

LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2
MODUL 6
STRUCK & ARRAY



DISUSUN OLEH:
ANASTASIA ADINDA NARENDRA INDRIANTO
103112400085
S1 IF-12-01
DOSEN:
Dimas Fanny Hebrasianto Permadi, S.ST., M.KOM

S1 TEKNIK INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2025

I. DASAR TEORI

1. Pengertian Bahasa Pemrograman Golang

Golang adalah bahasa pemrograman yang memiliki sejumlah kelebihan yang tidak dimiliki bahasa pemrograman yang lainnya. Hadirnya bahasa pemrograman Go Language (Golang) semakin dirasakan oleh para pengembang. Tidak heran jika banyak orang yang mulai belajar bahasa pemrograman yang satu ini.

Golang merupakan bahasa pemrograman yang dibuat Google dan tujuannya untuk menyempurnakan bahasa pemrograman yang ada, seperti C, Python dan yang lainnya. Golang bisa jadi pilihan yang tepat saat membuat aplikasi baru.

2. Pengertian Input dan Output

- i. *Input* atau masukan adalah data yang diberikan ke dalam program. Masukan ini bisa berasal dari berbagai sumber, seperti pengguna melalui keyboard, mouse, atau suara, dan juga bisa berasal dari sensor atau perangkat lain yang terhubung ke komputer. Dalam pemrograman, input diperlakukan sebagai bahan baku yang akan diproses oleh program.
- ii. *Output* atau keluaran adalah hasil yang diproduksi oleh program setelah mengolah input. Output bisa berbentuk tampilan pada layar, cetakan pada printer, suara, ataupun penyimpanan data ke file. Inti dari program yang kita tulis sebenarnya adalah menghasilkan output yang bermakna dari input yang diberikan.

3. Pengertian Tipe Data

Tipe data adalah kategori data yang akan diproses oleh sebuah program komputer. Dengan tipe data, program dapat mengetahui cara menangani data dengan benar, seperti melakukan operasi matematika pada angka atau memanipulasi string untuk tujuan tertentu. Secara sederhana, pengertian tipe data adalah kategori data yang akan diproses oleh sebuah program komputer. Dengan tipe data, program dapat mengetahui cara menangani data dengan benar, seperti melakukan operasi matematika pada angka atau memanipulasi string untuk tujuan tertentu. Contoh paling sederhana dari tipe data adalah tipe data **integer** yang digunakan untuk menyimpan angka bulat atau tipe data **string** yang digunakan untuk menyimpan rangkaian karakter. Penggunaan tipe data yang benar dalam suatu program memastikan bahwa data diolah secara tepat dan mengurangi risiko kesalahan atau bug dalam program tersebut.

4. Fungsi Tipe Data

- i. **Menentukan Jenis Nilai:** Tipe data memberi tahu program jenis nilai yang akan disimpan dalam variabel. Misalnya, jika kita ingin menyimpan angka, kita menggunakan tipe data integer atau float, sedangkan untuk menyimpan teks, kita menggunakan tipe data string. Ini membantu program memahami bagaimana cara menangani data tersebut.
- ii. **Efisiensi Penggunaan Memori:** Setiap tipe data memerlukan jumlah memori yang berbeda. Jika kita memilih tipe data yang tepat, program bisa menggunakan memori lebih efisien. Misalnya, menggunakan tipe data yang lebih kecil untuk angka yang tidak terlalu besar akan menghemat ruang di memori.
- iii. **Menjamin Konsistensi Data:** Dengan menentukan tipe data, kita memastikan bahwa variabel hanya bisa menyimpan jenis nilai yang sesuai. Misalnya, jika kita mendeklarasikan variabel sebagai tipe integer, program tidak akan bisa memasukkan teks atau jenis data lainnya ke dalamnya. Ini membantu menjaga agar data tetap konsisten dan sesuai dengan yang diharapkan.
- iv. **Memudahkan Operasi pada Data:** Tipe data juga menentukan operasi apa saja yang bisa dilakukan pada data. Misalnya, kita bisa melakukan perhitungan

matematika pada angka, tetapi kita tidak bisa melakukan hal yang sama pada teks. Jadi, dengan menentukan tipe data yang tepat, kita bisa melakukan operasi yang sesuai dengan jenis data yang kita miliki.

5. Pengertian If-Else dan Fungsinya

If-else adalah struktur percabangan dalam pemrograman yang digunakan untuk mengeksekusi kode berdasarkan suatu kondisi. Jika kondisi dalam if bernilai true, maka blok kode di dalamnya akan dijalankan. Jika tidak, program akan memeriksa kondisi dalam else if sebagai alternatif. Jika semua kondisi false, maka else akan dijalankan sebagai pilihan terakhir. Dengan menggunakan if-else, program dapat mengambil keputusan dan menjalankan instruksi yang sesuai berdasarkan kondisi yang diberikan.

If-else berfungsi untuk mengontrol alur program dengan mengeksekusi kode berdasarkan suatu kondisi. If digunakan untuk memeriksa apakah suatu kondisi bernilai true, jika ya, maka blok kode di dalamnya akan dijalankan. Jika tidak, program dapat menggunakan else-if untuk mengevaluasi beberapa kondisi tambahan secara berurutan. Jika semua kondisi false, maka else akan dieksekusi sebagai pilihan terakhir. Selain itu, terdapat if-else bersarang, yang memungkinkan pengecekan kondisi di dalam kondisi lain untuk menangani keputusan yang lebih kompleks.

6. Pengertian While Loop dan Fungsinya

While loop adalah metode perulangan yang mengeksekusi blok kode selama kondisi yang diberikan bernilai true dan akan berhenti ketika kondisi berubah menjadi false. Perulangan ini sangat berguna dalam kasus di mana jumlah iterasi belum diketahui secara pasti, seperti membaca input hingga valid atau menjalankan suatu proses hingga syarat tertentu terpenuhi. Fungsinya adalah;

- i. **Menjalankan Perulangan Berdasarkan Kondisi** – While loop memastikan suatu proses terus berjalan selama kondisi bernilai **true**.
- ii. **Mengatasi Iterasi yang Tidak Diketahui Jumlahnya** – Digunakan ketika jumlah perulangan tidak bisa ditentukan sejak awal, seperti menunggu input yang valid.
- iii. **Mengoptimalkan Kontrol Program** – Membantu mengelola eksekusi kode agar hanya berjalan saat kondisi tertentu terpenuhi, meningkatkan efisiensi program.

7. Pengertian Fungsi dan Manfaatnya

Fungsi merupakan rangkaian instruksi yang menghasilkan suatu nilai dengan memetakan input ke output tertentu. Dalam pemrograman, suatu subprogram dikategorikan sebagai fungsi apabila memiliki deklarasi tipe nilai yang dikembalikan dan menggunakan kata kunci return dalam tubuhnya. Fungsi digunakan dalam berbagai situasi, seperti menetapkan nilai ke suatu variabel melalui assignment, menjadi bagian dari suatu ekspresi, atau digunakan sebagai argumen dalam subprogram lain. Karena fungsi selalu menghasilkan nilai, penamaannya sebaiknya bersifat deskriptif dan mencerminkan hasil yang diberikan, seperti *median*, *rerata*, *nilaiTerbesar*, atau *selesai*.

Function memungkinkan programmer membagi kode menjadi segmen-segmen kecil yang lebih terkelola, masing-masing melakukan bagian tertentu dari tugas yang lebih besar. Tidak hanya membantu dalam mengorganisasi kode secara lebih efisien, hal ini juga memudahkan pemeliharaan dan pengujian kode.

8. Pengertian Prosedur dan Manfaatnya

Prosedur adalah kumpulan instruksi yang dikemas menjadi satu kesatuan untuk menyederhanakan kode dalam program besar. Prosedur tidak mengembalikan nilai karena tidak memiliki deklarasi tipe nilai kembalian dan tidak menggunakan kata kunci *return*. Ketika dipanggil, prosedur langsung memberikan efek pada program utama, seperti halnya instruksi dasar atau fungsi bawaan (*built-in*). Nama prosedur sebaiknya menggunakan kata kerja atau kata yang menggambarkan proses, seperti *cetak*, *hitungRerata*, atau *cariNilai*. Hal ini bertujuan agar lebih mudah dipahami dan sesuai dengan fungsinya dalam program. Dengan menggunakan prosedur, kode program menjadi lebih terstruktur, mudah dibaca, dan dikelola.

Manfaat utama prosedur adalah meningkatkan keterbacaan dan keteraturan kode, sehingga lebih mudah dikelola dan diperbaiki. Dengan memecah program menjadi bagian-bagian kecil, prosedur juga membantu dalam mengurangi pengulangan kode (code redundancy), meningkatkan efisiensi, serta mempermudah debugging dan pengembangan program. Hal ini membuat program lebih modular dan fleksibel untuk diperbarui di masa mendatang.

9. Pengertian Rekursif dan Fungsinya

Rekursi adalah teknik dalam pemrograman di mana sebuah fungsi memanggil dirinya sendiri untuk menyelesaikan masalah yang lebih kecil dari masalah utama.

Pendekatan ini sering digunakan untuk menyelesaikan permasalahan yang dapat dipecah menjadi submasalah serupa, seperti pencarian dalam struktur data pohon atau perhitungan faktorial. Agar tidak berjalan tanpa henti, rekursi memerlukan kondisi dasar yang menentukan kapan fungsi harus berhenti memanggil dirinya sendiri.

Fungsi rekursif memiliki keunggulan dalam menyederhanakan kode untuk masalah yang memiliki pola berulang, seperti algoritma Divide and Conquer atau pemrosesan data yang bersifat hierarkis. Namun, jika tidak dirancang dengan baik, rekursi dapat menyebabkan penggunaan memori yang berlebihan atau bahkan error karena stack overflow. Oleh karena itu, memahami kapan dan bagaimana menggunakan rekursi dengan efisien sangat penting dalam pemrograman.

10. Pengertian Struct dan Array

Struct adalah kumpulan yang terdiri dari elemen-elemen dengan tipe data yang heterogen atau tidak sama. Struktur dapat dideklarasikan menggunakan kata kunci 'struct', dan menggunakan '.' (operator titik) untuk mengakses elemen-elemen tersebut. Array mengacu pada koleksi yang terdiri dari elemen-elemen yang homogen, yaitu bertipe data yang sama. Array dideklarasikan menggunakan '[']. Array menggunakan subskrip '[']' (kurung siku) untuk mengakses elemen-elemennya. Pada dasarnya, array adalah pointer yang menunjuk ke elemen pertama dari sebuah koleksi.

II. GUIDED

1. Guide 1

Source Code:

```
//Anastasia Adinda Narendra Indrianto
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:\Semester 2\ALPRO2 CODING\103112400085_MODUL6> go run "d:\Semester 2\ALPRO2 CODING\103112400085_MODUL6\Guided\1.go"
=====
Toko Sembako Makmur
Jl. Raya No. 123, Jakarta
Tanggal: 25-04-2025 20:21
=====
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:\Semester 2\ALPRO2 CODING\103112400085_MODUL6>

```

Deskripsi Program:

Program ini membuat struk belanja otomatis. Pengguna dapat memasukkan informasi toko, tanggal, dan item yang dibeli (nama, harga, dan jumlah). Program kemudian menghitung total belanja dan mencetak struk yang berisi informasi toko, tanggal, rincian item, serta total harga. Struk ini menggunakan metode untuk menghitung total harga dan mencetaknya dengan format yang terstruktur.

2. Guide 2

Source Code:

```
//Anastasia Adinda Narendra Indrianto
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:\Semester 2\ALPRO2 CODING\103112400085_MODUL6> go run "d:\Semester 2\ALPRO2 CODING\103112400085_MODUL6\Guided\2.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 D:\Semester 2\ALPRO2 CODING\103112400085_MODUL6>

```

Deskripsi Program:

Program ini menghitung dan menampilkan data nilai mahasiswa. Pertama, program menampilkan nilai individu setiap mahasiswa, kemudian menghitung rata-rata nilai, nilai tertinggi, dan terendah. Selain itu, program juga menunjukkan contoh penggunaan array 2 dimensi dengan menampilkan nilai ujian mahasiswa dalam dua mata kuliah (Matematika dan Bahasa).

III. UNGUIDED

1. Unguided 1

Source Code:

```
//Anastasia Adinda Narendra Indrianto
package main
import (
    "fmt"
    "math"
)

type Lingkaran struct {
    x, y, r float64
}
type Titik struct {
    x, y float64
}
func hitungJarakAnastasia(x1, y1, x2, y2 float64) float64 {
    return math.Sqrt(math.Pow(x1-x2, 2) + math.Pow(y1-y2, 2))
}
func titikDalamLingkaran(lingkaran Lingkaran, titik Titik) bool {
    return hitungJarakAnastasia(lingkaran.x, lingkaran.y, titik.x, titik.y) <= lingkaran.r
}

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

    fmt.Print("Masukkan pusat dan radius lingkaran 1 (x, y, radius): ")
    fmt.Scan(&l1.x, &l1.y, &l1.r)
    fmt.Print("Masukkan pusat dan radius lingkaran 2 (x, y, radius): ")
    fmt.Scan(&l2.x, &l2.y, &l2.r)
    fmt.Println("Masukkan koordinat titik Anastasia (x, y):")
    fmt.Scan(&t.x, &t.y)
    lingkaran1 := titikDalamLingkaran(l1, t)
    lingkaran2 := titikDalamLingkaran(l2, t)

    if lingkaran1 && lingkaran2 {
        fmt.Println("Titik di dalam lingkaran 1 dan 2")
    } else if lingkaran1 {
        fmt.Println("Titik di dalam lingkaran 1")
    } else if lingkaran2 {
        fmt.Println("Titik di dalam lingkaran 2")
    } else {
        fmt.Println("Titik di luar lingkaran 1 dan 2")
    }
}
```

Output:

```
PS D:\Semester 2\ALPRO2 CODING\103112400085_MODUL6>
PS D:\Semester 2\ALPRO2 CODING\103112400085_MODUL6> go run "d:\Semester 2\ALPRO2 CODING\103112400085_MODUL6\Unguided\1.go"
Masukkan pusat dan radius lingkaran 1 (x, y, radius): 1 1 5
Masukkan pusat dan radius lingkaran 2 (x, y, radius): 8 8 4
Masukkan koordinat titik Anastasia (x, y):
2 2
Titik di dalam lingkaran 1
PS D:\Semester 2\ALPRO2 CODING\103112400085_MODUL6> go run "d:\Semester 2\ALPRO2 CODING\103112400085_MODUL6\Unguided\1.go"
Masukkan pusat dan radius lingkaran 1 (x, y, radius): 1 2 3
Masukkan pusat dan radius lingkaran 2 (x, y, radius): 4 5 6
Masukkan koordinat titik Anastasia (x, y):
7 8
Titik di dalam lingkaran 2
PS D:\Semester 2\ALPRO2 CODING\103112400085_MODUL6> go run "d:\Semester 2\ALPRO2 CODING\103112400085_MODUL6\Unguided\1.go"
Masukkan pusat dan radius lingkaran 1 (x, y, radius): 5 10 15
Masukkan pusat dan radius lingkaran 2 (x, y, radius): -15 4 20
Masukkan koordinat titik Anastasia (x, y):
0 0
Titik di dalam lingkaran 1 dan 2
PS D:\Semester 2\ALPRO2 CODING\103112400085_MODUL6> go run "d:\Semester 2\ALPRO2 CODING\103112400085_MODUL6\Unguided\1.go"
Masukkan pusat dan radius lingkaran 1 (x, y, radius): 1 1 5
Masukkan pusat dan radius lingkaran 2 (x, y, radius): 8 8 4
Masukkan koordinat titik Anastasia (x, y):
15 20
Titik di luar lingkaran 1 dan 2
PS D:\Semester 2\ALPRO2 CODING\103112400085_MODUL6> 
```

Deskripsi Program:

Program ini memeriksa apakah sebuah titik berada di dalam satu atau dua lingkaran yang diberikan. Pengguna memasukkan koordinat pusat dan radius kedua lingkaran, serta koordinat titik yang diuji. Program menghitung jarak titik dari pusat lingkaran dan membandingkannya dengan radius. Hasilnya menunjukkan apakah titik berada di dalam satu atau kedua lingkaran, atau di luar keduanya.

2. Unguided 2

Source Code:

```
//Anastasia Adinda Narendra Indrianto
package main

import "fmt"

func AnastasiaHitungRataRata(arr []int) float64 {
    var total int
    for _, nilai := range arr {
        total += nilai
    }
    return float64(total) / float64(len(arr))
}

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

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

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

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

    arr := make([]int, jumlahElemen)
    for i := 0; i < jumlahElemen; i++ {
        fmt.Printf("Masukkan elemen ke-%d: ", i+1)
        fmt.Scan(&arr[i])
    }
}
```

```

    }

    fmt.Println("\nIsi array:", arr)
    tampilkanIndeks(arr)
    fmt.Printf("Rata-rata: %.2f\n", AnastasiaHitungRataRata(arr))
    fmt.Printf("Simpangan baku (tanpa akar): %.2f\n", hitungSimpanganBaku(arr))
}

```

Output:

```

PS D:\Semester 2\ALPRO2 CODING\103112400085_MODUL6> go run "d:\Semester 2\ALPRO2 CODING\103112400085_MODUL6\Unguided\2.go"
Masukkan jumlah elemen: 3
Masukkan elemen ke-1: 1
Masukkan elemen ke-2: 2
Masukkan elemen ke-3: 3

Isi array: [1 2 3]
Indeks ganjil: 2
Indeks genap: 1 3
Rata-rata: 2.00
Simpangan baku (tanpa akar): 0.67
PS D:\Semester 2\ALPRO2 CODING\103112400085_MODUL6> 

```

Deskripsi Program:

Program ini memungkinkan pengguna untuk mengisi sebuah array dengan jumlah elemen yang ditentukan, kemudian melakukan berbagai operasi pada array tersebut. Setelah array diisi, program akan menampilkan seluruh elemen dalam array, serta elemen-elemen dengan indeks ganjil dan genap. Program juga menghitung dan menampilkan rata-rata nilai dari elemen array tersebut, serta simpangan baku atau standar deviasi yang menggambarkan sebaran data. Dengan demikian, program ini memberikan gambaran statistik dasar dari sekumpulan data dalam array yang diinput oleh pengguna.

3. Unguided 3

Source Code:

```
//Anastasia Adinda Narendra Indrianto
package main

import "fmt"

func main() {
    var AnastasiaTimSatu, timDua string
    var skorTimSatu, skorTimDua int
    var hasilPertandingan []string

    fmt.Print("Masukkan nama Tim 1: ")
    fmt.Scanln(&AnastasiaTimSatu)
    fmt.Print("Masukkan nama Tim 2: ")
    fmt.Scanln(&timDua)

    hitungTanding := 1
    for {
        fmt.Printf("Masukkan skor untuk pertandingan %d (Tim 1 vs Tim 2): ",
            hitungTanding)
        fmt.Scan(&skorTimSatu, &skorTimDua)

        if skorTimSatu < 0 || skorTimDua < 0 {
            break
        }

        if skorTimSatu > skorTimDua {
            hasilPertandingan = append(hasilPertandingan, AnastasiaTimSatu)
        } else if skorTimDua > skorTimSatu {
            hasilPertandingan = append(hasilPertandingan, timDua)
        } else {
            hasilPertandingan = append(hasilPertandingan, "Seri")
        }
        hitungTanding++
    }

    for i, hasil := range hasilPertandingan {
        fmt.Printf("Hasil pertandingan %d: %s\n", i+1, hasil)
    }
    fmt.Println("Pertandingan selesai")
}
```

Output:

```
PS D:\Semester 2\ALPRO2 CODING\103112400085_MODUL6> go run "d:\Semester 2\ALPRO2 CODING\103112400085_MODUL6\Unguided\3.go"
Masukkan nama Tim 1: aku
Masukkan nama Tim 2: dia
Masukkan skor untuk pertandingan 1 (Tim 1 vs Tim 2): 2 0
Masukkan skor untuk pertandingan 2 (Tim 1 vs Tim 2): 1 2
Masukkan skor untuk pertandingan 3 (Tim 1 vs Tim 2): 2 2
Masukkan skor untuk pertandingan 4 (Tim 1 vs Tim 2): 0 1
Masukkan skor untuk pertandingan 5 (Tim 1 vs Tim 2): 3 2
Masukkan skor untuk pertandingan 6 (Tim 1 vs Tim 2): 1 0
Masukkan skor untuk pertandingan 7 (Tim 1 vs Tim 2): 5 2
Masukkan skor untuk pertandingan 8 (Tim 1 vs Tim 2): 2 3
Masukkan skor untuk pertandingan 9 (Tim 1 vs Tim 2): -1 2
Hasil pertandingan 1: aku
Hasil pertandingan 2: dia
Hasil pertandingan 3: Seri
Hasil pertandingan 4: dia
Hasil pertandingan 5: aku
Hasil pertandingan 6: aku
Hasil pertandingan 7: aku
Hasil pertandingan 8: dia
Pertandingan selesai
PS D:\Semester 2\ALPRO2 CODING\103112400085_MODUL6> |
```

Deskripsi Program:

Program ini digunakan untuk mencatat hasil pertandingan antara dua tim. Pengguna akan memasukkan nama kedua tim dan skor pertandingan untuk setiap ronde. Program akan terus meminta input skor hingga salah satu skor negatif dimasukkan, menandakan akhir dari pertandingan. Setelah itu, program akan menampilkan hasil setiap pertandingan, apakah tim satu menang, tim dua menang, atau hasil seri.

4. Unguided 4

Source Code:

```
//Anastasia Adinda Narendra Indrianto
package main

import "fmt"

const NMAX int = 127
type tabel [NMAX]rune
func AnastasiaIsiTeks(t *tabel, jumlah *int) {
    var karakter rune
    *jumlah = 0
    fmt.Print("Masukkan teks: ")
    for {
        fmt.Scanf("%c", &karakter)
        if karakter == '.' || *jumlah >= NMAX {
            break
        }
        if karakter != ' ' && karakter != '\n' {
            (*t)[*jumlah] = karakter
            *jumlah++
        }
    }
}
func tampilkanArray(t tabel, jumlah int) {
    fmt.Print("Teks yang dimasukkan: ")
    for i := 0; i < jumlah; i++ {
        fmt.Printf("%c ", t[i])
    }
    fmt.Println()
}
func balikArray(t *tabel, jumlah int) {
    for i := 0; i < jumlah/2; i++ {
        t[i], t[jumlah-1-i] = t[jumlah-1-i], t[i]
    }
}
func cekPalindrom(t tabel, jumlah int) bool {
    var salinan tabel
    copy(salinan[:], t[:])
    balikArray(&salinan, jumlah)
    for i := 0; i < jumlah; i++ {
        if t[i] != salinan[i] {
            return false
        }
    }
    return true
}
```

```

func main() {
    var teks tabel
    var panjang int
    AnastasiaIsiTeks(&teks, &panjang)

    tampilkanArray(teks, panjang)
    balikArray(&teks, panjang)
    tampilkanArray(teks, panjang)

    fmt.Print("Apakah teks palindrom? ")
    if cekPalindrom(teks, panjang) {
        fmt.Println("Ya")
    } else {
        fmt.Println("Tidak")
    }
}

```

Output:

```

PS D:\Semester 2\ALPRO2 CODING\103112400085_MODUL6> go run "d:\Semester 2\ALPRO2 CODING\103112400085_MODUL6\Unguided\4.go"
Masukkan teks: KATAK.
Teks yang dimasukkan: K A T A K
Teks yang dimasukkan: K A T A K
Apakah teks palindrom? Ya
PS D:\Semester 2\ALPRO2 CODING\103112400085_MODUL6> go run "d:\Semester 2\ALPRO2 CODING\103112400085_MODUL6\Unguided\4.go"
Masukkan teks: SENANG.
Teks yang dimasukkan: S E N A N G
Teks yang dimasukkan: G N A N E S
Apakah teks palindrom? Tidak
PS D:\Semester 2\ALPRO2 CODING\103112400085_MODUL6> 

```

Deskripsi Program:

Program ini meminta pengguna untuk memasukkan sebuah teks karakter demi karakter hingga pengguna memasukkan titik (.). Teks yang dimasukkan kemudian disalin ke dalam array dan ditampilkan. Program ini juga akan membalikkan urutan teks dan menampilkan teks yang telah dibalik. Selanjutnya, program memeriksa apakah teks yang dimasukkan merupakan palindrom (apakah teks tersebut sama jika dibaca dari depan maupun belakang). Hasil pemeriksaan palindrom ditampilkan dengan pesan "Ya" jika palindrom, atau "Tidak" jika tidak palindrom.

IV. KESIMPULAN

1. Program Struk Belanja Otomatis

- Program ini dirancang untuk menghasilkan struk belanja otomatis berdasarkan input dari pengguna, seperti informasi toko, tanggal, dan item yang dibeli.
- Program ini menghitung total belanja dan mencetak struk yang terstruktur dengan rincian item dan harga.

2. Program Penghitungan Nilai Mahasiswa

- Program ini digunakan untuk menghitung dan menampilkan nilai mahasiswa, termasuk rata-rata, nilai tertinggi, dan nilai terendah.
- Penggunaan array dua dimensi memungkinkan untuk menampilkan nilai ujian mahasiswa dalam dua mata kuliah.

3. Program Pengecekan Posisi Titik dalam Lingkaran

- Program ini memeriksa apakah sebuah titik berada di dalam satu atau dua lingkaran dengan cara menghitung jarak titik dari pusat lingkaran dan membandingkannya dengan radius.
- Hasilnya menunjukkan apakah titik berada di dalam lingkaran atau di luar lingkaran tersebut.

4. Program Pengolahan Array dan Statistik Data

- Program ini memungkinkan pengguna untuk mengisi array dengan sejumlah elemen, kemudian melakukan berbagai operasi pada array seperti menampilkan elemen berdasarkan indeks tertentu, menghitung rata-rata, dan simpangan baku.
- Program ini memberikan gambaran statistik dasar dari data yang diinput oleh pengguna.

5. Program Hasil Pertandingan

- Program ini mencatat hasil pertandingan antara dua tim, dengan meminta input skor untuk setiap ronde dan kemudian menampilkan hasil apakah tim satu menang, tim dua menang, atau hasil seri.

6. Program Palindrom Teks

- Program ini meminta input teks karakter demi karakter dari pengguna, kemudian memeriksa apakah teks tersebut merupakan palindrom dengan membandingkan teks asli dan teks yang dibalik.

Dari hasil implementasi berbagai program ini, terlihat bahwa pendekatan berbasis fungsi dan struktur data yang tepat sangat membantu dalam menyelesaikan permasalahan secara efisien. Teknik yang digunakan memudahkan pengguna dalam mendapatkan hasil yang diinginkan dengan output yang jelas dan terstruktur. Namun, perlu diperhatikan bahwa pengelolaan input dan pengolahan data secara efektif sangat penting untuk mencegah kesalahan dan memastikan hasil yang akurat.

V. REFRENSI

<https://sko.dev/wiki/input-dan-output>

<https://codingstudio.id/blog/golang-adalah/>

<https://dif.telkomuniversity.ac.id/tipe-data-pemrograman/>

<https://rpubs.com/maulidyarahmah/829044>

https://repository.unikom.ac.id/62967/1/Materi%20Pertemuan%204_Labview%201%2BWhile%20Loop%20%2B%20Shift%20Register.pdf

<https://www.revou.co/kosakata/function>

<https://sko.dev/wiki/fungsi>

<https://drive.google.com/file/d/1tMnxOLAYoMqLB9cKyeJHwyFjmyVtKqMV/view?usp=sharing>

<https://janabadra.ac.id/2023/algoritma-rekursi/#:~:text=Rekursi%20adalah%20sebuah%20metode%20pengulangan,seperti%20definisi%20fungsi%20pada%20umumnya.>

<https://www.tutorialspoint.com/difference-between-array-and-structure>