

**LAPORAN PRAKTIKUM  
ALGORITMA PEMROGRAMAN 2**

**MODUL 10**

**PENCARIAN NILAI EKSTRIM PADA HIMPUNAN DATA**



**Universitas  
Telkom**

Oleh:

**WILDAN DAFFA' HAKIM PUTRA ANTARA**

**2311102055**

**S1IF-11-02**

**S1 TEKNIK INFORMATIKA**

**UNIVERSITAS TELKOM**

**PURWOKERTO**

**2024**

## I. DASAR TEORI

Suatu nilai pada selang tertentu dapat dikatakan nilai ekstrim jika nilai tersebut mencapai nilai maksimumnya (paling besar) atau nilai minimumnya (nilai paling kecilnya). Dalam melakukan pencarian nilai ekstrim, apabila kita menggunakan algoritma sequential search hal itu dapat dilakukan dengan cukup mudah. Berikut tahapannya :

1. Gunakan indeks paling awal (0) atau posisi elemen pertama sebagai nilai pembanding pertama.
2. Lakukan perulangan sebanyak n-1 kali (data kedua sampai terakhir) untuk melakukan perbandingan apakah ada nilai yang lebih ekstrim (min/max) dari nilai pembanding.
3. Jika ada, maka tukar nilai pembanding dengan nilai saat ini

Dalam Pseudocode	Dalam Golang
<pre>max ← 1 i ← 2 while i ≤ n do     if a[i] &gt; a[max] then         max ← i     endif i ← i+1 endwhile</pre>	<pre>max = 0 i = 1 for i &lt; n {     if a[i] &gt; a[max] {         max = i     }     i=i+1 }</pre>

1. Catatan : pada golang max bernilai 0 karena indeks suatu array dimulai dari nol, dan nilai yang digunakan pada max bukan nilai dari array melainkan posisi index nilai ekstrim

## II. GUIDED

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

```
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)
}

```

```

λ MacBook-Air-Daffa Modul 10 → go run "/Users/daffahakim/Documents/
Masukkan jumlah anak kelinci: 3
Masukkan berat anak kelinci:
2 3 1
Berat terkecil: 1.00
Berat terbesar: 3.00
λ MacBook-Air-Daffa Modul 10 → █

```

Pada program ini kita mencari berat badan kelinci yang paling minimum dan maximum dari banyaknya kelinci yang kita inputkan. Pada program main, dideklarasikan dua variabel N bertipe integer dan berat berupa array dengan tipe data float dengan banyaknya elemen 1000. Setelah itu, pengguna diminta menginputkan jumlah anak kelinci lalu, dilakukan perulangan sebanyak jumlahnya. Selama perulangan, pengguna menginputkan masing – masing berat badan yang akan disimpan pada array berat indeks ke i (nilai i sesuai dengan iterasi saat itu).

Setelah penginputan sebanyak jumlah selesai, dideklarasikan variable min dan max dengan isi yaitu nilai array berat dengan indeks ke – 0 sebagai acuan pembandingan pertama. Lalu, dilakukan perulangan sebanyak jumlah Dimana selama perulangan akan dicek apakah nilai berat indeks ke – i lebih kecil dari nilai min sekarang. Jika benar, maka nilai min sekarang diganti dengan nilai berat indeks ke i saat itu (mencari nilai ekstrim minimum). Dan proses serupa dilakukan untuk mencari nilai ekstrim maksimum yaitu dengan membandingkan apakah array berat indeks ke – i lebih besar daripada nilai maksimum. Setelah perulangan selesai, nilai min dan max akan dicetak dengan format decimal 2 angka dibelakang koma.

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.

```
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 ke
    atas jika x tidak habis 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()
}
```

```

λ MacBook-Air-Daffa Modul 10 → go run "/Users/daffahakim/Documents/Kuliah/SMT
Masukkan jumlah ikan dan kapasitas wadah: 3 2
Masukkan berat tiap ikan:
3.1 2.1 5.6
Total berat tiap wadah:
5.20 5.60
Rata-rata berat tiap wadah:
2.60 2.80

```

Program ini digunakan untuk memetakan ada berapa ikan yang ada di wadah dengan total berat dan rata – rata beratnya berdasarkan jumlah dan kapasitas yang di inputkan di awal.

Pada fungsi main, di deklarasikan variabel x dan y dengan tipe integer untuk menyimpan banyaknya ikan dan kapasitas ikan (banyak ikan) di setiap toplesnya. Lalu, pengguna diminta menginputkan jumlah dan kapasitas yang akan disimpan pada variabel x dan y, setelahnya dibuat array dengan nama berat dengan fungsi make bertipe data float, dengan ukuran x, lalu dilakukan pengisian array dengan for loop sebanyak x yan pada setiap iterasinya akan dilakukan fmt.Scan (mengambil input user) dan dimasukkan ke berat indeks ke-i. setelah user menginputkan dan perulangan selesai, jumlah wadah ditentukan dengan rumus  $(x+y-1)/y$  yang akan menghasilkan banyaknya wadah untuk menampung ikan, hasil perhitungan ini dipakai untuk membuat array baru dengan nama totalBeratWadah.

Setelah itu, dilakukan perulangan sebanyak x untuk menjumlah berat ikan di setiap wadah dimana indexnya merupakan hasil bagi  $i/y$  (integer division). Setelah selesai perulangan total berat dari wadah ditampilkan, sedangkan untuk rata rata maka total berat dibagi dengan kapasitas ikan dalam wadah. Semua angka yang ditampilkan menggunakan format dua angka dibelakang koma (%.2f)

### III. UNGUIDED

1. 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.

```

package main

import "fmt"

type arrBalita [100]float64

```

```

func hitungMinMax(arrBerat arrBalita, n int, bMin, bMax
*float64) {
    *bMin = arrBerat[0]
    *bMax = arrBerat[0]

    for i := 1; i < n; i++ {
        if *bMax < arrBerat[i] {
            *bMax = arrBerat[i]
        }

        if *bMin > arrBerat[i] {
            *bMin = arrBerat[i]
        }
    }
}

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

func main() {
    var dataBalita arrBalita
    var n int
    var max, min float64
    fmt.Print("Masukan banyak data berat balita : ")
    fmt.Scan(&n)
    for i := 0; i < n; i++ {
        fmt.Print("Masukan berat balita ke-", i+1,
": ")
        fmt.Scanln(&dataBalita[i])
    }

    hitungMinMax(dataBalita, n, &min, &max)

    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",
rerata(dataBalita, n))
}

```

```

Rerata berat balita: 21.00 kg
λ MacBook-Air-Daffa Modul 10 → go run "/Users/daffahakim/Document
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
λ MacBook-Air-Daffa Modul 10 →

```

Program ini digunakan untuk mengetahui nilai ekstrim (minimum dan maksimum) serta rata rata dari setiap berat balita yang diinputkan. Program ini memiliki 1 fungsi yaitu rerata untuk menghitung rata - rata dan 1 prosedur yaitu hitungMinMax untuk mencari nilai ekstrim.

Pada fungsi main, dideklarasikan variable dataBalita dengan tipe data bentukan arrBalita yang merupakan array dengan banyak elemen 100 dengan tipe float64 lalu n untuk menentukan banyaknya elemen pada array dan min, max untuk menampung nilai berat balita paling ringan dan berat yang akan disimpan pada variabel n. Setelah itu, pengguna diminta memasukkan ada berapa balita yang data beratnya mau diinputkan. setelah itu, dilakukan perulangan sebanyak n untuk memasukkan data berat balita ke array dataBalita. Setelah selesai perulangan dipanggil prosedur hitungMinMax dengan parameter formal arrBerat bertipe data arrBalita, n dengan tipe integer, dan bMin, bMax dengan tipe data float64 (reference).

Pada fungsi hitungMinMax akan dilakukan pencarian nilai ekstrim, sebelum melakukan pencarian nilai bMin dan bMax diubah dulu menjadi nilai array dataBalita indeks ke – 0. Setelah itu dilakukan perulangan jika ditemukan nilai yang lebih kecil dari bMin maka nilai bMin akan diganti nilai array saat perulangan itu. Begitupun untuk bMax jika ditemukan nilai yang lebih besar maka nilai bMax diganti dengan nilai array saat itu. Parameter formal ini diisi dengan parameter aktual dataBalita, n, reference dari min, dan reference dari max. setelah itu nilai variable min dan max dicetak mewakili bobot minimum dan maksimum. Pada bagian akhir dicetak juga rata – rata bobot bayi dengan memanggil fungsi rerata yang memiliki parameter formal array dengan nama arrBerat dengan tipe data arrBalita dan n yang merupakan banyaknya balita.

Pada fungsi rerata akan dilakukan perulangan sebanyak n dimana setiap perulangan akan mengambil nilai elemen ke – i pada array arrBerat dan menambahkannya pada sum. Setelah perulangan selesai, sebagai

Kembalian, hasil pertambahan semua elemen array dibagi dengan nilai  $n$  yang menghasilkan nilai rata – rata semua bayi. Parameter aktual dari fungsi ini adalah nilai variable `dataBalita` dan `n`.