

**LAPORAN PRAKTIKUM  
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL XI  
PENCARIAN NILAI EKSTRIM PADA HIMPUNAN DATA**



**Disusun Oleh :**

**Aji Noto Sutrisno (2311102262)**

**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

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

Berikut adalah notasi dalam pseudocode dan Bahasa Go, Misalnya untuk pencarian nilai maksimum :

	Notasi Algoritma	Notasi dalam bahasa Go
1	$\text{max} \leftarrow 1$	$\text{max} = 0$
2	$i \leftarrow 2$	$i = 1$
3	while $i \leq n$ do	for $i < n$ {
4	if $a[i] > a[\text{max}]$ then	if $a[i] > a[\text{max}]$ {
5	$\text{max} \leftarrow i$	$\text{max} = i$
6	endif	}
7	$i \leftarrow i + 1$	$i = i + 1$
8	endwhile	}

## II. 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

```
package main

import "fmt"

func inputArr(x []float32, n int) {
    var berat float32
    for i := 0; i < n; i++ {
        fmt.Printf("Input berat ke-%d : ", i+1)
        fmt.Scan(&berat)
        x[i] = berat
    }
}

func NilaiMaxMin(x []float32, n int) (float32, float32)
{
    max := x[0]
    min := x[0]
    for i := 1; i < n; i++ {
        if max < x[i] {
            max = x[i]
        }
        if min > x[i] {
            min = x[i]
        }
    }
    return max, min
}
```

```

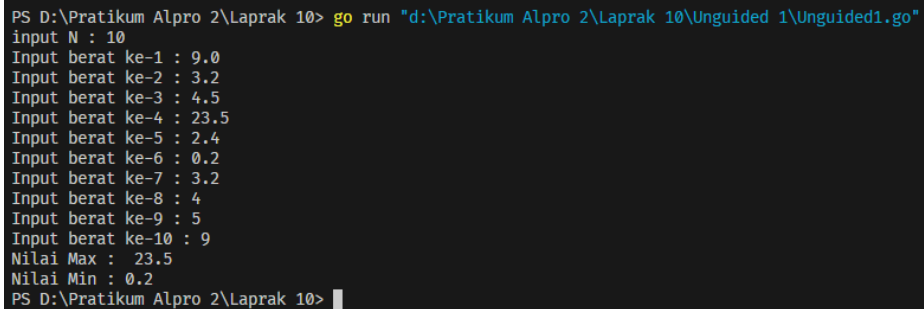
func main() {
    var N int
    var kelinci [999]float32
    fmt.Print("input N : ")
    fmt.Scan(&N)
    inputArr(kelinci[:N], N)

    max, min := NilaiMaxMin(kelinci[:N], N)

    fmt.Println("Nilai Max : ", max)
    fmt.Print("Nilai Min : ", min)
}

```

## Screenshoot Output



```

PS D:\Pratikum Alpro 2\Laprak 10> go run "d:\Pratikum Alpro 2\Laprak 10\Unguided 1\Unguided1.go"
input N : 10
Input berat ke-1 : 9.0
Input berat ke-2 : 3.2
Input berat ke-3 : 4.5
Input berat ke-4 : 23.5
Input berat ke-5 : 2.4
Input berat ke-6 : 0.2
Input berat ke-7 : 3.2
Input berat ke-8 : 4
Input berat ke-9 : 5
Input berat ke-10 : 9
Nilai Max : 23.5
Nilai Min : 0.2
PS D:\Pratikum Alpro 2\Laprak 10>

```

## Deskripsi Program

Program ini dibuat untuk menghitung berat badan kelinci. Program ini menambahkan data sesuai inputan **N** yang berupa panjang dari array, dan data tersebut ditambahkan ke array menggunakan func `inputArr(x []float32, n int)` setelah ditambahkan program akan mencari nilai maksimum dari nilai array yang telah diinputkan dengan cara melakukan perbandingan dari nilai awal hingga nilai akhir, dengan menggunakan func `NilaiMaxMin(x []float32, n int) (float32, float32)` fungsi ini memiliki kondisi `if max < x[i]`, dan `if min > x[i]` kemudian fungsi akan mereturn nilai max dan min.

## 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 dimasukkan ke dalam wadah. Baris kedua terdiri dari sejumlah x bilangan riil 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

### Sourcecode

```
package main

import "fmt"

func InputArr(ikan []float32, jumlah int) {
    for i := 0; i < jumlah; i++ {
        fmt.Printf("Masukkan berat ikan ke-%d: ", i+1)
        fmt.Scan(&ikan[i])
    }
}

func DistribusiIkan(ikan []float32, x, y int) {
    totalWadah := x / y
    sisa := x % y
    if sisa > 0 {
        totalWadah++
    }

    fmt.Printf("Jumlah wadah yang diperlukan: %d\n",
totalWadah)

    for i := 0; i < totalWadah; i++ {
        awal := i * y
        akhir := awal + y
        if akhir > x {
            akhir = x
        }
    }
}
```

```

        // Hitung total berat ikan di wadah
        totalBerat := float32(0)
        for j := awal; j < akhir; j++ {
            totalBerat += ikan[j]
        }

        // Hitung rata-rata berat ikan di wadah
        jumlahIkanDiWadah := akhir - awal
        rataRata := totalBerat /
float32(jumlahIkanDiWadah)

        // Tampilkan total berat dan rata-rata
        fmt.Printf("\nTotal berat ikan di wadah ke-%d:
%.2f\n", i+1, totalBerat)
        fmt.Printf("Berat rata-rata ikan di wadah ke-
%d: %.2f\n", i+1, rataRata)
    }
}

func main() {
    var x, y int
    fmt.Print("Banyak ikan (x): ")
    fmt.Scan(&x)
    fmt.Print("Maksimal per Wadah (y): ")
    fmt.Scan(&y)

    if x <= 0 || y <= 0 {
        fmt.Println("Banyak ikan dan kapasitas wadah
harus lebih dari 0.")
        return
    }

    ikan := make([]float32, x)
    InputArr(ikan, x)
    DistribusiIkan(ikan, x, y)
}

```

## Screenshoot Output

```
PS D:\Pratikum Alpro 2\Laprak 10> go run "d:\Pratikum Alpro 2\Laprak 10\Unguided 2\Unguided2.go"
Banyak ikan (x): 5
Maksimal per Wadah (y): 2
Masukkan berat ikan ke-1: 10
Masukkan berat ikan ke-2: 32
Masukkan berat ikan ke-3: 42
Masukkan berat ikan ke-4: 2
Masukkan berat ikan ke-5: 34
Jumlah wadah yang diperlukan: 3

Total berat ikan di wadah ke-1: 42.00
Berat rata-rata ikan di wadah ke-1: 21.00

Total berat ikan di wadah ke-2: 44.00
Berat rata-rata ikan di wadah ke-2: 22.00

Total berat ikan di wadah ke-3: 34.00
Berat rata-rata ikan di wadah ke-3: 34.00
PS D:\Pratikum Alpro 2\Laprak 10> █
```

## Deskripsi Program

Program ini dibuat untuk mencari berat total ikan dan rata rata setiap ikan pada masing masing wadah, pembagian wadah tersebut menggunakan cara,  $x$  dibagi  $y$ , jika habis maka jumlah tersebut adalah total wadah, apabila sisa, maka akan menambah 1 wadah lagi, misalnya pada output diatas, pengguna memiliki total ikan yaitu 5, dan maksimal setiap wadahnya itu 2, maka total wadahnya 2,2,1 yaitu 3. Untuk mencari jumlah berat dan rata-rata masing masing wadah program akan menghitung array dari indeks awal hingga indeks maksimal wadah, pada kasus diatas maksimalnya 2 yaitu indeks ke 0 hingga indeks ke 1.

### 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.

```
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

```
package main

import "fmt"

type arrBalita [100]float64

func hitungMinMax(arrBerat *arrBalita, n int) (float64, float64) {
    for i := 0; i < n; i++ {
        fmt.Printf("Masukkan Berat balita ke-%d : ",
i+1)
        fmt.Scan(&arrBerat[i])
    }
    Max := arrBerat[0]
    Min := arrBerat[0]
    for j := 1; j < n; j++ {
        if Max < arrBerat[j] {
            Max = arrBerat[j]
        }
        if Min > arrBerat[j] {
            Min = arrBerat[j]
        }
    }
    return Max, Min
}
```



```

    }
    if Min > arrBerat[j] {
        Min = arrBerat[j]
    }
}
return Max, Min
}

func rerata(arrBerat *arrBalita, n int) float64 {
    var total float64 = 0
    for i := 0; i < n; i++ {
        total += arrBerat[i]
    }
    return total / float64(n)
}

func main() {
    var arrBerat arrBalita
    var n int

    fmt.Print("Masukkan banyak data balita: ")
    fmt.Scan(&n)

    if n <= 0 {
        fmt.Println("Jumlah balita harus lebih dari
0.")
        return
    }

    Max, Min := hitungMinMax(&arrBerat, n)
    Rerata := rerata(&arrBerat, n)

    fmt.Printf("\nBerat balita minimum : %.2f\n", Min)
    fmt.Printf("\nBerat balita maksimum: %.2f\n", Max)
    fmt.Printf("\nRerata berat balita : %.2f\n",
Rerata)
}

```

## Screenshoot Output

```

PS D:\Pratikum Alpro 2\Laprak 10> go run "d:\Pratikum Alpro 2\Laprak 10\Unguided 3\Unguided3.go"
Masukkan banyak data 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
Berat balita maksimum: 9.90
Rerata berat balita : 6.38
PS D:\Pratikum Alpro 2\Laprak 10> █

```

## Deskripsi Program

Program ini dibuat untuk menghitung data berat balita, dan program tersebut mencari berat balita minimum, berat balita maksimum, dan rerata berat balita dari data yang diinputkan pengguna. Program ini meminta pengguna untuk menginputkan banyaknya data(**n**) kemudian pengguna memasukan inputan sebanyak data(**n**) kemudian program akan menghitung nilai maksimumnya dan nilai minimum kemudian program akan mencari rerata data dari data yang diinputkan.

Program ini memiliki 2 fungsi yaitu `func hitungMinMax(arrBerat *arrBalita, n int) (float64, float64)` dan `func rerata(arrBerat *arrBalita, n int) float64`