

**LAPORAN PRAKTIKUM
ALGORITMA PEMROGRAMAN 2**

MODUL X

PENCARIAN NILAI EKSTRIM PADA HIMPUNAN DATA



Oleh:

ALIFATUS SHABRINA AMALIA

NIM:

2311102225

S1 TEKNIK INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2024

I. DASAR TEORI

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 kenilai 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 ekstrim tersebut dengan data yang dicek.
- 3) Apabila semua data telah dicek, maka nilai ekstrim yang dimiliki adalah valid.

II. GUIDED

1. Guided 1

Source code

```
package main

import (
    "fmt"
)

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

    fmt.Print("Masukkan jumlah anak kelinci : ")
    fmt.Scan(&N)

    fmt.Println("Masukkan berat anak kelinci : ")
    for i := 0; i < N; i++ {
        fmt.Scan(&berat[i])
    }

    // Inisialisasi nilai min dan max dengan elemen pertama
    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]

    }

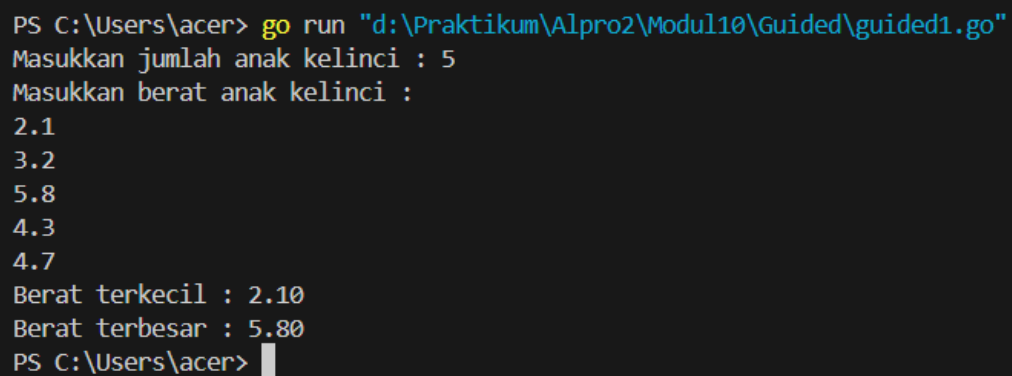
}

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

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

}
```

Screenshoot program



```
PS C:\Users\acer> go run "d:\Praktikum\Alpro2\Modul10\Guided\guided1.go"
Masukkan jumlah anak kelinci : 5
Masukkan berat anak kelinci :
2.1
3.2
5.8
4.3
4.7
Berat terkecil : 2.10
Berat terbesar : 5.80
PS C:\Users\acer> █
```

Deskripsi program

1. Jumlah anak kelinci : Program meminta pengguna memasukkan jumlah anak kelinci, dalam contoh ini 5.
2. Berat masing-masing kelinci : Pengguna memasukkan berat satu per satu:
 - Kelinci 1: 2.1
 - Kelinci 2: 3.2
 - Kelinci 3: 5.8
 - Kelinci 4: 4.3
 - Kelinci 5: 4.7
3. Hasil perhitungan :

- Berat terkecil dihitung dengan membandingkan setiap elemen, hasilnya adalah 2.10 (kelinci pertama).
- Berat terbesar dihitung dengan cara yang sama, hasilnya adalah 5.80 (kelinci ketiga).

Program menampilkan hasil akhir berupa berat terkecil dan terbesar dengan dua angka desimal.

2. Guided 2

Source code

```
package main

import (
    "fmt"
    "math"
)

func main() {
    var x, y int

    fmt.Printf("Masukkan jumlah ikan dan kapasitas wadah: ")
    fmt.Scan(&x, &y)

    // Inisialisasi slice berat
    berat := make([]float64, x)

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

    // Hitung jumlah wadah dengan pembulatan ke atas
```

```

jumlahWadah := int(math.Ceil(float64(x) / float64(y)))

totalBeratWadah := make([]float64, jumlahWadah)

// Distribusi berat ikan ke wadah
for i := 0; i < x; i++ {
    indeksWadah := i / y
    totalBeratWadah[indeksWadah] += berat[i]
}

// Output total berat tiap wadah
fmt.Println("Total berat tiap wadah:")
for i, total := range totalBeratWadah {
    fmt.Printf("Wadah %d: %.2f\n", i+1, total)
}

// Output rata-rata berat tiap wadah
fmt.Println("Rata-rata berat tiap wadah:")
for i, total := range totalBeratWadah {
    // Jumlah ikan dalam wadah terakhir bisa kurang dari
    kapasitas maksimal

    jumlahIkan := y

    if i == jumlahWadah-1 && x%y != 0 {
        jumlahIkan = x % y
    }

    rataRata := total / float64(jumlahIkan)

    fmt.Printf("Wadah %d: %.2f\n", i+1, rataRata)
}
}

```

Screenshoot program

```
PS C:\Users\acer> go run "d:\Praktikum\Alpro2\Modul10\Guided\guided2.go"
Masukkan jumlah ikan dan kapasitas wadah: 5 3
Masukkan berat tiap ikan:
1.8
2.5
4.3
3.1
2.6
Total berat tiap wadah:
Wadah 1: 8.60
Wadah 2: 5.70
Rata-rata berat tiap wadah:
Wadah 1: 2.87
Wadah 2: 2.85
PS C:\Users\acer> █
```

Deskripsi program

1. Jumlah ikan: 5, kapasitas per wadah: 3, sehingga diperlukan 2 wadah.
 - Wadah 1 menampung 3 ikan pertama: 1.8, 2.5, 4.3.
 - Wadah 2 menampung 2 ikan sisanya: 3.1, 2.6.
2. Perhitungan total berat tiap wadah:
 - Wadah 1: $1.8 + 2.5 + 4.3 = 8.60$
 - Wadah 2: $3.1 + 2.6 = 5.70$
3. Perhitungan rata-rata berat tiap wadah:
 - Wadah 1: $8.60 \div 3 = 2.87$
 - Wadah 2: $5.70 \div 2 = 2.85$

Program ini berhasil menghitung total berat dan rata-rata berat untuk tiap wadah berdasarkan input berat ikan dan kapasitas wadah.

III. UNGUIDED

1. Unguided 1

Source code

```
package main

import (
    "fmt"
)

type arrBalita225 [100]float64

func hitungMinMax(arrBerat arrBalita225, n int, bMin, bMax
*float64) {
    *bMin = arrBerat[0]
    *bMax = arrBerat[0]
    for i := 1; i < n; i++ {
        if arrBerat[i] < *bMin {
            *bMin = arrBerat[i]
        }
        if arrBerat[i] > *bMax {
            *bMax = arrBerat[i]
        }
    }
}

func rerata(arrBerat arrBalita225, n int) float64 {
    var total float64
    for i := 0; i < n; i++ {
```



```
        total += arrBerat[i]
    }

    return total / float64(n)
}

func main() {
    var n int

    var berat arrBalita225

    var bMin, bMax float64

    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(&berat[i])
    }

    hitungMinMax(berat, n, &bMin, &bMax)

    rata := rerata(berat, n)

    fmt.Printf("Berat balita minimum: %.2f kg\n", bMin)
    fmt.Printf("Berat balita maksimum: %.2f kg\n", bMax)
    fmt.Printf("Rerata berat balita: %.2f kg\n", rata)
}
```

Screenshoot program

```
PS D:\Praktikum\Alpro2\Modul10\Unguided> go run "d:\Praktikum\Alpro2\Modul10\Unguided\unguided1.go"
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 D:\Praktikum\Alpro2\Modul10\Unguided> █
```

Deskripsi program

1. Tipe Data dan Fungsi:

- arrBalita: Array dengan kapasitas 100 untuk menyimpan berat balita.
- Fungsi hitungMinMax:
 - Menghitung berat minimum dan maksimum dari array.
- Fungsi rerata:
 - Menghitung rata-rata berat balita.

2. Alur Program:

- Program meminta pengguna memasukkan jumlah balita (n) dan berat balita satu per satu.
- Berat minimum dan maksimum dihitung menggunakan hitungMinMax melalui parameter pointer.
- Rata-rata dihitung menggunakan fungsi rerata.
- Program menampilkan berat minimum, maksimum, dan rata-rata.