

**LAPORAN PRAKTIKUM**  
**ALGORITMA DAN PEMROGRAMAN 2**  
**MODUL 10**  
**PENCARIAN NILAI EKSTRIM PADA HIMPUNAN DATA**



Oleh:

ABID FADHILAH MUSTOFA

103112400046

12 IF 01

**S1 TEKNIK INFORMATIKA**  
**TELKOM UNIVERSITY PURWOKERTO**

**2025**

## **I. DASAR TEORI**

### **10.1 Ide Pencarian Nilai Max/Min**

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.

### **10.2 Pencarian Nilai Ekstrim pada Array Bertipe Data Dasar**

Misalnya terdefinisi sebuah array of integer dengan kapasitas 2023, dan array berisi sejumlah N bilangan bulat, kemudian pencarian nilai terkecil dilakukan pada array tersebut.

### **10.3 Pencarian Nilai Ekstrim pada Array Bertipe Data Terstruktur**

Pada kasus yang lebih kompleks pencarian ekstrim dapat juga dilakukan, misalnya mencari data mahasiswa dengan nilai terbesar, mencari lagu dengan durasi terlama, mencari pembalap yang memiliki catatan waktu balap tercepat, dan sebagainya.

## II. GUIDED

Source Code + Screenshot hasil program beserta penjelasan

Contoh 1

```
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)
}
```

### Screenshots Output

```
PS C:\Users\HUAWEI\OneDrive\Dokumen\ALPRO 2\103112400046_MODUL7> go run "c:
Masukkan jumlah tanaman: 3
Masukkan tinggi tanaman ke-1 (cm): 5
Masukkan tinggi tanaman ke-2 (cm): 6
Masukkan tinggi tanaman ke-3 (cm): 7

Tinggi tanaman tertinggi: 7.00 cm
Tinggi tanaman terpendek: 5.00 cm
PS C:\Users\HUAWEI\OneDrive\Dokumen\ALPRO 2\103112400046_MODUL7> █
```

// Foto hasil dari menjalankan code

Deskripsi: Program di atas digunakan untuk mencatat dan menganalisis data tinggi sejumlah tanaman. Pengguna diminta memasukkan jumlah tanaman dan tinggi masing-masing tanaman dalam satuan sentimeter. Data tinggi tersebut disimpan dalam array, lalu program mencari nilai tertinggi dan terendah dari seluruh data yang dimasukkan. Hasil akhirnya adalah tampilan tinggi tanaman tertinggi dan terpendek dalam format dua angka desimal.

## Contoh 2

```
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)
}
```

### Screenshots Output

```
PS C:\Users\HUAWEI\OneDrive\Dokumen\ALPRO 2\103112400046_MODUL7> go run "c:\Us
Masukkan jumlah buku dan jumlah buku per rak: 4 2

Masukkan harga setiap buku (dalam ribuan Rp):
Harga buku ke-1: 10000
Harga buku ke-2: 25000
Harga buku ke-3: 50000
Harga buku ke-4: 55000

Rata-rata harga per rak:
Rak ke-1: 17500.00 ribu rupiah
Rak ke-2: 52500.00 ribu rupiah
PS C:\Users\HUAWEI\OneDrive\Dokumen\ALPRO 2\103112400046_MODUL7> █
```

// Foto hasil dari menjalankan code

Deskripsi: Program di atas digunakan untuk menghitung rata-rata harga buku per rak dan menemukan harga buku termahal serta termurah dari sekumpulan buku. Pengguna diminta memasukkan jumlah buku (x) dan jumlah buku per rak (y). Kemudian, program akan menerima input harga masing-masing buku dan membagi buku-buku tersebut ke dalam rak sesuai kapasitas yang ditentukan. Program menghitung rata-rata harga per rak dan menyimpannya dalam array `hargaRataRata`. Selain itu, program juga mencari harga buku termahal dan termurah dengan melakukan perbandingan terhadap seluruh harga buku. Hasil akhir yang ditampilkan adalah rata-rata harga per rak, harga buku termahal, dan harga buku termurah, semuanya dalam satuan ribuan Rupiah dengan dua angka desimal.

### Contoh 3

```
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)
}
```

### Screenshots Output

```
PS C:\Users\HUAWEI\OneDrive\Dokumen\ALPRO 2\103112400046_MODUL7> go run "c
Masukkan jumlah siswa: 5

Masukkan nilai ujian masing-masing siswa:
Nilai siswa ke-1: 78
Nilai siswa ke-2: 84
Nilai siswa ke-3: 90
Nilai siswa ke-4: 67
Nilai siswa ke-5: 72

Nilai terendah: 67
Nilai tertinggi: 90
Rata-rata kelas: 78.20
Jumlah siswa di atas rata-rata: 2
PS C:\Users\HUAWEI\OneDrive\Dokumen\ALPRO 2\103112400046_MODUL7> █
```

// Foto hasil dari menjalankan code

Deskripsi: Program di atas digunakan untuk menghitung nilai terendah, tertinggi, rata-rata, dan jumlah siswa yang memiliki nilai di atas rata-rata dari sejumlah siswa yang mengikuti ujian. Pengguna diminta untuk memasukkan jumlah siswa dan nilai ujian masing-masing siswa, yang kemudian disimpan dalam array. Program menghitung total nilai semua siswa, lalu menggunakan total tersebut untuk menghitung rata-rata nilai kelas. Selanjutnya, program mencari nilai terendah dan tertinggi serta menghitung berapa banyak siswa yang memiliki nilai di atas rata-rata. Hasil akhir yang ditampilkan adalah nilai terendah, nilai tertinggi, rata-rata nilai kelas, dan jumlah siswa dengan nilai di atas rata-rata.



### III. UNGUIDED

Source Code + Screenshot hasil program beserta penjelasan

Soal 1

```
//ABID FDAHILAH M
//103112400046
package main

import (
    "fmt"
)

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

    var berat [1000]float64
    min, max := 0.0, 0.0

    for i := 0; i < n; i++ {
        fmt.Printf("Berat ke-%d (kg): ", i+1)
        fmt.Scan(&berat[i])
        if i == 0 || berat[i] < min {
            min = berat[i]
        }
        if berat[i] > max {
            max = berat[i]
        }
    }

    fmt.Printf("\nTerkecil: %.2f kg\nTerbesar: %.2f kg\n", min, max)
}
```

### Screenshots Output

```
PS C:\Users\HUAWEI\OneDrive\Dokumen\ALPRO 2\103112400046_MODUL7> go run "c:\Users\HUAWEI\OneDrive\Dokumen\ALPRO 2\103112400046_MODUL7\main.go"
Jumlah anak kelinci: 3
Berat ke-1 (kg): 39
Berat ke-2 (kg): 23
Berat ke-3 (kg): 21

Terkecil: 21.00 kg
Terbesar: 39.00 kg
PS C:\Users\HUAWEI\OneDrive\Dokumen\ALPRO 2\103112400046_MODUL7>
```

// Foto hasil dari menjalankan code

Deskripsi: Program di atas berfungsi untuk mencari dan menampilkan berat terkecil dan terbesar dari sejumlah anak kelinci. Pengguna diminta memasukkan jumlah anak kelinci dan berat masing-masing, lalu program menyimpan data tersebut dalam array. Setelah itu, program membandingkan semua nilai dalam array untuk menentukan berat minimum dan maksimum, yang kemudian ditampilkan ke layar.

## Soal 2

```
//ABID FADHILAH M
//103112400046
package main

import (
    "fmt"
)

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

    var beratIkan [1000]float64
    fmt.Println("\nMasukkan berat
setiap ikan (dalam kg):")
    for i := 0; i < x; i++ {
        fmt.Printf("Berat ikan ke-%d: ",
i+1)
        fmt.Scan(&beratIkan[i])
    }

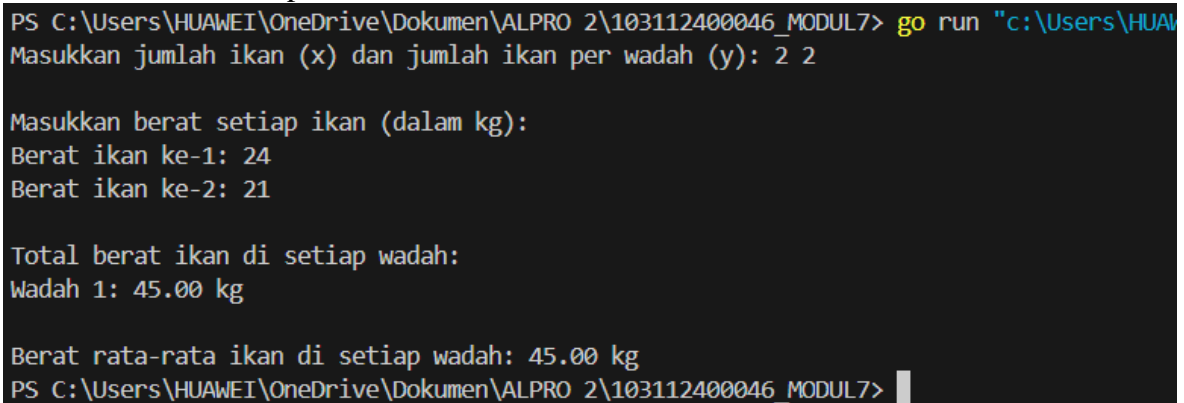
    var totalBeratWadah []float64
    for i := 0; i < x; i += y {
        total := 0.0
        for j := i; j < i+y && j < x; j++ {
            total += beratIkan[j]
        }
        totalBeratWadah =
append(totalBeratWadah, total)
    }

    fmt.Println("\nTotal berat ikan di
setiap wadah:")
    for i, total := range totalBeratWadah
{
        fmt.Printf("Wadah %d: %.2f kg\n",
i+1, total)
    }

    totalBerat := 0.0
    for _, berat := range
totalBeratWadah {
        totalBerat += berat
    }
}
```

```
totalBerat += berat
}
rataRata := totalBerat / float64(len(totalBeratWadah))
fmt.Printf("\nBerat rata-rata ikan di setiap wadah: %.2f kg\n", rataRata)
}
```

#### Screenshots Output



```
PS C:\Users\HUAWEI\OneDrive\Dokumen\ALPRO 2\103112400046_MODUL7> go run "c:\Users\HUAWEI\OneDrive\Dokumen\ALPRO 2\103112400046_MODUL7\main.go"
Masukkan jumlah ikan (x) dan jumlah ikan per wadah (y): 2 2

Masukkan berat setiap ikan (dalam kg):
Berat ikan ke-1: 24
Berat ikan ke-2: 21

Total berat ikan di setiap wadah:
Wadah 1: 45.00 kg

Berat rata-rata ikan di setiap wadah: 45.00 kg
PS C:\Users\HUAWEI\OneDrive\Dokumen\ALPRO 2\103112400046_MODUL7>
```

// Foto hasil dari menjalankan code

Deskripsi: Program di atas digunakan untuk menghitung total dan rata-rata berat ikan dalam beberapa wadah. Pengguna memasukkan jumlah ikan (`x`) dan kapasitas jumlah ikan per wadah (`y`), lalu memasukkan berat masing-masing ikan. Program membagi ikan ke dalam beberapa wadah sesuai kapasitas yang ditentukan, menghitung total berat per wadah, dan akhirnya menampilkan total berat masing-masing wadah beserta rata-rata berat per wadah. Program juga memvalidasi input agar jumlah ikan tidak melebihi 1000 dan kapasitas wadah lebih dari 0.

### Soal 3

```
//ABID FADHILAH M
//103112400046
package main

import (
    "fmt"
)

func main() {
    var n int
    var berat [100]float64
    var bMin, bMax, total float64

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

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

    rataRata := total / float64(n)

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

```
PS C:\Users\HUAWEI\OneDrive\Dokumen\ALPRO 2\103112400046_MODUL7> go run "c:\Users\HUAWEI\On
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 C:\Users\HUAWEI\OneDrive\Dokumen\ALPRO 2\103112400046_MODUL7>
```

### Screenshots Output

// Foto hasil dari menjalankan code

Deskripsi: Program di atas digunakan untuk menghitung berat minimum, maksimum, dan rata-rata dari sejumlah balita. Data berat disimpan dalam array bertipe `arrBalita`, dan pengguna diminta untuk memasukkan jumlah serta berat masing-masing balita. Fungsi `hitungMinMax` digunakan untuk menentukan berat terkecil dan terbesar, sedangkan fungsi `rerata` menghitung rata-rata berat dari seluruh data. Hasil akhirnya ditampilkan dalam satuan kilogram dengan dua angka di belakang koma.

#### **IV. KESIMPULAN**

## **V. REFERENSI**

Modul Praktikum Algoritma dan Pemrograman 2. (2025). *Modul 10: Pencarian Nilai Ekstrem Pada Himpunan Data*. Fakultas Informatika, Telkom University.