

**LAPORAN PRAKTIKUM  
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL 10**

**PENCARIAN NILAI EKSTRIM PADA HIMPUNAN DATA**



Oleh:

Achmad Zulvan Nur Hakim

103112400070

IF-12-01

**S1 TEKNIK INFORMATIKA  
TELKOM UNIVERSITY PURWOKERTO**

**2025**

## **I. DASAR TEORI**

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 dicek.
- 3) Apabila semua data telah dicek, maka nilai ekstrim yang dimiliki adalah valid. Algoritma rekursif terdiri dari dua komponen utama:

## II. GUIDED

Code 1:

```
package main

import "fmt"

func main() {
    var n int
    fmt.Print("Masukan jumlah tanaman : ")
    fmt.Scan(&n)

    var tinggiTanaman [500]float64
    for i := 0; i < n; i++ {
        fmt.Printf("Masukan tinggi tanaman ke-%d (cm): ", i+1)
        fmt.Scan(&tinggiTanaman[i])
    }
    min, max := tinggiTanaman[0], tinggiTanaman[0]
    for i := 1; i < n; i++ {
        if tinggiTanaman[i] < min {
            min = tinggiTanaman[i]
        }
        if tinggiTanaman[i] > max {
            max = tinggiTanaman[i]
        }
    }
    fmt.Printf("Tinggi tanaman terpendek: %.2f cm\n", min)
    fmt.Printf("Tinggi tanaman tertinggi: %.2f cm\n", max)
}
```

Output:

```
PS D:\103112400070> go run "d:\103112400070\103112400070_Modul7\Guided\1\1.go"
Masukan jumlah tanaman : 5
Masukan tinggi tanaman ke-1 (cm): 10
Masukan tinggi tanaman ke-2 (cm): 11
Masukan tinggi tanaman ke-3 (cm): 12
Masukan tinggi tanaman ke-4 (cm): 13
Masukan tinggi tanaman ke-5 (cm): 14
Tinggi tanaman terpendek: 10.00 cm
Tinggi tanaman tertinggi: 14.00 cm
```

Penjelasan:

Program ini meminta pengguna memasukkan jumlah dan tinggi beberapa tanaman, lalu menyimpannya dalam array. Setelah itu, program menghitung dan menampilkan tinggi tanaman terpendek dan tertinggi dari data yang dimasukkan.

Code 2:

```
package main

import "fmt"

func main() {
    var x, y int
    fmt.Print("Masukan jumlah buku dan jumlah buku per rak: ")
    fmt.Scan(&x, &y)
    var hargaBuku [500]float64
    fmt.Println("\nMasukan harga setiap buku (dalam ribuan Rp):")
    for i := 0; i < x; i++ {
        fmt.Scan(&hargaBuku[i])
    }

    var hargaRataRata []float64
    for i := 0; i < x; i++ {
        total := 0.0
        for j := i; j < i+y && j < x; j++ {
            total += hargaBuku[j]
        }
        hargaRataRata = append(hargaRataRata, total/float64(y))
    }
    min, max := hargaBuku[0], hargaBuku[0]
    for i := 1; i < x; i++ {
        if hargaBuku[i] < min {
            min = hargaBuku[i]
        }
        if hargaBuku[i] > max {
            max = hargaBuku[i]
        }
    }

    fmt.Printf("\nRata-rata harga per rak: ")
    for _, avg := range hargaRataRata {
        fmt.Printf("%.2f ", avg)
    }
    fmt.Printf("\nHarga termahal: %.2f\n", max)
    fmt.Printf("Harga termurah: %.2f\n", min)
}
```

Output:

```
PS D:\103112400070> go run "d:\103112400070\103112400070_Modul7\Guided\2\2.go"
Masukan jumlah buku dan jumlah buku per rak: 6 3

Masukan harga setiap buku (dalam ribuan Rp):
10000
11000
12000
13000
14000
15000

Rata-rata harga per rak: 11000.00 12000.00 13000.00 14000.00 9666.67 5000.00
Harga termahal: 15000.00
Harga termurah: 10000.00
```

Penjelasan:

Program ini membaca harga beberapa buku dan menghitung rata-rata harga buku per rak berdasarkan jumlah buku per rak. Selain itu, program juga menentukan harga buku termahal dan termurah dari seluruh data yang dimasukkan.

Code 3:

```
package main

import "fmt"

func main() {
    var n int
    fmt.Print("Masukkan jumlah siswa: ")
    fmt.Scan(&n)

    var nilaiSiswa [500]float64
    var totalNilai float64 = 0
    fmt.Println("\nMasukkan nilai ujian masing-masing siswa:")
    for i := 0; i < n; i++ {
        fmt.Scan(&nilaiSiswa[i])
        totalNilai += nilaiSiswa[i]
    }
    rataRata := totalNilai / float64(n)

    min, max := nilaiSiswa[0], nilaiSiswa[0]
    var diAtasRataRata int = 0
    for _, nilai := range nilaiSiswa[:n] {
        if nilai < min {
            min = nilai
        }
        if nilai > max {
            max = nilai
        }
        if nilai > rataRata {
            diAtasRataRata++
        }
    }
    fmt.Printf("\nNilai terendah: %.2f\n", min)
    fmt.Printf("Nilai tertinggi: %.2f\n", max)
    fmt.Printf("Rata-rata nilai: %.2f\n", rataRata)
    fmt.Printf("Jumlah siswa di atas rata-rata: %d\n", diAtasRataRata)
}
```

Output:

```
PS D:\103112400070> go run "d:\103112400070\103112400070_Modul7\Guided\3\3.go"
Masukkan jumlah siswa: 5

Masukkan nilai ujian masing-masing siswa:
85
68
74
91
59

Nilai terendah: 59.00
Nilai tertinggi: 91.00
Rata-rata nilai: 75.40
Jumlah siswa di atas rata-rata: 2
```

Penjelasan:

Program ini berfungsi untuk menghitung statistik dari nilai ujian sejumlah siswa, seperti nilai tertinggi, nilai terendah, dan rata-rata. Selain itu, program juga menentukan berapa banyak siswa yang memperoleh nilai di atas rata-rata.

### III. UNGUIDED

Code 1:

```
// Achmad Zulvan Nur Hakim 103112400070
package main

import "fmt"

func main() {
    var K int
    fmt.Print("Jumlah kelinci: ")
    fmt.Scan(&K)

    beratK := make([]float64, K)

    for i := 0; i < K; i++ {
        fmt.Printf("Berat kelinci ke-%d: ", i+1)
        fmt.Scan(&beratK[i])
    }

    min, max := beratK[0], beratK[0]
    for i := 1; i < K; i++ {
        if beratK[i] < min {
            min = beratK[i]
        }
        if beratK[i] > max {
            max = beratK[i]
        }
    }

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

Output:

```
PS D:\103112400070> go run "d:\103112400070\103112400070_Modul7\Unguided\1\1.go"
Jumlah kelinci: 5
Berat kelinci ke-1: 5.1
Berat kelinci ke-2: 4.8
Berat kelinci ke-3: 4.4
Berat kelinci ke-4: 5.3
Berat kelinci ke-5: 5.6
Berat terkecil: 4.40
Berat terbesar: 5.60
```

Penjelasan:

Program ini digunakan untuk mencatat dan menganalisis data berat beberapa kelinci. Program akan menampilkan berat kelinci terkecil dan terbesar dari seluruh data yang dimasukkan.



Code 2:

```
// Achmad Zulvan Nur Hakim 103112400070
package main

import "fmt"

func main() {
    var x, y int
    var beratIkan [1000]float64

    fmt.Print("Jumlah ikan : ")
    fmt.Scan(&x)
    fmt.Print("Kapasitas wadah : ")
    fmt.Scan(&y)

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

    jmlWadah := x / y
    if x%y != 0 {
        jmlWadah++
    }

    totalPWadah := make([]float64, jmlWadah)
    for i := 0; i < x; i++ {
        wadah := i / y
        totalPWadah[wadah] += beratIkan[i]
    }

    fmt.Print("Total berat per wadah: ")
    for i := 0; i < jmlWadah; i++ {
        fmt.Printf("%.2f ", totalPWadah[i])
    }
    fmt.Println()

    var total float64
    for i := 0; i < x; i++ {
        total += beratIkan[i]
    }
    avg := total / float64(x)

    fmt.Printf("Rata-rata berat ikan: %.2f\n", avg)
}
```

Output:

```
PS D:\103112400070> go run "d:\103112400070\103112400070_Modul7\Unguided\2\2.go"
Jumlah ikan : 6
Kapasitas wadah : 3
Masukkan berat ikan ke-1: 1.1
Masukkan berat ikan ke-2: 1.3
Masukkan berat ikan ke-3: 0.9
Masukkan berat ikan ke-4: 1.3
Masukkan berat ikan ke-5: 0.8
Masukkan berat ikan ke-6: 1.2
Total berat per wadah: 3.30 3.30
Rata-rata berat ikan: 1.10
```

Penjelasan:

Program ini digunakan untuk mengelompokkan sejumlah ikan ke dalam wadah berdasarkan kapasitas tertentu dan menghitung total berat ikan per wadah. Selain itu, program juga menghitung rata-rata berat seluruh ikan yang dimasukkan.

Code 3:

```
// Achmad Zulvan Nur Hakim 103112400070
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 arrBerat[i] < *bMin {
            *bMin = arrBerat[i]
        }
        if arrBerat[i] > *bMax {
            *bMax = arrBerat[i]
        }
    }
}

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

func main() {
    var data arrBalita
    var n int
    var min, max float64

    fmt.Print("Banyak data berat balita: ")
    fmt.Scan(&n)

    for i := 0; i < n; i++ {
        fmt.Printf("Berat balita ke-%d: ", i+1)
        fmt.Scan(&data[i])
    }
}
```

```

hitungMinMax(data, n, &min, &max)
avg := rerata(data, n)

fmt.Printf("Berat minimum: %.2f kg\n", min)
fmt.Printf("Berat maksimum: %.2f kg\n", max)
fmt.Printf("Rerata berat : %.2f kg\n", avg)
}

```

Output:

```

PS D:\103112400070> go run "d:\103112400070\103112400070_Modul7\Unguided\3\3.go"
Banyak data berat balita: 6
Berat balita ke-1: 5.8
Berat balita ke-2: 6.2
Berat balita ke-3: 6.8
Berat balita ke-4: 5.5
Berat balita ke-5: 5.2
Berat balita ke-6: 6.5
Berat minimum: 5.20 kg
Berat maksimum: 6.80 kg
Rerata berat : 6.00 kg

```

Penjelasan:

Program ini berfungsi untuk mengolah data berat badan balita, termasuk mencari berat minimum dan maksimum serta menghitung rata-ratanya. Dengan menggunakan array dan pointer, program memproses data yang dimasukkan pengguna secara efisien dan terstruktur.

#### **IV. KESIMPULAN**

## **V. REFERENSI**

*MODUL 10. PENCARIAN NILAI EKSTRIM PADA HIMPUNAN DATA*