

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

MODUL XI

PENCARIAN NILAI EXTREM PADA HIMPUNAN DATA



Disusun Oleh :

Avrizal Setyo Aji Nugroho

2311102145

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

Pencarian adalah proses yang biasa dilakukan dalam berbagai aspek kehidupan sehari-hari. Misalnya, mencari file di komputer, mencari teks dalam dokumen, atau mencari buku di rak perpustakaan. Pencarian nilai ekstrem, yang berarti nilai terkecil atau terbesar dalam set data, adalah salah satu jenis pencarian yang penting. Nilai ekstrem bermanfaat untuk berbagai jenis analisis data, seperti menentukan rentang, rata-rata, atau menemukan data yang tidak biasa (outlier).

Konsep dasar algoritma pencarian nilai ekstrem sangat sederhana. Proses dimulai dengan menjadikan elemen pertama dalam kumpulan data sebagai nilai ekstrem sementara, baik untuk minimum maupun maksimum. Kemudian, setiap elemen berikutnya dibandingkan dengan nilai ekstrem tersebut secara berurutan. Jika ditemukan elemen dengan nilai minimum atau maksimum yang lebih kecil, maka nilai ekstrem tersebut diperbarui dengan elemen tersebut. Setelah semua elemen data diperiksa, prosedur ini diulang. Pada akhirnya, nilai tertinggi yang disimpan dalam algoritma adalah hasil pencarian yang valid.

II. UNGUIDED

1. Sourcecode

```
package main

import (
    "fmt"
)

func main() {
    var berat_145 [1000]float64

    var n int
    fmt.Print("Masukkan jumlah anak kelinci: ")
    fmt.Scan(&n)

    if n <= 0 || n > 1000 {
        fmt.Println("Jumlah anak kelinci antara 1 dan 1000.")
        return
    }

    fmt.Println("Masukkan berat anak kelinci (spasi):")
    for i := 0; i < n; i++ {
        fmt.Scan(&berat_145[i])
    }

    minBerat := berat_145[0]
    maxBerat := berat_145[0]

    for i := 1; i < n; i++ {
        if berat_145[i] < minBerat {
            minBerat = berat_145[i]
        }
        if berat_145[i] > maxBerat {
            maxBerat = berat_145[i]
        }
    }

    fmt.Printf("Berat terkecil: %.2f\n", minBerat)
    fmt.Printf("Berat terbesar: %.2f\n", maxBerat)
}
```

Screenshoot Output

```
PS D:\praktikum alpro> go run "d:\praktikum alpro\Modul11\Unguided1.go"
Masukkan jumlah anak kelinci: 5
Masukkan berat anak kelinci (spasi):
2.5 3.5 4.5 5.5 6.5
Berat terkecil: 2.50
Berat terbesar: 6.50
```

Deskripsi Program

Program di atas menggunakan algoritma pencarian nilai ekstrem untuk menentukan berat anak kelinci terkecil dan terbesar. Program meminta jumlah anak kelinci (n), melakukan validasi antara 1 dan 1000, dan kemudian menerima (n) data berat yang disimpan dalam array. Program menginisialisasi berat terkecil ("minBerat") dan berat terbesar ("maxBerat") dengan nilai berat pertama. Kemudian, setiap berat baru dibandingkan dengan nilai minimum dan maksimum saat ini, dan jika nilainya lebih kecil, "minBerat" diperbarui. Setelah iterasi selesai, program mencetak berat terkecil dan terbesar dalam format dua angka desimal.

2. Sourcecode

```
package main

import (
    "fmt"
)

func main() {
    var x, y int
    fmt.Print("Input (X) jumlah ikan dan (y) kapasitas
    ikan per wadah : ")
    fmt.Scan(&x, &y)

    if x <= 0 || y <= 0 || x > 1000 {
        fmt.Println("Input tidak valid!")
        return
    }

    fmt.Println("Input berat ikan :")
    beratIkan_145 := make([]float64, x)
    for i := 0; i < x; i++ {
        fmt.Scan(&beratIkan_145[i])
    }

    var totalWadah_145 []float64
    var totalBerat_145 float64
    for i := 0; i < x; i++ {
        if i%y == 0 {
            totalWadah_145 = append(totalWadah_145,
0)
        }
        totalWadah_145[len(totalWadah_145)-1] +=
beratIkan_145[i]
        totalBerat_145 += beratIkan_145[i]
    }
}
```

```

rataRata := totalBerat_145 /
float64(len(totalWadah_145))

fmt.Println("Total berat per wadah:")
for i, berat := range totalWadah_145 {
    fmt.Printf("Wadah %d: %.2f\n", i+1, berat)
}
fmt.Printf("Berat rata-rata per wadah: %.2f\n",
rataRata)
}

```

Screenshoot Output

```

Input (X) jumlah ikan dan (y) kapasitas ikan per wadah : 6 3
Input berat ikan :
5 20 5.9 3.5 3 2.1
Total berat per wadah:
Wadah 1: 30.90
Wadah 2: 8.60
Berat rata-rata per wadah: 19.75
PS D:\praktikum alpro>

```

Deskripsi Program

Program di atas menggunakan algoritma pencarian nilai ekstrem untuk menentukan jumlah ikan (x) dan kapasitas maksimal per wadah (y) untuk menghitung berat total ikan per wadah dan berat rata-rata ikan di semua wadah. setelah memasukkan jumlah ikan, kapasitas wadah, dan berat masing-masing ikan, program mendistribusikan ikan ke masing-masing wadah sesuai kapasitasnya. Setiap kali kapasitas wadah sebelumnya penuh, berat ikan dimasukkan ke dalam wadah baru. Kemudian program menghitung berat total semua ikan dan berat rata-rata per wadah, dan hasilnya adalah total berat ikan di setiap wadah dan berat rata-rata dalam dua angka desimal.

3. Sourcecode

```

package main

import (
    "fmt"
)

type arrBalita [100]float64

func hitungMinMax(arrBerat arrBalita, n int, bMin
*float64, bMax *float64) {

```

```

    *bMin = arrBerat[0]
    *bMax = arrBerat[0]

    for i := 0; i < n; i++ {
        if arrBerat[i] < *bMin {
            *bMin = arrBerat[i]
        }
        if arrBerat[i] > *bMax {
            *bMax = arrBerat[i]
        }
    }
}

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

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

    if n > 100 || n <= 0 {
        fmt.Println("Jumlah data harus antara 1 hingga
100.")
        return
    }

    var dataBerat_145 arrBalita
    for i := 0; i < n; i++ {
        fmt.Printf("input berat balita ke-%d: ", i+1)
        fmt.Scan(&dataBerat_145[i])
    }

    var bMin, bMax float64
    hitungMinMax(dataBerat_145, n, &bMin, &bMax)
    rataRata := hitungRataRata(dataBerat_145, n)

    fmt.Printf("Berat 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)
}

```

Screenshoot Output

```
PS D:\praktikum alpro> go run "d:\praktikum alpro\Modul11\Unguided3.go"
Input banyak data berat balita: 4
input berat balita ke-1: 5.3
input berat balita ke-2: 6.2
input berat balita ke-3: 4.1
input 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 D:\praktikum alpro> █
```

Deskripsi Program

Program di atas menggunakan algoritma pencarian nilai ekstrem untuk menghitung berat minimum, maksimum, dan rata-rata dari balita yang dimasukkan oleh pengguna. Pengguna diminta untuk memasukkan jumlah data balita (n) dengan validasi antara 1 dan 100, kemudian menyimpan berat masing-masing balita dalam array. Fungsi "hitungMinMax" menghitung berat minimum dan maksimum dengan membandingkan setiap elemen array, dan fungsi "hitungRataRata" menghitung berat rata-rata balita dengan menjumlahkan seluruh data dan membaginya dengan jumlah data. Hasilnya adalah berat minimum, maksimum, dan rata-rata yang ditentukan.