

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



DISUSUN OLEH:
Muhammad Shabrian Fadly
103112400087
S1 IF-12-01

PROGRAM STUDI S1 TEKNIK INFORMATIKA
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2024/2025

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 Guided 1:

```
//103112400087_Muhammad Shabrian

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", max)

    fmt.Printf("\nTinggi tanaman Terpendek: %.2f cm", min)

}
```

Output :

```
PS D:\Coding manja> go run "d:\Coding manja\semester2\modul7\tempCodeRunnerFile.go"
masukan jumlah tanaman: 4
masukan tinggi tanaman ke -1(cm) : 1
masukan tinggi tanaman ke -2(cm) : 2
masukan tinggi tanaman ke -3(cm) : 3
masukan tinggi tanaman ke -4(cm) : 4

Tinggi Tanaman Tertinggi: 4.00 cm
Tinggi tanaman Terpendek: 1.00 cm
```

Deskripsi Program: Program ini digunakan untuk mencatat tinggi beberapa tanaman. Setelah semua tinggi tanaman dimasukkan, program akan menampilkan tinggi tanaman yang paling pendek dan yang paling tinggi. Prosesnya dilakukan dengan membandingkan satu per satu tinggi tanaman dan mencari nilai terkecil dan terbesar.

Source Code Guided 2:

```
//103112400087_Muhammad Shabrian

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("\nHarga Termurah:%2f Rp\n ", min)
}

```

Output:

```

PS D:\Coding manja> go run "d:\Coding manja\semester2\modul7\guided2.go"
masukkan jumlah buku dan jumlah buku per rak: 3
3

masukkan harga setiap buku (dalam ribuan Rp):
30000
12000
15000

Rata-rata harga per rak: 19000.00
Harga Termahal: 30000.00 Rp

Harga Termurah: 12000.00 Rp

```

Deskripsi Program: Program ini mencatat harga buku dan membaginya ke dalam beberapa rak. Setiap rak menampung sejumlah buku sesuai jumlah yang ditentukan. Program menghitung rata-rata harga buku per rak, serta mencari harga buku termahal dan termurah secara keseluruhan. Data dihitung satu per satu berdasarkan urutan input.

Source code guided 3 :

```
//103112400087_Muhammad Shabrian

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("\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 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 D:\Coding manja> go run "d:\Coding manja\semester2\modul7\guided3.go"
masukkan jumlah siswa: 3

masukan nilai ujian masing-masing siswa:
80 85 65

Nilai terendah: 65
nilai tertinggi: 85
rata-rata kelas : 76.67
jumlah siswa di atas rata-rata: 2

```

Deskripsi Program: Program ini digunakan untuk mencatat nilai ujian dari beberapa siswa. Setelah semua nilai dimasukkan, program akan menampilkan nilai tertinggi, nilai terendah, rata-rata nilai, serta menghitung berapa banyak siswa yang nilainya di atas rata-rata.

III. UNGUIDED

Source Code Unguided 1:

```
//103112400087_Muhammad Shabrian

package main

import "fmt"

func main() {

    var n int

    var berat [1000]float64


    fmt.Print("Masukkan jumlah anak kelinci: ")

    fmt.Scan(&n)


    fmt.Print("Masukkan berat anak kelinci ke-1: ")

    fmt.Scan(&berat[0])

    min, max := berat[0], berat[0]


    for i := 1; i < n; i++ {

        fmt.Printf("Masukkan berat anak kelinci ke-%d: ", i+1)

        fmt.Scan(&berat[i])


        if berat[i] < min {

            min = berat[i]

        }

        if berat[i] > max {

            max = berat[i]

        }

    }

}
```

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

    fmt.Printf("Berat terbesar: %.2f kg\n", max)

}
```

Output:

```
PS D:\Coding manja> go run "d:\Coding manja\semester2\modul7\unguided1.go"
Masukkan jumlah anak kelinci: 4
Masukkan berat anak kelinci ke-1: 3
Masukkan berat anak kelinci ke-2: 4
Masukkan berat anak kelinci ke-3: 5
Masukkan berat anak kelinci ke-4: 1
Berat terkecil: 1.00 kg
Berat terbesar: 5.00 kg
PS D:\Coding manja>
```

Deskripsi Program: Program ini digunakan untuk mencatat berat beberapa anak kelinci. Setelah data berat dimasukkan, program akan mencari dan menampilkan berat terkecil dan berat terbesar dari seluruh anak kelinci yang diinput. Proses pencarian dilakukan dengan cara membandingkan setiap data berat satu per satu.

Source Code Unguided 2 :

```
//103112400087_Muhammad Shabrian

package main

import "fmt"

func main() {

    var berat [1000]float64

    var x, y int

    fmt.Print("Masukkan jumlah ikan dan kapasitas wadah: ")

    fmt.Scan(&x, &y)

    for i := 0; i < x; i++ {

        fmt.Scan(&berat[i])

    }

    jumlahWadah := (x + y - 1) / y

    totalPerWadah := make([]float64, jumlahWadah)

    for i := 0; i < x; i++ {

        wadahKe := i / y

        totalPerWadah[wadahKe] += berat[i]

    }

    total := 0.0

    for i := 0; i < jumlahWadah; i++ {

        fmt.Printf("Wadah ke-%d: %.2f kg\n", i+1, totalPerWadah[i])

    }

}
```

```
        total += totalPerWadah[i]
    }

    rata2 := total / float64(jumlahWadah)

    fmt.Printf("Rata-rata berat per wadah: %.2f kg\n", rata2)
}
```

Output:

```
PS D:\Coding manja> go run "d:\Coding manja\semester2\modul7\unguided2.go"
Masukkan jumlah ikan dan kapasitas wadah: 4 3
1
2
3
5
Wadah ke-1: 6.00 kg
Wadah ke-2: 5.00 kg
Rata-rata berat per wadah: 5.50 kg
```

Deskripsi Program: Program ini digunakan untuk menghitung total berat ikan yang dimasukkan ke dalam beberapa wadah berdasarkan jumlah ikan dan kapasitas tiap wadah. Program juga menghitung rata-rata berat per wadah. Proses pembagian ikan dilakukan sesuai urutan input, dan perhitungan dilakukan secara bertahap untuk tiap kelompok ikan.

Source Code unguided 3:

```
//103112400087_Muhammad Shabrian

package main

import "fmt"

func main() {
    var n int
    var berat [100]float64

    fmt.Print("Masukan banyak data berat balita: ")
    fmt.Scan(&n)

    total := 0.0
    min, max := 0.0, 0.0

    for i := 0; i < n; i++ {
        fmt.Printf("Masukan berat balita ke-%d: ", i+1)
        fmt.Scan(&berat[i])
        total += berat[i]

        if i == 0 || berat[i] < min {
            min = berat[i]
        }

        if i == 0 || berat[i] > max {
```

```
        max = berat[i]
    }
}

fmt.Printf("Berat balita minimum: %.2f kg\n", min)

fmt.Printf("Berat balita maksimum: %.2f kg\n", max)

fmt.Printf("Rata-rata berat balita: %.2f kg\n", total/float64(n))
}
```

Output:

```
PS D:\Coding manja> go run "d:\Coding manja\semester2\modul7\unguided3.go"
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
Rata-rata berat balita: 11.00 kg
```

Deskripsi Program: Program ini digunakan untuk mencatat data berat balita. Setelah semua data diinput, program akan menampilkan berat minimum, maksimum, dan rata-rata dari seluruh balita. Proses pencarian minimum dan maksimum dilakukan sambil membaca data, sehingga lebih efisien.

IV. KESIMPULAN

Dari praktikum kali ini, saya belajar bahwa pencarian nilai terkecil, terbesar, maupun rata-rata dalam data bisa dilakukan dengan cara yang sederhana namun efektif. Kita cukup menyimpan nilai awal sebagai acuan, lalu bandingkan dengan data-data berikutnya satu per satu. Konsep ini bisa dipakai dalam banyak kasus di dunia nyata, misalnya mencari berat teringan dari hewan, membagi barang ke dalam wadah, atau mencatat pertumbuhan anak.

Melalui penggunaan array dan logika perulangan, kita bisa memproses data dalam jumlah banyak secara cepat dan efisien. Praktikum ini membantu saya memahami cara menyusun program yang tidak hanya benar, tapi juga lebih sederhana dan mudah dibaca.

V. REFERENSI

Modul praktikum algoritma dan pemrograman 2: Modul 10 – Pencarian nilai ekstrim pada himpunan data.