

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



Oleh:

DWI OKTA SURYANINGRUM

103112400066

12-IF-01

**S1 TEKNIK INFORMATIKA**  
**TELKOM UNIVERSITY PURWOKERTO**  
**2025**

## **I. DASAR TEORI**

### **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.

## II. GUIDED

Source Code + Screenshot hasil program beserta penjelasan

### a. Guided 1

```
1 // DWI OKTA SURYANINGRUM
2 // 103112400066
3
4 package main
5
6 import "fmt"
7
8 func main() {
9     var n int
10    fmt.Print("masukkan jumlah tanaman : ")
11    fmt.Scan(&n)
12
13    var tinggiTanaman [500]float64
14    for i := 0; i < n; i++ {
15        fmt.Printf("masukkan tinggi tanaman ke-%d (cm) : ", i+1)
16        fmt.Scan(&tinggiTanaman[i])
17    }
18
19    min, max := tinggiTanaman[0], tinggiTanaman[0]
20    for i := 1; i < n; i++ {
21        if tinggiTanaman[i] < min {
22            min = tinggiTanaman[i]
23        }
24        if tinggiTanaman[i] > max {
25            max = tinggiTanaman[i]
26        }
27    }
28    fmt.Printf("Tinggi tanaman terpendek adalah %.2f cm\n", min)
29    fmt.Printf("Tinggi tanaman tertinggi adalah %.2f cm\n", max)
30 }
```

Output :

```
mymac@mymacs-Air ALPRO SMT 2 % go run "/Users/mymac/Documents/ITTP/ALPRO SMT 2/MODUL 7/guided1.go"
masukkan jumlah tanaman : 2
masukkan tinggi tanaman ke-1 (cm) : 14.5
masukkan tinggi tanaman ke-2 (cm) : 13.9
Tinggi tanaman terpendek adalah 13.90 cm
Tinggi tanaman tertinggi adalah 14.50 cm
```

## b. Guided 2

```
1 // DWI OKTA SURYANINGRUM
2 // 1013112400066
3
4 package main
5
6 import "fmt"
7
8 func main() {
9     var x, y int
10    fmt.Print("masukan jumlah buku dan jumlah buku per rak: ")
11    fmt.Scan(&x, &y)
12
13
14    var hargaBuku [500]float64
15    fmt.Println("masukan harga setiap buku (dalam ribuan Rp.):")
16    for i := 0; i < x; i++ {
17        fmt.Scan(&hargaBuku[i])
18    }
19
20    var hargaRataRata []float64
21    for i := 0; i < x; i+= y {
22        total := 0.0
23        for j := i; j < i+y && j < x; j++ {
24            total += hargaBuku[j]
25        }
26        hargaRataRata = append(hargaRataRata, total/float64(y))
27    }
28
29    min, max := hargaRataRata[0], hargaRataRata[0]
30    for _, harga := range hargaBuku[:x] {
31        if harga < min {
32            min = harga
33        }
34        if harga > max {
35            max = harga
36        }
37    }
38
39    fmt.Printf("\n harga rata-rata per rak: ")
40    for _, avg := range hargaRataRata {
41        fmt.Printf("%.2f ", avg)
42    }
43    fmt.Printf("\n harga buku termahal: %.2f", max)
44    fmt.Printf("\n harga buku termurah: %.2f\n", min)
45 }
```

Output :

```
mymac@mymacs-Air ALPRO SMT 2 % go run "/Users/mymac/Documents/ITTP/ALPRO SMT 2/MODUL 7/guided2.go"
masukan jumlah buku dan jumlah buku per rak: 5 3
masukan harga setiap buku (dalam ribuan Rp.):
78000
70000
65000
88000
67000

harga rata-rata per rak: 71000.00 51666.67
harga buku termahal: 88000.00
harga buku termurah: 65000.00
```

### c. Guided 3

```
1 // DWI OKTA SURYANINGRUM
2 // 103112400066
3
4 package main
5
6 import "fmt"
7
8 func main() {
9     var n int
10    fmt.Print("masukkan jumlah siswa: ")
11    fmt.Scan(&n)
12
13    var nilaiSiswa [200]float64
14    var totalNilai float64 = 0
15
16    fmt.Println("masukan nilai ujian masing-masing siswa")
17    for i := 0; i < n; i++ {
18        fmt.Scan(&nilaiSiswa[i])
19        totalNilai += nilaiSiswa[i]
20    }
21
22    rataRata := totalNilai / float64(n)
23
24    min, max := nilaiSiswa[0], nilaiSiswa[0]
25    var diAtasRataRata int = 0
26    for _, nilai := range nilaiSiswa[:n]{
27        if nilai < min {
28            min = nilai
29        }
30        if nilai > max {
31            max = nilai
32        }
33        if nilai > rataRata {
34            diAtasRataRata++
35        }
36    }
37    fmt.Printf("Nilai terendah: %.2f\n", min)
38    fmt.Printf("Nilai tertinggi: %.2f\n", max)
39    fmt.Printf("Nilai rata-rata: %.2f\n", rataRata)
40    fmt.Printf("Jumlah siswa di atas rata-rata: %d\n", diAtasRataRata)
41
42 }
```

Output :

```
mymac@mymacs-Air ALPRO SMT 2 % go run "/Users/mymac/Documents/ITTP/ALPRO SMT 2/MODUL 7/guided3.go"
masukkan jumlah siswa: 3
masukan nilai ujian masing-masing siswa
90
88
98
Nilai terendah: 88.00
Nilai tertinggi: 98.00
Nilai rata-rata: 92.00
Jumlah siswa di atas rata-rata: 1
```

### III. UNGUIDED

#### a. Unguided 1

```
1 // DWI OKTA SURYANINGRUM
2 // 103112400066
3
4 package main
5
6 import (
7     "fmt"
8 )
9
10 func main() {
11     var N int
12     var berat [1000]float64 // Menyimpan jumlah anak kelinci yang akan ditimbang
13                               // Array untuk menyimpan berat anak kelinci (maksimal 1000 data)
14
15     // Input jumlah anak kelinci dari pengguna
16     fmt.Print("Masukkan jumlah anak kelinci: ")
17     fmt.Scan(&N) // Membaca jumlah anak kelinci
18
19     // Input berat masing-masing anak kelinci sebanyak N kali
20     for i := 0; i < N; i++ {
21         fmt.Print("Masukkan berat anak kelinci ke-%d: ", i+1)
22         fmt.Scan(&berat[i]) // Membaca berat anak kelinci dan menyimpannya ke array
23     }
24
25     // Inisialisasi nilai terkecil dan terbesar dengan elemen pertama
26     terkecil := berat[0] // Asumsikan berat pertama sebagai yang terkecil
27     terbesar := berat[0] // Asumsikan berat pertama sebagai yang terbesar
28
29     // Loop untuk mencari nilai terkecil dan terbesar
30     for i := 1; i < N; i++ {
31         if berat[i] < terkecil { // Jika berat ke-i lebih kecil dari yang terkecil sebelumnya
32             terkecil = berat[i] // Perbarui nilai terkecil
33         }
34         if berat[i] > terbesar { // Jika berat ke-i lebih besar dari yang terbesar sebelumnya
35             terbesar = berat[i] // Perbarui nilai terbesar
36         }
37     }
38
39     // Menampilkan hasil akhir: berat terkecil dan terbesar
40     fmt.Printf("Berat terkecil: %.2f\n", terkecil) // Output berat terkecil dengan 2 desimal
41     fmt.Printf("Berat terbesar: %.2f\n", terbesar) // Output berat terbesar dengan 2 desimal
42 }
```

Output :

```
● mymac@mymacs-Air ALPRO SMT 2 % go run "/Users/mymac/Documents/ITTP/ALPRO SMT 2/MODUL 7/unguided1.go"
Masukkan jumlah anak kelinci: 3
Masukkan berat anak kelinci ke-1: 1.5
Masukkan berat anak kelinci ke-2: 1.25
Masukkan berat anak kelinci ke-3: 1.75
Berat terkecil: 1.25
Berat terbesar: 1.75
```

## b. Unguided 2

```
1 // DWI OKTA SURYANINGRUM
2 // 10311240066
3
4 package main
5
6 import "fmt"
7
8 func main() {
9     var x, y int // x = jumlah ikan yang akan dihitung, y = kapasitas wadah
10    var berat [1000]float64 // Array untuk menyimpan berat tiap ikan, kapasitas maksimum 1000 ikan
11
12    // Meminta pengguna untuk memasukkan jumlah ikan dan kapasitas wadah
13    fmt.Println("Masukkan jumlah ikan dan kapasitas wadah: ")
14    fmt.Scan(&x, &y) // Membaca jumlah ikan dan kapasitas wadah yang dimasukkan pengguna
15
16    // Meminta pengguna untuk memasukkan berat masing-masing ikan
17    fmt.Println("Masukkan berat masing-masing ikan:")
18    for i := 0; i < x; i++ {
19        fmt.Scan(&berat[i]) // Membaca berat setiap ikan dan menyimpannya ke dalam array
20    }
21
22    // Menyiapkan wadah untuk menampung total berat ikan per wadah
23    var totalWadah []float64 // Slice untuk menyimpan total berat ikan dalam setiap wadah
24    currentTotal := 0.0 // Variabel untuk menghitung total berat ikan dalam wadah saat ini
25    count := 0 // Variabel untuk menghitung jumlah ikan dalam satu wadah
26
27    // Mengelompokkan ikan ke dalam wadah berdasarkan kapasitas dan menghitung total berat per wadah
28    for i := 0; i < x; i++ {
29        currentTotal += berat[i] // Menambahkan berat ikan ke total berat wadah saat ini
30        count++ // Menambahkan jumlah ikan yang sudah dimasukkan ke wadah saat ini
31
32        // Jika wadah sudah penuh atau ini adalah ikan terakhir, simpan total wadah dan reset
33        if count == y || i == x-1 {
34            totalWadah = append(totalWadah, currentTotal) // Simpan total berat wadah ke dalam slice
35            currentTotal = 0.0 // Reset total berat untuk wadah berikutnya
36            count = 0 // Reset jumlah ikan untuk wadah berikutnya
37        }
38    }
39
40    // Menampilkan total berat per wadah
41    fmt.Println("Total berat per wadah:")
42    for _, total := range totalWadah {
43        fmt.Printf("%.2f ", total) // Menampilkan total berat ikan dalam setiap wadah dengan format 2 desimal
44    }
45    fmt.Println()
46
47    // Menghitung rata-rata berat ikan per wadah
48    var sum float64
49    for _, total := range totalWadah {
50        sum += total // Menjumlahkan berat total setiap wadah
51    }
52    average := sum / float64(len(totalWadah)) // Menghitung rata-rata dari semua total wadah
53
54    // Menampilkan hasil rata-rata berat per wadah
55    fmt.Printf("Rata-rata berat per wadah: %.2f\n", average)
56 }
57
```

Output :

```
● mymac@mymacs-Air ALPRO SMT 2 % go run "/Users/mymac/Documents/ITTP/ALPRO SMT 2/MODUL 7/unguided2.go"
Masukkan jumlah ikan dan kapasitas wadah: 5 10
Masukkan berat masing-masing ikan:
2
3
2
4
5
Total berat per wadah:
16.00
Rata-rata berat per wadah: 16.00
```



### c. Unguided 3

```
1 // DWI OKTA SURYANIGNRUM
2 // 10311240066
3
4 package main
5
6 import "fmt"
7
8 // Mendefinisikan tipe arrBalita sebagai array dengan ukuran maksimum 100 elemen
9 type arrBalita [100]float64
10
11 // Fungsi untuk menghitung berat minimum dan maksimum dari array arrBerat
12 func hitungMinMax(arrBerat arrBalita, bMin, bMax *float64) {
13     // Menginisialisasi bMin dan bMax dengan nilai berat balita pertama
14     *bMin = arrBerat[0]
15     *bMax = arrBerat[0]
16
17     // Loop untuk mencari nilai minimum dan maksimum
18     for i := 0; i < len(arrBerat); i++ {
19         if arrBerat[i] != 0 {
20             if arrBerat[i] < *bMin {
21                 *bMin = arrBerat[i] // Jika ditemukan yang lebih kecil, set bMin
22             }
23             if arrBerat[i] > *bMax {
24                 *bMax = arrBerat[i] // Jika ditemukan yang lebih besar, set bMax
25             }
26         }
27     }
28
29     // Menampilkan hasil minimum dan maksimum
30     fmt.Printf("Berat balita minimum: %.2f kg\n", *bMin)
31     fmt.Printf("Berat balita maksimum: %.2f kg\n", *bMax)
32 }
33
34 // Fungsi untuk menghitung rata-rata berat balita
35 func rerata(arrBerat arrBalita) float64 {
36     var pembagi, jumlah float64
37
38     // Menghitung total berat dan jumlah data yang valid
39     for i := 0; i < len(arrBerat); i++ {
40         if arrBerat[i] > 0 {
41             jumlah += arrBerat[i]
42             pembagi++
43         }
44     }
45
46     // Menghitung dan mengembalikan rata-rata
47     return jumlah / pembagi
48 }
49
50 func main() {
51     var N int
52     var beratBalita arrBalita // Array untuk menampung data berat balita
53
54     // Meminta input dari pengguna tentang jumlah balita
55     fmt.Print("Masukkan banyak data berat balita: ")
56     fmt.Scan(&N)
57
58     // Memasukkan berat balita ke dalam array
59     for i := 0; i < N; i++ {
60         fmt.Printf("Masukkan berat balita ke-%d: ", i+1)
61         fmt.Scan(&beratBalita[i])
62     }
63
64     // Menghitung berat minimum dan maksimum
65     var min, max float64
66     hitungMinMax(beratBalita, &min, &max)
67
68     // Menghitung dan menampilkan rerata berat balita
69     fmt.Printf("Rata-rata: %.2f\n", rerata(beratBalita))
70 }
```

Output :

```
mymac@mymacs-Air ALPRO SMT 2 % go run "/Users/mymac/Documents/ITTP/ALPRO SMT 2/MODUL 7/tempCodeRunnerFile.go"
Masukkan banyak data berat balita: 4
Masukkan berat balita ke-1: 5.3
Masukkan berat balita ke-2: 6.2
Masukkan berat balita ke-3: 4.1
Masukkan berat balita ke-4: 9.9
Berat balita minimum: 4.10 kg
Berat balita maksimum: 9.90 kg
Rata-rata: 6.38
```