

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



Oleh:

MUHAMMAD FAUZAN

103112400064

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"
)

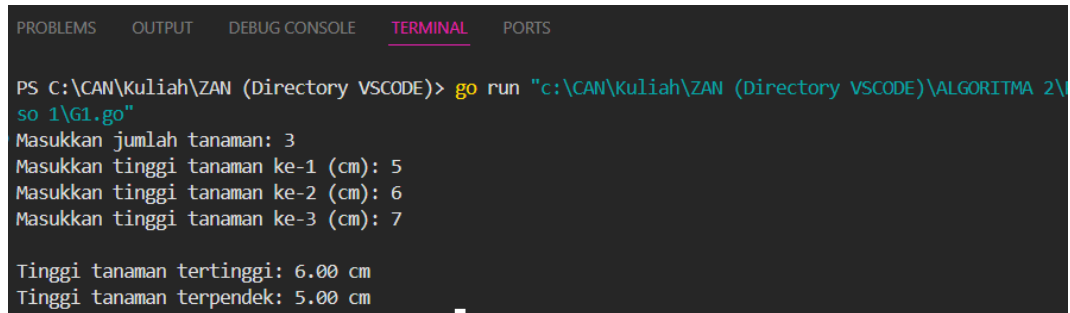
func main() {
    var n int
    fmt.Print("Masukkan jumlah tanaman: ")
    fmt.Scan(&n)

    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



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\CAN\Kuliah\ZAN (Directory VSCODE)> go run "c:\CAN\Kuliah\ZAN (Directory VSCODE)\ALGORITMA 2\so 1\G1.go"
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: 6.00 cm
Tinggi tanaman terpendek: 5.00 cm
```

// 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])
    }

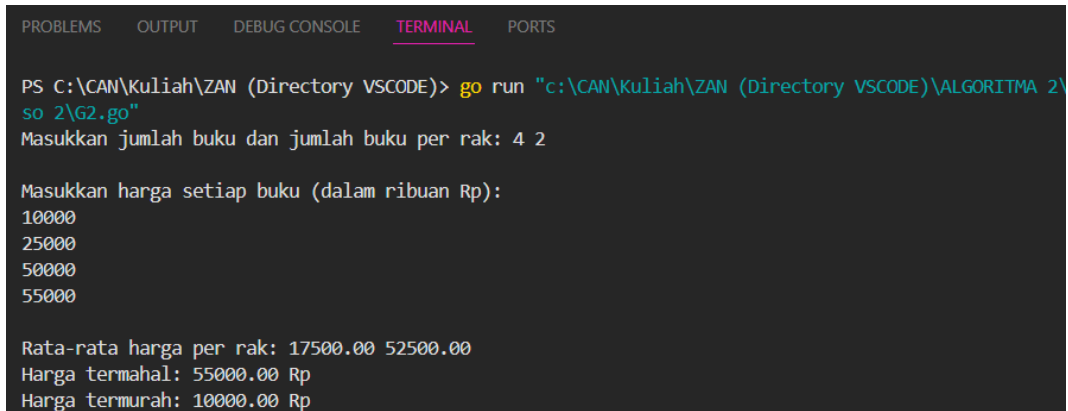
    var hargaRataRata []float64
    for i := 0; i < x; i += y {
        total := 0.0
        for j := i; j < i+y && j < x; j++ {
            total += hargaBuku[j]
        }
        hargaRataRata = append(hargaRataRata, total/float64(y))
    }

    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



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\CAN\Kuliah\ZAN (Directory VSCODE)> go run "c:\CAN\Kuliah\ZAN (Directory VSCODE)\ALGORITMA 2\so 2\G2.go"
Masukkan jumlah buku dan jumlah buku per rak: 4 2

Masukkan harga setiap buku (dalam ribuan Rp):
10000
25000
50000
55000

Rata-rata harga per rak: 17500.00 52500.00
Harga termahal: 55000.00 Rp
Harga termurah: 10000.00 Rp
```

// 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)

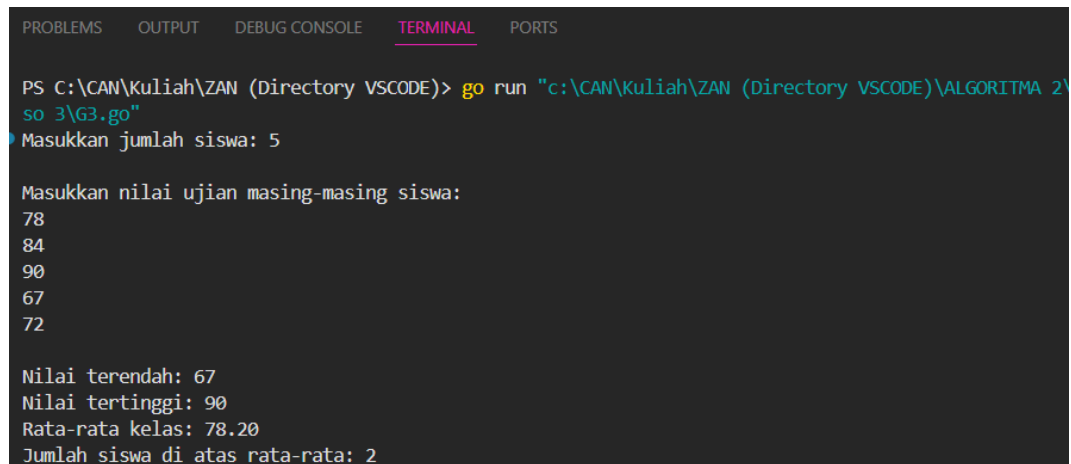
    var nilaiSiswa [200]float64
    var totalNilai float64 = 0

    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[:] {
        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



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\CAN\Kuliah\ZAN (Directory VSCODE)> go run "c:\CAN\Kuliah\ZAN (Directory VSCODE)\ALGORITMA 2\so 3\G3.go"
Masukkan jumlah siswa: 5

Masukkan nilai ujian masing-masing siswa:
78
84
90
67
72

Nilai terendah: 67
Nilai tertinggi: 90
Rata-rata kelas: 78.20
Jumlah siswa di atas rata-rata: 2
```

// 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

```
// MUHAMMAD FAUZAN
// 103112400064
package main

import "fmt"

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

    fmt.Print("Masukkan jumlah anak kelinci (maks 1000): ")
    fmt.Scan(&n)

    if n < 1 || n > 1000 {
        fmt.Println("Jumlah tidak valid! Harus antara 1 hingga 1000.")
        return
    }

    fmt.Println("\nMasukkan berat masing-masing anak kelinci:")
    for i := 0; i < n; i++ {
        fmt.Printf("Berat anak kelinci ke-%d: ", i+1)
        fmt.Scan(&berat[i])
    }

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

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

Screenshots Output

```
PS C:\CAN\Kuliah\ZAN (Directory VSCODE)> go run "c:\CAN\Kuliah\ZAN (Directory VSCODE)\ALGORITMA 2\1\103112400064_Unguided1.go"
Masukkan jumlah anak kelinci (maks 1000): 3

Masukkan berat masing-masing anak kelinci:
Berat anak kelinci ke-1: 39
Berat anak kelinci ke-2: 23
Berat anak kelinci ke-3: 21

Berat terkecil: 21.00 kg
Berat terbesar: 39.00 kg
```

// 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

```
//MUHAMMAD FAUZAN
//1013112400064
package main

import (
    "fmt"
)

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

    fmt.Print("Masukkan jumlah ikan dan jumlah ikan per wadah (1 dan 2):")
    fmt.Scan(&x, &y)

    if x <= 0 || x > 1000 || y <= 0 {
        fmt.Println("Nilai x harus 1–1000 dan y > 0")
        return
    }

    fmt.Println("Masukkan berat ikan:")
    for i := 0; i < x; i++ {
        fmt.Scan(&berat[i])
    }

    jumlahWadah := (x + y - 1) / y
    totalWadah := make([]float64, jumlahWadah)

    for idx := 0; idx < jumlahWadah; idx++ {
        start := idx * y
        end := start + y
        if end > x {
            end = x
        }
        for i := start; i < end; i++ {
            totalWadah[idx] += berat[i]
        }
    }

    fmt.Println("Total berat tiap wadah:")
    totalBeratSemua := 0.0
```

```

    for _, total := range totalWadah {
        fmt.Printf("%.2f ", total)
        totalBeratSemua += total
    }
    fmt.Println()

    rataRata := totalBeratSemua / float64(jumlahWadah)
    fmt.Printf("Berat rata-rata per wadah: %.2f\n", rataRata)
}

```

Screenshots Output

```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\CAN\Kuliah\ZAN (Directory VSCODE)> go run "c:\CAN\Kuliah\ZAN (Directory VSCODE)\ALGORITMA 2\
2\103112400064_Unguided2.go"
• Masukkan jumlah ikan dan jumlah ikan per wadah (1 dan 2): 2 2
Masukkan berat ikan:
24
21
Total berat tiap wadah:
45.00
Berat rata-rata per wadah: 45.00

```

// 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

```
// MUHAMMAD FAUZAN
// 103112400064
package main

import "fmt"

type arrBalita [100]float64

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

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

func main() {
    var n int
    var data arrBalita

    fmt.Print("Masukkan jumlah balita: ")
    fmt.Scan(&n)

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

    var min, max float64
```

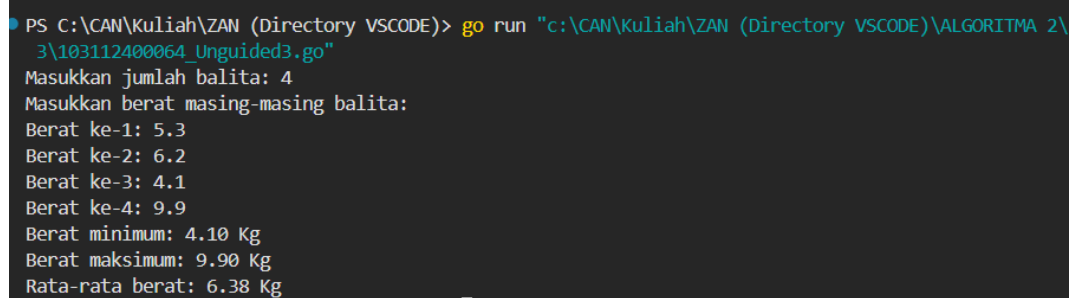
```

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

        fmt.Printf("Berat minimum: %.2f Kg\n", min)
        fmt.Printf("Berat maksimum: %.2f Kg\n", max)
        fmt.Printf("Rata-rata berat: %.2f Kg\n", rata)
    }

```

Screenshots Output



```

PS C:\CAN\Kuliah\ZAN (Directory VSCODE)> go run "c:\CAN\Kuliah\ZAN (Directory VSCODE)\ALGORITMA 2\
3\103112400064_Unguided3.go"
Masukkan jumlah balita: 4
Masukkan berat masing-masing balita:
Berat ke-1: 5.3
Berat ke-2: 6.2
Berat ke-3: 4.1
Berat ke-4: 9.9
Berat minimum: 4.10 Kg
Berat maksimum: 9.90 Kg
Rata-rata berat: 6.38 Kg

```

// 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 Ekstrim Pada Himpunan Data*. Fakultas Informatika, Telkom University.