

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2
MODUL 7
PENCARIAN NILAI EKSTRIM PADA HIMPUNAN DATA**



Oleh:

Muhamad Faza Fahri aziz

10311240072

IF-12-01

**S1 TEKNIK INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO**

2025

I. DASAR TEORI

Pencarian merupakan proses yang umum dilakukan dalam kehidupan sehari-hari. Pada modul ini, kita mempelajari salah satu jenis pencarian, yaitu pencarian nilai terkecil atau terbesar dari sekumpulan data, yang dikenal sebagai pencarian nilai ekstrem.

Ide algoritmanya sangat sederhana. Karena data harus diproses secara berurutan (sekuensial), maka nilai atau indeks dari data maksimum (atau minimum) yang telah diproses disimpan sementara untuk dibandingkan dengan data berikutnya. Nilai yang tetap bertahan hingga akhir proses pencarian adalah nilai ekstrem yang diinginkan.

Secara umum, algoritma pencarian nilai ekstrem dilakukan sebagai berikut:

1. Anggap data pertama sebagai nilai ekstrem awal.
2. Bandingkan nilai ekstrem dengan data kedua hingga data terakhir:
 - Jika nilai ekstrem tidak memenuhi syarat (misalnya ada data yang lebih besar), maka perbarui nilai ekstrem dengan data tersebut.
3. Setelah seluruh data diperiksa, nilai ekstrem yang tersisa adalah hasil yang valid.

Sebagai ilustrasi, berikut ini ditampilkan pseudocode dan implementasinya dalam bahasa Go untuk pencarian nilai maksimum.

II. GUIDED

Guided 1

```
// 103112400072
```

```
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 C:\Users\fazaf\OneDrive\ドキュメント\Tugas kuliah\SEMESTER 2\ALPRO 2\103112400072_Modul 7> go run 2\ALPRO 2\103112400072_Modul 7\GUIDED1\G1.go"
Masukkan jumlah tanaman: 3
Masukkan tinggi tanaman ke-1 (cm): 2
Masukkan tinggi tanaman ke-2 (cm): 3
Masukkan tinggi tanaman ke-3 (cm): 4

Tinggi tanaman tertinggi: 3.00 cm
Tinggi tanaman terpendek: 2.00 cm
PS C:\Users\fazaf\OneDrive\ドキュメント\Tugas kuliah\SEMESTER 2\ALPRO 2\103112400072_Modul 7> |
```

Deskripsi Program :

Program ini berfungsi untuk membaca dan menganalisis data tinggi dari sejumlah tanaman. Pengguna diminta untuk memasukkan jumlah tanaman dan tinggi masing-masing tanaman dalam satuan sentimeter. Seluruh data tinggi tanaman tersebut akan disimpan dalam sebuah array. Selanjutnya, program akan mencari dan menentukan nilai tertinggi serta terpendek dari data yang telah dimasukkan. Hasil akhir yang ditampilkan oleh program adalah tinggi tanaman tertinggi dan terpendek.

Guided 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)
}

```

Output

```
PS C:\Users\fazaf\OneDrive\ドキュメント\Tugas kuliah\SEMESTER 2\ALPRO 2\103112400072_Modul 7
7\GUIDED2\G2.go"
Masukkan jumlah buku dan jumlah buku per rak: 2 3

Masukkan harga setiap buku (dalam ribuan Rp):
200 65000

Rata-rata harga per rak: 21733.33
Harga termahal: 65000.00 Rp
Harga termurah: 200.00 Rp
PS C:\Users\fazaf\OneDrive\ドキュメント\Tugas kuliah\SEMESTER 2\ALPRO 2\103112400072_Modul 7
```

Deskripsi Program :

Program ini digunakan untuk menghitung dan menganalisis harga buku berdasarkan rak tempat penyimpanannya. Pengguna diminta untuk memasukkan jumlah total buku serta kapasitas buku per rak. Setelah itu, pengguna juga diminta menginput harga setiap buku, yang kemudian disimpan dalam sebuah array. Program akan menghitung rata-rata harga buku di setiap rak dan menyimpannya dalam sebuah slice. Selain itu, program juga akan mencari harga buku tertinggi dan terendah dari seluruh data yang telah dimasukkan. Hasil akhirnya adalah daftar rata-rata harga per rak, serta informasi mengenai harga buku paling mahal dan paling murah.

Guided 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)
}

```

Output

```
PS C:\Users\fazaf\OneDrive\ドキュメント\Tugas kuliah\SEMESTER 2\ALPRO 2\10311240072_Modul 7> go t
7\GUIDED3\G3.go"
Masukkan jumlah siswa: 2

Masukkan nilai ujian masing-masing siswa:
90
93

Nilai terendah: 90
Nilai tertinggi: 93
Rata-rata kelas: 91.50
Jumlah siswa di atas rata-rata: 1
PS C:\Users\fazaf\OneDrive\ドキュメント\Tugas kuliah\SEMESTER 2\ALPRO 2\10311240072_Modul 7> |
```

Deskripsi Program :

Program ini berfungsi untuk mengolah data nilai ujian siswa. Pengguna diminta untuk memasukkan jumlah siswa dan nilai ujian dari masing-masing siswa, yang kemudian disimpan ke dalam array. Setelah itu, program menghitung total nilai guna menentukan rata-rata nilai kelas. Selain menghitung rata-rata, program juga mencari nilai ujian tertinggi dan terendah, serta menghitung jumlah siswa yang mendapatkan nilai di atas rata-rata. Hasil akhir yang ditampilkan meliputi nilai terendah, nilai tertinggi, rata-rata kelas, dan jumlah siswa yang nilainya melampaui rata-rata.

III. UNGUIDED

Unguided 1


```
package main
```

```
import (  
    "fmt"  
)
```

```
func main() {
```

```
    var beratKelinci [1000]float64
```

```
    var n int  
    fmt.Scan(&n)
```

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

```
    for i := 0; i < n; i++ {  
        fmt.Scan(&beratKelinci[i])  
    }
```

```
    beratTerkecil := beratKelinci[0]  
    beratTerbesar := beratKelinci[0]
```

```
    for i := 1; i < n; i++ {  
        if beratKelinci[i] < beratTerkecil {  
            beratTerkecil = beratKelinci[i]  
        }  
        if beratKelinci[i] > beratTerbesar {  
            beratTerbesar = beratKelinci[i]  
        }  
    }
```

```
    fmt.Printf("%.2f %.2f\n", beratTerkecil, beratTerbesar)
```

```
}
```

Output

Output

```
PS C:\Users\fazaf\OneDrive\ドキュメント\Tugas
7\UNGUIDED1\UG1.go"
3
2 4 3
2.00 4.00
PS C:\Users\fazaf\OneDrive\ドキュメント\Tugas
```

Deskripsi Program :

Program ini digunakan untuk menentukan anak kelinci dengan berat paling ringan dan paling berat. Pengguna akan diminta untuk memasukkan jumlah anak kelinci beserta berat masing-masing kelinci. Setiap nilai berat yang dimasukkan akan dibaca dan disimpan dalam array. Selanjutnya, program akan membandingkan seluruh data berat untuk menemukan kelinci teringan dan terberat. Di akhir, program akan menampilkan berat anak kelinci terberat dan teringan dengan format dua angka di belakang koma.

Unguided 2

```

package main

import (
    "fmt"
)

func main() {

    var beratIkan [1000]float64

    var x, y int
    fmt.Scan(&x, &y)

    if x <= 0 || x > 1000 {
        fmt.Println("Jumlah ikan harus antara 1-1000")
        return
    }

    if y <= 0 {
        fmt.Println("Jumlah wadah harus positif")
        return
    }

    for i := 0; i < x; i++ {
        fmt.Scan(&beratIkan[i])
    }

    var totalBeratPerWadah [1000]float64
    var ikanPerWadah [1000]int

    for i := 0; i < x; i++ {
        wadahIdx := i % y
        totalBeratPerWadah[wadahIdx] += beratIkan[i]
        ikanPerWadah[wadahIdx]++
    }

    for i := 0; i < y; i++ {
        fmt.Printf("%.2f", totalBeratPerWadah[i])
        if i < y-1 {
            fmt.Print(" ")
        }
    }
    fmt.Println()

    var totalRataRata float64
    for i := 0; i < y; i++ {
        var rataRata float64
        if ikanPerWadah[i] > 0 {
            rataRata = totalBeratPerWadah[i] / float64(ikanPerWadah[i])
        }
        totalRataRata += rataRata
    }

    rataRataKeseluruhan := totalRataRata / float64(y)
    fmt.Printf("%.2f\n", rataRataKeseluruhan)
}

```

Output

```
PS C:\Users\fazaf\OneDrive\ドキュメント\Tugas kuliah\SEMESTER 2\ALPRO 2\103112400072_Modul 7> go run "c:\Us
4
2
4
3
2.6
3.1
6.60 6.10
3.17
PS C:\Users\fazaf\OneDrive\ドキュメント\Tugas kuliah\SEMESTER 2\ALPRO 2\103112400072_Modul 7>
```

Deskripsi Program :

Program ini digunakan untuk menghitung total dan rata-rata berat ayam yang dimasukkan ke dalam beberapa wadah berdasarkan kapasitas tertentu. Pengguna diminta untuk memasukkan jumlah ayam, kapasitas maksimum ayam per wadah, serta berat masing-masing ayam. Setelah itu, program akan menghitung total berat ayam dalam setiap wadah dan menghitung rata-rata beratnya. Di akhir, program akan menampilkan total dan rata-rata berat dari seluruh wadah yang tersedia

Unguided 3

```

1 package main
2
3 import (
4     "fmt"
5 )
6
7 type arrBalita [100]float64
8
9 func hitungMinMax(arrBerat arrBalita, bMin, bMax *float64) {
10     *bMin = arrBerat[0]
11     *bMax = arrBerat[0]
12
13     for i := range arrBerat {
14         if arrBerat[i] != 0 {
15             if arrBerat[i] < *bMin {
16                 *bMin = arrBerat[i]
17             }
18             if arrBerat[i] > *bMax {
19                 *bMax = arrBerat[i]
20             }
21         }
22     }
23 }
24
25 func rerata(arrBerat arrBalita, n int) float64 {
26     total := 0.0
27     for i := 0; i < n; i++ {
28         total += arrBerat[i]
29     }
30
31     if n > 0 {
32         return total / float64(n)
33     }
34     return 0
35 }
36
37 func main() {
38     var dataBalita arrBalita
39     var n int
40
41     fmt.Print("Masukkan banyak data berat balita : ")
42     fmt.Scan(&n)
43
44     if n <= 0 || n > 100 {
45         fmt.Println("Jumlah data harus antara 1-100")
46         return
47     }
48
49     for i := 0; i < n; i++ {
50         fmt.Printf("Masukkan berat balita ke-%d: ", i+1)
51         fmt.Scan(&dataBalita[i])
52     }
53
54     var min, max float64
55     hitungMinMax(dataBalita, &min, &max)
56
57     rataRata := rerata(dataBalita, n)
58
59     fmt.Printf("Berat balita minimum: %.2f kg\n", min)
60     fmt.Printf("Berat balita maksimum: %.2f kg\n", max)
61     fmt.Printf("Rerata berat balita: %.2f kg\n", rataRata)
62 }
63
64
65

```

Output

```
h\SEMESTER 2\ALPRO 2\103112400072_Modul 7\UNGUIDED3\UG3.go
Masukkan banyak data berat balita : 5
Masukkan berat balita ke-1: 4.7
Masukkan berat balita ke-2: 5.2
Masukkan berat balita ke-3: 6
Masukkan berat balita ke-4: 7.2
Masukkan berat balita ke-5: 6.9
Berat balita minimum: 4.70 kg
Berat balita maksimum: 7.20 kg
Rerata berat balita: 6.00 kg
```

Deskripsi Program : Program ini digunakan untuk menghitung berat minimum, maksimum, dan rata-rata dari sekumpulan data berat balita. Pengguna diminta memasukkan jumlah balita serta berat masing-masing balita. Semua data yang dimasukkan akan disimpan dalam array. Program menggunakan fungsi `hitungMax` untuk menentukan berat terendah dan tertinggi, serta fungsi `rerata` untuk menghitung nilai rata-rata dari seluruh data. Hasil akhirnya, program akan menampilkan berat minimum, maksimum, dan rata-ratanya dengan dua angka di belakang koma.

IV. KESIMPULAN

Dari modul ini, kita mempelajari cara mencari nilai minimum, maksimum, dan rata-rata dari sekumpulan data menggunakan bahasa pemrograman Go. Proses pencarian dilakukan dengan membandingkan setiap elemen data secara bertahap. Pembahasan ini membantu kita memahami mekanisme pencarian nilai ekstrem serta penerapannya dalam berbagai konteks, seperti data tinggi badan, berat, nilai ujian, maupun harga suatu barang.

MODUL 7 PRAKTIKUM ALPRO