LAPORAN PRAKTIKUM
ALGORITMA PEMROGRAMAN 2

MODUL 10

PENCARIAN NILAI EKSTRIM PADA HIMPUNAN DATA

Oleh:

DAMARA GALUH PEMBAYUN

2311102110

IF-11-02

S1 TEKNIK INFORMATIKA

INSTITUT TEKNOLOGI TELKOM PURWOKERTO

2024

I. DASAR TEORI

Dalam analisis data, istilah "nilai ekstrim" mengacu pada nilai tertinggi (maksimum) atau terendah (minimum) dalam kumpulan data tertentu. Identifikasi nilai ekstrim menawarkan banyak keuntungan, termasuk memberikan gambaran umum tentang sebaran data dalam statistik deskriptif dan membantu proses pengambilan keputusan dalam berbagai bidang, seperti teknik, bisnis, dan ilmu pengetahuan. Untuk metode pencarian nilai ekstrim, pemeriksaan manual atau pengurutan data cukup mudah; namun, algoritma komputasional seperti pencarian linear, pembagian, dan penaklukan, atau penggunaan struktur data khusus seperti heap menjadi lebih efisien. Selain itu, berbagai bahasa pemrograman mendukung penerapan pencarian nilai ekstrim dengan menyediakan fungsi atau library bawaan yang mempermudah proses komputasi. Dalam kehidupan nyata, nilai ekstrim memiliki aplikasi yang luas, mulai dari analisis data sederhana seperti menentukan suhu tertinggi dan terendah hingga penerapan dalam algoritma optimisasi dan machine learning.

Beberapa hal perlu diperhatikan saat mencari nilai ekstrim. Pertama, efisiensi algoritma sangat penting, terutama untuk data dalam skala besar. Kedua, akurasi data sangat penting untuk mendapatkan hasil yang valid. Terakhir, pemilihan metode dan bahasa pemrograman yang tepat sangat mempengaruhi keberhasilan implementasi algoritma. Selain itu, hasil analisis nilai ekstrim dapat dipengaruhi oleh adanya outlier atau data yang sangat berbeda dari data lainnya. Oleh karena itu, memahami karakteristik data dengan baik dan memilih metode yang tepat sangat penting untuk melakukan analisis nilai ekstrim.

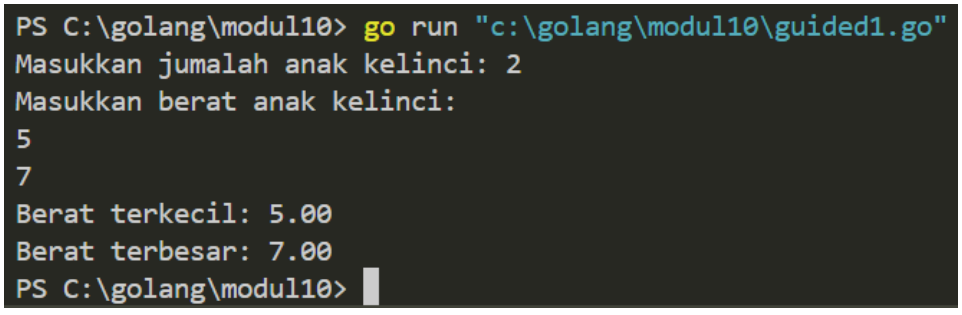
II. GUIDED

Guided 1

SourchCode

```
package main

import (
    "fmt"
)

func main() {
    var N int
    var berat [1000]float64

    fmt.Print("Masukkan jumalah anak kelinci: ")
    fmt.Scan(&N)

    fmt.Println("Masukkan berat anak kelinci: ")
    for i := 0; i < N; i++ {
        fmt.Scan(&berat[i])
    }

    // Inisialisasi nilai min dan max dengan elemen
pertama
    min := berat[0]
    max := berat[0]

    for i := 1; i < N; i++ {
        if berat[i] < min {
            min = berat[i]
        }
        if berat[i] > max {
            max = berat[i]
        }
    }

    fmt.Printf("Berat terkecil: %.2f\n", min)
    fmt.Printf("Berat terbesar: %.2f\n", max)
}
```

Hasil ScreenShoot



Guided 2

SourchCode

```
package main

import (
    "fmt"
)

func main() {
    var x, y int
    fmt.Print("Masukkan jumlah ikan dan kapasistas wadah:
")
    fmt.Scan(&x, &y)

    berat := make([]float64, x)
    fmt.Println("Masukkan bearat tiap ikan:")
    for i := 0; i < x; i++ {
        fmt.Scan(&berat[i])
    }

    jumlahWadah := (x + y - 1) / y // Pembulatan keatas
jika x tidak dibagi y
    totalBeratWadah := make([]float64, jumlahWadah)

    for i := 0; i < x; i++ {
        indeksWadah := i / y
        totalBeratWadah[indeksWadah] += berat[i]
    }

    // Output total berat tiap wadah
    fmt.Println("Total berat tiap wadah: ")
    for _, total := range totalBeratWadah {
        fmt.Printf("%.2f", total)
    }
    fmt.Println()

    // Output rata-rata berat tiap wadah
    fmt.Println("Rata-rata berat tiap wadah: ")
    for _, total := range totalBeratWadah {
        rataRata := total / float64(y)
        fmt.Printf("%.2f", rataRata)
    }
    fmt.Println()
}
```

Hasil ScreenShoot

```
PS C:\golang\modul10> go run "c:\golang\modul10\guided2.go"
Masukkan jumlah ikan dan kapasistas wadah: 7 8
Masukkan bearat tiap ikan:
7
9
8
7
5
7
9
Total berat tiap wadah:
52.00
Rata-rata berat tiap wadah:
6.50
PS C:\golang\modul10>
```

III. UNGUIDED

Unguided 1

SourchCode

```
package main

import "fmt"

// arrBalita adalah tipe data array dengan ukuran maksimal
100 elemen,
// yang menyimpan nilai berat balita dalam tipe float64.
type arrBalita [100]float64

// hitungMinMax menghitung nilai minimum dan maksimum dari
array berat balita.
// Parameter:
//   arrBerat: array yang berisi data berat balita.
//   bMin: pointer ke variabel yang akan menyimpan nilai
minimum.
//   bMax: pointer ke variabel yang akan menyimpan nilai
maksimum.
func hitungMinMax(arrBerat arrBalita, bMin *float64, bMax
*float64) {
    *bMin = arrBerat[0]
    *bMax = arrBerat[0]

    for _, berat := range arrBerat {
        if berat < *bMin {
            *bMin = berat
        }
```

```
        if berat > *bMax {
            *bMax = berat
        }
    }
}

// rerata menghitung nilai rata-rata dari array berat
balita.
// Parameter:
//   arrBerat: array yang berisi data berat balita.
// Return:
//   nilai rata-rata dari array berat balita.
func rerata(arrBerat arrBalita) float64 {
    var total float64
    for _, berat := range arrBerat {
        total += berat
    }
    return total / float64(len(arrBerat))
}

func main() {
    var n int
    fmt.Print("Masukkan banyak data berat balita: ")
    fmt.Scan(&n)

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

    var min, max float64
    hitungMinMax(beratBalita, &min, &max)

    rata := rerata(beratBalita)

    fmt.Printf("Berat balita minimum: %.2f kg\n", min)
    fmt.Printf("Berat balita maksimum: %.2f kg\n", max)
    fmt.Printf("Rerata berat balita: %.2f kg\n", rata)
}
```

Hasil ScreenShoot

```
PS C:\golang\modul10> go run "c:\golang\modul10\unguided1.go"
Masukkan banyak data berat balita: 6
Masukkan berat balita ke-1: 5
Masukkan berat balita ke-2: 4
Masukkan berat balita ke-3: 6
Masukkan berat balita ke-4: 7
Masukkan berat balita ke-5: 4
Masukkan berat balita ke-6: 3
Berat balita minimum: 0.00 kg
Berat balita maksimum: 7.00 kg
Rerata berat balita: 0.29 kg
PS C:\golang\modul10> go run "c:\golang\modul10\guided1.go"
```

Deskripsi

Kode Go dibuat untuk menghitung statistik dasar dari data berat badan balita, seperti nilai terkecil, terbesar, dan rata-rata. Ini dilakukan dengan meminta pengguna memasukkan data berat badan kemudian memprosesnya menggunakan fungsi yang telah ditetapkan untuk menemukan nilai minimum, maksimum, dan rata-rata. Sederhananya, program ini mirip dengan kalkulator kecil yang khusus untuk menghitung berat badan balita.