

LAPORAN PRAKTIKUM
ALGORITMA PEMROGRAMAN 2
MODUL 10
PENCARIAN NILAI EKSTRIM PADA HIMPUNAN DATA



Oleh :

Dimas Bagus Firmansyah

2311102002

S1 IF 11 02

S1 TEKNIK INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2024

I. DASAR TEORI

10.1 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. 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.

10.2 pencarian Nilai Ekstrim pada Array Bertipe Data Dasar

```
5  type arrInt [2023]int
..  ...
15
16 func terkecil_1(tabInt arrInt, n int) int {
17  /* mengembalikan nilai terkecil yang terdapat di dalam tabInt yang berisi n
18  bilangan bulat */
19      var min int = tabInt[0]          // min berisi data pertama
20      var j int = 1                    // pencarian dimulai dari data berikutnya
21      for j < n {
22          if min > tabInt[j] {          // pengecekan apakah nilai minimum valid
23              min = tabInt[j]          // update nilai minimum dengan yang valid
24          }
25          j = j + 1
26      }
27      return min                       // returnkan nilai minimumnya
28 }
```

10.3 Pencarian Nilai Ekstrim pada Array Data Terstruktur

```
..  ...
5  type mahasiswa struct {
..      nama, nim, kelas, jurusan string
..      ipk float64
..  }
..  type arrMhs [2023]mahasiswa
..  ...
15
16 func IPK_1(T arrMhs, n int) float64 {
17  /* mengembalikan ipk terkecil yang dimiliki mahasiswa pada array T yang berisi
18  n mahasiswa */
19      var tertinggi float64 = T[0].ipk
20      var j int = 1
21      for j < n {
22          if tertinggi < T[j].ipk {
23              tertinggi = T[j].ipk
24          }
25          j = j + 1
26      }
27      return tertinggi
28 }
```

II. GUIDED

GUIDED 1

```
package main

import "fmt"

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

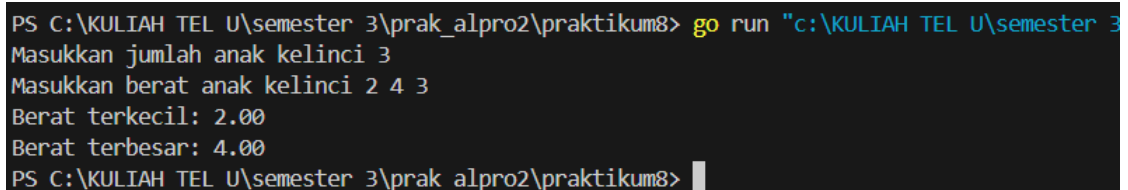
    fmt.Print("Masukkan jumlah anak kelinci ")
    fmt.Scan(&N)
    fmt.Print("Masukkan berat anak kelinci ")
    for i := 0; i < N; i++ {
        fmt.Scan(&berat[i])
    }

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

    for i := 0; 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:\KULIAH TEL U\semester 3\prak_alpro2\praktikum8> go run "c:\KULIAH TEL U\semester 3
Masukkan jumlah anak kelinci 3
Masukkan berat anak kelinci 2 4 3
Berat terkecil: 2.00
Berat terbesar: 4.00
PS C:\KULIAH TEL U\semester 3\prak_alpro2\praktikum8> █
```

DESKRIPSI PROGRAM

Berdasarkan input pengguna, program ini menghitung berat terkecil dan terbesar dari sekelompok anak kelinci. Pengguna harus memasukkan jumlah anak kelinci (*N*) dan berat masing-masing kelinci. Program menghitung berat kelinci dalam array dan menggunakan iterasi untuk menghitung dan membandingkan nilai minimum dan maksimum. Dalam format dua angka desimal, hasil akhir menunjukkan berat terkecil dan terbesar.

GUIDED 2

```
package main

import "fmt"

func main() {
    var x, y int
    fmt.Print("Masukkan jumlah ikan 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])
    }

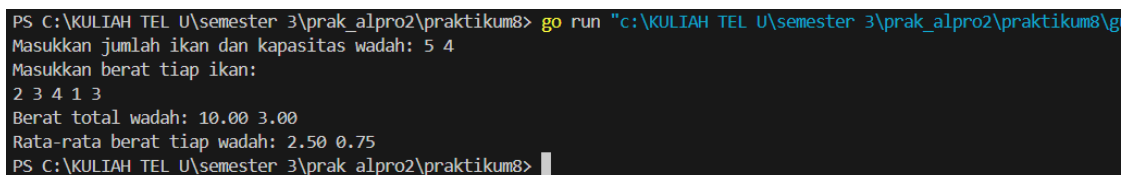
    jumlahWadah := (x + y - 1) / y
    totalBeratWadah := make([]float64, jumlahWadah)

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

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

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

SCREENSHOOT PROGRAM



```
PS C:\KULIAH TEL U\semester 3\prak_alpro2\praktikum8> go run "c:\KULIAH TEL U\semester 3\prak_alpro2\praktikum8\g
Masukkan jumlah ikan dan kapasitas wadah: 5 4
Masukkan berat tiap ikan:
2 3 4 1 3
Berat total wadah: 10.00 3.00
Rata-rata berat tiap wadah: 2.50 0.75
PS C:\KULIAH TEL U\semester 3\prak_alpro2\praktikum8> █
```

DESKRIPSI PROGRAM

Program ini menghitung jumlah ikan dan kapasitas maksimum setiap wadah berdasarkan jumlah ikan dan berat masing-masing ikan. Pengguna memberikan jumlah ikan (*x*), kapasitas wadah (*y*), dan berat masing-masing ikan. Program kemudian menghitung jumlah wadah yang dibutuhkan, membagi ikan secara berurutan ke dalam masing-masing wadah, dan menghitung total berat ikan di setiap wadah. Selanjutnya, program menghitung rata-rata berat per wadah berdasarkan jumlah ikan

III. UNGUIDED

UNGUIDED 1

```
package main

import (
    "fmt"
)

// Tipe data array untuk berat balita
type arrBalita [100]float64

// Fungsi untuk menghitung berat minimum dan maksimum
func hitungMinMax(arrBerat []float64, bMin, bMax *float64) {
    *bMin = arrBerat[0]
    *bMax = arrBerat[0]
    for _, berat := range arrBerat {
        if berat < *bMin {
            *bMin = berat
        }
        if berat > *bMax {
            *bMax = berat
        }
    }
}

// Fungsi untuk menghitung rata-rata berat balita
func rataRata(arrBerat []float64) float64 {
    total := 0.0
    for _, berat := range arrBerat {
        total += berat
    }
    return total / float64(len(arrBerat))
}

func main() {
    var n int
    fmt.Print("Masukan banyak data berat balita: ")
    fmt.Scan(&n)

    var arrBerat []float64
    for i := 0; i < n; i++ {
        var berat float64
        fmt.Printf("Masukan berat balita ke-%d: ", i+1)
        fmt.Scan(&berat)
        arrBerat = append(arrBerat, berat)
    }

    var bMin, bMax float64
    hitungMinMax(arrBerat, &bMin, &bMax)

    rata := rataRata(arrBerat)

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

SCREENSHOOT PROGRAM

```
PS C:\KULIAH TEL U\semester 3\prak_alpro2\praktikum8> go run "c:\KULIAH TEL U\semester 3\prak_alpro2\pra
Masukan banyak data berat balita: 5
Masukan berat balita ke-1: 20.6
Masukan berat balita ke-2: 25.7
Masukan berat balita ke-3: 22.4
Masukan berat balita ke-4: 30.0
Masukan berat balita ke-5: 28.1
Berat balita minimum: 20.60 kg
Berat balita maksimum: 30.00 kg
Berat rata-rata balita: 25.36 kg
PS C:\KULIAH TEL U\semester 3\prak_alpro2\praktikum8> |
```

DESKRIPSI PROGRAM

Tiga hal utama—berat minimum, berat maksimum, dan rata-rata—digunakan oleh program untuk menghitung berat balita. Untuk memulai, pengguna diminta untuk memasukkan jumlah balita (*n*) yang akan digunakan untuk input data; setelah itu, mereka diminta untuk memasukkan berat setiap balita satu per satu, yang disimpan dalam slice "arrBerat". Setelah data berat balita terkumpul, program akan menggunakan fungsi "hitungMinMax" dan "rataRata" untuk menghitung nilai berat minimum dan maksimum dengan menggunakan pointer. Nilai berat rata-rata dihitung dengan menjumlahkan semua. Selanjutnya, program menampilkan berat balita terkecil (minimum), terbesar (maksimum), dan rata-rata dalam bentuk angka desimal.