

LAPORAN
PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2
MODUL 6
STRUCT & ARRAY



Oleh:

NAMA: MOHAMMAD REYHAN ARETHA FATIN

NIM: 103112400078

KELAS: IF-12-01

S1 TEKNIK INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2025

I. DASAR TEORI

1. Tipe Bentuk

Tipe bentuk memungkinkan pemrograman untuk mendefinisikan suatu tipe data baru pada suatu bahasa pemrograman. Tipe bentuk 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

Struct atau struktur data adalah cara untuk mengelompokkan beberapa data menjadi satu kesatuan. Misalnya, untuk mempresentasikan waktu, kita bisa buat struct waktu yang berisi jam, menit, dan detik. Dengan begitu, semua data terkait waktu bisa dikelola bersama-sama dalam satu variabel.

2. Array

Array adalah struktur data yang terdiri atas elemen-elemen homogen yang tersusun dalam urutan tertentu dan diakses menggunakan indeks. Di Go, ukuran array bersifat statis ditentukan saat deklarasi dan tidak bisa diubah saat runtime

3. Slice dan Map: Struktur Data Dinamis

Slice sebagai alternatif array dinamis. Slice adalah referensi ke array dasar dengan ukuran fleksibel. Slice dapat diperbesar dengan fungsi append, membuat pengelolaan koleksi data menjadi lebih fleksibel dibandingkan array biasa. Selain slice, map adalah struktur data asosiatif yang menyimpan pasangan kunci-nilai. Tidak seperti array yang diindeks dengan integer, map menggunakan kunci dari berbagai tipe data.

GUIDED 1

SOURCE CODE:

```
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:\ALGORITMA PROGRAMING\PraktikumSem2\103112400078_MODUL6> go run "d:\ALGORITMA PROGRAMING\PraktikumSem2\103112400078_MODUL6\main.go"
=====
Toko Sembako Makmur
Jl. Raya No. 123, Jakarta
Tanggal: 25-04-2025 13:25
=====
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:\ALGORITMA PROGRAMING\PraktikumSem2\103112400078_MODUL6>

```

DEKSRIPSI:

Program ini merupakan simulasi pencetakan struk belanja menggunakan bahasa Go yang memanfaatkan struct untuk merepresentasikan data barang (Item) dan struk belanja (Receipt). Program ini menyimpan informasi seperti nama toko, tanggal transaksi, daftar barang belanjaan, dan total harga. Melalui method `CalculateTotal()`, program menghitung total belanja dari harga dan jumlah tiap item, lalu menampilkannya dalam format struk yang rapi melalui `PrintReceipt()`. Program ini mencerminkan penerapan konsep struktur data dan pemrograman berorientasi objek secara sederhana dan fungsional dalam konteks sistem kasir.

GUIDED 2

SOURCE CODE:

```
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:\ALGORITMA PROGRAMING\PraktikumSem2\103112400078_MODUL6> go run "d:\ALGORITMA PROGRAMING\Prak
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:\ALGORITMA PROGRAMING\PraktikumSem2\103112400078_MODUL6>

```

DEKSRIPSI:

Program ini menampilkan pengolahan data nilai mahasiswa menggunakan array satu dimensi dan dua dimensi. Pertama, program mendeklarasikan array berisi lima nilai mahasiswa, lalu mencetak tiap nilai, menghitung rata-rata, serta menentukan nilai tertinggi dan terendah. Selanjutnya, program juga menampilkan contoh array dua dimensi yang merepresentasikan nilai ujian tiga mahasiswa dalam dua mata kuliah, yaitu Matematika dan Bahasa, dan mencetak nilai masing-masing mahasiswa secara terstruktur. Program ini menunjukkan pemanfaatan array, perulangan, dan logika dasar untuk analisis data sederhana.

II. UNGUIDED

UNGUIDED 1

SOURCE CODE

```
package main
// Mohammad Reyhan Aretha Fatin
// 103112400078
import (
    "fmt"
    "math"
)

type Titik struct {
    X, Y float64
}

type Lingkaran struct {
    Pusat Titik
    Radius float64
}

func hitungJarak(a, b Titik) float64 {
    return math.Sqrt(math.Pow(a.X-b.X, 2) + math.Pow(a.Y-b.Y, 2))
}

func diDalamLingkaran(l Lingkaran, t Titik) bool {
    return hitungJarak(l.Pusat, t) <= l.Radius
}

func main() {
    var l1, l2 Lingkaran
    var titik Titik

    fmt.Print("Masukkan koordinat dan radius lingkaran 1 (x y r): ")
    fmt.Scan(&l1.Pusat.X, &l1.Pusat.Y, &l1.Radius)

    fmt.Print("Masukkan koordinat dan radius lingkaran 2 (x y r): ")
    fmt.Scan(&l2.Pusat.X, &l2.Pusat.Y, &l2.Radius)

    fmt.Print("Masukkan koordinat titik (x y): ")
    fmt.Scan(&titik.X, &titik.Y)

    dalamL1 := diDalamLingkaran(l1, titik)
    dalamL2 := diDalamLingkaran(l2, titik)
```

```

switch {
case dalamL1 && dalamL2:
    fmt.Println("Titik di dalam lingkaran 1 dan 2")
case dalamL1:
    fmt.Println("Titik di dalam lingkaran 1")
case dalamL2:
    fmt.Println("Titik di dalam lingkaran 2")
default:
    fmt.Println("Titik di luar lingkaran 1 dan 2")
}
}

```

OUTPUT

```

PS D:\ALGORITMA PROGRAMING\PraktikumSem2\103112400078_MODUL6> go run "d:\ALGORITMA PROGRAMING\PraktikumSem2\103112400078_MODUL6\main.go"
Masukkan koordinat dan radius lingkaran 1 (x y r): 1 1 5
Masukkan koordinat dan radius lingkaran 2 (x y r): 8 8 4
Masukkan koordinat titik (x y): 2 2
Titik di dalam lingkaran 1
PS D:\ALGORITMA PROGRAMING\PraktikumSem2\103112400078_MODUL6> go run "d:\ALGORITMA PROGRAMING\PraktikumSem2\103112400078_MODUL6\main.go"
Masukkan koordinat dan radius lingkaran 1 (x y r): 1 2 3
Masukkan koordinat dan radius lingkaran 2 (x y r): 4 5 6
Masukkan koordinat titik (x y): 7 8
Titik di dalam lingkaran 2
PS D:\ALGORITMA PROGRAMING\PraktikumSem2\103112400078_MODUL6> go run "d:\ALGORITMA PROGRAMING\PraktikumSem2\103112400078_MODUL6\main.go"
Masukkan koordinat dan radius lingkaran 1 (x y r): 5 10 15
Masukkan koordinat dan radius lingkaran 2 (x y r): -15 4 20
Masukkan koordinat titik (x y): 0 0
Titik di dalam lingkaran 1 dan 2
PS D:\ALGORITMA PROGRAMING\PraktikumSem2\103112400078_MODUL6> go run "d:\ALGORITMA PROGRAMING\PraktikumSem2\103112400078_MODUL6\main.go"
Masukkan koordinat dan radius lingkaran 1 (x y r): 1 1 5
Masukkan koordinat dan radius lingkaran 2 (x y r): 8 8 4
Masukkan koordinat titik (x y): 15 20
Titik di luar lingkaran 1 dan 2
PS D:\ALGORITMA PROGRAMING\PraktikumSem2\103112400078_MODUL6> 

```

DEKSRIPSI

Program ini ditujukan untuk menentukan posisi relatif sebuah titik terhadap dua buah lingkaran. Dengan memanfaatkan struct untuk merepresentasikan titik dan lingkaran, program membaca input berupa koordinat pusat dan radius dua lingkaran serta koordinat titik yang diuji. Program menghitung jarak titik terhadap masing-masing pusat lingkaran menggunakan rumus Euclidean, lalu memeriksa apakah titik tersebut berada di dalam, di luar, atau hanya di dalam salah satu dari dua lingkaran. Hasil akhirnya dicetak ke layar sebagai deskripsi posisi titik terhadap kedua lingkaran tersebut.

UNGUIDED 2

SOURCE CODE

```
package main
//Mohammad Reyhan Aretha Fatin
// 103112400078
import "fmt"

func rataRata(reyhan []int) float64 {
    var sum int
    for _, v := range reyhan {
        sum += v
    }
    return float64(sum) / float64(len(reyhan))
}

func simpanganBaku(reyhan []int) float64 {
    mean := rataRata(reyhan)
    var sumSquares float64
    for _, v := range reyhan {
        selisih := float64(v) - mean
        sumSquares += selisih * selisih
    }
    return sumSquares / float64(len(reyhan))
}

func frekuensi(reyhan []int, val int) int {
    count := 0
    for _, v := range reyhan {
        if v == val {
            count++
        }
    }
    return count
}

func main() {
    var n int
    fmt.Print("Masukkan jumlah elemen: ")
    fmt.Scan(&n)

    reyhan := make([]int, n)
    for i := 0; i < n; i++ {
        fmt.Printf("Masukkan elemen ke-%d: ", i)
        fmt.Scan(&reyhan[i])
    }
}
```

```

    }

    fmt.Println("\nIsi array:", reyhan)

    fmt.Print("Indeks ganjil: ")
    for i := 1; i < n; i += 2 {
        fmt.Print(reyhan[i], " ")
    }
    fmt.Println()

    fmt.Print("Indeks genap: ")
    for i := 0; i < n; i += 2 {
        fmt.Print(reyhan[i], " ")
    }
    fmt.Println()

    fmt.Printf("Rata-rata: %.2f\n", rataRata(reyhan))
    fmt.Printf("Simpangan baku (tanpa akar): %.2f\n", simpanganBaku(reyhan))

    var x int
    fmt.Print("Masukkan angka untuk dihitung frekuensinya: ")
    fmt.Scan(&x)
    fmt.Println("Frekuensi angka", x, ":", frekuensi(reyhan, x))
}

```

OUTPUT

```

PS D:\ALGORITMA PROGRAMING\PraktikumSem2\103112400078_MODUL6> go run "d:\ALGORITMA PROGRAMING\Praktikum
Masukkan jumlah elemen: 3
Masukkan elemen ke-0: 1
Masukkan elemen ke-1: 2
Masukkan elemen ke-2: 3

Isi array: [1 2 3]
Indeks ganjil: 2
Indeks genap: 1 3
Rata-rata: 2.00
Simpangan baku (tanpa akar): 0.67
Masukkan angka untuk dihitung frekuensinya: 2
Frekuensi angka 2 : 1
PS D:\ALGORITMA PROGRAMING\PraktikumSem2\103112400078_MODUL6>

```

DEKSRIPSI

Program yang digunakan untuk mengolah data array bilangan bulat. Program meminta pengguna memasukkan sejumlah elemen, lalu menyimpannya dalam array. Setelah itu, program menampilkan isi array, memisahkan dan mencetak elemen dengan indeks ganjil dan genap, menghitung rata-rata, dan menghitung simpangan baku (tanpa akar) dari elemen-elemen tersebut. Selain itu, program juga menyediakan fitur untuk menghitung frekuensi kemunculan suatu bilangan tertentu dalam array. Seluruh perhitungan dilakukan secara manual tanpa menggunakan library eksternal seperti math.

UNGUIDED 3

SOURCE CODE

```
package main
//Mohammad Reyhan Aretha Fatin
//103112400078
import "fmt"
func tampilkanHasil(pemenang []string) {
    fmt.Println("\n=== Hasil Pertandingan ===")
    for i, hasil := range pemenang {
        fmt.Printf("Hasil %d: %s\n", i+1, hasil)
    }
    fmt.Println("Pertandingan selesai.")
}

func main() {
    var clubA, clubB string
    var skorA, skorB int
    var hasilPertandingan []string
    nomorPertandingan := 1

    fmt.Print("Klub A: ")
    fmt.Scan(&clubA)
    fmt.Print("Klub B: ")
    fmt.Scan(&clubB)

    for {
        fmt.Printf("Pertandingan ke-%d (format: skorA skorB): ",
nomorPertandingan)
        fmt.Scan(&skorA, &skorB)

        if skorA < 0 || skorB < 0 {
            break
        }
        if skorA > skorB {
            hasilPertandingan = append(hasilPertandingan, clubA)
        } else if skorB > skorA {
            hasilPertandingan = append(hasilPertandingan, clubB)
        } else {
            hasilPertandingan = append(hasilPertandingan, "Draw")
        }
        nomorPertandingan++
    }
    tampilkanHasil(hasilPertandingan)
}
```

OUTPUT

```
PS D:\ALGORITMA PROGRAMING\PraktikumSem2\103112400078_MODUL6> go run "d:\ALGORITMA PROGRAMING\PraktikumSem2\103112400078_MODUL6\main.go"
Klub A: MU
Klub B: Inter
Pertandingan ke-1 (format: skorA skorB): 2 0
Pertandingan ke-2 (format: skorA skorB): 1 2
Pertandingan ke-3 (format: skorA skorB): 2 2
Pertandingan ke-4 (format: skorA skorB): 1 0
Pertandingan ke-5 (format: skorA skorB): 3 2
Pertandingan ke-6 (format: skorA skorB): 1 0
Pertandingan ke-7 (format: skorA skorB): 5 2
Pertandingan ke-8 (format: skorA skorB): 2 3
Pertandingan ke-9 (format: skorA skorB): -1 2

=== Hasil Pertandingan ===
Hasil 1: MU
Hasil 2: Inter
Hasil 3: Draw
Hasil 4: MU
Hasil 5: MU
Hasil 6: MU
Hasil 7: MU
Hasil 8: Inter
Pertandingan selesai.
PS D:\ALGORITMA PROGRAMING\PraktikumSem2\103112400078_MODUL6>
```

DEKSRIPSI

Program ini merupakan simulasi rekap hasil pertandingan dua klub sepak bola. Program meminta input nama dua klub, lalu menerima skor dari masing-masing pertandingan secara berulang. Setiap hasil pertandingan ditentukan berdasarkan skor: jika skor klub A lebih tinggi, klub A menang; jika sebaliknya, klub B menang; dan jika skornya sama, hasilnya dianggap imbang (Draw). Proses input terus berlangsung hingga salah satu skor bernilai negatif sebagai tanda berhenti. Seluruh hasil pertandingan disimpan dalam array dan ditampilkan secara berurutan melalui fungsi khusus `tampilkanHasil`, sehingga memudahkan pengguna melihat rekap siapa yang menang di setiap pertandingan.

UNGUIDED 4

SOURCE CODE

```
package main
// Mohammad Reyhan Aretha Fatin
//103112400078
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 m int
    isiArray(&tab, &m)
    balikArray(&tab, m)
    cetakArray(tab, m)
    fmt.Print("Palindrom: ", palindrom(tab, m))
}

```

OUTPUT

```

PS D:\ALGORITMA PROGRAMING\PraktikumSem2\103112400078_MODUL6> go run "d:\ALGORITMA PROGRAMING\PraktikumSem2\103112400078_MODUL6\main.go"
Teks: K A T A K .
Reverse teks: K A T A K
Palindrom: true
PS D:\ALGORITMA PROGRAMING\PraktikumSem2\103112400078_MODUL6> go run "d:\ALGORITMA PROGRAMING\PraktikumSem2\103112400078_MODUL6\main.go"
Teks: S E N A N G .
Reverse teks: G N A N E S
Palindrom: false
PS D:\ALGORITMA PROGRAMING\PraktikumSem2\103112400078_MODUL6>

```

DEKSRIPSI

Program ini berfungsi untuk membaca sebuah teks karakter demi karakter hingga karakter titik (.) sebagai penanda akhir input, kemudian menyimpan karakter tersebut dalam array bertipe rune dengan kapasitas maksimum 127 elemen. Program akan menghapus spasi dan newline dari input, lalu membalik urutan isi array dan mencetak hasil teks yang telah dibalik. Selain itu, program juga memeriksa apakah urutan karakter tersebut membentuk palindrom, yaitu susunan karakter yang tetap sama meskipun dibaca dari depan atau belakang. Pengecekan dilakukan dengan membandingkan array asli dengan salinannya yang telah dibalik. Hasil akhir berupa teks yang dibalik dan informasi apakah teks tersebut palindrom atau tidak.

III. KESIMPULAN

Praktikum ini menunjukkan secara nyata bagaimana pemanfaatan konsep dasar pemrograman seperti array, slice, dan tipe bentukan (struct) dapat diaplikasikan untuk menyelesaikan berbagai persoalan komputasional dalam kehidupan sehari-hari. Mulai dari menentukan posisi titik terhadap lingkaran, melakukan analisis nilai numerik, hingga mengevaluasi apakah suatu kata merupakan palindrom, seluruh studi kasus dalam praktikum ini memperkuat pemahaman terhadap logika algoritmik dan manipulasi struktur data. Selain melatih keterampilan teknis dalam menulis kode yang efisien dan modular, praktikum ini juga menekankan pentingnya representasi data yang tepat untuk menghasilkan solusi yang akurat, mudah dibaca, dan dapat dikembangkan lebih lanjut.

REFERENSI

MODUL 6 STRUCT & ARRAY ALGORITMA PEMOGRAMAN 2