

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL 6
STRUCT & ARRAY**



Oleh:

DWI OKTA SURYANINGRUM

103112400066

12-IF-01

**S1 TEKNIK INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO**

2025

I. DASAR TEORI

Tipe bentukan memungkinkan pemrograman untuk mendefinisikan suatu tipe data baru pada suatu

bahasa pemrograman. Tipe bentukan 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".

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. Berbeda dengan bahasa pemrograman lain. kesamaan tipe dari dua variabel berjenis structure 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.

3) Array

Array mempunyai ukuran (jumlah elemen) yang tetap (statis) selama eksekusi program, sehingga jumlah elemen array menjadi bagian dari deklarasi variabel dengan tipe array.

4) 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

Source Code + Screenshot hasil program beserta penjelasan

a. Guided 1

```
1 package main
2
3 import (
4     "fmt"
5     "time"
6 )
7
8 // Struct untuk barang dalam struk belanja
9 type Item struct {
10     Name    string
11     Price   float64
12     Quantity int
13 }
14
15 // Struct untuk struk belanja
16 type Receipt struct {
17     StoreInfo string
18     Date      time.Time
19     Items     []Item
20     TotalAmount float64
21 }
22
23 // Method untuk menghitung total harga semua item
24 func (r *Receipt) CalculateTotal() {
25     var total float64
26     for _, item := range r.Items {
27         total += item.Price * float64(item.Quantity)
28     }
29     r.TotalAmount = total
30 }
31
32 // Method untuk mencetak struk belanja
33 func (r *Receipt) PrintReceipt() {
34     fmt.Println("=====")
35     fmt.Println(r.StoreInfo)
36     fmt.Println("Tanggal:", r.Date.Format("02-01-2006 15:04"))
37     fmt.Println("=====")
38     fmt.Printf("%-15s %-10s %-8s %-10s\n", "Item", "Harga", "Jumlah", "Total")
39     fmt.Println("=====")
40
41     for _, item := range r.Items {
42         itemTotal := item.Price * float64(item.Quantity)
43         fmt.Printf("%-15s Rp%-9.2f %-8d Rp%-9.2f\n", item.Name, item.Price, item.Quantity, itemTotal)
44     }
45
46     fmt.Println("=====")
47     fmt.Printf("%-35s Rp%-9.2f\n", "Total Belanja:", r.TotalAmount)
48     fmt.Println("=====")
49     fmt.Println("Terima kasih telah berbelanja!")
50 }
51
52 func main() {
53     receipt := Receipt{
54         StoreInfo: "Toko Sembako Makmur\nJl. Raya No. 123, Jakarta",
55         Date:      time.Now(),
56         Items: []Item{
57             {Name: "Beras", Price: 12000, Quantity: 5},
58             {Name: "Gula", Price: 15000, Quantity: 2},
59             {Name: "Minyak", Price: 20000, Quantity: 1},
60             {Name: "Telur", Price: 2000, Quantity: 10},
61         },
62     }
63
64     receipt.CalculateTotal()
65     receipt.PrintReceipt()
66 }
67
68 /*
69 Program ini dibuat untuk menampilkan struk belanja sederhana menggunakan bahasa Go,
70 dengan pendekatan yang menyerupai proses nyata saat kita berbelanja di toko.
71
72 Di dalam program, terdapat dua struct utama, yaitu Item dan Receipt. Struct Item
73 merepresentasikan barang belanjaan dengan atribut nama barang, harga per satuan,
74 dan jumlah yang dibeli. Struct Receipt menyimpan informasi mengenai toko, tanggal
75 transaksi, daftar belanjaan, serta total harga yang harus dibayar.
76
77 Program akan menghitung total biaya belanja dengan menjumlahkan hasil perkalian
78 antara harga dan jumlah setiap barang melalui fungsi CalculateTotal(). Setelah itu,
79 fungsi PrintReceipt() digunakan untuk mencetak struk yang menampilkan detail toko,
80 tanggal, daftar item belanja beserta harga dan jumlahnya, total per item, hingga
81 total keseluruhan belanja.
82
83 Output akhir dari program menyerupai struk yang biasanya kita terima setelah
84 bertransaksi di kasir. Program ini menggambarkan bagaimana konsep struct dan method
85 di Go bisa digunakan untuk membuat simulasi transaksi yang terstruktur dan informatif.
86 */
87
```

Output :

```
mymac@192 ALPRO SMT 2 % go run "/Users/mymac/Documents/ITTP/ALPRO SMT 2/MODUL 6/guided1.go"
=====
Toko Sembako Makmur
Jl. Raya No. 123, Jakarta
Tanggal: 25-04-2025 22:22
=====
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!
```

b. Guided 2

```
1 package main
2
3 import (
4     "fmt"
5 )
6
7 func main() {
8     // Deklarasi dan inisialisasi array nilai mahasiswa
9     nilaiMahasiswa := []int{85, 90, 78, 88, 95}
10
11     fmt.Println("Data Nilai Mahasiswa:")
12     fmt.Println("=====")
13
14     // Menampilkan nilai per mahasiswa
15     for i, nilai := range nilaiMahasiswa {
16         fmt.Printf("Mahasiswa %d: %d\n", i+1, nilai)
17     }
18
19     // Menghitung rata-rata nilai
20     var total int
21     for _, nilai := range nilaiMahasiswa {
22         total += nilai
23     }
24     rataRata := float64(total) / float64(len(nilaiMahasiswa))
25
26     fmt.Println("=====")
27     fmt.Printf("Rata-rata nilai: %.2f\n", rataRata)
28
29     // Mencari nilai tertinggi dan terendah
30     tertinggi := nilaiMahasiswa[0]
31     terendah := nilaiMahasiswa[0]
32
33     for _, nilai := range nilaiMahasiswa {
34         if nilai > tertinggi {
35             tertinggi = nilai
36         }
37         if nilai < terendah {
38             terendah = nilai
39         }
40     }
41
42     fmt.Printf("Nilai tertinggi: %d\n", tertinggi)
43     fmt.Printf("Nilai terendah: %d\n", terendah)
44
45     // Contoh array 2 dimensi
46     fmt.Println("\nContoh Array 2 Dimensi:")
47     fmt.Println("=====")
48
49     // Nilai ujian mahasiswa dalam 2 mata kuliah (Matematika, Bahasa)
50     nilaiUjian := [3][2]int{
51         {80, 85},
52         {90, 75},
53         {70, 95},
54     }
55
56     // Menampilkan nilai ujian per mahasiswa
57     fmt.Println("Nilai Ujian Mahasiswa (Matematika, Bahasa):")
58     for i, nilai := range nilaiUjian {
59         fmt.Printf("Mahasiswa %d: Matematika = %d, Bahasa = %d\n", i+1, nilai[0], nilai[1])
60     }
61 }
```

Output :

```
mymac@192 ALPRO SMT 2 % go run "/Users/mymac/Documents/ITTP/ALPRO SMT 2/MODUL 6/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
```

III. UNGUIDED

a. Unguided 1

```
1 // DWI OKTA SURYANINGRUM
2 // 103112400066
3
4 package main
5
6 import "fmt"
7
8 // mendefinisikan struct untuk Titik (x,y) dan Lingkaran (pusat Titik, radius)
9 type Titik struct { X, Y int }
10 type Lingkaran struct { Pusat Titik; Radius int }
11
12 func main() {
13     // mendeklarasi variabel untuk input
14     var l1, l2 Lingkaran
15     var p Titik
16
17     // membaca input lingkaran 1 (x y radius)
18     fmt.Scan(&l1.Pusat.X, &l1.Pusat.Y, &l1.Radius)
19
20     // membaca input lingkaran 2 (x y radius)
21     fmt.Scan(&l2.Pusat.X, &l2.Pusat.Y, &l2.Radius)
22
23     // membaca input titik yang akan dicek (x y)
24     fmt.Scan(&p.X, &p.Y)
25
26     // membuat fungsi untuk cek apakah titik di dalam lingkaran
27     cekLingkaran := func(p Titik, l Lingkaran) bool {
28         dx := p.X - l.Pusat.X // selisih koordinat x
29         dy := p.Y - l.Pusat.Y // selisih koordinat y
30         // Jika jarak^2 <= radius^2 maka titik ada di dalam
31         return dx*dx + dy*dy <= l.Radius*l.Radius
32     }
33
34     // cek posisi titik terhadap kedua lingkaran
35     inL1 := cekLingkaran(p, l1) // apakah di lingkaran 1?
36     inL2 := cekLingkaran(p, l2) // apakah di lingkaran 2?
37
38     // menentukan output berdasarkan beberapa kondisi
39     switch {
40     case inL1 && inL2:
41         fmt.Println("Titik di dalam lingkaran 1 dan 2")
42     case inL1:
43         fmt.Println("Titik di dalam lingkaran 1")
44     case inL2:
45         fmt.Println("Titik di dalam lingkaran 2")
46     default:
47         fmt.Println("Titik di luar lingkaran 1 dan 2")
48     }
49 }
```

Output :

```
● mymac@192 ALPRO SMT 2 % go run "/Users/mymac/Documents/ITTP/ALPRO SMT 2/MODUL 6/unguided1.go"
1 1 5
8 8 4
2 2
Titik di dalam lingkaran 1
● mymac@192 ALPRO SMT 2 % go run "/Users/mymac/Documents/ITTP/ALPRO SMT 2/MODUL 6/unguided1.go"
1 2 3
4 5 6
7 8
Titik di dalam lingkaran 2
● mymac@192 ALPRO SMT 2 % go run "/Users/mymac/Documents/ITTP/ALPRO SMT 2/MODUL 6/unguided1.go"
5 10 15
-15 4 20
0 0
Titik di dalam lingkaran 1 dan 2
● mymac@192 ALPRO SMT 2 % go run "/Users/mymac/Documents/ITTP/ALPRO SMT 2/MODUL 6/unguided1.go"
1 1 5
8 8 4
15 20
Titik di luar lingkaran 1 dan 2
❖ mymac@192 ALPRO SMT 2 %
```


b. Unguided 2

```
1 // DWI OKTA SURYANINGRUM
2 // 103112400066
3
4 package main
5
6 import (
7     "fmt"
8     "math"
9 )
10
11 // Struct ArrayHandler menampung array dan jumlah elemen aktif
12 type ArrayHandler struct {
13     data [100]int // Array berkapasitas maksimal 100 elemen
14     length int    // Menyimpan jumlah elemen yang sedang aktif
15 }
16
17 // Menampilkan seluruh isi array
18 func (a *ArrayHandler) TampilkanSemua() {
19     fmt.Println("Isi array:")
20     for i := 0; i < a.length; i++ {
21         fmt.Printf("%d ", a.data[i]) // Cetak setiap elemen array
22     }
23     fmt.Println()
24 }
25
26 // Menampilkan elemen dengan indeks ganjil
27 func (a *ArrayHandler) TampilkanIndeksGanjil() {
28     fmt.Println("Elemen dengan indeks ganjil:")
29     for i := 1; i < a.length; i += 2 { // Mulai dari indeks 1, lompat 2
30         fmt.Printf("data[%d] = %d\n", i, a.data[i])
31     }
32 }
33
34 // Menampilkan elemen dengan indeks genap
35 func (a *ArrayHandler) TampilkanIndeksGenap() {
36     fmt.Println("Elemen dengan indeks genap:")
37     for i := 0; i < a.length; i += 2 { // Mulai dari indeks 0, lompat 2
38         fmt.Printf("data[%d] = %d\n", i, a.data[i])
39     }
40 }
```



```
1 // Menampilkan elemen dengan indeks kelipatan x
2 func (a *ArrayHandler) TampilkanIndeksKelipatan(x int) {
3     fmt.Printf("Elemen dengan indeks kelipatan %d:\n", x)
4     for i := 0; i < a.length; i++ {
5         if i%x == 0 { // Cek apakah indeks merupakan kelipatan x
6             fmt.Printf("data[%d] = %d\n", i, a.data[i])
7         }
8     }
9 }
10
11 // Menghapus elemen pada indeks tertentu
12 func (a *ArrayHandler) HapusPadaIndeks(indeks int) {
13     for i := indeks; i < a.length-1; i++ {
14         a.data[i] = a.data[i+1] // Geser elemen ke kiri
15     }
16     a.length-- // Kurangi panjang array
17     fmt.Println("Setelah penghapusan:")
18     a.TampilkanSemua() // Tampilkan isi array setelah elemen dihapus
19 }
20
21 // Menghitung rata-rata
22 func (a *ArrayHandler) RataRata() float64 {
23     total := 0
24     for i := 0; i < a.length; i++ {
25         total += a.data[i] // Jumlahkan seluruh elemen
26     }
27     return float64(total) / float64(a.length) // Hitung rata-rata
28 }
29
30 // Menghitung simpangan baku (standar deviasi)
31 func (a *ArrayHandler) SimpanganBaku() float64 {
32     rata := a.RataRata() // Hitung rata-rata dulu
33     var total float64
34     for i := 0; i < a.length; i++ {
35         total += math.Pow(float64(a.data[i])-rata, 2) // (x - rata)^2
36     }
37     return math.Sq
```

```

1 // Menampilkan frekuensi dari suatu bilangan
2 func (a *ArrayHandler) Frekuensi(nilai int) int {
3     count := 0
4     for i := 0; i < a.length; i++ {
5         if a.data[i] == nilai { // Jika elemen sama dengan nilai yang dicari
6             count++
7         }
8     }
9     return count // Kembalikan jumlah kemunculan
10 }
11
12 func main() {
13     var n int
14     handler := ArrayHandler{} // Inisialisasi struct ArrayHandler
15
16     fmt.Print("Masukkan jumlah elemen array: ")
17     fmt.Scan(&n)
18
19     // Input elemen array dari pengguna
20     for i := 0; i < n; i++ {
21         fmt.Printf("Masukkan elemen ke-%d: ", i)
22         fmt.Scan(&handler.data[i])
23         handler.length++ // Tambah panjang array setiap input
24     }
25
26     handler.TampilkanSemua() // a. Tampilkan semua isi array
27     handler.TampilkanIndeksGanjil() // b. Tampilkan indeks ganjil
28     handler.TampilkanIndeksGenap() // c. Tampilkan indeks genap
29
30     var x int
31     fmt.Print("Masukkan nilai x untuk indeks kelipatan: ")
32     fmt.Scan(&x)
33     handler.TampilkanIndeksKelipatan(x) // d. Tampilkan indeks kelipatan x
34
35     var indeks int
36     fmt.Print("Masukkan indeks yang ingin dihapus: ")
37     fmt.Scan(&indeks)
38     handler.HapusPadaIndeks(indeks) // e. Hapus elemen di indeks tertentu
39
40     fmt.Printf("Rata-rata: %.2f\n", handler.RataRata()) // f. Hitung rata-rata
41     fmt.Printf("Simpangan baku: %.2f\n", handler.SimpanganBaku()) // g. Hitung standar deviasi
42
43     var cari int
44     fmt.Print("Masukkan bilangan untuk dicari frekuensinya: ")
45     fmt.Scan(&cari)
46     fmt.Printf("Frekuensi %d = %d\n", cari, handler.Frekuensi(cari)) // h. Hitung frekuensi nilai
47 }

```

Output :

```

mymac@192 ALPRO SMT 2 % go run "/Users/mymac/Documents/ITTP/ALPRO SMT 2/MODUL 6/unguided2.go"
Masukkan jumlah elemen array: 2
Masukkan elemen ke-0: 3
Masukkan elemen ke-1: 5
Isi array:
3 5
Elemen dengan indeks ganjil:
data[1] = 5
Elemen dengan indeks genap:
data[0] = 3
Masukkan nilai x untuk indeks kelipatan: 2
Elemen dengan indeks kelipatan 2:
data[0] = 3
Masukkan indeks yang ingin dihapus: 2
Setelah penghapusan:
Isi array:
3
Rata-rata: 3.00
Simpangan baku: 0.00
Masukkan bilangan untuk dicari frekuensinya: 2
Frekuensi 2 = 0

```

c. Unguided 3

```
1 // DWI OKTA SURYANINGRUM
2 // 103112400066
3
4 package main
5
6 import "fmt"
7
8 // Struct Pertandingan menyimpan data dua klub dan daftar pemenang dari beberapa pertandingan
9 type Pertandingan struct {
10     KlubA    string // Menyimpan nama klub A
11     KlubB    string // Menyimpan nama klub B
12     Pemenang []string // Slice untuk menyimpan hasil pemenang setiap pertandingan
13 }
14
15 func main() {
16     var p Pertandingan // Inisialisasi variabel struct Pertandingan
17
18     // Input nama klub
19     fmt.Print("Klub A : ")
20     fmt.Scan(&p.KlubA) // Meminta input nama Klub A
21     fmt.Print("Klub B : ")
22     fmt.Scan(&p.KlubB) // Meminta input nama Klub B
23
24     // Proses input skor pertandingan berulang
25     for i := 1; ; i++ { // Loop tanpa batas, dihentikan dengan kondisi tertentu
26         var skorA, skorB int // Variabel untuk menyimpan skor tiap klub
27         fmt.Printf("Pertandingan %d : ", i)
28         fmt.Scan(&skorA, &skorB) // Input skor Klub A dan Klub B
29
30         // Validasi skor negatif untuk menghentikan input
31         if skorA < 0 || skorB < 0 {
32             fmt.Println("Pertandingan selesai")
33             break // Keluar dari loop jika skor negatif dimasukkan
34         }
35
36         // Menentukan pemenang berdasarkan skor
37         switch {
38             case skorA > skorB:
39                 p.Pemenang = append(p.Pemenang, p.KlubA) // Klub A menang
40                 fmt.Printf("Hasil %d : %s\n", i, p.KlubA)
41             case skorB > skorA:
42                 p.Pemenang = append(p.Pemenang, p.KlubB) // Klub B menang
43                 fmt.Printf("Hasil %d : %s\n", i, p.KlubB)
44             default:
45                 p.Pemenang = append(p.Pemenang, "Draw") // Skor imbang
46                 fmt.Printf("Hasil %d : Draw\n", i)
47         }
48     }
49
50     // Menampilkan daftar pemenang dari semua pertandingan
51     fmt.Println("\nDaftar Klub Pemenang:")
52     for i, pemenang := range p.Pemenang {
53         fmt.Printf("Pertandingan %d: %s\n", i+1, pemenang)
54     }
55 }
```

Output :

```
mymac@192 ALPR0 SMT 2 % go run "/Users/mymac/Documents/ITTP/ALPR0 SMT 2/MODUL 6/unguided3.go"
Klub A : MU
Klub B : inter
Pertandingan 1 : 2 0
Hasil 1 : MU
Pertandingan 2 : 1 2
Hasil 2 : inter
Pertandingan 3 : 2 2
Hasil 3 : Draw
Pertandingan 4 : 0 1
Hasil 4 : inter
Pertandingan 5 : 3 2
Hasil 5 : MU
Pertandingan 6 : 1 0
Hasil 6 : MU
Pertandingan 7 : 5 2
Hasil 7 : MU
Pertandingan 8 : 2 3
Hasil 8 : inter
Pertandingan 9 : -1 2
Pertandingan selesai

Daftar Klub Pemenang:
Pertandingan 1: MU
Pertandingan 2: inter
Pertandingan 3: Draw
Pertandingan 4: inter
Pertandingan 5: MU
Pertandingan 6: MU
Pertandingan 7: MU
Pertandingan 8: inter
```

d. Unguided 4

```
1 // DWI OKTA SURYANINGRUM
2 // 103112400066
3
4 package main
5
6 import "fmt"
7
8 const NMAX int = 127 // Konstanta untuk ukuran maksimum array
9
10 type tabel [NMAX]rune // Deklarasi tipe array rune dengan ukuran NMAX
11
12 // membuat fungsi untuk mengisi array dengan input dari user
13 func isiArray(t *tabel, n *int) {
14     var input rune
15     *n = 0 // Inisialisasi panjang array
16     fmt.Print("Teks : ")
17
18     // Loop untuk membaca input hingga menemukan titik atau mencapai kapasitas maksimum
19     for {
20         fmt.Scanf("%c", &input) // membaca input per karakter
21
22         // melewati spasi dan newline
23         if input == ' ' || input == '\n' {
24             continue
25         }
26
27         // berhenti jika menemukan titik atau melebihi kapasitas array
28         if input == '.' || *n >= NMAX {
29             break
30         }
31
32         t[*n] = input // menyimpan karakter ke array
33         (*n)++        // menambah counter panjang array
34     }
35 }
36
37 // membuat fungsi untuk mencetak isi array ke layar
38 func cetakArray(t tabel, n int) {
39     // mencetak setiap karakter dipisahkan spasi
40     for i := 0; i < n; i++ {
41         fmt.Printf("%c ", t[i])
42     }
43     fmt.Println() // membuat baris baru setelah selesai
44 }
```

```

1 // membuat fungsi untuk membalikkan urutan elemen dalam array
2 func balikanArray(t *tabel, n int) {
3     // membalik array dengan menukar elemen dari depan dan belakang
4     for i := 0; i < n/2; i++ {
5         temp := t[i] // menyimpan sementara elemen depan
6         t[i] = t[n-1-i] // mengganti elemen depan dengan belakang
7         t[n-1-i] = temp // mengganti elemen belakang dengan depan
8     }
9 }
10
11 // membuat fungsi untuk memeriksa apakah array adalah palindrom
12 func palindrom(t tabel, n int) bool {
13     // membandingkan elemen dari kedua ujung
14     for i := 0; i < n/2; i++ {
15         if t[i] != t[n-1-i] { // jika ada yang tidak sama
16             return false // bukan palindrom
17         }
18     }
19     return true // jika semua sama, palindrom
20 }
21
22 func main() {
23     var tab tabel // array untuk menyimpan karakter
24     var m int // variabel untuk menyimpan panjang array
25
26     isiArray(&tab, &m) // memanggil fungsi untuk mengisi array
27     tabCopy := tab // membuat salinan array untuk pengecekan palindrom
28
29     // membalik array dan mencetak hasilnya
30     fmt.Print("Reverse teks : ")
31     balikanArray(&tab, m)
32     cetakArray(tab, m)
33
34     // mencetak dan menampilkan apakah palindrom
35     fmt.Print("Palindrom ? ")
36     if palindrom(tabCopy, m) {
37         fmt.Println("true")
38     } else {
39         fmt.Println("false")
40     }
41 }

```

Output :

```

● mymac@192 ALPRO SMT 2 % go run "/Users/mymac/Documents/ITTP/ALPRO SMT 2/MODUL 6/unguided4.go"
Teks : S E N A N G.
Reverse teks : G N A N E S
Palindrom ? false
○ mymac@192 ALPRO SMT 2 %
● mymac@192 ALPRO SMT 2 % go run "/Users/mymac/Documents/ITTP/ALPRO SMT 2/MODUL 6/unguided4.go"
Teks : K A T A K.
Reverse teks : K A T A K
Palindrom ? true
❖ mymac@192 ALPRO SMT 2 %

```

IV. KESIMPULAN

Tipe bentukan dalam bahasa pemrograman, khususnya dalam bahasa Go, memberikan fleksibilitas tinggi dalam pengelolaan data dengan memungkinkan pemrogram mendefinisikan tipe data baru sesuai kebutuhan. Tipe bentukan terdiri dari dua jenis utama, yaitu **alias (type)** dan **struct (record)**. Alias digunakan untuk memberikan nama baru yang lebih familiar atau deskriptif terhadap tipe data yang sudah ada, sehingga membuat kode lebih mudah dipahami.

Sementara itu, struct memungkinkan pengelompokan beberapa nilai yang memiliki keterkaitan dalam satu entitas yang utuh, sangat berguna dalam representasi objek atau entitas nyata, seperti data mahasiswa, barang, atau pertandingan. Struct juga mendukung pendekatan pemrograman modular dan terstruktur karena dapat didefinisikan sebagai tipe baru dan digunakan berulang kali tanpa menulis ulang field-fieldnya. Selain itu, penggunaan **array** sebagai struktur data linear membantu dalam menyimpan sejumlah data homogen secara statis, sedangkan **map** menawarkan fleksibilitas indeks dinamis untuk pengelolaan data asosiatif. Dengan memahami dan memanfaatkan tipe bentukan ini, pemrogram dapat menulis kode yang lebih rapi, efisien, dan mudah untuk dikembangkan.