

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL 6
STRUCT & ARRAY**



Oleh:

HISYAM NURDIATMOKO

103112400049

IF-12-01

**S1 TEKNIK INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO**

2025

I. DASAR TEORI

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 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.

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.

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.

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 Guided 1:

```
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:\1_Matkul\Alpro2\week8_array\103112400049_MODULE6> go run "d:\1_Matkul\Alpro2\week8_array\103112400049_MODULE6\
o =====
Toko Sembako Makmur
Jl. Raya No. 123, Jakarta
Tanggal: 25-04-2025 14:43
=====
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 D:\1_Matkul\Alpro2\week8_array\103112400049_MODULE6>

```

Deskripsi :

Program Go Guided 1 ini untuk mencetak struk belanja. Terdapat dua struct utama, yaitu Item yang merepresentasikan barang dengan atribut nama, harga, dan jumlah, serta Receipt yang berisi informasi toko, tanggal, daftar item, dan total harga belanja. Program ini juga memiliki dua metode: CalculateTotal untuk menghitung total harga berdasarkan harga dan jumlah setiap item, serta PrintReceipt yang mencetak struk dengan informasi barang yang dibeli, harga, jumlah, total per item, dan total belanja keseluruhan. Di dalam fungsi main,

sebuah struk belanja dibuat dengan data toko dan daftar barang, kemudian outputnya adalah total harga dihitung dan struk.

Source code Guided 2 :

```
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:\1_Matkul\Alpro2\week8_array\103112400049_MODUL6> go run "d:\1_Matkul\Alpro2\week8_array\103112400049_MODUL6\
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 D:\1_Matkul\Alpro2\week8_array\103112400049_MODUL6>
```

Deskripsi :

Program Go Guided 2 ini bertujuan untuk mengelola dan menampilkan data nilai mahasiswa. Pertama, program mendeklarasikan dan menginisialisasi array satu dimensi berisi nilai mahasiswa, kemudian menampilkan nilai setiap mahasiswa dengan menggunakan loop for dan range. Selanjutnya, program menghitung rata-rata nilai mahasiswa, nilai tertinggi, dan nilai terendah dengan melakukan iterasi melalui array dan menyimpan hasilnya dalam variabel yang sesuai. Setelah itu, program melanjutkan dengan contoh array dua dimensi untuk menyimpan nilai ujian mahasiswa di dua mata kuliah (Matematika dan Bahasa). Program menampilkan nilai untuk masing-masing mata kuliah dari setiap mahasiswa menggunakan loop yang berbeda.

III. UNGUIDED

Source code Unguided 1 :

```
package main

//103112400049 Hisyam Nurdiatmoko

import (
    "fmt"
    "math"
)

type titik struct {
    x, y int
}

type lingkaran struct {
    cx, cy, r int
}

func jarak(p, q titik) float64 {
    return math.Sqrt(float64((p.x-q.x)*(p.x-q.x) + (p.y-q.y)*(p.y-q.y)))
}

func didalam(c lingkaran, p titik) bool {
    return jarak(titik{c.cx, c.cy}, p) <= float64(c.r)
}

func main() {
    var c1, c2 lingkaran
    var hisyam titik
    fmt.Scan(&c1.cx, &c1.cy, &c1.r)
    fmt.Scan(&c2.cx, &c2.cy, &c2.r)
    fmt.Scan(&hisyam.x, &hisyam.y)
    diC1 := didalam(c1, hisyam)
    diC2 := didalam(c2, hisyam)
    if diC1 && diC2 {
        fmt.Println("Titik di dalam lingkaran 1 dan 2")
    } else if diC1 {
        fmt.Println("Titik di dalam lingkaran 1")
    } else if diC2 {
        fmt.Println("Titik di dalam lingkaran 2")
    } else {
        fmt.Println("Titik di luar lingkaran 1 dan 2")
    }
}
```


Output:

```
PS D:\1_Matkul\Alpro2\week8_array\103112400049_MODUL6> go run "d:\1_Matkul\Alpro2\week8_array\103112400049_MODUL6\
1 1 5
8 8 4
2 2
Titik di dalam lingkaran 1
PS D:\1_Matkul\Alpro2\week8_array\103112400049_MODUL6> go run "d:\1_Matkul\Alpro2\week8_array\103112400049_MODUL6\
1 2 3
4 5 6
7 8
Titik di dalam lingkaran 2
PS D:\1_Matkul\Alpro2\week8_array\103112400049_MODUL6> go run "d:\1_Matkul\Alpro2\week8_array\103112400049_MODUL6\
5 10 15
-15 4 20
0 0
Titik di dalam lingkaran 1 dan 2
PS D:\1_Matkul\Alpro2\week8_array\103112400049_MODUL6> go run "d:\1_Matkul\Alpro2\week8_array\103112400049_MODUL6\
1 1 5
8 8 4
15 20
Titik di luar lingkaran 1 dan 2
PS D:\1_Matkul\Alpro2\week8_array\103112400049_MODUL6> 
```

Deskripsi:

Program Go Unguided 1 ini untuk menentukan apakah sebuah titik berada di dalam satu atau kedua lingkaran. Terdapat dua struct yang didefinisikan: titik untuk merepresentasikan koordinat titik (x, y) dan lingkaran yang memiliki atribut pusat koordinat (cx, cy) dan radius (r). Program ini juga menggunakan dua fungsi: jarak untuk menghitung jarak antara dua titik menggunakan rumus Pythagoras, dan didalam untuk mengecek apakah titik berada di dalam lingkaran, berdasarkan apakah jarak antara titik dan pusat lingkaran lebih kecil atau sama dengan radius lingkaran.

Di dalam fungsi main, program menerima input untuk dua lingkaran dan sebuah titik, kemudian memeriksa apakah titik tersebut berada di dalam lingkaran pertama, kedua, atau keduanya dengan menggunakan fungsi didalam. Hasilnya kemudian dicetak untuk memberi tahu posisi titik tersebut terkait dengan kedua lingkaran.

Source code Unguided 2 :

```
package main

//103112400049 Hisyam Nurdiatmoko
import "fmt"

func hitungRataRata(hisyam []int) float64 {
    var total int
    for _, num := range hisyam {
        total += num
    }
    return float64(total) / float64(len(hisyam))
}

func hitungSimpanganBaku(hisyam []int) float64 {
    rata := hitungRataRata(hisyam)
    var kuadratSelisih float64
    for _, num := range hisyam {
        selisih := float64(num) - rata
        kuadratSelisih += selisih * selisih
    }
    return kuadratSelisih / float64(len(hisyam))
}

func hitungFrekuensi(hisyam []int, target int) int {
    count := 0
    for _, num := range hisyam {
        if num == target {
            count++
        }
    }
    return count
}

func tampilkanIndeks(hisyam []int) {
    fmt.Print("Indeks ganjil: ")
    for i := 1; i < len(hisyam); i += 2 {
        fmt.Print(hisyam[i], " ")
    }
    fmt.Println()

    fmt.Print("Indeks genap: ")
    for i := 0; i < len(hisyam); i += 2 {
        fmt.Print(hisyam[i], " ")
    }
}
```

```

    fmt.Println()
}

func main() {
    var jumlahElemen int
    fmt.Print("Masukkan jumlah elemen: ")
    fmt.Scan(&jumlahElemen)
    hisyam := make([]int, jumlahElemen)
    for i := 0; i < jumlahElemen; i++ {
        fmt.Printf("Masukkan elemen ke-%d: ", i+1)
        fmt.Scan(&hisyam[i])
    }
    fmt.Println("\nIsi array:", hisyam)
    tampilkanIndeks(hisyam)
    fmt.Printf("Rata-rata: %.2f\n", hitungRataRata(hisyam))
    fmt.Printf("Simpangan baku (tanpa akar): %.2f\n",
    hitungSimpanganBaku(hisyam))
    var angkaUntukFrekuensi int
    fmt.Print("Masukkan angka untuk dihitung frekuensinya: ")
    fmt.Scan(&angkaUntukFrekuensi)
    fmt.Println("Frekuensi angka", angkaUntukFrekuensi, ":",
    hitungFrekuensi(hisyam, angkaUntukFrekuensi))
}

```

Output:

```

PS D:\1_Matkul\Alpro2\week8_array\103112400049_MODULE6> go run "d:\1_Matkul\Alpro2\week8_array\103112400049_MODULE6\
Masukkan jumlah elemen: 5
Masukkan elemen ke-1: 1
Masukkan elemen ke-2: 2
Masukkan elemen ke-3: 3
Masukkan elemen ke-4: 4
Masukkan elemen ke-5: 5

Isi array: [1 2 3 4 5]
Indeks ganjil: 2 4
Indeks genap: 1 3 5
Rata-rata: 3.00
Simpangan baku (tanpa akar): 2.00
Masukkan angka untuk dihitung frekuensinya: 2
Frekuensi angka 2 : 1
PS D:\1_Matkul\Alpro2\week8_array\103112400049_MODULE6>

```

Deskripsi:

Program Go Unguided 2 ini berfungsi untuk mengolah dan menganalisis sebuah array berisi angka dengan beberapa operasi statistik dasar. Program ini memiliki beberapa fungsi utama: `hitungRataRata` untuk menghitung rata-rata elemen-elemen array, `hitungSimpanganBaku` untuk menghitung simpangan baku (varians) dari array, dan `hitungFrekuensi` untuk menghitung berapa kali sebuah angka muncul dalam array. Fungsi lainnya, `tampilkanIndeks`, digunakan untuk menampilkan elemen-elemen yang ada pada indeks ganjil dan genap array. Dalam fungsi `main`, program meminta pengguna untuk memasukkan jumlah elemen array

dan nilai-nilai elemen tersebut, kemudian menampilkan beberapa informasi: isi array, elemen pada indeks ganjil dan genap, rata-rata, simpangan baku (tanpa akar), dan frekuensi kemunculan angka tertentu yang diminta pengguna. Program ini memberikan cara yang efisien untuk menganalisis data dalam array di Go dan menghitung statistik dasar yang umum digunakan.

Source code Unguided 3 :

```
package main

//103112400049 Hisyam Nurdiatmoko

import "fmt"

func main() {
    var klubA, klubB string
    var skor1, skor2 int
    var pemenang_hisyam []string
    var pertandingan_103112400049 int = 1

    fmt.Print("Klub A: ")
    fmt.Scan(&klubA)
    fmt.Print("Klub B: ")
    fmt.Scan(&klubB)

    for {
        fmt.Printf("Pertandingan ke-%d : ", pertandingan_103112400049)
        fmt.Scan(&skor1, &skor2)
        if skor1 < 0 || skor2 < 0 {
            break
        }
        if skor1 > skor2 {
            pemenang_hisyam = append(pemenang_hisyam, klubA)
        } else if skor1 < skor2 {
            pemenang_hisyam = append(pemenang_hisyam, klubB)
        } else {
            pemenang_hisyam = append(pemenang_hisyam, "Draw")
        }
        pertandingan_103112400049++
    }
    for i, result := range pemenang_hisyam {
        fmt.Printf("Hasil %d: %s\n", i+1, result)
    }
    fmt.Println("Pertandingan selesai")
}
```

Output:

```
PS D:\1_Matkul\Alpro2\week8_array\103112400049_MODUL6> go run "d:\1_Matkul\Alpro2\week8_array\103112400049_MODUL6\
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 D:\1_Matkul\Alpro2\week8_array\103112400049_MODUL6> |
```

Deskripsi:

Program Go Unguided 3 ini digunakan untuk mencatat hasil pertandingan antara dua klub sepak bola. Program ini meminta input nama dua klub (klubA dan klubB) serta skor untuk masing-masing pertandingan antara kedua klub. Program kemudian terus menerima input skor untuk setiap pertandingan hingga skor yang dimasukkan bernilai negatif, yang menandakan akhir input pertandingan.

Setiap hasil pertandingan dibandingkan, dan berdasarkan skor yang dimasukkan, pemenang dari pertandingan (atau "Draw" jika skor sama) ditambahkan ke dalam slice pemenang_hisyam. Setelah semua pertandingan selesai, program akan menampilkan hasil dari setiap pertandingan yang telah dilakukan.

Program ini berjalan dalam loop yang terus meminta input skor pertandingan hingga kondisi berhenti (skor negatif), dan kemudian mencetak hasil dari setiap pertandingan yang terjadi.

Source code Unguided 4 :

```
package main

//103112400049 Hisyam Nurdiamoko

import "fmt"

const NMAX int = 127

type tabel [NMAX]rune

func isiArray(t *tabel, n *int) {
    var Teks rune
    *n = 0
    fmt.Print("Teks    : ")
    for {
        fmt.Scanf("%c", &Teks)
        if Teks == '.' || *n >= NMAX {
            break
        }
        if Teks != ' ' && Teks != '\n' {
            (*t)[*n] = Teks
            *n++
        }
    }
}

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

func balikanArray(t *tabel, n int) {
    for i := 0; i < n/2; i++ {
        t[i], t[n-1-i] = t[n-1-i], t[i]
    }
}

func palindrom(t tabel, n int) bool {
    var tabClone tabel
    copy(tabClone[:], t[:])
```

```

    balikArray(&tabClone, n)
    for i := 0; i < n; i++ {
        if t[i] != tabClone[i] {
            return false
        }
    }
    return true
}

func main() {
    var tab tabel
    var hisyam int
    isiArray(&tab, &hisyam)
    fmt.Print("Teks   : ")
    for i := 0; i < hisyam; i++ {
        fmt.Printf("%c ", tab[i])
    }
    fmt.Println()
    balikArray(&tab, hisyam)
    cetakArray(tab, hisyam)
    fmt.Print("Palindrom   ? ")
    if palindrom(tab, hisyam) {
        fmt.Println("true")
    } else {
        fmt.Println("false")
    }
}

```

Output:

```

PS D:\1_Matkul\Alpro2\week8_array\103112400049_MODUL6> go run "d:\1_Matkul\Alpro2\week8_array\103112400049_MODUL6\
Teks           : K A T A K .
Teks           : K A T A K
Reverse teks   : K A T A K
Palindrom      ? true
PS D:\1_Matkul\Alpro2\week8_array\103112400049_MODUL6> go run "d:\1_Matkul\Alpro2\week8_array\103112400049_MODUL6\
Teks           : S E N A N G .
Teks           : S E N A N G
Reverse teks   : G N A N E S
Palindrom      ? false
PS D:\1_Matkul\Alpro2\week8_array\103112400049_MODUL6>

```

Deskripsi:

Program Go ini dirancang untuk mengolah teks dengan membalikkan urutan karakter dan memeriksa apakah teks tersebut merupakan palindrom. Program dimulai dengan mendeklarasikan tipe data tabel yang merupakan array untuk menyimpan karakter hingga 127 elemen. Fungsi isiArray digunakan untuk meminta input teks dari pengguna, di mana karakter-karakter dimasukkan satu per satu hingga tanda titik (.) atau batas array tercapai. Setelah teks dimasukkan, fungsi cetakArray digunakan untuk menampilkan teks dalam urutan normal,

sedangkan fungsi `balikanArray` membalikkan urutan karakter dalam array. Fungsi `palindrom` memeriksa apakah teks tersebut dapat dibaca sama dari depan dan belakang dengan membandingkan array asli dengan array yang telah dibalik. Fungsi utama `main` menjalankan seluruh proses ini: menerima input, menampilkan teks, membalikkan dan mencetaknya, serta memeriksa apakah teks tersebut palindrom. Program ini memberikan cara untuk memanipulasi dan memeriksa teks dalam format yang sederhana.

IV. KESIMPULAN

Kesimpulan dari laporan praktikum ini adalah bahwa pemrograman dengan menggunakan **struct** dan **array** sangat berguna untuk mengelola data yang terstruktur dan kompleks. Dengan **struct**, kita dapat mengelompokkan data yang memiliki relasi tertentu dalam satu unit, seperti yang terlihat dalam contoh aplikasi struk belanja yang menggunakan struct untuk mendefinisikan item dan struk belanja. Di sisi lain, **array** memungkinkan kita untuk menyimpan data dalam bentuk koleksi dengan ukuran tetap, sedangkan **slice** dan **map** menawarkan fleksibilitas yang lebih besar, seperti dalam contoh pengelolaan data nilai mahasiswa yang menggunakan array dan array dua dimensi. Praktikum ini menunjukkan pentingnya pemahaman terhadap penggunaan struct dan array dalam bahasa pemrograman Go untuk mengelola data secara efisien, melakukan analisis, serta mempermudah implementasi berbagai jenis algoritma dalam pemrograman.

V. REFERENSI

Modul 6 Struct dan Array. Praktikum Algoritma Pemrograman 2