

**LAPORAN PRAKTIKUM
PEMROGRAMAN BERORIENTASI OBJEK
MODUL X
PENCARIAN NILAI EKSTRIM PADA HIMPUNAN DATA**



Oleh:

AULIA RADIX PUTRA WINARKO

2311102056

S1IF-11-02

**S1 TEKNIK INFORMATIKA
INSTITUT TEKNOLOGI TLKOM PURWOKERTO**

2024

I. DASAR TEORI

A. 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 matics lab

B. Pencarian Nilai Ekstrim pada Array Bertipe Data Dasar

Misalnya terdefinisi sebuah array of integer dengan kapasitas 2023, dan array berisi sejumlah N bilangan bulat, kemudian pencarian nilai terkecil dilakukan pada array tersebut. Perhatikan potongan program dalam bahasa Go berikut ini

C. Pencarian 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. Sebagai

contoh misalnya terdapat array yang digunakan untuk menyimpan data mahasiswa, kemudian terdapat fungsi IPK yang digunakan untuk mencari data mahasiswa dengan IPK tertinggi.

II. GUIDED

1. Source Code

```
package main

import ("fmt")

func main() {
    var N int
    var berat [1000]float64

    fmt.Print("Masukan Jumlah Anak Kelinci: ")
    fmt.Scan(&N)

    fmt.Println("Masukan Berat Anak Kelinci: ")
    for i := 0; i < N; i++{
        fmt.Scan(&berat[i])
    }

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

    for i := 1; i <N; i++ {
        if berat [i] <min {
            min = berat [i]
        }
        if berat[i] > max {
            max = berat[i]
        }
    }
    fmt.Printf("Berat Kelonci: %.2f\n", min)
    fmt.Printf("Berat Terbesar: %.2f\n", max)
}
```

Output

```

PS C:\Users\user\go\src\Praktikum Alpro2\modul 10> go run
Masukan Jumlah Anak Kelinci: 3
Masukan Berat Anak Kelinci:
1
3
4
Berat Kelonci: 1.00
Berat Terbesar: 4.00
PS C:\Users\user\go\src\Praktikum Alpro2\modul 10>

```

Keterangan

Dibutuhkan input bobot beberapa bayi kelinci dan dari data tersebut menentukan dan menampilkan bobot minimum dan maksimum. Program ini meminta nomor kelinci dan mengumpulkan bobot setiap kelinci ke dalam array bobot. Setelah semua data dimasukkan, program membandingkan setiap nilai dalam larik untuk menemukan nilai minimum dan maksimum, yang dikeluarkan sebagai hasil akhir.

2. Source Code

```

package main
import (
    "fmt"
)

func main() {
    var x, y int
    fmt.Print("masukkan jumlah inkan dan
    kapasitas wadah: ")
    fmt.Scan(&x, &y)

    berat := make([]float64, x)
    fmt.Println("masukkan berat tiap ikan:
    ")
    for i := 0; i < x; i++ {
        fmt.Scan(&berat[i])
    }

    jumlahWadah := (x + y - 1) / y
    totalBeratWadah := make([]float64,
    jumlahWadah)

    for i := 0; i < x; i++ {
        indekswadah := i / y

```

```

        totalBeratWadah[indekswadah] +=
berat[i]
    }

    fmt.Println("total berat tiap wadah: ")
    for _, total := range totalBeratWadah {
        fmt.Printf("%.2f", total)
    }
    fmt.Println()

    fmt.Println("rata rata berat tiap
wadah: ")
    for _, total := range totalBeratWadah {
        ratarata := total / float64(y)
        fmt.Printf("%.2f", ratarata)
    }
    fmt.Println()
}

```

Output

```

PS C:\Users\user\go\src\Praktikum Alpro2\modul 10> go run "c:
masukkan jumlah ikan dan kapasitas wadah: 1 3
masukkan berat tiap ikan:
12
total berat tiap wadah:
12.00
rata rata berat tiap wadah:
4.00
PS C:\Users\user\go\src\Praktikum Alpro2\modul 10>

```

Keterangan

Kelompokkan ikan menurut beratnya ke dalam beberapa wadah dengan kapasitas tertentu. Program meminta pengguna untuk memasukkan jumlah ikan, kapasitas wadah, dan berat masing-masing ikan. Setelah data dimasukkan, program menghitung jumlah kontainer yang dibutuhkan dan mendistribusikan ikan ke kontainer sesuai kapasitas yang ditentukan. Hasil akhir menampilkan berat total ikan di setiap wadah dan rata-rata berat per wadah, memberikan informasi akurat mengenai distribusi berat ikan.

III. UNGUIDED

1. Source Code

```
package main

import "fmt"

const maxData = 100

type datas [maxData]float64

func minMax(data datas, n int) (float64, float64) {
    min := data[0]
    max := data[0]
    for i := 1; i < n; i++ {
        if data[i] < min {
            min = data[i]
        }
        if data[i] > max {
            max = data[i]
        }
    }
    return min, max
}

func rataRata(data datas, n int) float64 {
    var sum float64
    for i := 0; i < n; i++ {
        sum += data[i]
    }
    return sum / float64(n)
}

func main() {
    var data datas
    var n int
    fmt.Print("Masukan banyak data berat balita: ")
    fmt.Scan(&n)
    for i := 0; i < n; i++ {
        fmt.Printf("Masukan berat balita ke-%d: ", i+1)
        fmt.Scan(&data[i])
    }
    min, max := minMax(data, n)
```

```
    fmt.Printf("\nBerat balita minimum:
%.2f kg\n", min)
    fmt.Printf("Berat balita maksimum: %.2f
kg\n", max)
    fmt.Printf("Rerata berat balita: %.2f
kg\n", rataRata(data, n))
}
```

Output

```
PS C:\Users\user\go\src\Praktikum Alpro2\modul 10> go run
Masukan banyak data berat balita: 4
Masukan berat balita ke-1: 5.3
Masukan berat balita ke-2: 6.2
Masukan berat balita ke-3: 4.1
Masukan berat balita ke-4: 9.9

Berat balita minimum: 4.10 kg
Berat balita maksimum: 9.90 kg
Rerata berat balita: 6.38 kg
PS C:\Users\user\go\src\Praktikum Alpro2\modul 10> █
```

Keterangan

Tubuh bayi kemudian menghitung berat minimum, maksimum, dan rata-rata dari data input pengguna. Pengguna diminta untuk memasukkan jumlah bayi dan berat badannya masing-masing. Program ini menggunakan dua fungsi utama: minMax dan avg. Perhitungan berat minimum, maksimum, dan rata-rata ditampilkan pada Formulir. Desain program yang modular dan efisien membuatnya cocok untuk analisis berat badan dasar pada anak kecil. Selain itu, hal ini dapat dikembangkan lebih lanjut untuk tujuan lain, seperti pemantauan status kesehatan dan gizi kelompok anak kecil.