

**LAPORAN PRAKTIKUM  
ALGORITMA DAN PEMROGRAMAN II  
MODUL X  
“PENCARIAN NILAI EKSTRIM PADA HIMPUNAN DATA”**



Oleh:

**ZAHRINA ANTIKA MALAHATI**

**2311102109**

**IF 11 02**

**S1 TEKNIK INFORMATIKA  
TELKOM UNIVERSITY PURWOKERTO  
2024**

## I. DASAR TEORI

Nilai ekstrim dalam suatu himpunan data adalah nilai tertinggi (maksimum) atau nilai terendah (minimum) dari semua elemen dalam himpunan tersebut. Pencarian nilai ekstrim merupakan salah satu operasi dasar dalam pengolahan data yang sering digunakan dalam berbagai aplikasi.

### ➤ 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 computer, pencarian suatu teks di dalam sebuah dokumen, pencarian buku pada rak buku, dan contoh lainnya.

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 ekstrim dengan data yang dicek.)
- 3.) Apabila semua data telah dicek, maka nilai ekstrim yang dimiliki adalah valid.

Notasi dalam bahasa Go,

```
max = 0
i = 1
for i < n {
    if a[i] < max {
        max = i
    }
    i = i + 1
}
```

### ➤ Pencarian Nilai ekstrim pada Array Bertipe Data Dasar

Pencarian nilai ekstrim pada array adalah proses menemukan nilai terbesar (maksimum) dan terkecil (minimum) dari sekumpulan data yang tersimpan dalam sebuah array. Array adalah struktur data yang terdiri dari kumpulan elemen dengan tipe data yang sama, diurutkan dalam urutan yang berurutan.

Misalnya terdefinisi sebuah array of integer dengan kapasitas 2023, dan array berisi sejumlah N bilangan bulat, kemudian pencarian nilai terkecil dilakukan pada array tersebut.

➤ **Pencarian Nilai ekstrim pada array Bertipe Data Terstruktur**

Pencarian nilai ekstrim dalam konteks ini berarti mencari elemen dengan nilai atribut tertentu yang paling besar (maksimum) atau paling kecil (minimum). Misalnya, jika kita memiliki array berisi data mahasiswa dengan atribut nama, NIM, dan nilai ujian, kita bisa mencari mahasiswa dengan nilai ujian tertinggi atau terendah.

## II. GUIDED

### ❖ Guided 1

- 1) 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.

### Source code

```
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])
    }

    // 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)
}
```

## Output

```
PS C:\PRAKALPRO2\MODUL10> go run "c:\PRAKALPRO2\MODUL10\guided1.go"
Masukkan jumlah anak kelinci: 5
Masukkan berat anak kelinci:
15
10
13
12
14
Berat terkecil: 10.00
Berat terbesar: 15.00
PS C:\PRAKALPRO2\MODUL10> █
```

## Deskripsi

Program adalah program dalam bahasa Go untuk mencari berat terkecil dan terbesar dari sekumpulan data berat anak kelinci. Program ini akan meminta pengguna memasukkan jumlah anak kelinci dan berat masing-masing. Setelah itu, program akan memproses data tersebut dan menampilkan berat terkecil dan terbesar yang ditemukan. Algoritma yang digunakan cukup sederhana namun efektif, yaitu dengan membandingkan setiap data dengan nilai minimum dan maksimum yang terus diperbarui.

## ❖ Guided 2

- 2) 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 dimasukkan 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 dimasukkan 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.

## Source code

```
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 // 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()
}

```

## Output

```

PS C:\PRAKALPRO2\MODUL10> go run "c:\PRAKALPRO2\MODUL10\guided2.go"
Masukkan jumlah ikan dan kapasitas wadah: 5 10
Masukkan berat tiap ikan:
2
3
4
5
6
Total berat tiap wadah:
20.00
Rata-rata berat tiap wadah:
2.00
PS C:\PRAKALPRO2\MODUL10>

```

## Deskripsi

Program adalah program dalam bahasa Go untuk mendistribusikan sejumlah ikan ke dalam beberapa wadah dengan kapasitas yang sama. Program ini akan meminta pengguna memasukkan jumlah ikan dan kapasitas maksimal setiap wadah. Selanjutnya, program akan membagi ikan-ikan tersebut secara merata ke dalam wadah-wadah tersebut. Setelah itu, program akan menghitung dan menampilkan total berat ikan serta rata-rata berat ikan di setiap wadah. Algoritma yang digunakan cukup sederhana, namun efektif untuk menyelesaikan masalah pembagian beban secara merata.

### III. UNGUIDED

#### ❖ Unguided 1

#### Soal

- 3) 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
```

Halaman 73 | Modul Praktikum Algoritma dan Pemrograman 2

```
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
```

#### Source code

```
package main

import (
    "fmt"
)

// Tipe data arrBalita
type arrBalita [100]float64
```

```

// Fungsi untuk menghitung nilai minimum dan maksimum dari
array
func hitungMinMax(arrBerat arrBalita, n int, bMin, bMax
*float64) {
    *bMin = arrBerat[0]
    *bMax = arrBerat[0]
    for i := 1; i < n; i++ {
        if arrBerat[i] < *bMin {
            *bMin = arrBerat[i]
        }
        if arrBerat[i] > *bMax {
            *bMax = arrBerat[i]
        }
    }
}

// Fungsi untuk menghitung rata-rata berat balita dalam
array
func hitungRataRata(arrBerat arrBalita, n int) float64 {
    total := 0.0
    for i := 0; i < n; i++ {
        total += arrBerat[i]
    }
    return total / float64(n)
}

func main() {
    var n int
    var arrBerat arrBalita
    var bMin, bMax float64

    // Input jumlah data balita
    fmt.Print("Masukkan banyak data berat balita: ")
    fmt.Scan(&n)

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

    // Hitung nilai minimum, maksimum, dan rata-rata
    hitungMinMax(arrBerat, n, &bMin, &bMax)
    rataRata := hitungRataRata(arrBerat, n)

    // Output hasil
    fmt.Printf("\nBerat balita minimum: %.2f kg\n", bMin)
    fmt.Printf("Berat balita maksimum: %.2f kg\n", bMax)
    fmt.Printf("Rata-rata berat balita: %.2f kg\n",
rataRata)
}

// Zahrina Antika Malahati_2311102109

```



## Output

```
PS C:\PRAKALPRO2\MODUL10> go run "c:\PRAKALPRO2\MODUL10\unguided1.go"
Masukkan banyak data berat balita: 4
Masukkan berat balita ke-1: 5.30
Masukkan berat balita ke-2: 6.20
Masukkan berat balita ke-3: 4.10
Masukkan berat balita ke-4: 9.90

Berat balita minimum: 4.10 kg
Berat balita maksimum: 9.90 kg
Rata-rata berat balita: 6.38 kg
PS C:\PRAKALPRO2\MODUL10> █
```

## Deskripsi

Program ini adalah program dalam bahasa Go untuk melakukan analisis data sederhana terhadap kumpulan berat badan balita. Program ini mampu menghitung nilai statistik dasar seperti nilai minimum, maksimum, dan rata-rata dari data yang diberikan. Setelah pengguna memasukkan jumlah data balita dan nilai berat badan masing-masing, program akan memproses data tersebut menggunakan fungsi `hitungMinMax` dan `hitungRataRata`. Fungsi `hitungMinMax` akan mencari nilai berat badan terkecil dan terbesar, sedangkan fungsi `hitungRataRata` akan menghitung rata-rata dari semua berat badan. Hasil perhitungan kemudian akan ditampilkan ke layar sehingga pengguna dapat dengan mudah melihat rangkuman statistik dari data berat badan balita yang telah dimasukkan.