

**LAPORAN PRAKTIKUM  
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL 7**

**PENCARIAN NILAI EKSTRIM PADA HIMPUNAN DATA**



Oleh:

**PRATAMA BINTANG DANISWARA**

103112400051

IF-12-01

**S1 TEKNIK INFORMATIKA  
TELKOM UNIVERSITY PURWOKERTO**

**2025**

## **I. DASAR TEORI**

### **1. Konsep Pencarian Nilai Min/Max**

Pencarian merupakan sebuah kegiatan yang umum dilakukan dalam kehidupan sehari-hari. Misalnya dalam kehidupan nyata, proses pencarian bisa sangat bervariasi, contohnya pencarian file dalam folder komputer, pencarian kata dalam dokumen, pencarian buku di rak, dan sebagainya. Di modul ini, kita akan mempelajari salah satu algoritma pencarian nilai terkecil atau terbesar dalam kumpulan data, yang dikenal dengan pencarian nilai ekstrem.

Algoritma ini cukup sederhana. Karena data perlu diproses satu per satu, maka nilai pada indeks tertentu akan dibandingkan dengan nilai lain dalam data, dan nilai terbesar atau terkecil akan disimpan sementara hingga algoritma selesai berjalan. Nilai akhir yang disimpan adalah nilai maksimum atau minimum yang dicari.

### **2. Pencarian Nilai Ekstrem dalam Array Bertipe Data Sederhana**

Misalnya telah didefinisikan sebuah array bertipe integer dengan kapasitas 2023, dan array tersebut diisi dengan N buah bilangan bulat. Maka pencarian nilai terkecil atau terbesar bisa dilakukan terhadap array tersebut.

### **3. Pencarian Nilai Ekstrem dalam Array Bertipe Data Terstruktur**

Dalam kasus yang lebih kompleks, pencarian nilai ekstrem tetap bisa dilakukan. Misalnya mencari mahasiswa dengan nilai tertinggi, mencari lagu dengan durasi terlama, atau mencari pembalap dengan catatan waktu tercepat, dan sebagainya.

## II. Guided Contoh1

```
package main

import (
    "fmt"
)

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

    var tinggiTanaman [500] float64
    for i := 0; i < n; i++ {
        fmt.Printf("Masukkan 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("\nTinggi tanaman tertinggi: %.2f cm\n", max)
    fmt.Printf("Tinggi tanaman terpendek: %.2f cm\n", min)
}
```

### Output

```
PS D:\Coding\ALPRO\103112400051_MODUL7> go
ALPRO\103112400051_MODUL7\103112400051_Guide
Masukkan jumlah tanaman: 3
Masukkan tinggi tanaman ke-1 (cm): 4
Masukkan tinggi tanaman ke-2 (cm): 5
Masukkan tinggi tanaman ke-3 (cm): 6

Tinggi tanaman tertinggi: 5.00 cm
Tinggi tanaman terpendek: 4.00 cm
PS D:\Coding\ALPRO\103112400051_MODUL7> |
```

Deskripsi : Program ini digunakan untuk membaca data tinggi beberapa tanaman dari input pengguna, lalu menentukan dan menampilkan tinggi tanaman tertinggi dan terpendek. Pengguna diminta memasukkan jumlah tanaman, kemudian memasukkan tinggi masing-masing tanaman satu per satu. Setelah semua data dimasukkan, program menghitung nilai maksimum dan minimum dari tinggi tanaman yang diberikan, lalu mencetak hasilnya ke layar.

## Contoh2

```
package main

import (
    "fmt"
)

func main() {
    var x, y int
    fmt.Print("Masukkan jumlah buku dan jumlah buku per rak: ")
    fmt.Scan(&x, &y)

    var hargaBuku [500]float64
    fmt.Println("\nMasukkan 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 += y {
        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 _, harga := range hargaBuku[:x] {
        if harga < min {
            min = harga
        }
        if harga > max {
            max = harga
        }
    }

    fmt.Printf("\nRata-rata harga per rak: ")
```

```

    for _, avg := range hargaRataRata {
        fmt.Printf("%.2f ", avg)
    }
    fmt.Printf("\nHarga termahal: %.2f Rp\n", max)
    fmt.Printf("Harga termurah: %.2f Rp\n", min)
}

```

### Output :

```

PS D:\Coding\ALPRO\103112400051_MODUL7> go run "d:
Masukkan jumlah buku dan jumlah buku per rak: 3 2

Masukkan harga setiap buku (dalam ribuan Rp):
10000
20000
30000

Rata-rata harga per rak: 15000.00 15000.00
Harga termahal: 30000.00 Rp
Harga termurah: 10000.00 Rp
PS D:\Coding\ALPRO\103112400051_MODUL7>

```

**Deskripsi :** Program ini membantu pengguna menghitung rata-rata harga buku di setiap rak, serta mencari buku dengan harga paling mahal dan paling murah. Pengguna cukup memasukkan jumlah buku, jumlah buku per rak, dan harga masing-masing buku. Setelah itu, program akan menampilkan rata-rata harga per rak, lalu menunjukkan harga buku tertinggi dan terendah.

### Contoh3

```

package main

import (
    "fmt"
)

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

    var nilaiSiswa [200]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: %.0f\n", min)
    fmt.Printf("Nilai tertinggi: %.0f\n", max)
    fmt.Printf("Rata-rata kelas: %.2f\n", rataRata)
    fmt.Printf("Jumlah siswa di atas rata-rata: %d\n",
diAtasRataRata)

}

```

### Output :

```

PS D:\Coding\ALPRO\103112400051_MODUL7> go
Masukkan jumlah siswa: 3

Masukkan nilai ujian masing-masing siswa:
80
90
100

Nilai terendah: 80
Nilai tertinggi: 100
Rata-rata kelas: 90.00
Jumlah siswa di atas rata-rata: 1
PS D:\Coding\ALPRO\103112400051_MODUL7>

```

**Deskripsi :** Program ini membantu pengguna mengolah data nilai ujian siswa dengan mudah. Setelah pengguna memasukkan jumlah siswa dan nilai masing-masing, program akan menghitung dan menampilkan nilai tertinggi, terendah, rata-rata kelas, serta jumlah siswa yang nilainya di atas rata-rata. Hasilnya memberikan gambaran singkat tentang performa keseluruhan siswa dalam satu kelas.

### III. Unguided

#### Soal1

```
package main

import "fmt"

func main() {
    var n int
    var berat [1000]float64

    fmt.Print("Jumlah anak kelinci: ")
    fmt.Scan(&n)

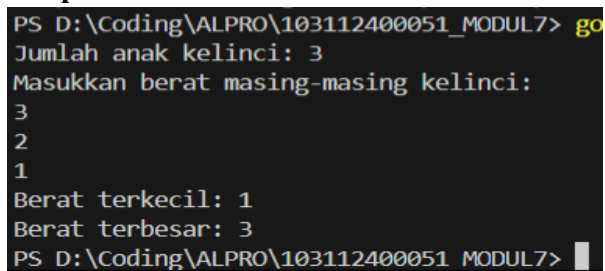
    fmt.Println("Masukkan berat masing-masing kelinci:")
    for i := 0; i < n; i++ {
        fmt.Scan(&berat[i])
    }

    min := berat[0]
    max := berat[0]

    for i := 0; i < n; i++ {
        if berat[i] < min {
            min = berat[i]
        }
        if berat[i] > max {
            max = berat[i]
        }
    }

    fmt.Println("Berat terkecil:", min)
    fmt.Println("Berat terbesar:", max)
}
```

#### Output :



```
PS D:\Coding\ALPRO\103112400051_MODUL7> go
Jumlah anak kelinci: 3
Masukkan berat masing-masing kelinci:
3
2
1
Berat terkecil: 1
Berat terbesar: 3
PS D:\Coding\ALPRO\103112400051 MODUL7>
```

**Deskripsi :** Program ini digunakan untuk mengetahui berat terkecil dan terbesar dari sejumlah anak kelinci. Pengguna diminta memasukkan jumlah kelinci, lalu berat masing-masing kelinci satu per satu. Setelah semua data dimasukkan, program membandingkan setiap berat untuk mencari nilai paling kecil dan paling besar, lalu menampilkannya sebagai hasil akhir. Program ini membantu memudahkan identifikasi variasi berat dalam kelompok kelinci.

## Soal2

```
package main

import "fmt"

func main() {
    var a, b int
    var data [1000]float64
    var hasil [1000]float64

    fmt.Print("Masukkan jumlah ikan dan wadah: ")
    fmt.Scan(&a, &b)

    fmt.Println("Berat ikannya:")
    for i := 0; i < a; i++ {
        fmt.Scan(&data[i])
    }

    for i := 0; i < a; i++ {
        pos := i % b
        hasil[pos] = hasil[pos] + data[i]
    }

    fmt.Println("Isi wadah:")
    for i := 0; i < b; i++ {
        fmt.Print(hasil[i], " ")
    }
    fmt.Println()

    var jum float64 = 0
    for i := 0; i < b; i++ {
        jum = jum + hasil[i]
    }
    rata := jum / float64(b)
    fmt.Println("Rata-rata:", rata)
}
```



### Output :

```
PS D:\Coding\ALPRO\103112400051_MODUL7> go
Masukkan jumlah ikan dan wadah: 3 3
Berat ikannya:
3 2 1
Isi wadah:
3 2 1
Rata-rata: 2
PS D:\Coding\ALPRO\103112400051_MODUL7> █
```

**Deskripsi :** Program ini membagi ikan ke beberapa wadah secara bergiliran berdasarkan urutan, lalu menampilkan isi masing-masing wadah dan menghitung rata-ratanya. Pengguna cukup memasukkan jumlah ikan, jumlah wadah, dan berat tiap ikan.

### Soal3

```
package main

import "fmt"

func hitungMinMax(arr []float64, n int, bMin *float64, bMax
*float64) {
    min := arr[0]
    max := arr[0]
    for i := 1; i < n; i++ {
        if arr[i] < min {
            min = arr[i]
        }
        if arr[i] > max {
            max = arr[i]
        }
    }
    *bMin = min
    *bMax = max
}

func rerata(arr []float64, n int) float64 {
    total := 0.0
    for i := 0; i < n; i++ {
        total += arr[i]
    }
    return total / float64(n)
}

func main() {
    var n int
```

```

fmt.Print("Masukan banyak data berat balita : ")
fmt.Scanln(&n)

var berat [100]float64

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

var min, max float64
hitungMinMax(berat[:], n, &min, &max)
rata := rerata(berat[:], n)

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", rata)
}

```

#### Output :

```

PS D:\Coding\ALPRO\103112400051_MODUL7> go
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
PS D:\Coding\ALPRO\103112400051_MODUL7>

```

**Deskripsi :** Program ini digunakan untuk mencatat data berat badan beberapa balita yang dimasukkan oleh pengguna, kemudian menyimpannya ke dalam array. Setelah semua data dimasukkan, program akan memproses data tersebut menggunakan subprogram untuk menentukan berat balita paling ringan, paling berat, dan menghitung rata-rata dari seluruh data yang telah dimasukkan.

#### **IV. Kesimpulan**

## **V. Referensi**

Modul 7 PENCARIAN NILAI EKSTRIM PADA HIMPUNAN DATA.  
Praktikum Algoritma Pemrograman 2