

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL XI
NILAI EKSTRIM**



Disusun Oleh :

Maulisa Elvita Sari / 2311102259

IF-11-05

Dosen Pengampu :

Arif Amrulloh, S.Kom., M.Kom.

**PROGRAM STUDI S1 TEKNIK INFORMATIKA
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2024**

I. DASAR TEORI

Nilai ekstrim dalam pemrograman mengacu pada nilai minimum dan maksimum yang dapat diwakili oleh tipe data tertentu. Dalam konteks algoritma atau struktur data, nilai ekstrim sering digunakan untuk:

- Menyimpan hasil perhitungan minimum atau maksimum.
- Menginisialisasi variable sebelum proses iterasi pencarian nilai optimal.

Adapun beberapa tipe data dan rentang nilai ekstrim dalam go, yaitu tipe bilangan bulat, tipe bilangan decimal (floating-point), dan tipe lainnya. Contoh penggunaan dalam kehidupan sehari-hari, misalnya pencarian file di dalam directory computer, pencarian suatu teks di dalam sebuah dokumen, pencarian buku pada rak buku, dan lainnya.

UNGUIDED

1. Soal Studi Case

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.

Sourcecode

```
package main

import (
    "fmt"
    "math"
)

func main() {
    var weights [1000]float64

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

    if N <= 0 || N > 1000 {
        fmt.Println("Jumlah anak kelinci harus antara 1 hingga 1000.")
        return
    }

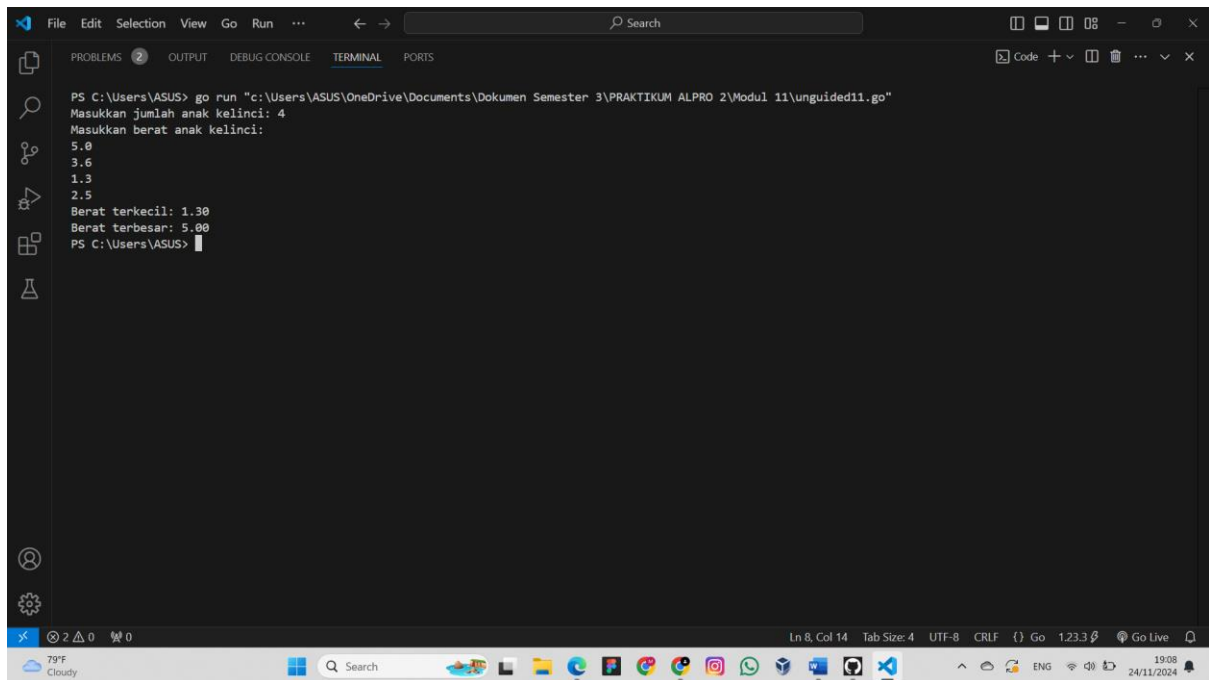
    fmt.Println("Masukkan berat anak kelinci:")
    for i := 0; i < N; i++ {
        fmt.Scan(&weights[i])
    }

    minWeight := math.MaxFloat64
    maxWeight := -math.MaxFloat64

    for i := 0; i < N; i++ {
        if weights[i] < minWeight {
            minWeight = weights[i]
        }
        if weights[i] > maxWeight {
            maxWeight = weights[i]
        }
    }

    fmt.Printf("Berat terkecil: %.2f\n", minWeight)
    fmt.Printf("Berat terbesar: %.2f\n", maxWeight)
}
```

Screenshoot Output



```
PS C:\Users\ASUS> go run "c:\Users\ASUS\OneDrive\Documents\Dokumen Semester 3\PRAKTIKUM ALPRO 2\Modul 11\unguided11.go"
Masukkan jumlah anak kelinci: 4
Masukkan berat anak kelinci:
5.0
3.6
1.3
2.5
Berat terkecil: 1.30
Berat terbesar: 5.00
PS C:\Users\ASUS>
```

Deskripsi Program

(deskripsikan program apa yang dibuat, memakai algoritma, dan cara kerja program sampai ke output yang dihasilkan dengan bahasa sendiri)

Program di atas merupakan program menggunakan bahasa Go yang dimana digunakan untuk mencatat berat anak kelinci yang akan dijual ke pasar, dengan tujuan untuk menerima jumlah anak kelinci dan berat masing-masing anak kelinci, dapat menentukan berat anak kelinci dari yang terkecil dan terbesar, terakhir menampilkan berat terkecil dan terbesar dari data yang dimasukkan.

2. Soal Studi Case

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 dijual.

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.

Informatics lab

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

Sourcecode

```
package main

import (
    "fmt"
)

func main() {
    var x, y int
    fmt.Print("Masukkan jumlah ikan (x) dan jumlah ikan per wadah (y): ")
    fmt.Scan(&x, &y)

    if x <= 0 || y <= 0 || x > 1000 {
        fmt.Println("Masukan tidak valid. Pastikan x > 0, y > 0, dan x <= 1000.")
        return
    }

    var weights [1000]float64

    fmt.Println("Masukkan berat ikan:")
    for i := 0; i < x; i++ {
```

```
        fmt.Scan(&weights[i])
    }

    var containerWeights []float64
    var currentWeight float64

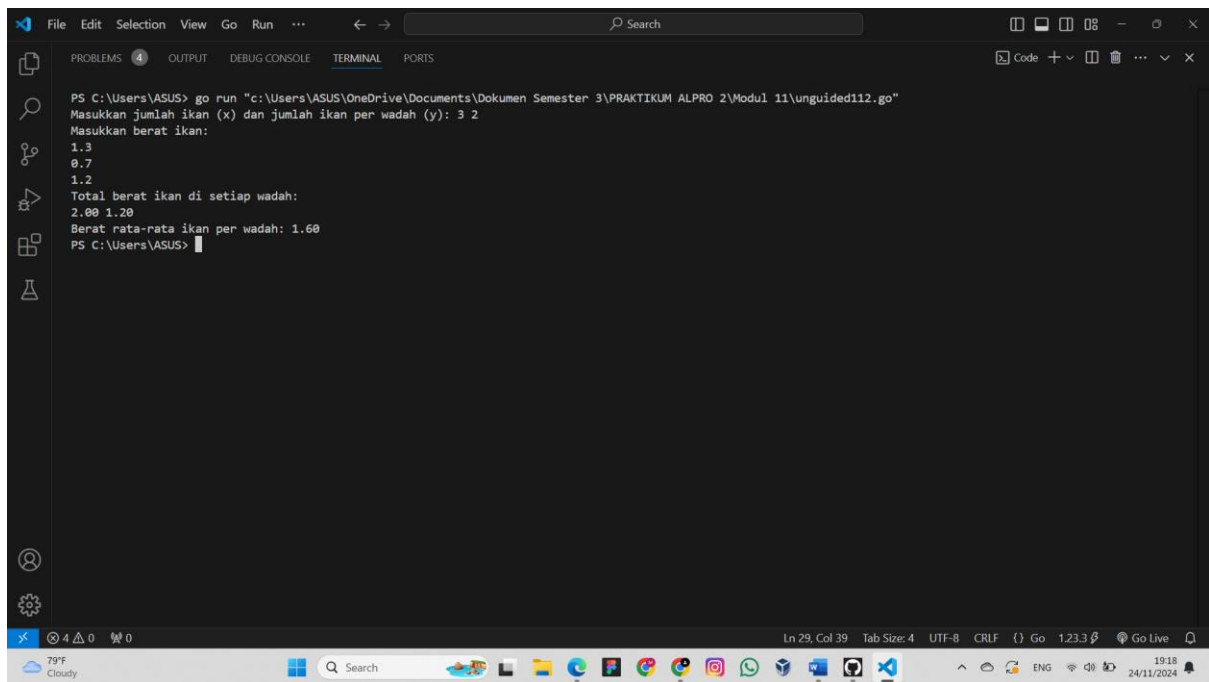
    for i := 0; i < x; i++ {
        currentWeight += weights[i]
        if (i+1)%y == 0 || i == x-1 {
            containerWeights = append(containerWeights, currentWeight)
            currentWeight = 0
        }
    }

    fmt.Println("Total berat ikan di setiap wadah:")
    for _, weight := range containerWeights {
        fmt.Printf("%.2f ", weight)
    }
    fmt.Println()

    totalWeight := 0.0
    for _, weight := range containerWeights {
        totalWeight += weight
    }
    averageWeight := totalWeight / float64(len(containerWeights))

    fmt.Printf("Berat rata-rata ikan per wadah: %.2f\n", averageWeight)
}
```

Screenshoot Output



```
PS C:\Users\ASUS> go run "c:\Users\ASUS\OneDrive\Documents\Dokumen Semester 3\PRAKTIKUM ALPRO 2\Modul 11\unguided112.go"
Masukkan jumlah ikan (x) dan jumlah ikan per wadah (y): 3 2
Masukkan berat ikan:
1.3
0.7
1.2
Total berat ikan di setiap wadah:
2.00 1.20
Berat rata-rata ikan per wadah: 1.60
PS C:\Users\ASUS>
```

Deskripsi Program

(deskripsikan program apa yang dibuat, memakai algoritma, dan cara kerja program sampai ke output yang dihasilkan dengan bahasa sendiri)

Program di atas merupakan program menggunakan bahasa Go yang dimana digunakan untuk menentukan total berat ikan setiap wadah dan menghitung berat rata-rata ikan di seluruh wadah. Program ini membantu pengguna untuk membagi ikan ke dalam wadah sesuai kapasitas yang ditentukan pengguna, lalu menampilkan hasil dari berat total berat per wadah serta berat rata-rata berat ikan per wadah.

3. Soal Studi Case

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
```


Sourcecode

```
package main

import (
    "fmt"
)

type arrBalita [100]float64

func hitungMinMax(arrBerat arrBalita, 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 hitungRata(arrBerat arrBalita, N int) float64 {
    var total float64 = 0
    for i := 0; i < N; i++ {
        total += arrBerat[i]
    }
    return total / float64(N)
}

func main() {
    var arrBerat arrBalita
    var N int

    fmt.Print("Masukkan banyak data berat balita: ")
    fmt.Scan(&N)

    if N <= 0 || N > 100 {
        fmt.Println("Jumlah data balita tidak valid. Harus antara 1 hingga 100.")
        return
    }

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

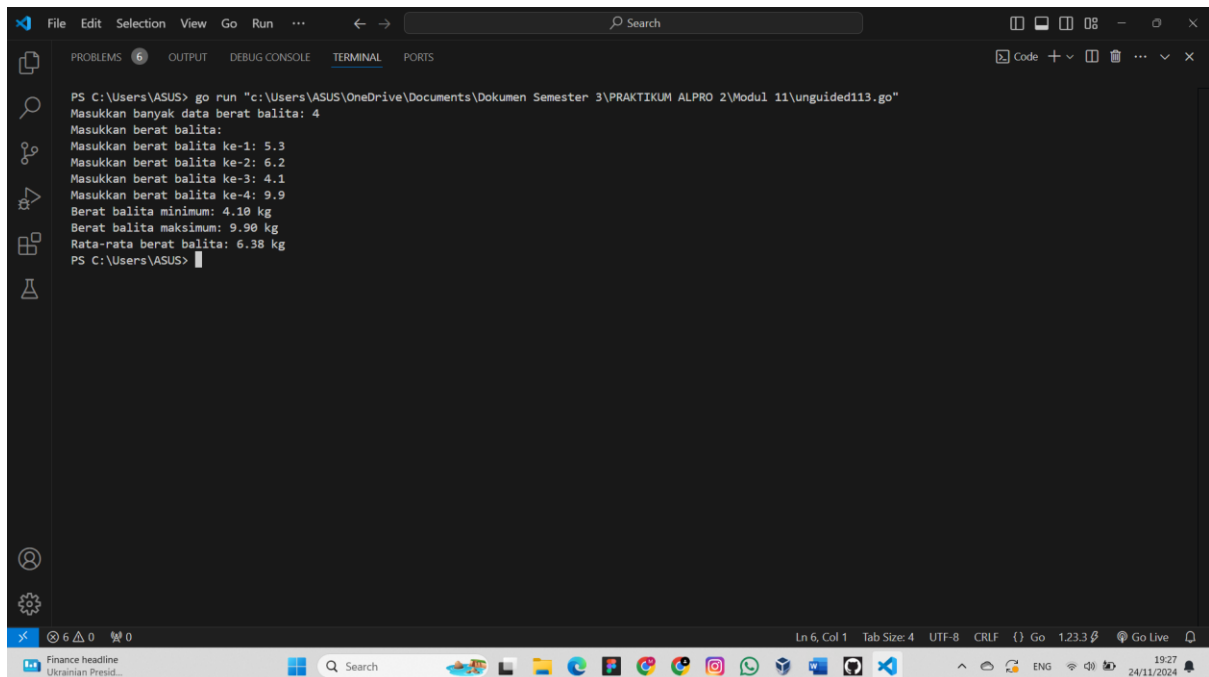
    var bMin, bMax float64

    hitungMinMax(arrBerat, N, &bMin, &bMax)

    rataRata := hitungRata(arrBerat, N)

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

Screenshoot Output



```
PS C:\Users\ASUS> go run "c:\Users\ASUS\OneDrive\Documents\Dokumen Semester 3\PRAKTIKUM ALPRO 2\Modul 11\unguided113.go"
Masukkan banyak data berat balita: 4
Masukkan berat balita:
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
Rata-rata berat balita: 6.38 kg
PS C:\Users\ASUS>
```

Deskripsi Program

(deskripsikan program apa yang dibuat, memakai algoritma, dan cara kerja program sampai ke output yang dihasilkan dengan bahasa sendiri)

Program di atas merupakan program menggunakan bahasa Go yang dimana digunakan untuk mencatat data berat balita yang ditimbang di posyandu, serta menghitung berat balita terkecil, terbesar, dan rata-ratanya. Program ini memiliki struktur modular dengan fungsi-fungsi yang memisahkan logika perhitungan dari logika utama, sehingga mudah dibaca dan dikelola. Dimulai dengan pengguna diminta memasukkan jumlah balita yang akan dimasukkan ke dalam program, selanjutnya memvalidasi bahwa jumlah data tidak kurang dari 1 dan tidak lebih dari 100 (sesuai kapasitas array "arrBalita"), pengguna diminta memasukkan berat balita satu per satu hingga mencapai jumlah yang diinputkan sebelumnya. Setelah selesai, maka program akan mencetak hasil perhitungan berupa berat balita terkecil, berat balita terbesar, dan rata-rata berat balita.

KESIMPULAN

Nilai ekstrem, yang mencakup nilai minimum dan maksimum dalam sebuah kumpulan data, adalah konsep penting dalam pemrograman, terutama ketika bekerja dengan algoritma pencarian atau perhitungan statistik. Dalam bahasa pemrograman Go, tipe data seperti `int` dan `float64` memiliki batas nilai ekstrem yang berguna untuk berbagai aplikasi, seperti pencarian data optimal atau validasi input.