

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL 6
STRUCT & ARRAY**



Oleh:

BERTHA ADELA

103112400041

IF-12-01

**S1 TEKNIK INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO**

2025

I. DASAR TEORI

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".

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 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.

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

Slice: Array dalam Go juga dapat mempunyai ukuran yang dinamik. (Tidak digunakan di kelas Algoritma Pemrograman). Deklarasinya mirip dengan deklarasi array, tetapi jumlah elemennya dikosongkan.

Fungsi built-in **len** dapat digunakan untuk mengetahui ukuran slice. Fungsi lain, **cap**, dapat digunakan untuk mengetahui total tempat yang disediakan untuk slice tersebut. Fungsi built-in **append** dapat digunakan untuk menambahkan elemen ke suatu slice, dan bila perlu memperbesar tempat untuk slice tersebut.

II. GUIDED

• GUIDED 1

Code:

```
SMT2 > Pertemuan6 > 103112400041_Guided1.go > Receipt > Items

1  package main
2
3  import (
4      "fmt"
5      "time"
6  )
7  //Struct untuk barang dalam struk belanja
8  type Item struct {
9      Name string
10     Price float64
11     Quantity int
12 }
13 //Struct untuk struk belanja
14 type Receipt struct {
15     StoreInfo string
16     Date time.Time
17     Items []Item
18     TotalAmount float64
19 }
20 //Method untuk menghitung total harga semua item
21 func (r *Receipt) CalculateTotal() {
22     var total float64
23     for _, item := range r.Items {
24         total += item.Price * float64(item.Quantity)
25     }
26     r.TotalAmount = total
27 }
28 //Method untuk mencetak struk belanja
29 func (r Receipt) PrintReceipt() {
30     fmt.Println("=====")
31     fmt.Println(r.StoreInfo)
32     fmt.Println("Tanggal:", r.Date.Format("02-01-2006 15:04"))
33     fmt.Println("=====")
34     fmt.Println("=====")
35     fmt.Printf("%-15s %-10s %-8s %-10s\n", "Item", "Harga", "Jumlah", "Total")
36     fmt.Println("-----")
37
38     for _, item := range r.Items {
39         itemTotal := item.Price * float64(item.Quantity)
40         fmt.Printf("%-15s Rp%-9.2f %-8d Rp%-9.2f\n", item.Name, item.Price, item.Quantity, itemTotal)
41     }
42
43     fmt.Println("=====")
44     fmt.Printf("%-35s Rp%-9.2f\n", "Total Belanja:", r.TotalAmount)
45     fmt.Println("=====")
46     fmt.Println("Terima kasih telah berbelanja!")
47 }
48
49 func main() {
50     receipt := Receipt{
51         StoreInfo: "Toko Sembako Makmur\nJl. Raya No. 123, Jakarta",
52         Date:      time.Now(),
53         Items: []Item{
54             {Name: "Beras", Price: 12000, Quantity: 5},
55             {Name: "Gula", Price: 15000, Quantity: 2},
56             {Name: "Minyak", Price: 20000, Quantity: 1},
57             {Name: "Telur", Price: 2000, Quantity: 10},
58         },
59     }
60
61     receipt.CalculateTotal()
62     receipt.PrintReceipt()
63 }
```

Output:

```
PS C:\Users\levina\OneDrive\Documents\golang> go run "c:\Users\levina\OneDrive\Documents\golang\main.go"
=====
Toko Sembako Makmur
Jl. Raya No. 123, Jakarta
Tanggal: 25-04-2025 20:41
=====
=====
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!
PS C:\Users\levina\OneDrive\Documents\golang> 
```

Penjelasan:

Program diatas berguna untuk membuat *shopping receipt* di toko sembako makmur, program mengkalkulasikan total harga barang yang telah dibeli.

• GUIDED 2

Code:

```
SMT2 > Pertemuan6 > 103112400041_Guided2.go > main
1  package main
2
3  import (
4      "fmt"
5  )
6
7  func main() {
8      // Deklarasi dan inisialisasi array nilai mahasiswa
9      nilaiMahasiswa := [5]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:

```
PS C:\Users\levina\OneDrive\Documents\golang> go run "c:\Users\levina\OneDrive\Documents\golang\array2.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
PS C:\Users\levina\OneDrive\Documents\golang>
```

Penjelasan:

Program diatas adalah contoh array. Program menghitung rata-rata nilai mahasiswa, mencari nilai tertinggi dan terendah, serta program menampilkan array 2 dimensi menggunakan nilai ujian mahasiswa.

III. UNGUIDED

- UNGUIDED 1

Code:

```
SMT2 > Pertemuan6 > -go 103112400041_Unguided1.go > Didalam
1 //BERTHA ADELA
2 //103112400041
3 package main
4 import (
5     "fmt"
6     "math"
7 )
8 type Titik struct {
9     x, y float64
10 }
11
12 type Lingkaran struct {
13     pusat Titik
14     radius float64
15 }
16 func main() {
17     var lingkaran1, lingkaran2 Lingkaran
18     var titik Titik
19
20     fmt.Scanln(&lingkaran1.pusat.x, &lingkaran1.pusat.y, &lingkaran1.radius)
21     fmt.Scanln(&lingkaran2.pusat.x, &lingkaran2.pusat.y, &lingkaran2.radius)
22     fmt.Scanln(&titik.x, &titik.y)
23
24     if Didalam(lingkaran1, titik) && Didalam(lingkaran2, titik) {
25         fmt.Print("Titik di dalam lingkaran 1 dan 2")
26     } else if Didalam(lingkaran1, titik) {
27         fmt.Print("Titik di dalam lingkaran 1")
28     } else if Didalam(lingkaran2, titik) {
29         fmt.Print("Titik di dalam lingkaran 2")
30     } else {
31         fmt.Print("Titik di luar lingkaran 1 dan 2")
32     }
33 }
34
35 func Jarak(p, q Titik) float64 {
36     return math.Sqrt((q.x-p.x)*(q.x-p.x) + (q.y-p.y)*(q.y-p.y))
37 }
38
39 func Didalam(c Lingkaran, p Titik) bool {
40     return Jarak(c.pusat, p) <= c.radius
41 }
```

Output:

```
PS C:\Users\levina\OneDrive\Documents\golang> go run "c:\Users\levina\OneDrive\Documents\golang\nguided1.go"
1 1 5
8 8 4
2 2
Titik di dalam lingkaran 1
PS C:\Users\levina\OneDrive\Documents\golang> go run "c:\Users\levina\OneDrive\Documents\golang\nguided1.go"
1 2 3
4 5 6
7 8
Titik di dalam lingkaran 2
PS C:\Users\levina\OneDrive\Documents\golang> go run "c:\Users\levina\OneDrive\Documents\golang\nguided1.go"
5 10 15
-15 4 20
0 0
Titik di dalam lingkaran 1 dan 2
PS C:\Users\levina\OneDrive\Documents\golang> go run "c:\Users\levina\OneDrive\Documents\golang\nguided1.go"
1 1 5
8 8 4
15 20
Titik di luar lingkaran 1 dan 2
PS C:\Users\levina\OneDrive\Documents\golang> 
```

Penjelasan:

Program ini berguna untuk mencari apakah sebuah titik koordinat termasuk ke dalam lingkaran 1, lingkaran 2, keduanya, atau diluar keduanya.

• UNGUIDED 2

Code:

```
SMT2 > Pertemuan6 > go 103112400041_Unguided2.go > ...
1 //BERTHA ADELA
2 //103112400041
3 package main
4
5 import (
6     "fmt"
7     "math"
8 )
9
10 func main() {
11     var n int
12     fmt.Print("Masukkan jumlah array: ")
13     fmt.Scan(&n)
14
15     data := make([]int, n)
16
17     for i := 0; i < n; i++ {
18         fmt.Printf("Masukkan elemen ke-%d: ", i+1)
19         fmt.Scan(&data[i])
20     }
21
22     fmt.Println("\nSeluruh isi array:")
23     for i := 0; i < len(data); i++ {
24         fmt.Printf("indeks[%d] = %d\n", i+1, data[i])
25     }
26
27     fmt.Println("\nElemen dengan indeks ganjil:")
28     for i := 0; i < len(data); i++ {
29         if (i+1)%2 == 1 {
30             fmt.Printf("indeks[%d] = %d\n", i+1, data[i])
31         }
32     }
33
34     fmt.Println("\nElemen dengan indeks genap:")
35     for i := 0; i < len(data); i++ {
36         if (i+1)%2 == 0 {
37             fmt.Printf("indeks[%d] = %d\n", i+1, data[i])
38         }
39     }
40
41     var x int
42     fmt.Print("\nMasukkan nilai x untuk mencari indeks kelipatan x: ")
43     fmt.Scan(&x)
44     fmt.Printf("Elemen dengan indeks kelipatan %d:\n", x)
45     for i := 0; i < len(data); i++ {
46         if x != 0 {
47             if (i+1)%x == 0 {
48                 fmt.Printf("indeks[%d] = %d\n", i+1, data[i])
49             }
50         }
51     }
```

```

53     var hapus int
54     fmt.Print("\nMasukkan indeks yang ingin dihapus: ")
55     fmt.Scan(&hapus)
56     hapus = hapus - 1
57
58     if hapus >= 0 && hapus < len(data) {
59         var temp []int
60         for i := 0; i < len(data); i++ {
61             if i != hapus {
62                 temp = append(temp, data[i])
63             }
64         }
65         data = temp
66         fmt.Println("Isi array setelah dihapus:")
67         for i := 0; i < len(data); i++ {
68             fmt.Printf("indeks[%d] = %d\n", i+1, data[i])
69         }
70     } else {
71         fmt.Println("Indeks tidak valid.")
72     }
73
74     var jumlah int
75     for i := 0; i < len(data); i++ {
76         jumlah += data[i]
77     }
78     var rata2 float64 = float64(jumlah) / float64(len(data))
79     fmt.Printf("\nRata-rata: %.2f\n", rata2)
80

```

```

81     var total float64
82     for i := 0; i < len(data); i++ {
83         selisih := float64(data[i]) - rata2
84         total += selisih * selisih
85     }
86     var deviasi float64 = math.Sqrt(total / float64(len(data)))
87     fmt.Printf("Standar Deviasi: %.2f\n", deviasi)
88
89     var cari int
90     fmt.Print("\nMasukkan angka yang ingin dicari frekuensinya: ")
91     fmt.Scan(&cari)
92     var frek int
93     for i := 0; i < len(data); i++ {
94         if data[i] == cari {
95             frek++
96         }
97     }
98     fmt.Printf("Frekuensi angka %d: %d kali\n", cari, frek)
99 }
100

```

Output:

```
PS C:\Users\levina\OneDrive\Documents\golang> go run "c:\Users\levina\OneDrive\Documents\golang\nguided2.go"
Masukkan jumlah array: 5
Masukkan elemen ke-1: 32
Masukkan elemen ke-2: 32
Masukkan elemen ke-3: 34
Masukkan elemen ke-4: 32
Masukkan elemen ke-5: 12

Seluruh isi array:
indeks[1] = 32
indeks[2] = 32
indeks[3] = 34
indeks[4] = 32
indeks[5] = 12

Elemen dengan indeks ganjil:
indeks[1] = 32
indeks[3] = 34
indeks[5] = 12

Elemen dengan indeks genap:
indeks[2] = 32
indeks[4] = 32

Masukkan nilai x untuk mencari indeks kelipatan x: 2
Elemen dengan indeks kelipatan 2:
indeks[2] = 32
indeks[4] = 32

Masukkan indeks yang ingin dihapus: 5
Isi array setelah dihapus:
indeks[1] = 32
indeks[2] = 32
indeks[3] = 34
indeks[4] = 32

Rata-rata: 32.50
Standar Deviasi: 0.87

Masukkan angka yang ingin dicari frekuensinya: 32
Frekuensi angka 32: 3 kali
PS C:\Users\levina\OneDrive\Documents\golang>
```

Penjelasan:

Program ini digunakan untuk menampilkan keseluruhan isi dari array yang sudah diinputkan, menampilkan elemen-elemen array dengan indeks ganjil saja, menampilkan elemen-elemen array dengan indeks genap saja, menampilkan elemen-elemen array dengan indeks kelipatan bilangan x(input pengguna), menghapus elemen array pada indeks tertentu kemudian menampilkan keseluruhan untuk memastikan data yang dihapus tidak tampil, menampilkan rata-rata dari bilangan yang ada di dalam array, menampilkan standar deviasi atau simpangan baku dari bilangan yang ada di dalam array tersebut, dan menampilkan frekuensi dari suatu bilangan tertentu di dalam array yang telah diisi tersebut.

• UNGUIDED 3

Code:

```
SMT2 > Pertemuan6 > 103112400041_Unguided3.go > main
1 //BERTHA ADELA
2 //103112400041
3 package main
4
5 import "fmt"
6
7 func main() {
8     var clubA, clubB string
9     var score1, score2 int
10
11     fmt.Print("Klub A: ")
12     fmt.Scanln(&clubA)
13     fmt.Print("Klub B: ")
14     fmt.Scanln(&clubB)
15
16     match := 1
17     var results []string
18
19     for score1 != -1 {
20         fmt.Printf("Pertandingan ke-%d : ", match)
21         fmt.Scanln(&score1, &score2)
22
23         if score1 != -1 {
24             if score1 > score2 {
25                 results = append(results, fmt.Sprintf("Hasil %d : %s", match, clubA))
26             } else if score2 > score1 {
27                 results = append(results, fmt.Sprintf("Hasil %d : %s", match, clubB))
28             } else {
29                 results = append(results, fmt.Sprintf("Hasil %d : Draw", match))
30             }
31             match = match + 1
32         }
33     }
34
35     for _, result := range results {
36         fmt.Println(result)
37     }
38
39     fmt.Println("Pertandingan selesai")
40 }
```

Output:

```
PS C:\Users\levina\OneDrive\Documents\golang> go run "c:\Users\levina\OneDrive\Documents\golang\103112400041_Unguided3.go"
Klub A: MU
Klub B: Inter
Pertandingan ke-1 : 2 0
Pertandingan ke-2 : 1 2
Pertandingan ke-3 : 2 2
Pertandingan ke-4 : 0 1
Pertandingan ke-5 : 3 2
Pertandingan ke-6 : 1 0
Pertandingan ke-7 : 5 2
Pertandingan ke-8 : 2 3
Pertandingan ke-9 : -1 2
Hasil 1 : MU
Hasil 2 : Inter
Hasil 3 : Draw
Hasil 4 : Inter
Hasil 5 : MU
Hasil 6 : MU
Hasil 7 : MU
Hasil 8 : Inter
Pertandingan selesai
PS C:\Users\levina\OneDrive\Documents\golang>
```

Penjelasan:

Program ini menampilkan klub pemenang di setiap pertandingan.

• UNGUIDED 4

Code:

```
SMT2 > Pertemuan6 > -go 103112400041_Unguided4.go > ...
1 //BERTHA ADELA
2 //103112400041
3 package main
4
5 import (
6     "fmt"
7     "strings"
8 )
9
10 const NMAX int = 127
11
12 type tabel struct {
13     arr [NMAX]rune
14     m int
15 }
16
17 func isiArray(t *tabel) {
18     var c rune
19     t.m = 0
20     fmt.Print("Teks: ")
21     for t.m < NMAX && c != '.' {
22         fmt.Scanf("%c", &c)
23         if c != '.' {
24             t.arr[t.m] = c
25             t.m=t.m+1
26         }
27     }
28 }
29
30
31 func cetakArray(t tabel) {
32     for i := 0; i < t.m; i++ {
33         fmt.Printf("%c", t.arr[i])
34     }
35     fmt.Println()
36 }
37
38 func balikanArray(t *tabel) {
39     for i := 0; i < t.m/2; i++ {
40         t.arr[i], t.arr[t.m-i-1] = t.arr[t.m-i-1], t.arr[i]
41     }
42 }
43
44 func palindrom(t tabel) bool {
45     // Membersihkan input untuk mengabaikan spasi dan simbol
46     cleaned := ""
47     for i := 0; i < t.m; i++ {
48         if t.arr[i] != ' ' && t.arr[i] != '.' { // Abaikan spasi dan titik
49             cleaned += strings.ToLower(string(t.arr[i]))
50         }
51     }
52 }
```

```

53     // Periksa apakah string yang sudah dibersihkan adalah palindrom
54     length := len(cleaned)
55     for i := 0; i < length/2; i++ {
56         if cleaned[i] != cleaned[length-i-1] {
57             return false
58         }
59     }
60     return true
61 }
62
63 func main() {
64     var tab tabel
65     isiArray(&tab)
66     if palindrom(tab) {
67         fmt.Println("Palindrom:",palindrom(tab))
68     } else {
69         fmt.Println("Palindrom:",palindrom(tab))
70     }
71 }

```

Output:

```

PS C:\Users\levina\OneDrive\Documents\golang> go run "c:\Users\levina\OneDrive\Documents\golang\nguided4.go"
Teks: K A T A K .
Palindrom: true
PS C:\Users\levina\OneDrive\Documents\golang> go run "c:\Users\levina\OneDrive\Documents\golang\nguided4.go"
Teks: S E N A N G .
Palindrom: false
PS C:\Users\levina\OneDrive\Documents\golang>

```

Penjelasan:

Program di atas akan menyatakan apakah teks yang diinputkan palindrom atau tidak.

IV. KESIMPULAN

Manfaat array bermacam-macam, yaitu memudahkan dalam menyusun suatu data, membuat data menjadi lebih rapi dan mudah dilihat, mudah di akses, lebih sederhana namun tetap estetik, dan lebih efisien dalam penggunaannya.

REFERENSI

MODUL 6 STRUCT & ARRAY