkan semua elemen, elemen

dengan indeks ganjil dan genap, serta elemen dengan indeks kelipatan tertentu yang ditentukan oleh pengguna.

Selain itu, program ini memberi kesempatan kepada pengguna untuk menghapus elemen pada indeks tertentu dan menampilkan isi array setelah penghapusan dilakukan. Program juga menghitung dan menampilkan rata-rata serta simpangan baku (standar deviasi) dari elemen-elemen dalam array. Di akhir program, pengguna dapat mencari frekuensi kemunculan bilangan tertentu di dalam array.

3. UnGuided 3

Source Code :

```
// Felix Pedrosa V

package main

import (
    "fmt"
)

func main() {
    var teamA, teamB string
    var scoreA, scoreB int
    var results []string

    // Input nama klub
    fmt.Print("Masukkan nama Klub A: ")
    fmt.Scanln(&teamA)
    fmt.Print("Masukkan nama Klub B: ")
    fmt.Scanln(&teamB)

    matchNumber := 1
    for {
        fmt.Printf("Skor Pertandingan %d: ", matchNumber)
        fmt.Scan(&scoreA, &scoreB)

        // Menghentikan input jika skor negatif
        if scoreA < 0 || scoreB < 0 {
            break
        }
    }
}
```

```

    }

    // Menentukan hasil pertandingan
    if scoreA > scoreB {
        results = append(results, teamA)
    } else if scoreB > scoreA {
        results = append(results, teamB)
    } else {
        results = append(results, "Draw")
    }
    matchNumber++
}

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

```

Output :

```

PS D:\KULIAH\SEMESTER 2\Algoritma & Pemrograman 2\CODING - GOLANG - Alpro 2> go run "d:\KULIAH\SEMESTER 2\Algoritma & Pemrograman 2\CODING - GOLANG - Alpro 2\alpro2_week8_modul7\10311240056_Unguided3\10311240056_Unguided3.go"
Masukkan nama Klub A: MU
Masukkan nama Klub B: Inter
Skor Pertandingan 1: 2 0
Skor Pertandingan 2: 1 2
Skor Pertandingan 3: 2 2
Skor Pertandingan 4: 0 1
Skor Pertandingan 5: 3 2
Skor Pertandingan 6: 1 0
Skor Pertandingan 7: 5 2
Skor Pertandingan 8: 2 3
Skor Pertandingan 9: -1 2
Hasil Pertandingan 1: MU
Hasil Pertandingan 2: Inter
Hasil Pertandingan 3: Draw
Hasil Pertandingan 4: Inter
Hasil Pertandingan 5: MU
Hasil Pertandingan 6: MU
Hasil Pertandingan 7: MU
Hasil Pertandingan 8: Inter
Pertandingan selesai

```

Penjelasan Program :

Program diatas ditulis menggunakan bahasa Go yang bertujuan untuk merekam dan menampilkan hasil pertandingan antara dua klub sepak bola. Pengguna diminta untuk memasukkan nama kedua klub, diikuti dengan skor pertandingan yang berlangsung. Program akan terus meminta input skor hingga pengguna memasukkan nilai negatif, yang menandakan akhir dari proses input. Setelah setiap skor dimasukkan, program akan menentukan hasil pertandingan, apakah Klub A menang, Klub B menang,

atau jika hasilnyaimbang (draw) dan menyimpan informasi tersebut dalam sebuah daftar. Setelah semua skor tercatat, program akan menampilkan hasil dari setiap pertandingan yang telah dicatat.

4. UnGuided 4

Source Code :

```
// Felix Pedrosa V

package main

import (
    "fmt"
)

const NMAX int = 127

type table [NMAX]rune

// Fungsi untuk mengisi array dengan karakter
func fillArray(t *table, count *int) {
    var character rune
    *count = 0
    fmt.Print("Masukkan teks: ")
    for {
        fmt.Scanf("%c", &character)

        // Hentikan jika karakter adalah titik
        if character == '.' || *count >= NMAX {
            break
        }

        // Abaikan spasi dan newline
        if character != ' ' && character != '\n' {
            (*t)[*count] = character
            *count++
        }
    }
}
```



```

// Fungsi untuk mencetak array
func printArray(t table, count int) {
    for i := 0; i < count; i++ {
        fmt.Printf("%c ", t[i])
    }
    fmt.Println()
}

// Fungsi untuk membalikkan array
func reverseArray(t *table, count int) {
    for i := 0; i < count/2; i++ {
        t[i], t[count-1-i] = t[count-1-i], t[i]
    }
}

// Fungsi untuk memeriksa apakah array adalah palindrom
func isPalindrome(t table, count int) bool {
    var copyTable table
    copy(copyTable[:], t[:])
    reverseArray(&copyTable, count)
    for i := 0; i < count; i++ {
        if t[i] != copyTable[i] {
            return false
        }
    }
    return true
}

func main() {
    var charTable table
    var length int

    fillArray(&charTable, &length)

    fmt.Print("Teks: ")
    printArray(charTable, length)

    fmt.Printf("Apakah palindrom? %v\n", isPalindrome(charTable,
length))

```

```
}
```

Output :

```
PS D:\KULIAH\SEMESTER 2\Algoritma & Pemrograman 2\CODING - GOLANG - Alpro 2> go run "d:\KULIAH\SEMESTER 2\Algoritma & Pemrograman 2\CODING - GOLANG - Alpro 2\alpro2_week8_modul17\10311240056_Unguided4\10311240056_Unguided4.go"
Masukkan teks: K A T A K .
Teks: K A T A K
Apakah palindrom? true
PS D:\KULIAH\SEMESTER 2\Algoritma & Pemrograman 2\CODING - GOLANG - Alpro 2> go run "d:\KULIAH\SEMESTER 2\Algoritma & Pemrograman 2\CODING - GOLANG - Alpro 2\alpro2_week8_modul17\10311240056_Unguided4\10311240056_Unguided4.go"
Masukkan teks: S E N A N G .
Teks: S E N A N G
Apakah palindrom? false
```

Penjelasan Program :

Program diatas ditulis menggunakan bahasa Go yang bertujuan untuk mengelola dan menganalisis sekumpulan karakter yang dimasukkan oleh pengguna. Pengguna akan diminta untuk menginput teks karakter satu per satu hingga mencapai tanda titik (.) atau hingga batas maksimum karakter yang telah ditentukan. Karakter yang telah dimasukkan akan disimpan dalam sebuah array. Setelah semua karakter terinput, program akan menampilkan isi array tersebut. Selanjutnya, program akan memeriksa apakah urutan karakter dalam array tersebut membentuk palindrom, yaitu kata atau frasa yang dapat dibaca sama baik dari depan maupun dari belakang. Hasil pemeriksaan palindrom ini kemudian akan disampaikan kepada pengguna.

IV. KESIMPULAN

Modul ini memperkenalkan berbagai konsep penting dalam pemrograman Golang, khususnya mengenai tipe bentukan. Dalam konteks bahasa Go, tipe bentukan mencakup alias dan struktur (struct), yang memungkinkan para pemrogram untuk menciptakan tipe data baru yang lebih sesuai dengan kebutuhan aplikasi mereka. Dengan menggunakan alias, kita dapat mengganti nama tipe data yang ada menjadi lebih ringkas dan mudah dipahami. Sementara itu, struktur memungkinkan kita untuk mengelompokkan beberapa nilai yang saling terkait menjadi satu kesatuan yang terorganisir.

Selain itu, modul ini juga mengulas tentang array dan cara penggunaannya, baik itu array statis maupun dinamis. Di dalamnya terdapat juga fungsi-fungsi bawaan seperti `len`, `cap`, dan `append` yang memfasilitasi manipulasi array dengan lebih mudah. Konsep pemrograman yang lebih kompleks, seperti `map`, juga diperkenalkan, yang memberikan kebebasan untuk menggunakan kunci non-integer dalam mengakses nilai.

Melalui beragam contoh dan latihan praktis, modul ini bertujuan untuk memberikan pemahaman yang mendalam tentang cara efektif mengelola data dalam pemrograman, dengan memanfaatkan tipe bentukan dan struktur data yang tepat.

V. REFERENSI

Modul 7 - Praktikum Alpro 2

- A.24. Struct. (n.d.). Retrieved from
<https://dasarpemrogramangolang.novalagung.com/A-struct.html>
- A.15. Array. (n.d.). Retrieved from
<https://dasarpemrogramangolang.novalagung.com/A-array.html>
- A.16. Slice. (n.d.). Retrieved from
<https://dasarpemrogramangolang.novalagung.com/A-slice.html>