

**LAPORAN PRAKTIKUM  
PEMROGRAMAN BERORIENTASI  
OBJEK**

**MODUL X  
PENCARIAN NILAI MAX/MIN**



**Oleh :**

**NAMA : FAISAL KHOIRUDDIN**

**NIM : 2311102046**

**Kelas : IF-11-02**

**S1 TEKNIK INFORMATIKA  
INSTITUT TEKNOLOGI TELKOM 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. 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

### 1. Guided 1

Sebuah program digunakan untuk mendata berat anak kelinci yang akan dijual ke pasar. Program ini menggunakan array dengan kapasitas 1000 untuk menampung data berat anak kelinci yang akan dijual.

Masukan terdiri dari sekumpulan bilangan, yang mana bilangan pertama adalah bilangan bulat N yang menyatakan banyaknya anak kelinci yang akan ditimbang beratnya. Selanjutnya N bilangan riil berikutnya adalah berat dari anak kelinci yang akan dijual.

Keluaran terdiri dari dua buah bilangan riil yang menyatakan berat kelinci terkecil dan terbesar.

#### Source Code

```
package main

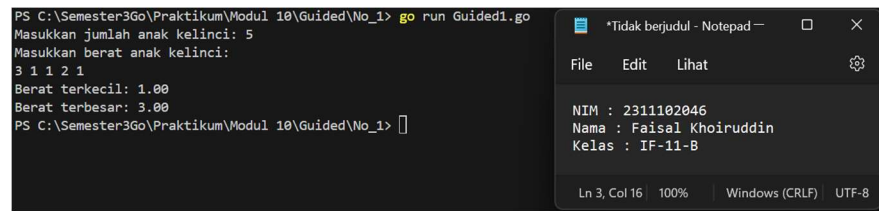
import (
    "fmt"
)

func main() {
    var N int
    var berat [1000]float64 // inisialisasi
    array data

    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]  
        // mengecek inputan array untuk setiap  
array yang ada  
        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)  
    }
```

Screenshots Output



```
PS C:\Semester3Go\Praktikum\Modul 10\Guided\No_1> go run Guided1.go
Masukkan jumlah anak kelinci: 5
Masukkan berat anak kelinci:
3 1 1 2 1
Berat terkecil: 1.00
Berat terbesar: 3.00
PS C:\Semester3Go\Praktikum\Modul 10\Guided\No_1>
```

\*Tidak berjudul - Notepad

File Edit Lihat

NIM : 2311102046  
Nama : Faisal Khoiruddin  
Kelas : IF-11-B

Ln 3, Col 16 100% Windows (CRLF) UTF-8

Deskripsi:

Program tersebut merupakan program untuk mendata berat anak kelinci yang akan dijual ke pasar. Pengguna diminta menginputkan banyaknya anak kelinci yang akan ditimbang beratnya, kemudian menginputkan berat dari anak kelinci menggunakan bilangan riil. Output dari program tersebut yaitu berat kelinci terkecil dan berat kelinci terbesar.

- package main → paket utama program golang
- import → mengimpor
- "fmt" → mengimpor fmt
- func main() { → merupakan fungsi utama
- var N int → deklarasi variabel N bertipe data integer
- var berat [1000]float64 → inisialisasi array data dengan kapasitas 1000
- fmt.Print("Masukkan jumlah anak kelinci: ") → menampilkan statement untuk masukkan jumlah anak kelinci
- fmt.Scan(&N) → membaca input dan menyimpan ke variabel N
- fmt.Println("Masukkan berat anak kelinci: ") → menampilkan statement untuk Masukkan berat anak kelinci:
- for i := 0; i < N; i++ { → perulangan for untuk membaca berat masing-masing kelinci
- fmt.Scan(&berat[i]) → membaca input berat dan menyimpan di



array berat

- `min := berat[0]` → inisialisasi nilai min dengan elemen pertama
- `max := berat[0]` → inisialisasi nilai max dengan elemen pertama
- `// mengecek inputan array untuk setiap array yang ada`
- `for i := 1; i < N; i++ {` → perulangan for untuk mengecek setiap elemen
- `if berat[i] < min {` → jika berat saat ini lebih kecil dari min
- `min = berat[i]` → update nilai min
- `if berat[i] > max {` → jika berat saat ini lebih besar dari min
- `max = berat[i]` → update nilai max
- `fmt.Printf("Berat terkecil: %.2f\n", min)` → menampilkan Berat terkecil
- `fmt.Printf("Berat terbesar: %.2f\n", max)` → menampilkan Berat terbesar

## 2. Guided 2

Sebuah program digunakan untuk menentukan tarif ikan yang akan dijual ke pasar. Program ini menggunakan array dengan kapasitas 1000 untuk menampung data berat ikan yang akan Masukan terdiri dari dua baris, yang mana baris pertama terdiri dari dua bilangan bulat x dan y. Bilangan x menyatakan banyaknya ikan yang akan dijual, sedangkan y adalah banyaknya ikan yang akan dimasukan ke dalam wadah. Baris kedua terdiri dari sejumlah x bilangan riil yang menyatakan banyaknya ikan yang akan dijual. Keluaran terdiri dari dua baris. Baris pertama adalah kumpulan bilangan riil yang menyatakan total berat ikan di setiap wadah (jumlah wadah tergantung pada nilai x dan y, urutan ikan yang dimasukan ke dalam wadah sesuai urutan pada

masuk baris ke-2). Baris kedua adalah sebuah bilangan riil yang menyatakan berat rata-rata ikan di setiap wadah.

### Source Code

```
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 //
pembulatan ke atas jika x tidak habis
dibagi y
    totalBeratWadah := make([]float64,
jumlahWadah)
```

```

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

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

```

## Screenshots Output

```

PS C:\Semester3Go\Praktikum\Modul 10\Guided\No_2> go run Guided2.go
Masukkan jumlah ikan dan kapasitas wadah: 9 3
Masukkan berat tiap ikan:
1 1 1 2 3 1 2 3
Total berat tiap wadah:
3.00 6.00 6.00
Rata-rata berat tiap wadah:
1.00 2.00 2.00
PS C:\Semester3Go\Praktikum\Modul 10\Guided\No_2>

```

\*Tidak berjudul - Notepad -

File Edit Lihat

NIM : 2311102046  
Nama : Faisal Khoiruddin  
Kelas : IF-11-B

Ln 3, Col 16 | 100% | Windows (CRLF) | UTF-8

## Deskripsi:

Program tersebut merupakan program untuk menentukan tarif ikan

yang akan dijual ke pasar. Program untuk menampilkan daftar masiswa. Pada program tersebut, pengguna diminta menginputkan dua bilangan bulat x dan y pada baris pertama. Bilangan x menyatakan banyaknya ikan yang akan dijual, sedangkan y adalah banyaknya ikan yang akan dimasukan ke dalam wadah. Kemudian meinginputkan bilangan riil yang menyatakan banyaknya ikan yang akan dijual. Program tersebut menampilkan kumpulan bilangan riil yang menyatakan total berat ikan di setiap wadah pada baris pertama dan bilangan riil yang menyatakan berat rata-rata ikan di setiap wadah pada baris kedua.

- `package main` → paket utama program golang
- `import "fmt"` → mengimpor fmt
- `func main() {` → merupakan fungsi utama
- `var x, y int` → deklarasi variabel x dan y bertipe data integer
- `fmt.Print("Masukkan jumlah ikan dan kapasitas wadah: ")` → menampilkan statement untuk Masukkan jumlah ikan dan kapasitas wadah
- `fmt.Scan(&x, &y)` → membaca input jumlah ikan dan kapasitas wadah
- `berat := make([]float64, x)` → membuat array berat dengan Panjang x
- `fmt.Println("Masukkan berat tiap ikan: ")` → menampilkan statement untuk Masukkan berat tiap ikan
- `for i := 0; i < x; i++ {` → perulangan for untuk membaca berat setiap ikam
- `fmt.Scan(&berat[i])` → membaca input berat ikan dan menyimpan di array berat

- $\text{jumlahWadah} := (x + y - 1) / y \rightarrow$  menghitung jumlah wadah dengan pembulatan ke atas jika x tidak habis dibagi y
- $\text{totalBeratWadah} := \text{make}([]\text{float64}, \text{jumlahWadah}) \rightarrow$  membuat array untuk menyimpan total berat per wadah
- $\text{for } i := 0; i < x; i++ \{ \rightarrow$  perulangan untuk mengalokasikan berat ikan ke wadah yang sesuai
- $\text{indeksWadah} := i / y \rightarrow$  menghitung indeks wadah berdasarkan indeks ikan
- $\text{totalBeratWadah}[\text{indeksWadah}] += \text{berat}[i] \rightarrow$  menambahkan berat ikan ke total berat wadah yang sesuai
- $\text{fmt.Println}(\text{"Total berat tiap wadah:"}) \rightarrow$  menampilkan statement Total berat tiap wadah
- $\text{for } \_, \text{total} := \text{range totalBeratWadah} \{ \rightarrow$  perulangan untuk mencetak total berat tiap wadah
- $\text{fmt.Printf}(\text{"%.2f "}, \text{total}) \rightarrow$  menampilkan total berat wadah
- $\text{fmt.Println}() \rightarrow$  mencetak baris baru
- $\text{fmt.Println}(\text{"Rata-rata berat tiap wadah:"}) \rightarrow$  menampilkan statement Rata-rata berat tiap wadah:
- $\text{for } \_, \text{total} := \text{range totalBeratWadah} \{ \rightarrow$  perulangan untuk menghitung dan mencetak rata-rata berat setiap wadah
- $\text{rataRata} := \text{total} / \text{float64}(y) \rightarrow$  menghitung rata-rata setiap wadah
- $\text{fmt.Printf}(\text{"%.2f "}, \text{rataRata}) \rightarrow$  menampilkan rata-rata
- $\text{fmt.Println}() \rightarrow$  mencetak baris baru

### III. Unguided

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

```
type arrBalita [100]float64

func hitungMinMax(arrBerat arrBalita; bMin, bMax *float64) {
/* I.S. Terdefinisi array dinamis arrBerat
Proses: Menghitung berat minimum dan maksimum dalam array
F.S. Menampilkan berat minimum dan maksimum balita */
...
}

function rerata (arrBerat arrBalita) real {
/* menghitung dan mengembalikan rerata berat balita dalam array */
...
}
```

Perhatikan sesi interaksi pada contoh berikut ini (teks bergaris bawah adalah input/read)

```
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
```

#### Source Code

```
package main

import "fmt"

type arrBalita [100]float64

func hitungMinMax(arrBerat arrBalita, n
    int, bMin, bMax *float64) {
```

```

    *bMin, *bMax = arrBerat[0], 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 arrBalita, n int)
    float64 {
    total := 0.0
    for i := 0; i < n; i++ {
        total += arrBerat[i]
    }
    return total / float64(n)
}

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

    if n > 100 {
        fmt.Println("Jumlah balita melebihi
        kapasitas array.")
        return
    }
}

```

```

    }

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

    var beratMin, beratMax float64
    hitungMinMax(beratBalita, n, &beratMin,
        &beratMax)

    rataRata := rerata(beratBalita, n)

    fmt.Printf("Berat balita minimum: %.2f
    kg\n", beratMin)
    fmt.Printf("Berat balita maksimum: %.2f
    kg\n", beratMax)
    fmt.Printf("Rerata berat balita: %.2f
    kg\n", rataRata)
}

```

### Screenshots Output

```

PS C:\Semester3Go\Praktikum\Modul 10\Unguided\No_1> go run Unguided1.go
Masukkan banyak data berat 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 minimum: 4.10 kg
Berat balita maksimum: 9.90 kg
Rerata berat balita: 6.38 kg
PS C:\Semester3Go\Praktikum\Modul 10\Unguided\No_1>

```

\*Tidak berjudul - Note...  
File Edit Lihat  
NIM : 2311102046  
Nama : Faisal Khoiruddin  
Kelas : IF-11-02  
Ln 3, Col 17 100% Windows (CRLF) UTF-8

### Deskripsi:

Program tersebut merupakan program untuk mencatat data berat



balita (dalam kg). Program tersebut meminta pengguna menginput banyak data berat balita kemudian menginput berat balita. Program tersebut menampilkan berat Balita minimum, berat balita maksimum rerata, dan berat balita.

- package main → paket utama pada program golang
- import "fmt" → import fmt
- type arrBalita [100]float64 → array dengan tipe balita untuk menyimpan berat balita dengan kapasitas 100
- func hitungMinMax(arrBerat arrBalita, n int, bMin, bMax \*float64) { → fungsi untuk hitung berat minimum dan maksimum dengan parameter arrBerat arrBalita, n int, bMin, bMax \*float64
- \*bMin, \*bMax = arrBerat[0], arrBerat[0] → inisialisasi berat minimum dan maksimum dengan nilai elemen pertama array
- for i := 1; i < n; i++ { → perulangan melalui elemen-elemen array dengan i := 1; i < n; i++(increment)
- if arrBerat[i] < \*bMin { → jika berat saat ini < berat minimum
- \*bMin = arrBerat[i] → update berat minimum
- if arrBerat[i] > \*bMax { → jika berat saat ini > berat maksimum
- \*bMax = arrBerat[i] → update berat maksimum
- func rerata(arrBerat arrBalita, n int) float64 { → fungsi untuk menghitung rerata berat balita dengan parameter arrBerat arrBalita, n int
- total := 0.0 → inisialisasi total berat sama dengan 0
- for i := 0; i < n; i++ { → perulangan for dengan i := 0; i < n; i++(increment)

- `total += arrBerat[i]` → menambahkan berat balita ke total
- `return total / float64(n)` → return total / float64(n)
- `func main() {` → merupakan fungsi utama
- `var n int` → deklarasi variabel n bertipe data int
- `fmt.Print("Masukkan banyak data berat balita: ")` → menampilkan pernyataan untuk Masukkan banyak data berat balita:
- `fmt.Scan(&n)` → membaca input dan menyimpan di variabel n
- `if n > 100 {` → mengecek jika jumlah balita melebihi kapasitas array
- `fmt.Println("Jumlah balita melebihi kapasitas array.")` → menampilkan pernyataan Jumlah balita melebihi kapasitas array.
- `return` → return
- `var beratBalita arrBalita` → deklarasi array untuk menyimpan berat balita
- `for i := 0; i < n; i++ {` → perulangan for untuk memasukkan berat masing-masing balita
- `fmt.Printf("Masukkan berat balita ke-%d: ", i+1)` → menampilkan Masukkan berat balita ke-n
- `fmt.Scan(&beratBalita[i])` → membaca input berat balita dan menyimpan di array
- `var beratMin, beratMax float64` → deklarasi variabel untuk menyimpan berat minimum dan maksimum
- `hitungMinMax(beratBalita, n, &beratMin, &beratMax)` → memanggil fungsi untuk menghitung berat minimum dan

maksimum

- `rataRata := rerata(beratBalita, n)` → menghitung rata-rata berat balita
- `fmt.Printf("Berat balita minimum: %.2f kg\n", beratMin)` → menampilkan berat balita minimum
- `fmt.Printf("Berat balita maksimum: %.2f kg\n", beratMax)` → menampilkan berat balita maksimum
- `fmt.Printf("Rerata berat balita: %.2f kg\n", rataRata)` → menampilkan berat balita maksimum