

**LAPORAN PRAKTIKUM
ALGORITMA PEMROGRAMAN 2**

**MODUL 7
STRUCT & ARRAY**



**Universitas
Telkom**

Oleh:

WILDAN DAFFA' HAKIM PUTRA ANTARA

2311102055

S1IF-11-02

S1 TEKNIK INFORMATIKA

UNIVERSITAS TELKOM

PURWOKERTO

2024

I. DASAR TEORI

1. Tipe Data Bentukan

Tipe data bentukan merupakan tipe data yang dibentuk dari satu atau beberapa elemen bertipe tertentu. Tipe data bentukan sendiri dibagi menjadi 2 yaitu alias dan struct

1.1. Alias (Type)

Tipe ini membuat suatu tipe data baru dengan melakukan *alias* kepada tipe data yang sudah ada (seperti memberikan nama lain untuk tipe data yang sudah ada).

Dalam Pseudocode	Dalam Golang
type <nama alias> <tipe data aslinya>	Sama seperti pada pseudocode

1.2. Struct

Struct digunakan untuk mengumpulkan beberapa tipe data menjadi satu tipe data. Dalam penggunaannya, yang dikumpulkan merupakan nilai / data yang memiliki hubungan atau relasi.

Dalam Pseudocode	Dalam Golang
type <nama alias> < <field1> <tipe data> <field2> <tipe data> <field3> <tipe data> ... >	type <nama struct> struct { <field1> <tipe data> <field2> <tipe data> <field3> <tipe data> }

2. Array

Array merupakan sebuah tipe data yang berisi elemen dengan tipe data sama dan memiliki ukuran yang statis atau tetap.

Dalam Pseudocode	Dalam Golang
------------------	--------------

<code><nama var>: array[0...n] of <type data></code>	<code>var <nama variable> [<ukuran>]<type data></code>
--	--

Dalam golang, selain array statis juga terdapat array dinamis atau disebut slice. Deklarasinya mirip dengan array biasa hanya saja pada bagian ukuran dikosongkan sehingga menjadi

```
var <nama variable> []<type data>

// dengan isi

var contoh = []int{ 2, 4, 5, 5, 7, 9 }
```

Atau kita dapat membuat slice dengan menggunakan make

```
var testMake []int = make([]int, 7)
```

Terdapat juga fungsi len dan cap. Len (len(arr)) berfungsi menghitung elemen yang ada pada array. Sedangkan cap (cap(arr)) digunakan untuk mengetahui total tempat yang disediakan slice.

Untuk menambahkan elemen ke array dapat menggunakan append, fungsi append ini juga akan memperbesar tempat slice jika diperlukan

```
<namaVar> = append(<nama var array>, <data append>)
```

Slice juga dapat melakukan *slicing* pada elemennya

```
namaVar = arr[start:end]
```

Ini akan mengambil semua elemen dimulai index ke-start dan sebelum index ke-end

Yang terakhir, ada Map. Map sendiri merupakan array dinamis yang memiliki key tidak harus integer namun bisa dengan tipe data lain. Key disini sama seperti indeks pada array umumnya.

Berikut contoh inisiasinya :

```

var data map[string]int

// langsung mendeklarasikan isinya

var data_namaLatin = map[string]string{ "padi":"orzya
sativa", "cabe":"capsium .f" }

// memanggil orzya dativa
fmt.Println(data_namaLatin["cabe"] )

//menambahkan data baru

data_namaLatin["tomat"] = "solonauum .l"

//mengubah data / update
data_namaLatin["cabe"] = "capsium frutescens"
// menghapus data dengan kunci
delete(data_namaLatin, "cabe")

```

II. GUIDED

1. Guided 1

```

package main

import (
    "fmt"
    "sort"
)

// Struktur untuk menampung data mahasiswa
type Mahasiswa struct {
    Nama      string
    Matematika int
    Fisika     int
    Kimia      int
    RataRata   float64
}

// Fungsi untuk menghitung rata - rata nilai tiap mahasiswa
func hitungRataRata(m *Mahasiswa) {
    total := m.Matematika + m.Fisika + m.Kimia
    m.RataRata = float64(total) / 3
}

func main() {
    // Array untuk menampung data mahasiswa
    mahasiswa := []Mahasiswa{
        {"Aldi", 85, 90, 80, 0},
        {"Budi", 70, 75, 80, 0},
        {"Cici", 90, 85, 95, 0},
        {"Doni", 60, 65, 70, 0},
    }
}

```

```

        {"Eka", 100, 95, 90, 0},
    }

    // Menghitung rata - rata nilai tiap mahasiswa
    for i := range mahasiswa {
        hitungRataRata(&mahasiswa[i])
    }

    // Mengurutkan mahasiswa berdasarkan nilai rata - rata
    (descending)
    sort.Slice(mahasiswa, func(i, j int) bool {
        return      mahasiswa[i].RataRata      >
mahasiswa[j].RataRata
    })

    // Menampilkan hasil
    fmt.Println("Peringkat mahasiswa berdasarkan rata -
rata nilai: ")
    for i, m := range mahasiswa {
        fmt.Printf("%d.    %s    -    Rata-rata:    %.2f
(Matematika: %d, Fisika: %d, Kimia: %d)\n", i+1, m>Nama,
m.RataRata, m.Matematika, m.Fisika, m.Kimia)
    }
}

```

```

λ MacBook-Air-Daffa Modul 7 → go run "/Users/daffahakim/Documents/Kuliah/SMT
Peringkat mahasiswa berdasarkan rata - rata nilai:
1. Eka - Rata-rata: 95.00 (Matematika: 100, Fisika: 95, Kimia: 90)
2. Cici - Rata-rata: 90.00 (Matematika: 90, Fisika: 85, Kimia: 95)
3. Aldi - Rata-rata: 85.00 (Matematika: 85, Fisika: 90, Kimia: 80)
4. Budi - Rata-rata: 75.00 (Matematika: 70, Fisika: 75, Kimia: 80)
5. Doni - Rata-rata: 65.00 (Matematika: 60, Fisika: 65, Kimia: 70)

```

Program ini digunakan untuk menghitung nilai rata – rata dari mata kuliah Matematika, Fisika, dan kimia dari setiap mahasiswa (nilai sudah disediakan di code). Lalu, setelahnya akan diurutkan berdasarkan rata ratanya. Program ini memiliki tipe data bentukan struct dengan nama Mahasiswa yang menampung nama dengan tipe data string, Matematika, Fisika, dan Kimia dengan tipe data integer, dan rata rata dengan tipe data Float.

Tipe data Mahasiswa tadi digunakan pada variable mahasiswa yang merupakan slice dengan isi struct Tipe Mahasiswa. Pada awal inisiasi rata rata masih 0 karena belum dihitung, setelah inisiasi variabel dilakukan perulangan sebanyak variabel dengan for-range sehingga perulangan nya hanya sebanyak isi mahasiswa. Setiap perulangan akan memanggil prosedur hitungRataRata yang memiliki parameter reference m dengan tipe data Mahasiswa. Ini digunakan untuk merubah nilai mahasiswa.RataRata setelah nilai rata – rata dihitung.

Setelah selesai mengkalkulasi rata – rata, mahasiswa diurutkan berdasarkan rata – ratanya menggunakan sort.Slice disini akan dicek apakah array index sebelumnya lebih besar dari array index setelahnya. Setelah diurutkan ditampilkan lagi array dari tipe mahasiswa dengan for-range

2. Unguided 2

```
package main

import "fmt"

func main() {
    // Membuat map dengan NIM sebagai kunci dan Nama
    // sebagai nilai
    mahasiswa := map[string]string{
        "20231001": "Andi",
        "20231002": "Budi",
        "20231003": "Cici",
    }

    // Menambahkan data baru ke map
    mahasiswa["20231004"] = "Dedi"

    // Menampilkan seluruh isi map dalam format kolom
    // dan baris
    fmt.Println("Daftar Mahasiswa:")
    fmt.Println("NIM\t\tNama")
    fmt.Println("-----")
    for nim, nama := range mahasiswa {
        fmt.Printf("%s\t%s\n", nim, nama)
    }

    // Mengakses data berdasarkan NIM
    nim := "20231002"
    fmt.Println("\nNama Mahasiswa dengan NIM", nim,
        "adalah", mahasiswa[nim])

    // Menghapus data berdasarkan NIM
    delete(mahasiswa, "20231003")

    // Menampilkan isi map setelah data dihapus dalam
    // format kolom dan baris
    fmt.Println("\nDaftar      Mahasiswa      setelah
    dihapus:")
    fmt.Println("NIM\t\tNama")
```

```

    fmt.Println("-----")
    for nim, nama := range mahasiswa {
        fmt.Printf("%s\t%s\n", nim, nama)
    }
}

```

```

λ MacBook-Air-Daffa Modul 7 → go run "/Users/daffahakim/Docume
Daftar Mahasiswa:
NIM          Nama
-----
20231001     Andi
20231002     Budi
20231003     Cici
20231004     Dedi

Nama Mahasiswa dengan NIM 20231002 adalah Budi

Daftar Mahasiswa setelah dihapus:
NIM          Nama
-----
20231001     Andi
20231002     Budi
20231004     Dedi
λ MacBook-Air-Daffa Modul 7 → █

```

Pada program ini ditampilkan list mahasiswa yang disimpan menggunakan tipe data bentukan map. Setelah itu dilakukan pemanggilan nama budi dengan key nimnya yaitu 20231002. Lalu dilakukan penghapusan dengan fungsi delete Dimana key yang di delete 20231003. Variabel yang menggunakan tipe data bentukan map adalah mahasiswa dimana keynya adalah string dan valuenya adalah string.

III. UNGUIDED

1. Suatu lingkaran didefinisikan dengan koordinat titik pusat (cx, cy) dengan radius r . Apabila diberikan dua buah lingkaran, maka tentukan posisi sebuah titik sembarang (x, y) berdasarkan dua lingkaran tersebut. Gunakan tipe bentukan titik untuk menyimpan koordinat, dan tipe bentukan lingkaran untuk menyimpan titik pusat lingkaran dan radiusnya. **Selengkapnya ada pada modul.**

```

package main

import (
    "fmt"

```

```

    "math"
)

type Titik struct {
    x, y int
}

type Lingkaran struct {
    radius float64
    titikPusat Titik
}

func getJarak(lingkaran Lingkaran, titikSembarang Titik)
float64 {
    return
    math.Sqrt(math.Pow(float64(lingkaran.titikPusat.x)-
float64(titikSembarang.x), 2) +
    math.Pow(float64(lingkaran.titikPusat.y)-
float64(titikSembarang.y), 2))
}

func cekDidalam(lingkaran Lingkaran, jarak float64) bool {
    return lingkaran.radius >= jarak
}

func main() {
    var lingkaran1, lingkaran2 Lingkaran
    var titikSembarang Titik
    var didalam1, didalam2 bool
    fmt.Scanln(&lingkaran1.titikPusat.x,
&lingkaran1.titikPusat.y, &lingkaran1.radius)
    fmt.Scanln(&lingkaran2.titikPusat.x,
&lingkaran2.titikPusat.y, &lingkaran2.radius)
    fmt.Scanln(&titikSembarang.x, &titikSembarang.y)
    didalam1 = cekDidalam(lingkaran1, getJarak(lingkaran1,
titikSembarang))
    didalam2 = cekDidalam(lingkaran2, getJarak(lingkaran2,
titikSembarang))
    switch {
    case didalam2 && didalam1:
        fmt.Print("Titik di dalam lingkaran 1 dan 2")
    case didalam1:
        fmt.Print("Titik di dalam lingkaran 1")
    case didalam2:
        fmt.Print("Titik di dalam lingkaran 2")
    default:
        fmt.Print("Titik di luar lingkaran 1 dan 2")
    }
}

```



```

20251004 DCD1
λ MacBook-Air-Daffa Modul 7 → go run "/Users/daffahakim/Documents/
0"
1 2 3
4 5 6
7 8
Titik di dalam lingkaran 2%
λ MacBook-Air-Daffa Modul 7 →

```

Program ini dilakukan untuk mengetahui apakah sebuah titik di koordinat sembarang berada dalam suatu lingkaran (atau keduanya). Program ini menggunakan tipe data bentukan Titik yang berisi x dan y yang bertipe data integer. Serta tipe bentukan Lingkaran yang berisi radius dengan tipe data float, dan titikPusat yang bertipe data Titik.

Pada fungsi main dideklarasikan variable lingkaran1 dan lingkaran2 dengan tipe data Lingkaran dan dideklarasikan variable titikSembarang dengan tipe data Titik. Selain itu, dideklarasikan variabel didalam1 dan didalam2 dengan tipe data Boolean untuk menyimpan penanda jika titik didalam lingkaran. Setelah itu, imputan user diminta untuk mengisi titik x, y, dan radius pada dua lingkaran dan setelahnya mengisi titik sembarang. Lalu dipanggil fungsi cekDidalam yang argumennya diisi dengan lingkaran dan hasil dari fungsi getJarak yang akan mengembalikan nilai dari rumus jarak. Fungsi cekDidalam akan mengembalikan nilai true jika jarak kurang atau sama dengan radius.

Lalu yang terakhir digunakan switch statement, untuk menampilkan output titik didalam atau diluar lingkaran. Titik hanya didalam suatu lingkaran atau kedua lingkaran jika nilai variable didalam1 atau didalam2 bernilai true.

2. Sebuah array digunakan untuk menampung sekumpulan bilangan bulat. Buatlah program yang digunakan untuk mengisi array tersebut sebanyak N elemen nilai. Asumsikan array memiliki kapasitas penyimpanan data sejumlah elemen tertentu. **Selengkapanya ada pada modul.**

```

package main

import (
    "fmt"
    "math"
)

```

```

func showMenu() {
    fmt.Println("1. menampilkan seluruh isi array")
    fmt.Println("2. Menampilkan elemen-elemen array dengan
indeks ganjil saja.")
    fmt.Println("3. Menampilkan elemen-elemen array dengan
indeks genap saja (asumsi indek ke-0 adalah genap).")
    fmt.Println("4. Menampilkan elemen-elemen array dengan
indeks kelipatan bilangan x")
    fmt.Println("5. Menghapus elemen array pada indeks
tertentu.")
    fmt.Println("6. Menampilkan rata-rata dari bilangan
yang ada di dalam array.")
    fmt.Println("7. Menampilkan standar deviasi atau
simpangan baku")
    fmt.Println("8. Menampilkan frekuensi dari suatu
bilangan")
    fmt.Println("9. Exit")
    fmt.Print("Pilih menu : ")
}

func showData(arr []int, skip int, expect int, skipZero
bool) {
    for index, data := range arr {
        if skipZero && index == 0 {
            continue
        }
        if index%skip == expect {
            fmt.Println("Index ke - ", index, " : ",
data)
        }
    }
}

func reRata(arr []int) float64 {
    var sum int
    for _, data := range arr {
        sum += data
    }
    return float64(sum) / float64(len(arr))
}

func simpBaku(arr []int) float64 {
    var sum float64
    for _, data := range arr {
        sum += math.Pow(float64(data)-reRata(arr), 2)
    }
    return math.Sqrt(float64(sum) / float64(len(arr)))
}

func frek(arr []int, bil int) int {
    var count int
    for _, data := range arr {
        if data == bil {

```

```

        count++
    }
}
return count
}

func removeIndex(index int, arr *[]int) {
    if len(*arr)-1 < index {
        fmt.Println("index out of range")
    } else {
        fmt.Println("item", (*arr)[index], "pada index", index, "telah dihapus\nisi Sekarang : ")
        *arr = append((*arr)[:index], (*arr)[index+1:]...)
        showData(*arr, 1, 0, false)
    }
}

func main() {
    var arraySize, input int
    fmt.Print("Masukkan Kapasitas array : ")
    fmt.Scan(&arraySize)
    var newArr []int = make([]int, arraySize)
    for index, _ := range newArr {
        fmt.Print("Masukkan index ke -", index, " : ")
        fmt.Scan(&newArr[index])
    }
    for input != 9 {
        showMenu()
        fmt.Scan(&input)
        switch input {
            case 1:
                showData(newArr, 1, 0, false)
            case 2:
                showData(newArr, 2, 1, false)
            case 3:
                showData(newArr, 2, 0, false)
            case 4:
                fmt.Print("Masukkan Kelipatan : ")
                fmt.Scan(&input)
                showData(newArr, input, 0, true)
            case 5:
                fmt.Print("Masukkan index yang dihapus : ")
                fmt.Scan(&input)
                removeIndex(input, &newArr)
            case 6:
                fmt.Println("Rerata dari isi array adalah :", reRata(newArr))
            case 7:
                fmt.Println("Deviasi dari isi array adalah :", simpBaku(newArr))
            case 8:

```

```

        fmt.Print("Masukkan bilangan yang akan
dicek : ")

        fmt.Scan(&input)
        fmt.Println("Frekuensi dari", input,
"pada array adalah :", frek(newArr, input))
        default:
            fmt.Println("Opsi salah")
        }
    }
}

```

```

λ MacBook-Air-Daffa Modul 7 → go run "/Users/daffahakim/Documents/Kuliah/SMT 3/Alpro2/Modul 7/ur
o"
Masukkan Kapasitas array : 4
Masukkan index ke -0 : 2
Masukkan index ke -1 : 4
Masukkan index ke -2 : 5
Masukkan index ke -3 : 6
1. menampilkan seluruh isi array
2. Menampilkan elemen-elemen array dengan indeks ganjil saja.
3. Menampilkan elemen-elemen array dengan indeks genap saja (asumsi index ke-0 adalah genap).
4. Menampilkan elemen-elemen array dengan indeks kelipatan bilangan x
5. Menghapus elemen array pada indeks tertentu.
6. Menampilkan rata-rata dari bilangan yang ada di dalam array.
7. Menampilkan standar deviasi atau simpangan baku
8. Menampilkan frekuensi dari suatu bilangan
9. Exit
Pilih menu : 1

```

1. Menampilkan keseluruhan isi dari array.

```

Pilih menu : 1
Index ke - 0 : 2
Index ke - 1 : 4
Index ke - 2 : 5
Index ke - 3 : 6

```

2. Menampilkan elemen-elemen array dengan indeks ganjil saja.

```

Pilih menu : 2
Index ke - 1 : 4
Index ke - 3 : 6

```

3. Menampilkan elemen-elemen array dengan indeks genap saja (asumsi index ke-0 adalah genap).

```

Pilih menu : 3
Index ke - 0 : 2
Index ke - 2 : 5

```

4. Menampilkan elemen-elemen array dengan indeks kelipatan bilangan x. x bisa diperoleh dari masukan pengguna.

```

Pilih menu : 4
Masukkan Kelipatan : 3
Index ke - 3 : 6
1. menampilkan seluruh isi array

```

5. Menghapus elemen array pada indeks tertentu, asumsi indeks yang hapus selalu valid. Tampilkan keseluruhan isi dari arraynya, pastikan data yang dihapus tidak tampil.

```
Pilih menu : 5
Masukkan index yang dihapus : 3
item 6 pada index 3 telah dihapus
isi Sekarang :
Index ke - 0 : 2
Index ke - 1 : 4
Index ke - 2 : 5
```

6. Menampilkan rata-rata dari bilangan yang ada di dalam array.

```
Pilih menu : 6
Rerata dari isi array adalah : 3.6666666666666665
1. menampilkan seluruh isi array
```

7. Menampilkan standar deviasi atau simpangan baku dari bilangan yang ada di dalam array tersebut.

```
Pilih menu : 7
Deviasi dari isi array adalah : 1.247219128924647
```

8. Menampilkan frekuensi dari suatu bilangan tertentu di dalam array yang telah diisi tersebut.

```
Pilih menu : 8
Masukkan bilangan yang akan dicek : 6
Frekuensi dari 6 pada array adalah : 0
```

Program ini memiliki beberapa fungsi yang digunakan untuk mengoperasikan elemen bilangan (integer) yang ada didalam array. Pada awal program, dideklarasikan arraySize dan input bertipe integer. arraySize digunakan menyimpan size array, input untuk menampung inputan user berupa menu atau parameter faktual tambahan yang diperlukan setiap menu. Setelah inputan arraySize diterima maka dibuat slice baru dengan nama newArr. Setelah itu user diminta memasukkan elemen ke index dengan for-range. Setelah itu dilakukan perulangan for yang akan menampilkan menu.

Pada case ke 1,2,3 dipanggil fungsi showData yang menerima parameter array, skip yang digunakan untuk value modulo, expect yang digunakan sebagai pembanding modulo, dan skipZero untuk melewati indeks nol mengingat bilangan 0 dimodulo apapun menghasilkan nol sehingga perlu ditambahkan ini agar tidak selalu muncul. Setelah itu output ditampilkan dengan perulangan for range yang dilakukan pengecekan apakah nilai index modulo nilai skip sama dengan nilai expect. Jika iya maka ditampilkan index dan

nilainya. Selain itu juga dilakukan pengecekan apakah skipZero bernilai true dan posisi index di 0 jika iya iterasi saat itu dilewati. Pada case ke 4 sama seperti case 1,2,3 hanya saja nilai skip ditentukan pengguna (mewakili kelipatan)

Pada case ke 5 memanggil fungsi removeIndex (parameter reference) dengan parameter array dan bilangan index dan diawal fungsi akan dilakukan pengecekan apakah index yang akan dihapus melebihi size jika iya maka akan ditampilkan bahwa index melebihi batas. Jika tidak akan dilakukan penghapusan memanfaatkan slicing pada slice.

Pada case ke 6 memanggil fungsi rerata yang akan melakukan penambahan semua elemen array memanfaatkan for-range dan membagi dengan banyaknya isi array (memanfaatkannya len()). Pada case ke 7 cara kerjanya hampir mirip case 6, Dimana ini memanfaatkan case 6 untuk mengambil rata rata dan menghitung standar deviasi sesuai rumus. Setiap sum yang dilakukan selama perulangan for-range disimpan pada variable sum.

Pada case 8, akan dipanggil fungsi frek dengan parameter arr (sebagai array yang mau dicek), dan bil (sebagai bilangan yang akan dicek frekuensi kemunculannya). Akan dilakukan perulangan for-range sepanjang array dan setiap perulangan akan dilakukan perbandingan apakah elemen di setiap iterasi nilainya sama dengan bilangan. Jika iya nilai variable count akan ditambah 1.

3. Sebuah program digunakan untuk menyimpan dan menampilkan nama-nama klub yang memenangkan pertandingan bola pada suatu grup pertandingan. Buatlah program yang digunakan untuk merekap skor pertandingan bola 2 buah klub bola yang berlaga.

Pertama-tama program meminta masukan nama-nama klub yang bertanding, kemudian program meminta masukan skor hasil pertandingan kedua klub tersebut. Yang disimpan dalam array adalah nama-nama klub yang menang saja.

Proses input skor berhenti ketika skor salah satu atau kedua klub tidak valid (negatif).

```

package main

import "fmt"

type Pertandingan struct {
    klubA      string
    klubB      string
    hasilPertandingan []string
}

func hitung(dataClub Pertandingan, skor1, skor2 int) string {
    if skor1 > skor2 {
        return dataClub.klubA
    } else if skor1 == skor2 {
        return "Draw"
    } else {
        return dataClub.klubB
    }
}

func showHasil(dataClub Pertandingan) {
    for i, data := range dataClub.hasilPertandingan {
        fmt.Printf("Hasil %d : %s\n", i+1, data)
    }
    fmt.Println("Pertandingan selesai")
}

func main() {
    var skor1, skor2, count int
    var dataTanding Pertandingan
    fmt.Print("Klub A : ")
    fmt.Scan(&dataTanding.klubA)
    fmt.Print("Klub B : ")
    fmt.Scan(&dataTanding.klubB)
    for {
        fmt.Printf("Pertandingan %d : ", count+1)
        fmt.Scan(&skor1, &skor2)
        if skor1 >= 0 && skor2 >= 0 {
            dataTanding.hasilPertandingan =
append(dataTanding.hasilPertandingan, hitung(dataTanding,
skor1, skor2))
            count++
        } else {
            break
        }
    }
    showHasil(dataTanding)
}

```

```

λ MacBook-Air-Daffa Modul 7 → go run "/Users/daffahakim/Documents/Ku
o"
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
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

```

Program ini bertujuan untuk mencatat hasil dari beberapa pertandingan antara dua klub sepak bola. Program menerima skor dari setiap pertandingan dan menentukan pemenangnya berdasarkan skor tersebut. Program juga menampilkan daftar hasil dari semua pertandingan setelah selesai.

Pada fungsi main dideklarasikan variable skor1 digunakan untuk menampung skor klubA sementara, skor2 digunakan untuk menampung skor klubB sementara, dan count untuk menghitung pertandingan seberapa banyak sekarang, selain itu terdapat variabel dataTanding dengan tipe data Pertandingan untuk menyimpan klub yang bertanding dan hasil pertandingannya. Setelah inisiasi program meminta inputan user untuk nama klub yang bertanding. Lalu, dilakukan perulangan Dimana setiap perulangan dilakukan pencatatan skor dan hasilnya akan langsung dimasukkan ke variable dataTanding.hasilPertandingan menggunakan append disini juga memanggil fungsi hitung dengan parameter dataClub, skor1, dan skor2 untuk mengetahui klub mana yang menang atau apakah draw. Jika nilai skor1 ataupun dua salah satunya negatif maka perulangan berhenti. Dan hasil ditampilkan dengan fungsi showHasil yang memiliki parameter dataClub yang nantinya akan melakukan perulangan for-range pada array yang berada di dataClub.hasilPertandingan sehingga hasilnya dapat tampil.

4. Sebuah array digunakan untuk menampung sekumpulan karakter, Anda diminta untuk membuat sebuah sub program untuk melakukan membalikkan urutan isi array dan memeriksa apakah membentuk palindrom.


```

package main

import "fmt"

const NMAX int = 127

type tabel [NMAX]rune

func isiArray(t *tabel, n *int) {
    var temp_inp rune
    for *n < NMAX {
        fmt.Scanf("%c", &temp_inp)
        if temp_inp == '.' {
            break
        } else if temp_inp == ' ' {
            continue
        }
        t[*n] = temp_inp
        *n++
    }
}

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

func balikanArray(t *tabel, n int) {
    var dat tabel

    for i := n - 1; i >= 0; i-- {
        dat[(n-1)-i] = t[i]
    }
    copy((*t)[:], dat[:])
}

func main() {
    var tab tabel
    var m int
    fmt.Print("Teks\t\t: ")
    isiArray(&tab, &m)
    balikanArray(&tab, m)
    fmt.Print("Reverse Teks\t: ")
    cetakArray(tab, m)
}

```

```

λ MacBook-Air-Daffa Modul 7 → go run "/Users/daffahakim/Documents/Kuliah
0"
Teks      : b w a s .
Reverse Teks : s a w b
λ MacBook-Air-Daffa Modul 7 →

```

Program ini menangkap setiap karakter yang kita inputkan. Setelah itu, dilakukan reverse (misalkan : tahu menjadi uhat). Program ini menggunakan tipe bentukan Bernama tabel yang merupakan sebuah array sepanjang 127 dengan tipe data rune (int32) untuk menyimpan character dalam bentuk bytes.

Terdapat prosedur isi_array yang memiliki parameter reference t yang bertipe table dan n. yang berupa int, fungsi ini akan memasukkan setiap character ke array selama masih kurang dari 127, memanfaatkan perulangan for dengan kondisi $n < NMAX$.

Terdapat juga prosedur cetak array yang akan menampilkan output per huruf secara horizontal dan memberikan newline setelah pencetakan selesai. Lalu terdapat juga fungsi balikkan array yang memiliki parameter reference t dengan tipe table dan parameter biasa n bertipe integer. Pada prosedur ini akan dilakukan pembalikan dengan loop, Dimana setiap loop nilai indeks kebalikan akan disimpan di array sementara dat lalu dipanggil isi copy untuk mengcopy semua slice dat ke t.

Pada fungsi main, dideklarasikan 2 variabel yaitu tab dengan tipe table dan m yang bertipe integer. Setelah itu dioutputkan "teks :." dan pemanggilan fungsi isi array dengan parameter factual berupa reference dari tab dan reference dari m. setelah pemanggilan selesai akan didapat banyaknya characternya (m) dan characternya (tab). Lalu dilakukan cetak pembalikan array dengan fungsi balikanArray dengan parameter aktual reference dari tab dan parameter factual berupa nilai m. lalu, dilakukan pencetakan array dengan cetakArray dengan parameter faktual nilai dari tab dan m diiringi kata "Reverse Teks: "

Modifikasi program tersebut dengan menambahkan fungsi palindrom. Tambahkan instruksi untuk memanggil fungsi tersebut dan menampilkan hasilnya pada program utama.

```
package main
```

```

import "fmt"

const NMAX int = 127

type tabel [NMAX]rune

func isiArray(t *tabel, n *int) {
    var temp_inp rune
    for *n < NMAX {
        fmt.Scanf("%c", &temp_inp)
        if temp_inp == '.' {
            break
        } else if temp_inp == ' ' {
            continue
        }
        t[*n] = temp_inp
        *n++
    }
}

func checkPolindrom(t tabel, n int) bool {
    for i := 0; i < n; i++ {
        if t[i] != t[(n-1)-i] {
            return false
        }
    }
    return true
}

func main() {
    var tab tabel
    var m int
    fmt.Print("Teks\t\t: ")
    isiArray(&tab, &m)
    fmt.Print("Palindrom\t?: ", checkPolindrom(tab, m))
}

```

```

λ MacBook-Air-Daffa Modul 7 → go run "/Users/daffahakim/Documents/Kuliah/SMT 3,
od.go"
Teks          : h e h e .
Palindrom     ? false%
λ MacBook-Air-Daffa Modul 7 →
λ MacBook-Air-Daffa Modul 7 → go run "/Users/daffahakim/Documents/Kuliah/SMT 3,
od.go"
Teks          : K A T A K .
Palindrom     ? true%

```

Pada modifikasi program ini inputan user dan variabel yang dideklarasikan tetap sama hanya saja prosedur `balikanArray` dan `cetakArray` diganti dengan `checkPolindrom`. Dimana menerima parameter formal `t` dengan tipe `tabel` dan `n` dengan tipe integer. Fungsi

ini akan mengecek dengan loop apakah nilai indeks ke-i sama dengan nilai indeks ke-(banyaknyaIsi - 1 - i) jika ditemukan satu perbedaan saja akan langsung dikembalikan false dan jika sampai akhir tidak ditemukan perbedaan akan dikembalikan true.

Pada fungsi utama, fungsi checkPolindrom akan dipanggil setelah isiArray Bersama kata “Polindrom ?” untuk menampilkan apakah hasilnya true atau false.