

**LAPORAN PRAKTIKUM  
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL VII  
STRUCT & ARRAY**



Oleh:

**FEROS PEDROSA VALENTINO**

103112400055

IF-12-01

**S1 TEKNIK INFORMATIKA  
TELKOM UNIVERSITY PURWOKERTO**

**2025**

## I. DASAR TEORI

Struct adalah tipe data yang memungkinkan pengelompokan elemen-elemen data yang berbeda dalam satu unit. Setiap elemen dalam struct disebut atribut yang bisa memiliki tipe data yang beda. Struct sering digunakan untuk menggambarkan objek dunia nyata dalam pemrograman.

Karakteristik struct:

- a. Tipe data heterogen  
Field dalam struct dapat memiliki tipe data yang berbeda, memungkinkan penggunaan tipe data yang lebih kompleks.
- b. Nama field  
Setiap field dalam struct memiliki nama yang unik.
- c. Notasi Titik  
Untuk mengakses atau memodifikasi field, kita menggunakan notasi titik (.).
- d. Komposisi Struct  
Struct mendukung komposisi (embedding), yang memungkinkan struct mengandung struct lain di dalamnya.

Array adalah struktur data yang digunakan untuk menyimpan sejumlah elemen dengan tipe data yang sama dalam satu variabel. Elemen-elemen dalam array disusun secara berurutan dan diakses menggunakan indeks. Array memiliki ukuran yang tetap, artinya kita perlu menentukan jumlah elemen yang akan disimpan di dalamnya saat array dideklarasikan. Dengan array kita bisa mengelola data dalam jumlah besar dengan cara efisien, karena array menyediakan akses langsung ke elemen berdasarkan indeks.

Karakteristik Array:

- a. Ukuran tetap  
Ukuran array di Go ditentukan saat deklarasi dan tidak dapat diubah selama runtime. Ini membuat array kurang fleksibel dibandingkan tipe data lain seperti slice.
- b. Tipe data Homogen  
Semua elemen dalam array harus memiliki tipe data yang sama, seperti int, string, atau float64.
- c. Indeks dimulai dari nol

Elemen pertama array berada pada indeks 0, elemen kedua di indeks 1, dan seterusnya.

d. Pengaksesan elemen

Elemen dalam array diakses menggunakan indeks dengan notasi `array[index]`.

## II. GUIDED

### 1. Guided 1

Source code:

```
//Feros Pedrosa Valentino

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 C:\ALPRO\semester2_alpro2\alpro2_modul7> go run "c:\ALPRO\semester2_alpro2\alpro2_modul7\coso1\coso1.go"
=====
Toko Sembako Makmur
Jl. Raya No. 123, Jakarta
Tanggal: 21-04-2025 11:56
=====
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!

```

#### Deskripsi Program:

Program diatas ditulis dalam bahasa go dan berfungsi untuk mencetak nota belanjaan. Program ini menggunakan dua struct utama yaitu Item yang merepresentasikan barang yang dibeli dengan atribut nama, harga, dan jumlah, serta Receipt yang merepresentasikan nota belanja dengan informasi tentang toko, tanggal transaksi, daftar barang, dan total belanja. Dalam struktur Receipt terdapat dua method penting yaitu CalculateTotal, yang bertugas menghitung total harga semua item dengan cara mengalikan harga tiap item dengan jumlahnya, lalu menjumlahkan hasilnya, serta PrintReceipt, yang mencetak struk belanja ke konsol dengan format terstruktur. Method ini menampilkan informasi tentang toko, tanggal, rincian setiap item, dan total belanja secara keseluruhan.

Di dalam fungsi main, objek Receipt diinisialisasi dengan rincian toko, tanggal saat ini, dan daftar item yang dibeli. Setelah total belanja dihitung dengan memanggil CalculateTotal, program akan mencetak struk belanja menggunakan PrintReceipt. Hal ini memberikan pengguna tampilan yang jelas dan rapi mengenai transaksi mereka.

## 2. Guided 2

Source code:

```
//Feros Pedrosa Valentino

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 {
```

```

II      terendah = nilai
II      }
IV     }
LV
VI     fmt.Printf("Nilai tertinggi: %d\n", tertinggi)
II     fmt.Printf("Nilai terendah: %d\n", terendah)
II
IX     // Contoh array 2 dimensi
LV     fmt.Println("\nContoh Array 2 Dimensi:")
LI     fmt.Println("=====")
II
II     // Nilai ujian mahasiswa dalam 2 mata kuliah
II     (Matematika, Bahasa)
IV     nilaiUjian := [3][2]int{
LV         {80, 85},
VI         {90, 75},
II         {70, 95},
II     }
IX
LV     // Menampilkan nilai ujian per mahasiswa
XI     fmt.Println("Nilai Ujian Mahasiswa (Matematika,
II     Bahasa):")
II     for i, nilai := range nilaiUjian {
II         fmt.Printf("Mahasiswa %d: Matematika = %d,
IV         Bahasa = %d\n", i+1, nilai[0], nilai[1])
XV     }

```

Output:

```

PS C:\ALPRO\semester2_alpro2\alpro2_modul7> go run "c:\ALPRO\semester2_alpro2\alpro2_modul7\coso2\coso2.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

```

Deskripsi program:

Program diatas ditulis dalam bahasa go dan berfungsi untuk mengelola data nilai mahasiswa. Pertama, deklarasikan dan inisialisasi sebuah array satu dimensi yang berisi nilai dari lima mahasiswa. Lalu program mencetak nilai masing-masing mahasiswa dengan loop. Kemudian, program menghitung rata-rata nilai dengan cara menjumlahkan semua



nilai dan membaginya dengan jumlah mahasiswa, kemudian hasilnya ditampilkan. Program juga akan mencari nilai tertinggi dan terendah dari array tersebut dengan membandingkan setiap nilai dalam loop. Setelah analisis nilai mahasiswa selesai, program beralih ke contoh array dua dimensi yang menyimpan nilai ujian mahasiswa dalam dua mata kuliah, yaitu Matematika dan Bahasa. Program kemudian menampilkan nilai ujian untuk setiap mahasiswa.

### III. UNGUIDED

#### 1. Unguided 1

Source code:

```
//Feros Pedrosa Valentino

package main

import (
    "fmt"
    "math"
)

type Titik struct {
    x int
    y int
}

type Lingkaran struct {
    Pusat Titik
    Radius int
}

func jarak(p1, p2 Titik) float64 {
    return math.Sqrt(float64((p1.x-p2.x)*(p1.x-p2.x) + (p1.y-p2.y)*(p1.y-p2.y)))
}

func adalahTitikDalamLingkaran(titik Titik, lingkaran Lingkaran) bool {
    return jarak(titik, lingkaran.Pusat) < float64(lingkaran.Radius)
}

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

    fmt.Scan(&l1.Pusat.x, &l1.Pusat.y, &l1.Radius)
    fmt.Scan(&l2.Pusat.x, &l2.Pusat.y, &l2.Radius)
    fmt.Scan(&t.x, &t.y)

    dalamLingkaran1 := adalahTitikDalamLingkaran(t, l1)
```

```

    dalamLingkaran2 := adalahTitikDalamLingkaran(t, l2)

    if dalamLingkaran1 && dalamLingkaran2 {

        fmt.Println("Titik di dalam lingkaran 1 dan 2")

    } else if dalamLingkaran1 {

        fmt.Println("Titik di dalam lingkaran 1")

    } else if dalamLingkaran2 {

        fmt.Println("Titik di dalam lingkaran 2")

    } else {

        fmt.Println("Titik di luar lingkaran 1 dan 2")

    }

}

```

### Output:

```

PS C:\ALPRO\semester2_alpro2\alpro2_array&struct> go run "c:\ALPRO\semester2_alpro2\alpro2_array&struct\unguided1\unguided1.go"
1 1 5
8 8 4
2 2
Titik di dalam lingkaran 1
PS C:\ALPRO\semester2_alpro2\alpro2_array&struct> go run "c:\ALPRO\semester2_alpro2\alpro2_array&struct\unguided1\unguided1.go"
1 2 3
4 5 6
7 8
Titik di dalam lingkaran 2
PS C:\ALPRO\semester2_alpro2\alpro2_array&struct> go run "c:\ALPRO\semester2_alpro2\alpro2_array&struct\unguided1\unguided1.go"
5 10 15
-15 4 20
0 0
Titik di dalam lingkaran 1 dan 2
PS C:\ALPRO\semester2_alpro2\alpro2_array&struct> go run "c:\ALPRO\semester2_alpro2\alpro2_array&struct\unguided1\unguided1.go"
1 1 5
8 8 4
15 20
Titik di luar lingkaran 1 dan 2

```

### Deskripsi program:

Program diatas ditulis dalam bahasa go dan berfungsi menentukan posisi sebuah titik terhadap dua lingkaran. Pertama definisikan dua struktur data yaitu Titik yang menyimpan koordinat x dan y, serta Lingkaran yang menyimpan pusat lingkaran dan radiusnya. Fungsi jarak digunakan untuk menghitung jarak antara dua titik berdasarkan rumus. Selanjutnya fungsi adalahTitikDalamLingkaran berguna untuk menentukan apakah suatu titik berada di dalam lingkaran dengan membandingkan jarak titik tersebut dari pusat lingkaran dengan radius lingkaran itu sendiri. Di bagian fungsi main, program meminta input dari pengguna untuk mendefinisikan dua lingkaran dan satu titik, kemudian program mengevaluasi dan mencetak hasil posisi titik tersebut. Terakhir program akan menentukan apakah titik berada di dalam satu atau kedua lingkaran, atau di luar keduanya dengan if-else.

## 2. Unguided 2

Source code:

```
//Feros Pedrosa

package main

import (
    "fmt"
    "math"
)

func main() {
    var N int

    fmt.Print("Masukkan jumlah elemen array: ")

    fmt.Scan(&N)

    array := make([]int, N)

    for i := 0; i < N; i++ {
        fmt.Printf("Masukkan elemen ke-%d: ", i)

        fmt.Scan(&array[i])
    }

    // a.

    fmt.Println("a. Isi dari semua elemen array:", array)

    // b.

    fmt.Print("b. Elemen dengan indeks ganjil: ")

    for i := 1; i < N; i += 2 {
        fmt.Print(array[i], " ")
    }

    fmt.Println()

    // c.

    fmt.Print("c. Elemen dengan indeks genap: ")

    for i := 0; i < N; i += 2 {
        fmt.Print(array[i], " ")
    }
}
```

```

fmt.Println()

// d.

var x int

fmt.Print("d. Masukkan bilangan x untuk indeks kelipatan: ")

fmt.Scan(&x)

fmt.Print("Indeks kelipatan ", x, ": ")

for i := 0; i < N; i++ {

    if i%x == 0 {

        fmt.Print(array[i], " ")

    }

}

fmt.Println()

// e.

var indexToDelete int

fmt.Print("e. Masukkan indeks untuk dihapus: ")

fmt.Scan(&indexToDelete)

array = append(array[:indexToDelete], array[indexToDelete+1:]...)

fmt.Println("Isi array setelah elemen dihapus:", array)

// f.

var sum int

for _, value := range array {

    sum += value

}

average := float64(sum) / float64(len(array))

fmt.Printf("f. Rata-rata dari bilangan yang ada: %.2f\n", average)

// g.

var variance float64

for _, value := range array {

    variance += math.Pow(float64(value)-average, 2)

}

variance /= float64(len(array))

stdDev := math.Sqrt(variance)

```

```

    fmt.Printf("g. Standar Deviasi atau simpangan baku: %.2f\n", stdDev)

    // h.

    var target int

    fmt.Print("h. Masukkan bilangan untuk dicari frekuensinya: ")

    fmt.Scan(&target)

    frequency := 0

    for _, value := range array {

        if value == target {

            frequency++

        }

    }

    fmt.Printf("Frekuensi dari bilangan %d: %d\n", target, frequency)
}

```

### Output:

```

PS C:\ALPRO\semester2_alpro2\alpro2_array&struct> go run "c:\ALPRO\semester2_alpro2\alpro2_array&struct\unguided2\unguided2.go"
Masukkan jumlah elemen array: 5
Masukkan elemen ke-0: 1
Masukkan elemen ke-1: 2
Masukkan elemen ke-2: 3
Masukkan elemen ke-3: 5
Masukkan elemen ke-4: 4
a. Isi dari semua elemen array: [1 2 3 5 4]
b. Elemen dengan indeks ganjil: 2 5
c. Elemen dengan indeks genap: 1 3 4
d. Masukkan bilangan x untuk indeks kelipatan: 1
Indeks kelipatan 1: 1 2 3 5 4
e. Masukkan indeks untuk dihapus: 3
Isi array setelah elemen dihapus: [1 2 3 4]
f. Rata-rata dari bilangan yang ada: 2.50
g. Standar Deviasi atau simpangan baku: 1.12
h. Masukkan bilangan untuk dicari frekuensinya: 2
Frekuensi dari bilangan 2: 1

```

### Deskripsi program:

Program diatas ditulis dalam bahasa go dan berfungsi mengelola dan menganalisis array bilangan bulat. Program dimulai dengan meminta pengguna untuk memasukkan jumlah elemen yang ingin diisikan ke dalam array. Setelah itu, pengguna diminta untuk mengisi elemen-elemen tersebut satu per satu. Program kemudian menampilkan semua elemen dalam array, serta memisahkan elemen berdasarkan indeks ganjil dan genap. Selanjutnya, pengguna dapat memasukkan angka tertentu untuk menentukan indeks kelipatan yang ingin ditampilkan. Program ini juga memperbolehkan pengguna untuk menghapus elemen pada posisi tertentu dan menampilkan isi array setelah penghapusan tersebut. Selain itu, program menghitung rata-rata dari elemen yang tersisa, serta varians dan standar deviasi dari elemen-elemen tersebut. Terakhir pengguna dapat mencari frekuensi kemunculan angka tertentu dalam array.

### 3. Unguided 3

Source code:

```
//Feros Pedrosa

package main

import "fmt"

func main() {

    var klubA, klubB string

    var skorA, skorB int

    var hasil []string

    fmt.Print("Klub A: ")

    fmt.Scanln(&klubA)

    fmt.Print("Klub B: ")

    fmt.Scanln(&klubB)

    for {

        fmt.Printf("Pertandingan %d: ", len(hasil)+1)

        fmt.Scan(&skorA, &skorB)

        if skorA < 0 || skorB < 0 {

            break

        }

        if skorA > skorB {

            hasil = append(hasil, klubA)

        } else if skorB > skorA {

            hasil = append(hasil, klubB)

        } else {

            hasil = append(hasil, "Draw")

        }

    }

    fmt.Println("\nDaftar klub yang memenangkan pertandingan:")

    for i, h := range hasil {

        fmt.Printf("Hasil %d: %s\n", i+1, h)

    }

}
```

```
fmt.Println("Pertandingan selesai")
}
```

### Output:

```
PS C:\ALPRO\semester2_alpro2\alpro2_array&struct> go run "c:\ALPRO\semester2_alpro2\alpro2_array&struct\unguided3\unguided3.go"
Klub A: MU
Klub B: Inter
Pertandingan 1: 2 0
Pertandingan 2: 1 2
Pertandingan 3: 2 2
Pertandingan 4: 0 1
Pertandingan 5: 3 2
Pertandingan 6: 1 0
Pertandingan 7: 5 2
Pertandingan 8: 2 3
Pertandingan 9: -1 2

Daftar Klub yang memenangkan pertandingan:
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
```

### Deskripsi program:

Program diatas ditulis dalam bahasa go dan berfungsi mencatat hasil pertandingan antara dua klub sepak bola. Pertama program akan meminta pengguna untuk memasukkan nama dua klub, yaitu Klub A dan Klub B. Selanjutnya, program akan memasuki sebuah loop di mana pengguna dapat secara berulang kali memasukkan skor pertandingan. Pengguna diminta untuk menginput skor untuk Klub A dan Klub B. Jika salah satu skor yang dimasukkan bernilai negatif, maka loop akan berhenti. Program kemudian akan membandingkan skor dari kedua klub: jika Klub A memiliki skor yang lebih tinggi, nama Klub A akan ditambahkan ke dalam daftar klub yang memenangkan pertandingan; jika Klub B yang lebih unggul, maka nama Klub B yang akan ditambahkan; dan jika skor kedua klub sama, ditambahkan kata "draw". Setelah proses loop selesai, program akan mencetak daftar hasil pertandingan yang telah dimenangkan oleh masing-masing klub, beserta hasil imbangnya, diakhiri dengan pesan bahwa pertandingan telah selesai.



#### 4. Unguided 4

Source code:

```
//Feros Pedrosa

package main

import "fmt"

const NMAX int = 127

type tabel [NMAX]rune

func isiArray(t *tabel, n *int) {

    fmt.Println("Masukkan karakter (gunakan titik untuk berhenti):")

    for i := 0; i < NMAX; i++ {

        var ch rune

        fmt.Scanf("%c", &ch)

        if ch == '.' {

            break

        }

        t[i] = ch

        *n++

    }

}

func cetakArray(t tabel, n int) {

    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-i-1] = t[n-i-1], t[i]

    }

}

func palindrom(t tabel, n int) bool {
```

```

    for i := 0; i < n/2; i++ {

        if t[i] != t[n-i-1] {

            return false

        }

    }

    return true
}

func main() {

    var tab tabel

    var m int

    isiArray(&tab, &m)

    balikanArray(&tab, m)

    fmt.Print("Teks: ")

    cetakArray(tab, m)

    if palindrom(tab, m) {

        fmt.Println("Palindrom? true")

    } else {

        fmt.Println("Palindrom? false")

    }

}

```

### Output:

```

PS C:\ALPRO\semester2_alpro2\alpro2_array&struct> go run "c:\ALPRO\semester2_alpro2\alpro2_array&struct\unguided4\unguided4.go"
• Masukkan karakter (gunakan titik untuk berhenti):
K A T A K.
Teks: K A T A K
Palindrom? true
PS C:\ALPRO\semester2_alpro2\alpro2_array&struct> go run "c:\ALPRO\semester2_alpro2\alpro2_array&struct\unguided4\unguided4.go"
• Masukkan karakter (gunakan titik untuk berhenti):
S E N A N G.
Teks: G N A N E S
Palindrom? false
PS C:\ALPRO\semester2_alpro2\alpro2_array&struct> go run "c:\ALPRO\semester2_alpro2\alpro2_array&struct\unguided4\unguided4.go"
• Masukkan karakter (gunakan titik untuk berhenti):
K A S U R R U S A K.
Teks: K A S U R R U S A K
Palindrom? true
PS C:\ALPRO\semester2_alpro2\alpro2_array&struct> go run "c:\ALPRO\semester2_alpro2\alpro2_array&struct\unguided4\unguided4.go"
• Masukkan karakter (gunakan titik untuk berhenti):
A P A.
Teks: A P A
Palindrom? true

```

### Deskripsi program:

Program diatas ditulis dalam bahasa go dan berfungsi mengelola dan menganalisis karakter dalam bentuk array. Pertama definisikan tipe data baru dengan nama tabel yang merupakan array dengan kapasitas maksimum 127 elemen bertipe rune. Fungsi

isiArray berguna untuk mengisi array dari pengguna di mana pengguna dapat memasukkan karakter berapapun itu hingga pengguna menuliskan titik (‘.’) untuk menghentikan input. Setelah array terisi, fungsi cetakArray digunakan untuk menampilkan semua isi karakter yang ada didalam array. Kemudian program akan membalikan urutan karakter yang ada di array dengan fungsi balikanArray yang melakukan pertukaran elemen dari kedua ujung array menuju ke tengah. Kemudian fungsi palindrom akan mengecek apakah urutan karakter dalam array membentuk sebuah palindrom, yaitu sebuah kata atau frasa yang dibaca sama dari depan maupun belakang. Di dalam fungsi main, program akan menginisialisasi array, mengisi array dengan karakter, membalikkan array, dan mencetak hasilnya. Terakhir, program akan memeriksa dan mencetak apakah urutan karakter tersebut merupakan palindrom atau tidak.

## **IV. KESIMPULAN**

Program yang melibatkan penggunaan struktur data, seperti struct dan array, memegang peranan penting dalam pengelolaan data yang kompleks dan terorganisir. Dengan memanfaatkan struct, kita dapat mengelompokkan elemen data yang berbeda dalam satu kesatuan, sehingga dapat mewakili objek dunia nyata dengan lebih efisien. Di sisi lain, array memberikan cara yang efektif untuk menyimpan dan mengakses sejumlah elemen dengan tipe data yang sama secara berurutan.

Modul ini juga mengilustrasikan penerapan konsep-konsep tersebut dalam bahasa pemrograman Go melalui berbagai contoh program, mulai dari pengelolaan data nilai mahasiswa hingga analisis karakter dalam bentuk array. Dengan pemahaman dan penerapan struktur data ini, kita dapat meningkatkan kemampuan dalam mengelola data, melakukan analisis statistik, dan mengembangkan aplikasi yang lebih kompleks serta fungsional.

## **V. REFERENSI**

MODUL 7. STRUCK & ARRAY - 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>