

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



Oleh:

HISYAM NURDIATMOKO

103112400049

IF-12-01

**S1 TEKNIK INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO**

2025

I. DASAR TEORI

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.

2. 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.

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 Guided 1:

```
package main

import "fmt"

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

    var tinggiTanaman [500]float64
    for i := 0; i < n; i++ {
        fmt.Printf("Masukan 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 D:\1_Matkul\Alpro2\week9_modul7\103112400049_MODUL7> go run "d:\1_Matkul\Alpro2\week9_modul7\
Masukan jumlah tanaman: 3
Masukan tinggi tanaman ke-1 (cm): 4
Masukan tinggi tanaman ke-2 (cm): 5
Masukan tinggi tanaman ke-3 (cm): 6

Tinggi tanaman tertinggi: 6.00 cm
Tinggi tanaman terpendek: 4.00 cm
PS D:\1_Matkul\Alpro2\week9_modul7\103112400049_MODUL7> █
```

Deskripsi :

Program GO Guided 1 berfungsi untuk mencatat serta menentukan tinggi tanaman tertinggi dan terpendek dari sejumlah tanaman yang dimasukkan oleh

pengguna. Program dimulai dengan meminta pengguna memasukkan jumlah tanaman, kemudian menggunakan array untuk menyimpan tinggi masing-masing tanaman. Setelah semua data dimasukkan, program melakukan iterasi untuk mencari nilai maksimum dan minimum dari array tersebut. Hasil akhir ditampilkan dalam bentuk tinggi tanaman tertinggi dan terpendek dengan format dua angka di belakang koma. Program ini berguna untuk analisis sederhana terhadap data tinggi tanaman.

Source code Guided 2 :

```
package main

import "fmt"

func main() {
    var x, y int
    fmt.Println("Masukan jumlah buku dan jumlah buku per rak: ")
    fmt.Scan(&x, &y)

    var hargaBuku [500]float64
    fmt.Println("\nMasukan 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 D:\1_Matkul\Alpro2\week9_modul7\103112400049_MODUL7> go run "d:\1_Matkul\Alpro2\week9_modul7\
Masukan jumlah buku dan jumlah buku per rak: 3 2

Masukan harga setiap buku (dalam ribuan Rp):
10000
20000
30000

Rata-rata harga per rak: 15000.00 15000.00
Harga termahal: 30000.00 Rp
Harga termurah: 10000.00 Rp
PS D:\1_Matkul\Alpro2\week9_modul7\103112400049_MODUL7> █
```

Deskripsi :

Program GO Guided 2 ini digunakan untuk menghitung rata-rata harga buku per rak serta mencari harga buku termahal dan termurah. Pengguna diminta memasukkan jumlah total buku dan kapasitas buku per rak, lalu menginput harga masing-masing buku. Program menghitung rata-rata harga untuk setiap rak dengan menjumlahkan harga buku dalam satu rak dan membaginya dengan kapasitas rak. Selain itu, program juga mencari harga buku paling mahal dan paling murah dari seluruh data yang dimasukkan. Hasil akhirnya berupa daftar rata-rata harga per rak, harga buku termahal, dan termurah dalam satuan ribuan rupiah.

Source code Guided 3:

```
package main

import "fmt"

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

    var nilaiSiswa [200]float64
    var totalNilai float64 = 0

    fmt.Println("\nMasukan 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 terendah: %.0f\n", max)
    fmt.Printf("Rata-rata kelas: %.2f\n", rataRata)
    fmt.Printf("Jumlah siswa di atas rata-rata: %d\n", diAtasRataRata)
}
```

Output:

```
PS D:\1_Matkul\Alpro2\week9_modul7\103112400049_MODUL7> go run "d:\1_Matkul\Alpro2\week9_modul7\
Masukan jumlah siswa: 3

Masukan nilai ujian masing-masing siswa:
80
90
100

Nilai terendah: 80
Nilai tertinggi: 100
Rata-rata kelas: 90.00
Jumlah siswa di atas rata-rata: 1
PS D:\1_Matkul\Alpro2\week9_modul7\103112400049_MODUL7> |
```

Deskripsi:

Program ini ditulis dalam bahasa Go dan digunakan untuk mengolah data nilai ujian siswa. Program meminta pengguna memasukkan jumlah siswa, lalu membaca nilai masing-masing siswa dan menghitung totalnya. Dari data tersebut, program menghitung rata-rata nilai kelas, serta mencari nilai tertinggi dan terendah. Selain itu, program juga menghitung berapa banyak siswa yang memiliki nilai di atas rata-rata. Hasil yang ditampilkan mencakup nilai terendah, nilai tertinggi, rata-rata nilai kelas, dan jumlah siswa yang nilainya melebihi rata-rata.

III. UNGUIDED

Source code Unguided 1 :

```
package main

//103112400049 Hisyam Nurdiamoko

import "fmt"

func main() {
    var hisyam int
    var berat [1000]float64
    fmt.Print("Masukkan jumlah anak kelinci: ")
    fmt.Scan(&hisyam)
    for i := 0; i < hisyam; i++ {
        fmt.Printf("Masukkan berat kelinci ke-%d (kg): ", i+1)
        fmt.Scan(&berat[i])
    }
    min := berat[0]
    max := berat[0]
    for i := 1; i < hisyam; i++ {
        if berat[i] < min {
            min = berat[i]
        }
        if berat[i] > max {
            max = berat[i]
        }
    }
    fmt.Printf("Berat kelinci terkecil: %.2f\n", min)
    fmt.Printf("Berat kelinci terbesar: %.2f\n", max)
}
```

Output:

```
PS D:\1_Matkul\Alpro2\week9_modul7\103112400049_MODUL7> go run "d:\1_Matkul\Alpro2\week9_modul7\
"
Masukkan jumlah anak kelinci: 3
Masukkan berat kelinci ke-1 (kg): 1
Masukkan berat kelinci ke-2 (kg): 2
Masukkan berat kelinci ke-3 (kg): 3
Berat kelinci terkecil: 1.00
Berat kelinci terbesar: 3.00
PS D:\1_Matkul\Alpro2\week9_modul7\103112400049_MODUL7> █
```

Deskripsi:

Program GO Unguided 1 ini berfungsi untuk mencatat serta menganalisis berat badan anak kelinci. Pengguna diminta memasukkan jumlah anak kelinci, lalu program menerima input berat masing-masing kelinci yang disimpan dalam array.

Setelah semua data dimasukkan, program mencari berat terkecil dan terbesar dengan membandingkan setiap elemen dalam array. Hasil akhirnya ditampilkan dalam bentuk berat kelinci terkecil dan terbesar dengan dua angka di belakang koma. Program ini berguna untuk pemantauan pertumbuhan atau kondisi fisik anak kelinci dalam suatu kelompok.

Source code Unguided 2 :

```
package main

//103112400049 Hisyam Nurdiatmoko

import "fmt"

func main() {
    var x, y int
    var beratIkanHisyam [1000]float64
    fmt.Print("Masukkan jumlah ikan dan jumlah ikan per wadah (x y): ")
    fmt.Scan(&x, &y)
    fmt.Print("Masukkan berat ikan:")
    for i := 0; i < x; i++ {
        fmt.Scan(&beratIkanHisyam[i])
    }
    jumlahWadah := x / y
    if x%y != 0 {
        jumlahWadah++
    }
    totalWadah := make([]float64, jumlahWadah)
    for i := 0; i < x; i++ {
        idx := i / y
        totalWadah[idx] += beratIkanHisyam[i]
    }
    fmt.Println("Total berat tiap wadah:")
    for _, total := range totalWadah {
        fmt.Printf("%.2f ", total)
    }
    fmt.Println()
    var totalSemua float64
    for _, total := range totalWadah {
        totalSemua += total
    }
    rataRata := totalSemua / float64(jumlahWadah)
    fmt.Printf("Rata-rata berat per wadah: %.2f\n", rataRata)
}
```

Output:

```
PS D:\1_Matkul\Alpro2\week9_modul7\103112400049_MODUL7> go run "d:\1_Matkul\Alpro2\week9_modul7\
Masukkan jumlah ikan dan jumlah ikan per wadah (x y): 3 3
Masukkan berat ikan:1
2
3
Total berat tiap wadah:
6.00
Rata-rata berat per wadah: 6.00
PS D:\1_Matkul\Alpro2\week9_modul7\103112400049_MODUL7> |
```

Deskripsi:

Program GO Unguided 2 ini digunakan untuk mengelompokkan ikan ke dalam beberapa wadah berdasarkan jumlah ikan per wadah, serta menghitung total dan rata-rata berat ikan di setiap wadah. Pengguna diminta memasukkan jumlah total ikan dan kapasitas ikan per wadah, lalu berat masing-masing ikan. Program menghitung jumlah wadah yang dibutuhkan dan mendistribusikan berat ikan ke wadah sesuai urutan input. Setelah itu, program menampilkan total berat per wadah dan menghitung rata-rata berat dari semua wadah. Program ini bermanfaat untuk pengelompokan dan analisis distribusi berat dalam sistem penyimpanan ikan atau logistik perikanan.

Source code Unguided 3 :

```
package main

//103112400049 Hisyam Nurdiamoko

import "fmt"

type arrBalita [100]float64

func hitungMinMax(arr arrBalita, n int, bMin, bMax *float64) {
    *bMin = arr[0]
    *bMax = 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 data arrBalita
    var hisyam int
    fmt.Print("Masukan banyak data berat balita : ")
    fmt.Scan(&hisyam)
    for i := 0; i < hisyam; i++ {
        fmt.Printf("Masukan berat balita ke-%d: ", i+1)
        fmt.Scan(&data[i])
    }
    var min, max float64
    hitungMinMax(data, hisyam, &min, &max)
    rata := rerata(data, hisyam)
    fmt.Printf("Berat balita minimum: %.2f kg\n", min)
    fmt.Printf("Berat balita maksimum: %.2f kg\n", max)
    fmt.Printf("Rerata berat balita: %.2f kg\n", rata)
```

```
}
```

Output:

```
PS D:\1_Matkul\Alpro2\week9_modul7\103112400049_MODUL7> go run "d:\1_Matkul\Alpro2\week9_modul7\
"
Masukan banyak data berat balita : 3
Masukan berat balita ke-1: 10
Masukan berat balita ke-2: 11
Masukan berat balita ke-3: 12
Berat balita minimum: 10.00 kg
Berat balita maksimum: 12.00 kg
Rerata berat balita: 11.00 kg
PS D:\1_Matkul\Alpro2\week9_modul7\103112400049_MODUL7> |
```

Deskripsi:

Program GO Unguided 3 ini digunakan untuk mengolah data berat badan balita, termasuk mencari berat minimum, maksimum, dan rata-rata. Program menggunakan tipe data array khusus bernama `arrBalita` dan memanfaatkan dua fungsi terpisah: `hitungMinMax` untuk mencari berat balita terendah dan tertinggi menggunakan pointer, serta `rerata` untuk menghitung nilai rata-rata. Pengguna diminta memasukkan jumlah data balita dan berat masing-masing, kemudian program menampilkan hasil analisis dalam satuan kilogram.

IV. KESIMPULAN

Praktikum Modul 7 ini mempelajari dan mengimplementasikan algoritma pencarian nilai ekstrem, yaitu nilai maksimum dan minimum dari sekumpulan data. Proses pencarian dilakukan baik pada array bertipe data dasar maupun array bertipe data terstruktur. Dengan menggunakan bahasa pemrograman Go, mahasiswa mempraktikkan bagaimana cara membaca data dari pengguna, menyimpannya dalam array, dan kemudian melakukan perbandingan untuk menemukan nilai ekstrim. Selain itu, mahasiswa juga mempelajari pengolahan data lebih lanjut seperti perhitungan rata-rata dan analisis distribusi nilai. Praktikum ini memperkuat pemahaman terhadap struktur data array, penggunaan perulangan, fungsi, serta pengelolaan input/output dalam konteks pencarian dan analisis data sederhana.

V. REFERENSI

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