LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2

MODUL 11

NILAI EKSTRIM

Disusun Oleh :

Annasya Maulafidatu Zahra / 2311102246

S1-IF-11-05

Dosen Pengampu :

Arif Amrulloh, S.Kom., M.Kom.

PROGRAM STUDI S1 TEKNIK INFORMATIKA

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2024

I. DASAR TEORI

Mencari nilai ekstrim, yaitu nilai maksimum dan minimum dari sekumpulan data, adalah salah satu operasi dasar dalam pemrograman yang sering digunakan dalam berbagai aplikasi. Dalam bahasa Go, kita dapat menggunakan struktur data seperti slice untuk menyimpan kumpulan angka. Slice adalah koleksi dinamis yang memungkinkan kita untuk menyimpan elemen dengan tipe data yang sama dan mengelola data dengan lebih fleksibel dibandingkan dengan array statis. Dengan memanfaatkan slice, kita dapat dengan mudah mengakses dan memanipulasi data yang diperlukan untuk mencari nilai ekstrim

Proses pencarian nilai maksimum dan minimum dalam sebuah slice dapat dilakukan dengan cara iteratif. Langkah pertama adalah menginisialisasi dua variabel untuk menyimpan nilai maksimum dan minimum, biasanya dengan menggunakan elemen pertama dari slice. Kemudian, kita melakukan iterasi melalui semua elemen dalam slice tersebut, membandingkan setiap elemen dengan nilai maksimum dan minimum yang sudah ada. Jika ditemukan elemen yang lebih besar dari nilai maksimum saat ini, maka kita memperbarui nilai maksimum. Sebaliknya, jika ditemukan elemen yang lebih kecil dari nilai minimum saat ini, maka kita memperbarui nilai minimum. Proses ini memastikan bahwa pada akhir iterasi, kita akan mendapatkan nilai ekstrim yang diinginkan.

Implementasi fungsi pencarian nilai ekstrim dalam Go sangat sederhana dan efisien. Dengan menggunakan loop `for`, kita dapat menjelajahi seluruh elemen dalam slice hanya dalam satu kali perjalanan (single pass), sehingga kompleksitas waktu dari algoritma ini adalah O(n), di mana n adalah jumlah elemen dalam slice. Pendekatan ini tidak hanya efektif tetapi juga mudah dipahami, menjadikannya pilihan yang baik untuk pengembang yang ingin melakukan analisis data dasar. Dengan demikian, kemampuan untuk menemukan nilai ekstrim menjadi alat penting dalam pengolahan data dan analisis statistik di berbagai bidang aplikasi.

I. UNGUIDED

1. Unguided 1

Soal Studi Case

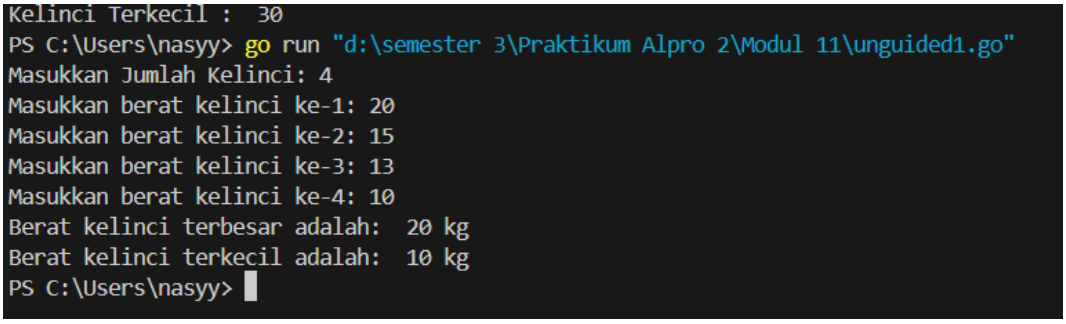
Sebuah program digunakan untuk mendata berat anak kelinci yang akan dijual ke pasar. Program ini menggunakan array dengan kapasitas 1000 untuk menampung data berat anak kelinci yang akan dijual.
Masukan terdiri dari sekumpulan bilangan, yang mana bilangan pertama adalah bilangan bulat
N yang menyatakan banyaknya anak kelinci yang akan ditimbang beratnya. Selanjutnya N bilangan riil berikutnya adalah berat dari anak kelinci yang akan dijual.Keluaran terdiri dari dua buah bilangan riil yang menyatakan berat kelinci terkecil dan terbesar

Sourcecode

```go
package main

import "fmt"

func main() {
    var jumlahKelinci int
    var berat [1000]float64
    var beratTerkecil, beratTerbesar float64

    fmt.Print("Masukkan Jumlah Kelinci: ")
    fmt.Scan(&jumlahKelinci)

    for i := 0; i < jumlahKelinci; i++ {
        fmt.Printf("Masukkan berat kelinci ke-%d: ", i+1)
        fmt.Scan(&berat[i])
    }

    beratTerkecil = berat[0]
    beratTerbesar = berat[0]
    for i := 1; i < jumlahKelinci; i++ {
        if beratTerbesar < berat[i] {
            beratTerbesar = berat[i]
        }

        if beratTerkecil > berat[i] {
            beratTerkecil = berat[i]
        }
    }
```

```
    fmt.Println("Berat kelinci terbesar adalah: ",
beratTerbesar, "kg")
    fmt.Println("Berat kelinci terkecil adalah: ",
beratTerkecil, "kg")
}
```

Screenshoot Output



Deskripsi Program

Program ini mencari berat kelinci terbesar dan terkecil dari sekumpulan data. Pertama, program meminta pengguna untuk memasukkan jumlah kelinci. Kemudian, pengguna diminta memasukkan berat masing-masing kelinci. Selanjutnya, program akan membandingkan setiap berat kelinci dengan berat terkecil dan terbesar yang sudah ditemukan sebelumnya. Setiap kali ditemukan berat yang lebih besar atau lebih kecil, nilai terkecil atau terbesar akan diperbarui. Akhirnya, program akan mencetak hasil perhitungan, yaitu berat kelinci terbesar dan terkecil. Sederhananya, program ini bekerja seperti seorang guru yang mencari nilai tertinggi dan terendah dari nilai ujian siswa.

2. Unguided 2

Soal Studi Case

Sebuah program digunakan untuk menentukan tarif ikan yang akan dijual ke pasar. Program ini menggunakan array dengan kapasitas 1000 untuk menampung data berat ikan yang akan dijual.

Masukan terdiri dari dua baris, yang mana baris pertama terdiri dari dua bilangan bulat x dan y. Bilangan x menyatakan banyaknya ikan yang akan dijual, sedangkan y adalah banyaknya ikan yang akan dimasukan ke dalam wadah. Baris kedua terdiri dari sejumlah x bilangan riil yang menyatakan banyaknya ikan yang akan dijual.

Keluaran terdiri dari dua baris. Baris pertama adalah kumpulan bilangan riil yang menyatakan total berat ikan di setiap wadah (jumlah wadah

tergantung pada nilai x dan y, urutan ikan yang dimasukan ke dalam wadah sesuai urutan pada masukan baris ke-2). Baris kedua adalah sebuah bilangan riil yang menyatakan berat rata-rata ikan di setiap wadah.

Sourcecode

```
package main

import (
    "fmt"
)

func validasi(jumlahIkan, kapasitasWadah int) bool {
    return jumlahIkan > 0 && kapasitasWadah > 0
}

func hitungTarif(jumlahIkan, kapasitasWadah int, beratIkan
[]float64) ([]float64, float64) {
    var totalBeratPerWadah []float64
    var totalKeseluruhan float64

    for i := 0; i < jumlahIkan; i += kapasitasWadah {
        end := i + kapasitasWadah
        if end > jumlahIkan {
            end = jumlahIkan
        }

        var totalBerat float64
        for j := i; j < end; j++ {
            totalBerat += beratIkan[j]
        }
        totalBeratPerWadah = append(totalBeratPerWadah,
totalBerat)
        totalKeseluruhan += totalBerat
    }

    rataRataPerWadah := totalKeseluruhan /
float64(len(totalBeratPerWadah))
    return totalBeratPerWadah, rataRataPerWadah
}

func main() {
    var jumlahIkan, kapasitasWadah int
```

```
    fmt.Print("Masukkan jumlah ikan yang akan dijual: ")
    fmt.Scan(&jumlahIkan)
    fmt.Print("Masukkan kapasitas ikan per wadah: ")
    fmt.Scan(&kapasitasWadah)

    if !validasi(jumlahIkan, kapasitasWadah) {
        fmt.Println("Jumlah ikan dan kapasitas wadah harus
lebih dari 0.")
        return
    }

    beratIkan := make([]float64, jumlahIkan)
    fmt.Println("Masukkan berat masing-masing ikan (kg):")
    for i := 0; i < jumlahIkan; i++ {
        fmt.Scan(&beratIkan[i])
    }

    totalBeratPerWadah, rataRataPerWadah :=
hitungTarif(jumlahIkan, kapasitasWadah, beratIkan)

    fmt.Println("\nTotal berat ikan di setiap wadah (kg):")
    for i, totalBerat := range totalBeratPerWadah {
        fmt.Printf("Wadah ke-%d: %.2f kg\n", i+1,
totalBerat)
    }
    fmt.Printf("\nRata-rata berat ikan per wadah: %.2f
kg\n", rataRataPerWadah)
}
```

Screenshoot Output

```
PS C:\Users\nasyy> go run "d:\semester 3\Praktikum Alpro 2\Modul 11\unguided2.go"
Masukkan jumlah ikan yang akan dijual: 3
Masukkan kapasitas ikan per wadah: 2
Masukkan berat masing-masing ikan (kg):
3
2
3

Total berat ikan di setiap wadah (kg):
Wadah ke-1: 5.00 kg
Wadah ke-2: 3.00 kg

Rata-rata berat ikan per wadah: 4.00 kg
PS C:\Users\nasyy>
```

Deskripsi Program

Program ini menghitung total berat ikan di setiap wadah dan rata-rata berat ikan per wadah berdasarkan jumlah ikan dan kapasitas wadah yang diberikan. Pengguna memasukkan jumlah ikan, kapasitas wadah, dan berat masing-masing ikan. Program kemudian menghitung total berat ikan untuk setiap wadah secara berurutan dan menampilkan hasilnya dan rata-rata berat ikan di semua wadah.

3. Unguided 3

Soal Studi Case

Pos Pelayanan Terpadu (posyandu) sebagai tempat pelayanan kesehatan perlu mencatat data berat balita (dalam kg). Petugas akan memasukkan data tersebut ke dalam array. Dari data yang diperoleh akan dicari berat balita terkecil, terbesar, dan reratanya. Buatlah program dengan spesifikasi subprogram sebagai berikut:

```
type arrBalita [100]float64

func hitungMinMax(arrBerat arrBalita; bMin, bMax *float64) {
/* I.S. Terdefinisi array dinamis arrBerat
  Proses: Menghitung berat minimum dan maksimum dalam array
  F.S. Menampilkan berat minimum dan maksimum balita */
    ...
}

function rerata (arrBerat arrBalita) real {
/* menghitung dan mengembalikan rerata berat balita dalam array */
    ...
}
```

Perhatikan sesi interaksi pada contoh berikut ini (**teks bergaris bawah adalah input/read**)

```
Masukan banyak data berat balita : 4
Masukan berat balita ke-1: 5.3
Masukan berat balita ke-2: 6.2
Masukan berat balita ke-3: 4.1
Masukan berat balita ke-4: 9.9
Berat balita minimum: 4.10 kg
Berat balita maksimum: 9.90 kg
Rerata berat balita: 6.38 kg
```

Sourcecode

```go
package main

import "fmt"

type arrBeratAnak [100]float64

func cariMinMax(beratAnak arrBeratAnak, beratTerkecil,
beratTerbesar *float64) {
    *beratTerkecil = beratAnak[0]
    *beratTerbesar = beratAnak[0]

    for i := 1; i < len(beratAnak); i++ {
        if beratAnak[i] != 0 {
            if beratAnak[i] < *beratTerkecil {
                *beratTerkecil = beratAnak[i]
            }
            if beratAnak[i] > *beratTerbesar {
                *beratTerbesar = beratAnak[i]
            }
        }
    }
}

func hitungRataRata(beratAnak arrBeratAnak, jumlahAnak int)
float64 {
    var totalBerat float64
    for i := 0; i < jumlahAnak; i++ {
        totalBerat += beratAnak[i]
    }
    return totalBerat / float64(jumlahAnak)
}

func main() {
    var beratAnak arrBeratAnak
    var jumlahAnak int
    var beratTerkecil, beratTerbesar float64

    fmt.Print("Masukkan jumlah anak : ")
    fmt.Scan(&jumlahAnak)

    for i := 0; i < jumlahAnak; i++ {
        fmt.Printf("Masukkan berat anak ke-%d (kg) : ", i+1)
        fmt.Scan(&beratAnak[i])
```

```
    }

    cariMinMax(beratAnak, &beratTerkecil, &beratTerbesar)

    rataRataBerat := hitungRataRata(beratAnak, jumlahAnak)

    fmt.Printf("\n= Hasil Analisis Data Berat Anak =\n")
    fmt.Printf("Berat Terkecil : %.2f kg\n", beratTerkecil)
    fmt.Printf("Berat Terbesar : %.2f kg\n", beratTerbesar)
    fmt.Printf("Berat Rata-rata : %.2f kg\n", rataRataBerat)
}
```

Screenshoot Output

```
Berat Rata-rata : 2.00 kg
PS C:\Users\nasyy> go run "d:\semester 3\Praktikum Alpro 2\Modul 11\unguided3.go"
Masukkan jumlah anak : 4
Masukkan berat anak ke-1 (kg) : 5
Masukkan berat anak ke-2 (kg) : 6
Masukkan berat anak ke-3 (kg) : 9
Masukkan berat anak ke-4 (kg) : 5

= Hasil Analisis Data Berat Anak =
Berat Terkecil : 5.00 kg
Berat Terbesar : 9.00 kg
Berat Rata-rata : 6.25 kg
PS C:\Users\nasyy>
```

Deskripsi Program

Program ini menghitung berat badan anak-anak. Pertama, program meminta pengguna untuk memasukkan jumlah anak dan berat masing-masing anak. Kemudian, program menggunakan fungsi cariMinMax untuk mencari berat badan anak terkecil dan terbesar. Fungsi hitungRataRata digunakan untuk menghitung rata-rata berat badan semua anak. Terakhir, program mencetak hasil perhitungan, yaitu berat badan terkecil, terbesar, dan rata-rata. Kode ini juga menggunakan tipe data arrBeratAnak yang merupakan array dengan panjang tetap untuk menyimpan data berat badan anak-anak. Selain itu, kode ini juga menggunakan pointer untuk mengubah nilai variabel dari dalam fungsi.