

**LAPORAN PRAKTIKUM
ALGORITMA PEMOGRAMAN 2**

MODUL 10

MATERI

PENCARIAN NILAI EKSTRIM PADA HIMPUNAN DATA



Oleh:

FEBRIAN FALIH ALWAFI

2311102181

S1F-11-02

S1 TEKNIK INFORMATIKA

UNIVERSITAS TELKOM PURWOKERTO

2024

I. DASAR TEORI

A. Ide pencarian nilai Max/Min

Pencarian adalah suatu proses yang lazim dilakukan di dalam kehidupan sehari-hari. Contoh penggunaannya dalam kehidupan nyata sangat beragam, misalnya pencarian file di dalam directory komputer, pencarian suatu teks di dalam sebuah dokumen, pencarian buku pada rak buku, dan contoh lainnya. Pertama pada modul ini akan dipelajari salah satu algoritma pencarian nilai terkecil atau terbesar pada sekumpulan data, atau biasa disebut pencarian nilai ekstrim. Ide algoritma sederhana sekali. Karena data harus diproses secara sekuensial, maka nilai atau indeks ke nilai maksimum dari data yang telah diproses disimpan untuk dibandingkan dengan data berikutnya. Nilai yang berhasil disimpan sampai algoritma tersebut berakhir adalah nilai maksimum yang dicari. Adapun algoritmanya secara umum adalah sebagai berikut:

- 1) Jadikan data pertama sebagai nilai ekstrim
- 2) Lakukan validasi nilai ekstrim dari data kedua hingga data terakhir.
 - Apabila nilai ekstrim tidak valid, maka update nilai ekstrims tersebut dengan dicek.
- 3) Apabila semua data telah dicek, maka nilai ekstrim yang dimiliki adalah valid.

B. Pencarian Nilai Ekstrim pada Array Bertipe Data Dasar

Misalnya terdefinisi sebuah array of integer dengan kapasitas 2023, dan array berisi sejumlah N bilangan bulat, kemudian pencarian nilai terkecil dilakukan pada array tersebut. Perhatikan potongan program dalam bahasa Go berikut ini :

```

type arrInt [2023]int
...

func terkecil_1(tabInt arrInt, n int) int {
/* mengembalikan nilai terkecil yang terdapat di dalam tabInt yang berisi n
bilangan bulat */
    var min int = tabInt[0]           // min berisi data pertama
    var j int = 1                     // pencarian dimulai dari data berikutnya
    for j < n {
        if min > tabInt[j] {          // pengecekan apakah nilai minimum valid
            min = tabInt[j]           // update nilai minimum dengan yang valid
        }
        j = j + 1
    }
    return min                         // returnkan nilai minimumnya
}

```

C. Pencarian Nilal Ekstrim pada Array Bertipe Data Terstruktur

Pada kasus yang lebih kompleks pencarian ekstrim dapat juga dilakukan, misalnya mencari data mahasiswa dengan nilai terbesar, mencari lagu dengan durasi terlama, mencari pembalap yang memiliki catatan waktu balap tercepat, dan sebagainya. Sebagai contoh misalnya terdapat array yang digunakan untuk menyimpan data mahasiswa, kemudian terdapat fungsi IPK yang digunakan untuk mencari data mahasiswa dengan IPK tertinggi.

```

...
type mahasiswa struct {
    nama, nim, kelas, jurusan string
    ipk float64
}
type arrMhs [2023]mahasiswa
...

func IPK_1(T arrMhs, n int) float64 {
/* mengembalikan ipk terkecil yang dimiliki mahasiswa pada array T yang berisi
n mahasiswa */
    var tertinggi float64 = T[0].ipk
    var j int = 1
    for j < n {
        if tertinggi < T[j].ipk {
            tertinggi = T[j].ipk
        }
        j = j + 1
    }
    return tertinggi
}

```

Apabila diperhatikan potongan program di atas, maka kita akan memperoleh ipk tertinggi, tetapi kita tidak memperoleh identitas mahasiswa dengan ipk tertinggi tersebut. Maka seperti penjelasan yang sudah diberikan sebelumnya, maka pencarian yang dilakukan bisa mengembalikan indeks mahasiswa dengan ipk tertinggi tersebut.

II. GUIDED

1. GUIDED 1

```
package main

import (
    "fmt"
)

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

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

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

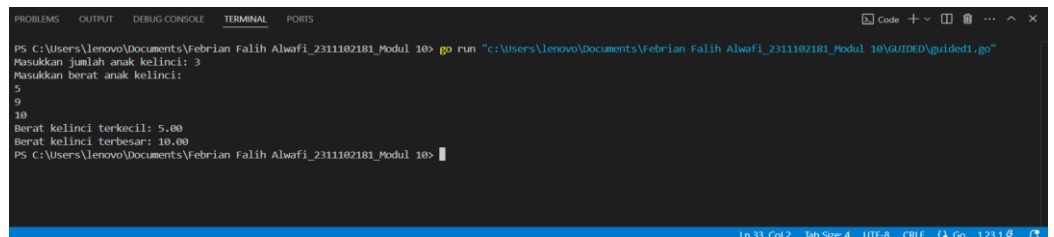
    min := berat[0]
    max := berat[0]

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

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

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

Output :



Deskripsi :

Program diatas untuk menentukan berat terkecil dan terbesar dari sejumlah anak kelinci berdasarkan input dari pengguna. Program dimulai dengan mengimpor paket fmt, yang digunakan untuk menangani input dan output. Kemudian, dua variabel utama dideklarasikan: N untuk menyimpan jumlah anak kelinci, dan berat, sebuah array dengan kapasitas maksimum 1000 elemen untuk menyimpan berat setiap anak kelinci dalam tipe data float64.

2. GUIDED 2

```
package main  
  
import "fmt"  
  
func main() {  
    var x, y int  
    fmt.Print("Masukkan jumlah ikan dan kapasitas  
wadah: ")  
    fmt.Scan(&x, &y)  
  
    berat := make([]float64, x)  
    fmt.Println("Masukkan berat tiap ikan:")  
    for i := 0; i < x; i++ {  
        fmt.Scan(&berat[i])  
    }  
}
```

```

    }

    jumlahWadah := (x + y - 1) / y
    totalBeratWadah := make([]float64,
jumlahWadah)
    jumlahIkanDiWadah := make([]int,
jumlahWadah)

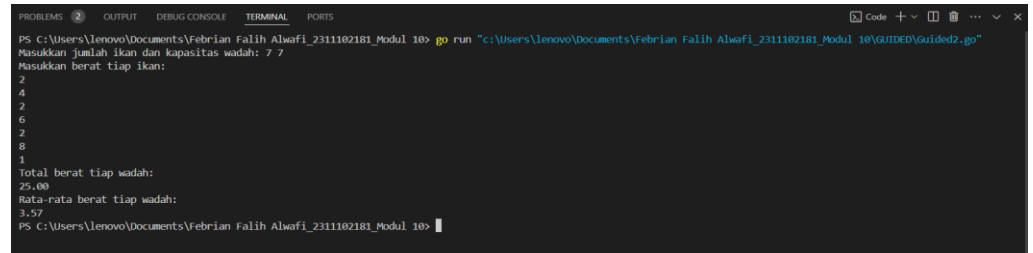
    for i := 0; i < x; i++ {
        indeksWadah := i / y
        totalBeratWadah[indeksWadah] += berat[i]
        jumlahIkanDiWadah[indeksWadah]++
    }

    // 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 i := 0; i < jumlahWadah; i++ {
        if jumlahIkanDiWadah[i] > 0 {
            rataRata := totalBeratWadah[i] /
float64(jumlahIkanDiWadah[i])
            fmt.Printf("%.2f ", rataRata)
        } else {
            fmt.Printf("0.00 ") // jika wadah kosong
        }
    }
    fmt.Println()
}

```

Output :



```
PS C:\Users\lenovo\Documents\Febrian Falih Alwafi_2311102181_Modul 10> go run "c:\Users\lenovo\Documents\Febrian Falih Alwafi_2311102181_Modul 10\GUIDED\guided2.go"
Masukkan jumlah ikan dan kapasitas wadah: 7 7
Masukkan berat tiap ikan:
2
4
2
6
2
8
1
Total berat tiap wadah:
25.00
Rata-rata berat tiap wadah:
3.57
PS C:\Users\lenovo\Documents\Febrian Falih Alwafi_2311102181_Modul 10>
```

Deskripsi :

Program diatas bertujuan untuk membagi sejumlah ikan ke dalam wadah, menghitung total berat ikan di setiap wadah, dan menghitung rata-rata berat ikan per wadah. Program dimulai dengan mengimpor paket fmt untuk menangani input dan output. Variabel x menyimpan jumlah ikan, sedangkan y menyimpan kapasitas maksimum jumlah ikan per wadah. Pengguna juga diminta memasukkan kedua nilai tersebut melalui input konsol.

III. UNGUIDED

Mengerjakan soal modul 10 nomor 3

3. Source Code

```
package main

import "fmt"

func hitungMinMax_2311102181(arrBerat []float64)
(float64, float64) {
    min, max := arrBerat[0], arrBerat[0]
    for _, berat := range arrBerat {
        if berat < min {
            min = berat
        }
        if berat > max {
            max = berat
        }
    }
}
```

```

    }
}
return min, max
}

func rataRata(arrBerat []float64) float64 {
    var total float64
    for _, berat := range arrBerat {
        total += berat
    }
    return total / float64(len(arrBerat))
}

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

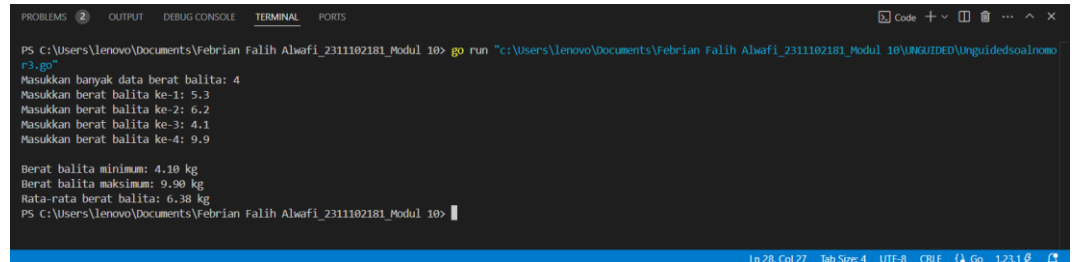
    balitas := make([]float64, jmlBalita)
    for i := 0; i < jmlBalita; i++ {
        fmt.Printf("Masukkan berat balita ke-%d: ", i+1)
        fmt.Scan(&balitas[i])
    }

    min, max := hitungMinMax_2311102181(balitas)

    fmt.Printf("\nBerat balita minimum: %.2f kg\n",
min)
    fmt.Printf("Berat balita maksimum: %.2f kg\n",
max)
    fmt.Printf("Rata-rata berat balita: %.2f kg\n",
rataRata(balitas))
}

```


Output :



```
PS C:\Users\lenovo\Documents\Febrian Falih Alwafi_2311102181_Modul 10> go run "c:\Users\lenovo\Documents\Febrian Falih Alwafi_2311102181_Modul 10\UNGUIDED\Unguidedsoalnomor3.go"
Masukkan banyak data berat balita: 4
Masukkan berat balita ke-1: 5.3
Masukkan berat balita ke-2: 6.2
Masukkan berat balita ke-3: 4.1
Masukkan berat balita ke-4: 9.9

Berat balita minimum: 4.10 kg
Berat balita maksimum: 9.90 kg
Rata-rata berat balita: 6.38 kg
PS C:\Users\lenovo\Documents\Febrian Falih Alwafi_2311102181_Modul 10>
```

Deskripsi :

Program ini digunakan untuk menghitung berat minimum, maksimum, dan rata-rata berat balita berdasarkan input dari pengguna. Program ini menggunakan slice untuk menyimpan data berat balita dan fungsi-fungsi terpisah untuk menghitung nilai minimum, maksimum, dan rata-rata. Pertama program dimulai dengan deklarasi fungsi `hitungMinMax`, yang menerima slice `arrBerat` sebagai parameter. Fungsi ini menginisialisasi nilai awal min dan max menggunakan elemen pertama dari slice. Kemudian, dengan menggunakan loop berbasis range, setiap berat dalam slice dibandingkan untuk memperbarui nilai min dan max jika ditemukan nilai yang lebih kecil atau lebih besar. Fungsi ini mengembalikan dua nilai, yaitu berat minimum dan maksimum.