

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2
MODUL 10
PENCARIAN NIKAI EKSTIM PADA HIMPUNAN DATA**



Oleh:

GALIH TRISNA

2311102050

IF-11-02

**S1 TEKNIK INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2024**

I. DASAR TEORI

Pencarian nilai ekstrem pada himpunan data adalah proses untuk menemukan elemen terkecil (minimum) dan elemen terbesar (maksimum) dalam kumpulan data tertentu. Proses ini penting dalam berbagai analisis data untuk mendapatkan wawasan tentang rentang nilai, distribusi data, dan outlier. Dalam bahasa pemrograman, pencarian nilai ekstrem sering diimplementasikan dengan perulangan melalui setiap elemen dalam array atau koleksi data dan membandingkan setiap elemen dengan variabel yang menyimpan nilai minimum atau maksimum sementara. Jika elemen saat ini lebih besar dari maksimum sementara, maka variabel maksimum diperbarui; jika lebih kecil dari minimum sementara, maka variabel minimum diperbarui.

Contoh

	Notasi Algoritma	Notasi dalam bahasa Go
1	<code>max ← 1</code>	<code>max = 0</code>
2	<code>i ← 2</code>	<code>i = 1</code>
3	<code>while i <= n do</code>	<code>for i < n {</code>
4	<code> if a[i] > a[max]</code>	<code> a[i] > a[max] {</code>
5	<code> max ← i</code>	<code> max = i</code>
6	<code> endif</code>	<code> }</code>
7	<code> i ← i + 1</code>	<code> i = i + 1</code>
8	<code>endwhile</code>	<code>}</code>

II. GUIDED

1. 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)
}
```

Screenshot

```
PS C:\Users\galih\Documents\belajar\Algoritma Pemrogramman 2> g
d\guided_1\guided_1.go"
Masukkan jumlah anak kelinci: 5
Masukkan berat anak kelinci:
4 5 6 7 8
Berat terkecil: 4.00
Berat terbesar: 8.00
PS C:\Users\galih\Documents\belajar\Algoritma Pemrogramman 2>
```

Penjelasan

Program diatas merupakan program untuk menemukan berat anak kelinci terkecil dan terbesar dari input pengguna. Program dimulai dengan mendeklarasikan array berat berukuran 1000 elemen dan variabel N untuk menyimpan jumlah anak kelinci. Pengguna diminta memasukkan jumlah anak kelinci (variabel N), setelah itu menginputkan berat masing-masing anak kelinci. Nilai berat disimpan dalam array berat. Variabel min dan max diinisialisasi dengan elemen pertama dari array. Kemudian, program menggunakan perulangan untuk membandingkan setiap elemen selanjutnya, memperbarui min jika ditemukan berat lebih kecil, dan max jika ditemukan berat lebih besar. Hasilnya ditampilkan dengan format dua desimal, menampilkan berat terkecil dan terbesar dari seluruh data anak kelinci.

2. 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 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()
}
```

Screenshoot

```
PS C:\Users\galih\Documents\belajar\Algoritma Pemrogramman 2> g
d\guided_2\guided_2.go"
Masukkan jumlah ikan dan kapasitas wadah: 5 5
Masukkan berat tiap ikan:
4 5 6 7 8
Total berat tiap wadah:
30.00
Rata-rata berat tiap wadah:
6.00
PS C:\Users\galih\Documents\belajar\Algoritma Pemrogramman 2> |
```

Penjelasan :

Program diatas merupakan program untuk menghitung total dan rata-rata berat ikan yang dimasukkan ke dalam wadah dengan kapasitas tertentu. Program meminta pengguna untuk input jumlah ikan (variabel x) dan kapasitas maksimal wadah (variabel y), dan kemudian pengguna menginputkan berat masing-masing ikan. Array berat digunakan untuk menyimpan data berat ikan, dan jumlahWadah dihitung dengan rumus $(x + y - 1) / y$ untuk memastikan pembulatan ke atas jika jumlah ikan tidak habis dibagi kapasitas wadah. Program mengisi array totalBeratWadah dengan menambahkan berat ikan ke wadah sesuai indeksinya. Setelah semua ikan dimasukkan ke dalam wadah, program mencetak total berat masing-masing wadah, lalu menghitung dan menampilkan rata-rata berat per wadah.

III. UNGUIDED

1. Source Code

```
package main

import (
    "fmt"
)

type arrBalita [100]float64

func hitungMinMax(arrBerat arrBalita, x int, bMin, bMax *float64){
    *bMin = arrBerat[0]
    *bMax = arrBerat[0]
    for i := 1; i < x; i++ {
        if arrBerat[i] > *bMax {
            *bMax = arrBerat[i]
        }
        if arrBerat[i] < *bMin {
            *bMin = arrBerat[i]
        }
    }
}

func rerata(arrBerat arrBalita, x int) float64{
    sum := 0.0
    for i := 0; i < x; i++ {
        sum += arrBerat[i]
    }
    rata_rata := sum / float64(x)
    return rata_rata
}

func main(){
    var x int
    var arrBalita[100] float64
    var bMin, bMax float64

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

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

    hitungMinMax(arrBalita, x, &bMin, &bMax)
    avg := rerata(arrBalita, x)

    fmt.Printf("Berat balita maksimal: %.2f kg\n", bMax)
    fmt.Printf("Berat balita minimum: %.2f kg\n", bMin)
    fmt.Printf("Rerata berat balita: %.2f kg", avg )
}
```

Screenshot

```
PS C:\Users\galih\Documents\belajar\Algoritma Pemrogramman 2>
ded\unguided_1.go"
Masukkan banyak data berat balita: 4
Masukkan berat balita ke-1: 12
Masukkan berat balita ke-2: 2
Masukkan berat balita ke-3: 3
Masukkan berat balita ke-4: 4

Berat balita maksimal: 12.00 kg
Berat balita minimum: 2.00 kg
Rerata berat balita: 5.25 kg
PS C:\Users\galih\Documents\belajar\Algoritma Pemrogramman 2>
```

Penjelasan

Program diatas merupakan program untuk menghitung berat maksimum, minimum, dan rata-rata berat balita berdasarkan data input dari pengguna. Struct `arrBalita` digunakan untuk menyimpan berat balita hingga 100 elemen. Fungsi `hitungMinMax` yaitu untuk menerima array berat balita, jumlah data `x`, dan juga pointer ke variabel `bMin` dan `bMax` untuk menyimpan hasil berat minimum dan maksimum. Pada program diatas terdapat logika perulangan untuk membandingkan setiap elemen dengan nilai `bMin` dan `bMax`, memperbarui nilainya jika ditemukan angka yang lebih besar atau lebih kecil. Fungsi `rerata` menghitung rata-rata dengan menjumlahkan semua elemen dalam array dan membagi hasilnya dengan `x`. Dalam fungsi `main` terdapat program untuk meminta jumlah data, membaca berat tiap balita, lalu menampilkan hasil berat maksimum, minimum, dan rata-rata dengan format dua desimal.