

**LAPORAN PRAKTIKUM  
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL 7**

**PENCARIAN NILAI EKSTRIM PADA HIMPUNAN DATA**



Oleh:

ARIEL AHNAF KUSUMA

103112400050

IF-12-01

**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.

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

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

### **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 GUIDED 1

### SOURCE CODE

```
GUIDED1.go > main
1 package main
2
3 import "fmt"
4
5 func main() {
6     var n int
7     fmt.Println("Masukkan jumlah tanaman: ")
8     fmt.Scan(&n)
9
10    var tinggiTanaman [500]float64
11    for i := 0; i < n; i++ {
12        fmt.Printf("Masukkan tinggi tanaman ke-%d (cm):", i+1)
13        fmt.Scan(&tinggiTanaman[i])
14    }
15
16    min, max := tinggiTanaman[0], tinggiTanaman[0]
17    for i := 1; i < n; i++ {
18        if tinggiTanaman[i] < min {
19            min = tinggiTanaman[i]
20        }
21        if tinggiTanaman[i] > max {
22            max = tinggiTanaman[i]
23        }
24    }
25    fmt.Printf("\nTinggi tanaman tertinggi %.2f cm\n", max)
26    fmt.Printf("Tinggi tanaman terpendek: %.2f cm\n", min)
27 }
28
```

### OUTPUT CODE

```
PS C:\ARIEL AHNAF KUSUMA\103112400050_Modul 10> go run "c:\ARIEL AHNAF KUSUMA\103112400050_Modul 10\GUIDED1.go"
Masukkan jumlah tanaman: 3
Masukkan tinggi tanaman ke-1 (cm): 2
Masukkan tinggi tanaman ke-2 (cm): 5
Masukkan tinggi tanaman ke-3 (cm): 8

Tinggi tanaman tertinggi: 5.00 cm
Tinggi tanaman terpendek: 2.00 cm
PS C:\ARIEL AHNAF KUSUMA\103112400050_Modul 10>
```

### DESKRIPSI CODE

Program diatas berfungsi untuk mencatat dan menganalisis tinggi tanaman dan rendah suatu tanaman

### GUIDED 2

### SOURCE CODE

```

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

```

## OUTPUT CODE

```

PS C:\ARIEL AHNAF KUSUMA\103112400050_Modul 10> go run "c:\ARIEL AHNAF KUSUMA\103112400050_Modul 10\GUIDED2.go"
Masukkan jumlah buku dan jumlah buku per rak: 3 4

Masukkan harga setiap buku (dalam ribuan Rp):
30000
40000
35000

Rata-rata harga per rak: 26250.00
Harga termahal: 40000.00 Rp
Harga termurah: 30000.00 Rp
PS C:\ARIEL AHNAF KUSUMA\103112400050_Modul 10>

```

## DESKRIPSI CODE

Program ini berfungsi untuk menghitung harga rata rata buku per/rak dan menentukan harganya dari yang termahal sampai termurah

## GUIDED 3

## SOURCE CODE

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

## OUTPUT CODE

```
PS C:\VARIEL AHNAF KUSUMA\103112400050_Modul 10> go run "c:\VARIEL AHNAF KUSUMA\103112400050_Modul 10\GUIDED3.go"
Masukkan jumlah siswa: 5

Masukkan nilai ujian masing-masing siswa:
65
88
100
73
91

Nilai terendah: 65
Nilai tertinggi: 100
Rata-rata kelas: 83.40
Jumlah siswa di atas rata-rata: 3
PS C:\VARIEL AHNAF KUSUMA\103112400050_Modul 10>
```

## DESKRIPSI CODE

Program ini berfungsi untuk menghitung nilai terendah, hingga tertinggi dan rata-rata siswa dan juga berapa jumlah siswa yang memiliki nilai di atas rata-rata.

### III. UNGUIDED

#### UNGUIDED 1

#### SOURCE CODE

```
1  package main
2
3  import (
4      "fmt"
5  )
6
7  // 103112400050
8  func main() {
9      var beratKelinci [1000]float64
10     var n int
11
12     fmt.Print("Masukkan jumlah anak kelinci (maksimal 1000): ")
13     fmt.Scan(&n)
14
15     if n < 1 || n > 1000 {
16         fmt.Println("Jumlah tidak valid! Harus antara 1 hingga 1000.")
17         return
18     }
19
20     fmt.Println("\nMasukkan berat masing-masing anak kelinci:")
21     for i := 0; i < n; i++ {
22         fmt.Printf("Berat anak kelinci ke-%d (kg): ", i+1)
23         fmt.Scan(&beratKelinci[i])
24     }
25
26     beratTerkecil := beratKelinci[0]
27     beratTerbesar := beratKelinci[0]
28
29     for i := 1; i < n; i++ {
30         if beratKelinci[i] < beratTerkecil {
31             beratTerkecil = beratKelinci[i]
32         }
33         if beratKelinci[i] > beratTerbesar {
34             beratTerbesar = beratKelinci[i]
35         }
36     }
37
38     fmt.Printf("\nBerat terkecil: %.2f kg\n", beratTerkecil)
39     fmt.Printf("Berat terbesar: %.2f kg\n", beratTerbesar)
40 }
```

#### OUTPUT CODE

```

PS C:\ARIEL AHNAF KUSUMA\103112400050_Modul 10> go run "c:\ARIEL AHNAF KUSUMA\103112400050_Modul 10\UNGUIDED
.go"
Masukkan jumlah anak kelinci (maksimal 1000): 4

Masukkan berat masing-masing anak kelinci:
Berat anak kelinci ke-1 (kg): 33
Berat anak kelinci ke-2 (kg): 27
Berat anak kelinci ke-3 (kg): 30
Berat anak kelinci ke-4 (kg): 25

Berat terkecil: 25.00 kg
Berat terbesar: 33.00 kg
PS C:\ARIEL AHNAF KUSUMA\103112400050_Modul 10>

```

## DESKRIPSI CODE

Program diatas ini berfungsi untuk mencari berat sejumlah anak kelinci dari yang terkecil hingga terbesar

## UNGUIDED 2

## SOURCE CODE

```

1  package main
2
3  //103112400050
4  import "fmt"
5
6  func main() {
7      var x, y int
8      var beratIkan [1000]float64
9      fmt.Print("Masukkan jumlah ikan dan jumlah ikan per wadah: ")
10     fmt.Scan(&x, &y)
11     fmt.Print("Masukkan berat ikan:")
12     for i := 0; i < x; i++ {
13         fmt.Scan(&beratIkan[i])
14     }
15     jumlahWadah := x / y
16     if x%y != 0 {
17         jumlahWadah++
18     }
19     totalWadah := make([]float64, jumlahWadah)
20     for i := 0; i < x; i++ {
21         idx := i / y
22         totalWadah[idx] += beratIkan[i]
23     }
24     fmt.Println("Total berat tiap wadah:")
25     for _, total := range totalWadah {
26         fmt.Printf("%.2f ", total)
27     }
28     fmt.Println()
29     var totalSemua float64
30     for _, total := range totalWadah {
31         totalSemua += total
32     }
33     rataRata := totalSemua / float64(jumlahWadah)
34     fmt.Printf("Rata-rata berat per wadah: %.2f\n", rataRata)
35 }

```



## OUTPUT CODE

```
PS C:\ARIEL AHMAF KUSUMA\103112400050_Modul 10> go run "c:\ARIEL AHMAF KUSUMA\103112400050_Modul 10\go.go"
Masukkan jumlah ikan dan jumlah ikan per wadah: 2 2
Masukkan berat ikan:23
20
Total berat tiap wadah:
43.00
Rata-rata berat per wadah: 43.00
```

## DESKRIPSI CODE

Program ini berfungsi untuk total rata-rata berat ikan dalam wadah berdasarkan jumlah ikan per wadahnya. Pengguna memasukkan memasukkan jumlah dan kapasitas jumlah ikan per wadah, dan masing masing berat ikan, lalu program akan membagi ikan dalam beberapa wadah sesuai apa yang ditentukan, lalu menampilkan berat masing masing wadah dengan rata-rata berat per wadah

## UNGUIDED 3

## SOURCE CODE

```

1  package main
2
3  import "fmt"
4
5  //103112400050
6  type arrBalita [100]float64
7
8  func hitungMinMax(data arrBalita, jumlah int, min, max *float64) {
9      *min = data[0]
10     *max = data[0]
11     for i := 1; i < jumlah; i++ {
12         if data[i] < *min {
13             *min = data[i]
14         }
15         if data[i] > *max {
16             *max = data[i]
17         }
18     }
19 }
20
21 func Rerata(data arrBalita, jumlah int) float64 {
22     var total float64
23     for i := 0; i < jumlah; i++ {
24         total += data[i]
25     }
26     return total / float64(jumlah)
27 }
28
29 func main() {
30     var data arrBalita
31     var jumlahData int
32
33     fmt.Print("Masukkan jumlah data berat balita: ")
34     fmt.Scan(&jumlahData)
35
36     for i := 0; i < jumlahData; i++ {
37         fmt.Printf("Masukkan berat balita ke-%d (kg): ", i+1)
38         fmt.Scan(&data[i])
39     }
40
41     var beratMin, beratMax float64
42     hitungMinMax(data, jumlahData, &beratMin, &beratMax)
43     rataRata := Rerata(data, jumlahData)
44
45     fmt.Printf("\nBerat minimum balita: %.2f kg\n", beratMin)
46     fmt.Printf("Berat maksimum balita: %.2f kg\n", beratMax)
47     fmt.Printf("Rata-rata berat balita: %.2f kg\n", rataRata)
48 }
49

```

OUTPUT CODE

```
PS C:\ARIEL AHNAF KUSUMA\103112400050_Modul 10> go run "c:\ARIEL AHNAF KUSUMA\103112400050_Modul 10\UNGUIDED3.go"
Masukkan jumlah data berat balita: 4
Masukkan berat balita ke-1 (kg): 9
Masukkan berat balita ke-2 (kg): 12
Masukkan berat balita ke-3 (kg): 10
Masukkan berat balita ke-4 (kg): 9.7

Berat minimum balita: 9.00 kg
Berat maksimum balita: 12.00 kg
Rata-rata berat balita: 10.18 kg
PS C:\ARIEL AHNAF KUSUMA\103112400050_Modul 10>
```

#### DESKRIPISI CODE

Program diatas berfungsi untuk mencari berat minimum, maksimum dan rata-rata dari data berat badan sejumlah balita, pengguna diminta untuk memasukkan berat dan jumlah masing masing balita

#### IV. KESIMPULAN

Pada modul ini kita mempelajari tentang algoritma pencarian nilai ekstrem dan mengimplementasikannya, yaitu nilai maksimum dan minimum dari sekumpulan data. Proses pencarian dilakukan baik pada array bertipe data dasar ataupun array bertipe data terstruktur. Mahasiswa juga mempelajari pengolahan data lebih lanjut seperti perhitungan rata-rata dan analisis distribusi nilai. Modul ini memperkuat pemahaman mahasiswa dalam struktur data array, penggunaan perulangan, fungsi, dan juga pengelolaan input/output pada konteks pencarian dan analisis data sederhana

## **V. REFERENSI**

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