

**LAPORAN PRAKTIKUM  
PEMROGRAMAN BERORIENTASI OBJEK  
MODUL 10  
PENCARIAN NILAI EXSTRIM PADA HIMPUNAN DATA**



Oleh:

MUHAMAD IHSAN

2311102077

IF - 11 – 02

**S1 TEKNIK INFORMATIKA  
TELKOM UNIVERSITY PURWOKERTO  
2024**

## I. DASAR TEORI

### 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 ekstrims tersebut dengan data yang dicek.
- 3) Apabila semua data telah dicek, maka nilai ekstrim yang dimiliki adalah valid, notis lob 3

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	}

## 2. 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" seperti penjelasan pada modul 9. Selanjutnya, pada penjelasan di awal bab 3 telah disampaikan bahwa pada pencarian yang terpenting adalah posisi atau Indeks dari nilai yang dicari dalam kumpulan data atau array. Oleh karena itu modifikasi pada program di atas dapat dilihat pada potongan 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  }
```

### 3. Pencarian Nilai 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.

```
.. ...
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 di atas, maka kita akan memperoleh ipk tertinggi, tetapi kita tidak memperoleh identitas mahasiswa dengan ipk tertinggi tersebut. Maka seperti penjelasan yang sudah diberikan sebelumnya, maka pencarian yang dilakukan bisa mengembalikan indeks mahasiswa dengan ipk tertinggi tersebut. Berikut ini adalah modifikasinya!

```
.. ...
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

### Guided 1

```
package main

import (
    "fmt"
)

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

    fmt.Println("masuka 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.Println("berat terkecil: %.2f", min)
    fmt.Println("berat terbesar: %.2f", max)
}
```

### Screenshot output

```

PS C:\Users\ASUS\OneDrive\Documents\SEMESTER 3 Tel-u\PRAKTIKUM ALPRO2\Golang> go run main.go
masuka jumlah anak kelinci:
2
masukan berat anak kelinci:
5
10
berat terkecil: %.2
5
berat terbesar: %.2
5
PS C:\Users\ASUS\OneDrive\Documents\SEMESTER 3 Tel-u\PRAKTIKUM ALPRO2\Golang>

```

## Deskripsi program

Program di atas menentukan berat terkecil dan terbesar dari anak-anak kelinci. Setelah pengguna memasukkan jumlah dan berat tiap anak kelinci, program membandingkan setiap berat untuk menemukan nilai terkecil dan terbesar, lalu menampilkannya dalam dua desimal.

## 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.Println("total berat tiap wadah: ")
    for _, total := range totalBeratWadah {

```

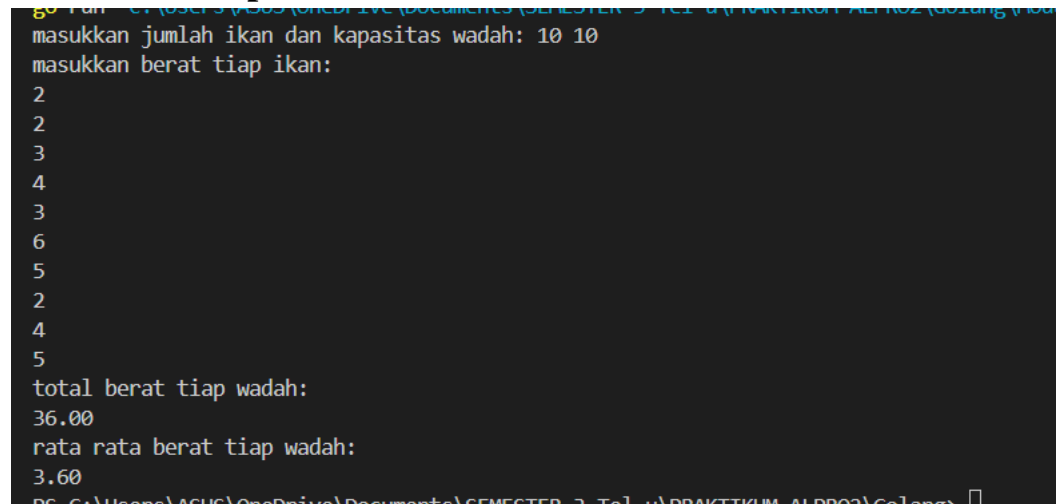
```

        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()
}

```

### Screenshot output



The screenshot shows the output of a Go program. It starts with a prompt 'masukkan jumlah ikan dan kapasitas wadah: 10 10' and 'masukkan berat tiap ikan:'. The user enters the following weights: 2, 2, 3, 4, 3, 6, 5, 2, 4, 5. The program then outputs 'total berat tiap wadah: 36.00' and 'rata rata berat tiap wadah: 3.60'.

```

golang: C:\Users\ASUS\OneDrive\Documents\SEMESTER 3_Tel_u\PPA\TEKNIK ALPRO2\Golang\100
masukkan jumlah ikan dan kapasitas wadah: 10 10
masukkan berat tiap ikan:
2
2
3
4
3
6
5
2
4
5
total berat tiap wadah:
36.00
rata rata berat tiap wadah:
3.60

```

### Deskripsi program

Program ini menghitung total dan rata-rata berat ikan di setiap wadah berdasarkan jumlah ikan dan kapasitas wadah yang dimasukkan pengguna. Program mendistribusikan berat ikan ke wadah, kemudian menampilkan total berat dan rata-rata berat per wadah dengan dua angka desimal.

### III. UNGUIDED

#### Soal Modul 10

#### Source code no 3

```
package main

import "fmt"

const maxData = 100

type datas [maxData] float64

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

    max := data[0]
    for i:=1; i < n; i++ {
        if max < data[i] {
            max = data[i]
        }
    }

    fmt.Printf("berat balita minimum: %.2f\n",min)
    fmt.Printf("berat balita maximum: %.2f\n",max)
}

func rataRata(data datas, n int) float64 {
    var sum,hasil float64

    for i:=0; i < n; i++ {
        sum += data[i]
    }
    hasil = float64(sum) / float64(n)

    return hasil
}
```



```

func main() {
    var data datas
    var n int

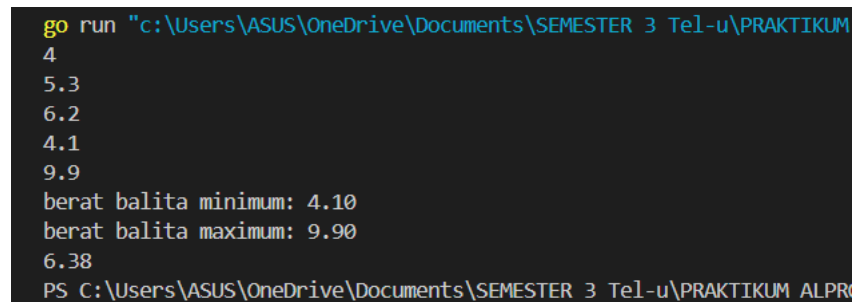
    fmt.Scan(&n)

    for i:=0; i < n; i++ {
        fmt.Scan(&data[i])
    }

    minMax(data,n)
    fmt.Printf("%.2f",rataRata(data,n))
}

```

## Screenshot output



```

go run "c:\Users\ASUS\OneDrive\Documents\SEMESTER 3 Tel-u\PRAKTIKUM
4
5.3
6.2
4.1
9.9
berat balita minimum: 4.10
berat balita maximum: 9.90
6.38
PS C:\Users\ASUS\OneDrive\Documents\SEMESTER 3 Tel-u\PRAKTIKUM ALPR

```

## Deskripsi

Program ini menghitung berat balita minimum, maksimum, dan rata-rata dari sekumpulan data. Pengguna memasukkan jumlah data dan berat balita, lalu program menggunakan fungsi minMax untuk mencari berat terkecil dan terbesar, serta fungsi rataRata untuk menghitung rata-rata berat balita. Hasilnya ditampilkan dengan format dua angka desimal.