

**LAPORAN PRAKTIKUM
PEMROGRAMAN BERORIENTASI OBJEK**

**MODUL 10
PENCARIAN NILAI EKSTRIM**



Disusun Oleh:

NAMA : MARIA DWI A

NIM : 2311102228

KELAS : S1- 1F – 11 – 02

S1 TEKNIK INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2024

I. DASAR TEORI

A. Ide Pencarian Nilai Max/Min

Pencarian adalah suatu proses yang umum 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.

Ide algoritma sederhana, karena data harus diproses secara sekuensial, maka nilai atau indeks ke nilai maksimum dari data yang telah diproses disimpan sampai algoritma tersebut berakhir adalah nilai maksimum yang dicari. Algoritma secara umum :

- 1) Jadikan nilai 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.

Berikut ini adalah notasi dalam pseudocode dan bahasa Go, misalnya untuk pencarian nilai terbesar atau maksimum :

	Notasi Algoritma	Notasi dalam bahasa Go
1	$\text{max} \leftarrow 1$	$\text{max} = 0$
2	$i \leftarrow 2$	$i = 1$
3	while $i \leq n$ do	for $i < n$ {
4	if $a[i] > a[\text{max}]$ then	if $a[i] > a[\text{max}]$ {
5	$\text{max} \leftarrow i$	$\text{max} = i$
6	endif	}
7	$i \leftarrow i + 1$	$i = i + 1$
8	endwhile	}

B. 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. Perhatikan potongan program dalam bahasa Go berikut ini :

```
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 }
```

Potongan program di atas sedikit berbeda dengan sebelumnya karena penggunaan indeks array pada bahasa Go di mulai dari nol atau "0" . Maka program di atas dapat dimodifikasi menjadi seperti program berikut ini!

```
..  ...
5  type arrInt [2023]int
..  ...
15
16 func terkecil_2(tabInt arrInt, n int) int {
17  /* mengembalikan indeks nilai terkecil yang terdapat di dalam tabInt yang berisi
18  n bilangan bulat */
19      var idx int = 0                  // idx berisi indeks data pertama
20      var j int = 1                    // pencarian dimulai dari data berikutnya
21      for j < n {
22          if tabInt[idx] > tabInt[j] { // pengecekan apakah nilai minimum valid
23              idx = j                  // update nilai minimum dengan yang valid
24          }
25          j = j + 1
26      }
27      return idx                       // returnkan indeks nilai minimumnya
28 }
```

C. Pencarian Nilai Ekstrim pada Array Bertipe Data Terstruktur

Pada kasus yang lebih kompleks pencarian nilai ekstrim dapat juga dilakukan, misalnya mencari dapat mahasiswa dengan nilai terbesar, mencari lagu dengan durasi terlama, mencari pembalap yang memiliki catatan waktu balap tercepat, dan sebagainya. Sebagai contoh misal terdapat array yang digunakan untuk menyimpan data mahasiswa, kemudian terdapat fungsi IPK yang digunakan untuk mencari data mahasiswa dengan IPK tertinggi.

```
..  ...
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 }
```

Apabila diperhatikan potongan program diatas, maka akan diperoleh nilai ipk tertinggi, tetapi kita tidak akan memperoleh identitas mahasiswa dengan ipk tertinggi tersebut. Maka program dapat dimodifikasi menjadi :

```

.. ...
5  type mahasiswa struct {
..      nama, nim, kelas, jurusan string
..      ipk float64
..  }
..  type arrMhs [2023]mahasiswa
..  ...
15
16  func IPK_2(T arrMhs, n int) int {
17      /* mengembalikan indeks mahasiswa yang memiliki ipk tertinggi pada array T yang
18      berisi n mahasiswa */
19      var idx int = 0
20      var j int = 1
21      for j < n {
22          if T[idx].ipk < T[j].ipk {
23              idx = j
24          }
25          j = j + 1
26      }
27      return idx
28  }

```

Sehingga melalui algoritma di atas, identitas mahasiswa dapat diperoleh, misalnya T[idx].nama, T[idx].nim, T[idx].kelas, hingga T[idx].jurusan.

II. GUIDED

1. Source Code

```
package main

import "fmt"

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

    fmt.Print("Masukkan jumlah anak kelici : ")
    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 min > berat[i] {
            min = berat[i]
        }
        if max < berat[i] {
            max = berat[i]
        }
    }
    fmt.Printf("Berat terkecil : %.2f\n", min)
    fmt.Printf("Berat terbesar : %.2f\n", max)
}
```

Screenshoot Program

```
PS D:\SEMESTER 3\LATIHAN GOLANG> go run "d:\SEMESTER 3\ALPRO 2 (PRAKTIK)"
Masukkan jumlah anak kelici : 5
Masukkan berat anak kelinci: 3.6 7.8 5.6 8.9 9.0

Berat terkecil : 3.60
Berat terbesar : 9.00
```

Deskripsi Program

Program diatas merupakan implementasi dari pencarian nilai ekstrim yang digunakan untuk mendata berat anak kelinci yang akan dijual. Program menggunakan array dengan kapasitas 1000 untuk menampung data. Program meminta user untuk menginputkan jumlah data yang akan disimpan beserta masing-masing berat data. Lalu program akan mencari nilai ekstrim dari data yaitu mencari serta menampilkan nilai minimal dan maksimal dari data yang diinputkan oleh user.

2. Source Code :

```
package main

import ("fmt")

func main() {
    var x, y int

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

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

```

Screenshoot Program

```

PS D:\SEMESTER 3\LATIHAN GOLANG> go run "d:\SEMESTER 3\ALPRO 2 (PRAK
Masukkan jumlah ikan dan kapasitas wadah: 5 10
Masukkan berat tiap ikan :
4.5 6.0 9.7 8.4 7.6

Total berat tiap wadah : 36.20
Rata-rata berat tiap wadah : 3.62

```

Deskripsi Program

Program diatas merupakan program yang digunakan untuk menampung data berat ikan dengan menggunakan array dengan kapasitas 1000. Program akan meminta user meinginputkan jumlah ikan dan kapasitas wadah. Lalu program akan menghitung dan menampilkan output total dari berat ikan yang diinputkan, serta rata-rata berat yang bisa ditampung didalam setiap wadah.

III. UNGUIDED Soal Modul 10

1. Pos Pelayanan Terpadu (posyandu) sebagai tempat pelayanan kesehatan perlu mencatat data berat balita (dalam kg). Petugas akan memasukkan ke dalam array. Dari data yang diperoleh akan dicari berat balita terkecil, terbesar, dan reratanya.

Source Code

```
package main

import "fmt"

type arrBalita [100]float64

func cariMin(N int, berat *arrBalita) float64{
    min := berat[0]

    for i := 0; i < N; i++ {
        if min > berat[i]{
            min = berat[i]
        }
    }
    return min
}

func cariMax(N int, berat *arrBalita)float64 {
    max := berat[0]

    for i := 0; i < N; i++ {
        if max < berat[i] {
            max = berat[i]
        }
    }
    return max
}

func rataRata(N int, berat *arrBalita) float64 {
    totalBerat := 0.0

    for i := 0; i < N; i++ {
        totalBerat += berat[i]
    }
}
```

```

        return totalBerat / float64(N)
    }

func main() {

    var jumlah int
    var berat arrBalita

    fmt.Println("\n~Program Menyimpan Data Posyandu~")
    fmt.Println("\n\nMasukkan jumlah balita : ")
    fmt.Scan(&jumlah)

    for i := 0; i < jumlah; i++ {
        fmt.Println("Masukkan berat balita ke- ", i+1, " : ")
        fmt.Scan(&berat[i])
    }

    minimal := cariMin(jumlah, &berat)
    maksimal := cariMax(jumlah, &berat)
    rata := rataRata(jumlah, &berat)

    fmt.Printf("\nBerat balita maksimum : %.2f kg\n",
minimal)
    fmt.Printf("Berat balita maksimum : %.2f kg\n",
maksimal)
    fmt.Printf("Rata-rata berat balita : %.2f kg\n\n",
rata)

}

```

Screenshoot Program

```

UM)\Maria Dwi A_2311102228_Modul 10\Unguided\unguided1_modul10.go"

~Program Menyimpan Data Posyandu~

Masukkan jumlah balita : 4
Masukkan berat balita ke- 1: 5.3
Masukkan berat balita ke- 2: 6.2
Masukkan berat balita ke- 3: 4.1
Masukkan berat balita ke- 4: 9.9

Berat balita maksimum : 4.10 kg
Berat balita maksimum : 9.90 kg
Rata-rata berat balita : 6.38 kg

```

Deskripsi Program

Program diatas suatu program yang digunakan untuk menampung data posyandu dengan menggunakan array. Program menggunakan type untuk menyimpan array balita dengan kapasitas 100. Program terdiri dari 3 fungsi untuk mencari nilai ekstrim yaitu fungsi untuk mencari nilai minimal, maksimal serta rata rata dari data array balita yang diinputkan oleh user. Program akan meminta user untuk menginputkan jumlah data yang akan disimpan beserta berat dari setiap data balita. Kemudian fungsi cariMin, cariMax, dan rataRata akan dipanggil untuk menampilkan outputan.