

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL XI
NILAI EKSTRIM**



Disusun Oleh :

Ghilbran Alfaries Pryma

\2311102267 S1IF-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 Ekstrem adalah nilai terbesar (maksimum) atau nilai terkecil (minimum) dari sekumpulan data. Misalnya, jika Anda memiliki angka-angka berikut:

`3, 7, 2, 9, 4`, maka:

- Nilai Maksimum (Max) adalah angka terbesar, yaitu `9`.
- Nilai Minimum (Min) adalah angka terkecil, yaitu `2`.

Dalam pemrograman Golang, mencari nilai maksimum atau minimum dengan membandingkan setiap elemen dalam kumpulan data (seperti array atau slice). Proses ini dilakukan dengan menggunakan loop untuk memeriksa elemen satu per satu.

Cara Kerja Pencarian Nilai Ekstrem

1. Mulai dengan nilai awal:

- Untuk mencari nilai maksimum, kita memulai dengan nilai terkecil yang mungkin, misalnya `-Infinity` atau nilai terkecil di komputer.
- Untuk mencari nilai minimum, kita memulai dengan nilai terbesar yang mungkin, misalnya `+Infinity`.

2. Periksa setiap angka satu per satu:

- Bandingkan angka yang sedang diperiksa dengan nilai awal.
- Jika angka lebih besar dari nilai maksimum saat ini, maka perbarui nilai maksimum.
- Jika angka lebih kecil dari nilai minimum saat ini, maka perbarui nilai minimum.

3. Lanjutkan hingga semua angka diperiksa.

II. UNGUIDED

Unguided I

1. Soal

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"
    "math"
)

func main() {
    // Deklarasi variabel untuk jumlah anak kelinci
    var jumlah int
    fmt.Print("Masukkan jumlah anak kelinci: ")
    fmt.Scan(&jumlah)

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

    // Deklarasi array untuk berat anak kelinci
```

```

beratKelinci := make([]float64, jumlah)
fmt.Println("Masukkan berat masing-masing anak kelinci:")

// Input berat anak kelinci
for i := 0; i < jumlah; i++ {
    fmt.Printf("Berat anak kelinci ke-%d: ", i+1)
    fmt.Scan(&beratKelinci[i])
}

// Inisialisasi berat terkecil dan terbesar
beratTerkecil := math.MaxFloat64
beratTerbesar := -math.MaxFloat64

// Cari berat terkecil dan terbesar
for _, berat := range beratKelinci {
    if berat < beratTerkecil {
        beratTerkecil = berat
    }
    if berat > beratTerbesar {
        beratTerbesar = berat
    }
}

// Tampilkan hasil
fmt.Printf("Berat terkecil: %.2f\n", beratTerkecil)
fmt.Printf("Berat terbesar: %.2f\n", beratTerbesar)
}

```

Screenshoot Output

```

PS D:\TUGAS SEMESTER 3\Praktikum alpro 2\Modul 11\Unguided 1> go run
Masukkan jumlah anak kelinci: 4
Masukkan berat masing-masing anak kelinci:
Berat anak kelinci ke-1: 5.00
Berat anak kelinci ke-2: 3.50
Berat anak kelinci ke-3: 3.90
Berat anak kelinci ke-4: 4.50
Berat terkecil: 3.50
Berat terbesar: 5.00
PS D:\TUGAS SEMESTER 3\Praktikum alpro 2\Modul 11\Unguided 1> █

```

Deskripsi Program

Program ini digunakan untuk mencatat dan menganalisis berat anak kelinci yang akan dijual ke pasar. Berikut adalah penjelasan langkah-langkahnya:

- **Meminta Jumlah Anak Kelinci:**
Program meminta pengguna untuk memasukkan jumlah anak kelinci yang akan didata. Jumlah ini harus berada dalam rentang 1 hingga 1000.
- **Validasi Input:**
Jika jumlah anak kelinci yang dimasukkan tidak valid (kurang dari 1 atau lebih dari 1000), program akan memberikan pesan kesalahan dan berhenti.
- **Memasukkan Berat Anak Kelinci:**
Program meminta pengguna untuk memasukkan berat masing-masing anak kelinci satu per satu. Berat ini disimpan ke dalam array.
- **Mencari Berat Terkecil dan Terbesar:**
Program melakukan pencarian berat terkecil dan terbesar dengan cara membandingkan setiap berat dalam array. Dua variabel digunakan:
 - ``beratTerkecil`` diinisialisasi dengan nilai terbesar yang mungkin (``math.MaxFloat64``).
 - ``beratTerbesar`` diinisialisasi dengan nilai terkecil yang mungkin (``-math.MaxFloat64``).
- **Menampilkan Hasil:**
Setelah pencarian selesai, program menampilkan berat terkecil dan terbesar dalam format dua angka di belakang koma.

Proses Kerja

- Program membaca input jumlah dan berat anak kelinci.
- Program memeriksa setiap berat untuk menentukan nilai terkecil dan terbesar.
- Program menampilkan hasil akhir berupa berat terkecil dan terbesar.

Program ini berguna untuk membantu peternak mencatat dan menganalisis data berat anak kelinci secara efisien.

Unguided II

2.Soal

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 dimasukkan 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 dimasukkan 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.

Source Code

```
package main

import ("fmt")

func main() {
    // Membaca jumlah ikan dan kapasitas wadah
    var x, y int
    fmt.Println("Masukkan jumlah ikan dan kapasitas wadah (x y):")
    fmt.Scan(&x, &y)

    // Validasi jumlah ikan dan kapasitas wadah
    if x <= 0 || y <= 0 || x > 1000 {
        fmt.Println("Jumlah ikan dan kapasitas wadah harus lebih
dari 0 dan jumlah ikan maksimal 1000.")
        return
    }

    // Membaca berat ikan
    beratIkan := make([]float64, x)
    fmt.Println("Masukkan berat ikan (dipisahkan spasi):")
    for i := 0; i < x; i++ {
        fmt.Scan(&beratIkan[i])
    }

    // Hitung total berat di setiap wadah
    jumlahWadah := (x + y - 1) / y // Rumus untuk menghitung
jumlah wadah (pembulatan ke atas)
    totalBeratPerWadah := make([]float64, jumlahWadah)

    for i := 0; i < x; i++ {
        indexWadah := i / y
        totalBeratPerWadah[indexWadah] += beratIkan[i]
    }

    // Hitung rata-rata berat per wadah
    totalRataRata := 0.0
    for _, totalBerat := range totalBeratPerWadah {
        totalRataRata += totalBerat
    }
    rataRataPerWadah := totalRataRata / float64(jumlahWadah)

    // Tampilkan hasil
    fmt.Println("Total berat di setiap wadah:")
}
```

```

    for _, totalBerat := range totalBeratPerWadah {
        fmt.Printf("%.2f ", totalBerat)
    }
    fmt.Println()
    fmt.Printf("Berat rata-rata per wadah: %.2f\n",
rataRataPerWadah)
}

```

Screenshoot Output

```

PS D:\TUGAS SEMESTER 3\Praktikum alpro 2\Modul 11\Unguided 2> go run
Masukkan jumlah ikan dan kapasitas wadah (x y):
10 7
Masukkan berat ikan (dipisahkan spasi):
2.50 2.32 2.34 2.45 3.22 2.80 2.88 2.78 2.89 2.91
Total berat di setiap wadah:
18.51 8.58
Berat rata-rata per wadah: 13.54
PS D:\TUGAS SEMESTER 3\Praktikum alpro 2\Modul 11\Unguided 2> 

```

Deskripsi Program

x: Jumlah ikan yang akan dijual.

y: Kapasitas maksimum ikan yang dapat dimuat dalam satu wadah.

Berisi x bilangan desimal yang menyatakan berat masing-masing ikan.

Output Program

Baris pertama: Total berat ikan di setiap wadah (dipisahkan spasi).

Baris kedua: Berat rata-rata ikan di setiap wadah.

Program ini digunakan untuk menghitung total berat ikan yang akan dimasukkan ke dalam wadah serta menghitung berat rata-rata ikan per wadah. Pada baris pertama input, program meminta dua bilangan bulat: `x`, yang menunjukkan jumlah ikan yang akan dijual, dan `y`, yang menunjukkan kapasitas maksimum ikan per wadah. Pada baris kedua, program menerima input berupa `x` bilangan desimal, yang masing-masing menyatakan berat ikan yang akan dijual.

Program akan menghitung jumlah wadah yang diperlukan dengan rumus pembulatan ke atas $\lceil (x + y - 1) / y \rceil$. Setiap ikan dimasukkan ke dalam wadah secara berurutan, dengan ikan pertama dimasukkan ke wadah pertama, ikan kedua ke wadah kedua, dan seterusnya, hingga kapasitas wadah tercapai. Setelah semua ikan dimasukkan, program menghitung total berat ikan di setiap wadah. Total berat di setiap wadah dihitung dengan menjumlahkan berat ikan-ikan yang ada di wadah tersebut. Kemudian, program menghitung berat rata-rata ikan di setiap wadah dengan cara menjumlahkan total berat dari semua wadah dan membaginya dengan jumlah wadah.

Hasil keluaran dari program ini terdiri dari dua baris. Baris pertama menampilkan total berat ikan di setiap wadah, dipisahkan oleh spasi. Baris kedua menampilkan berat rata-rata ikan di setiap wadah. Program ini berguna untuk mendata berat ikan dalam wadah secara efisien dan memberikan informasi yang jelas mengenai distribusi berat ikan serta rata-rata berat per wadah.

Unguided III

3. Soal

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 oarray F.S. Menampilkan berat minimum dan
maksimum balita */
}

function rerata (arrBerat arrBalita) real (
/* eenghitung 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"
)

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

// Subprogram untuk menghitung berat minimum dan maksimum
func hitungMinMax(arrBerat []float64) (float64, float64) {
    bMin := arrBerat[0]
    bMax := arrBerat[0]

    for _, berat := range arrBerat {
        if berat < bMin {
            bMin = berat
        }
        if berat > bMax {
            bMax = berat
        }
    }

    return bMin, bMax
}

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

// Program utama
func main() {
    var n int
    fmt.Print("Masukkan banyak data berat balita: ")
}
```

```

    fmt.Scan(&n)

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

    // Konversi array ke slice
    dataBalita := beratBalita[:n]

    // Hitung berat minimum dan maksimum
    bMin, bMax := hitungMinMax(dataBalita)

    // Hitung rerata
    rataRata := rerata(dataBalita)

    // Tampilkan hasil
    fmt.Printf("Berat balita minimum: %.2f kg\n", bMin)
    fmt.Printf("Berat balita maksimum: %.2f kg\n", bMax)
    fmt.Printf("Rerata berat balita: %.2f kg\n", rataRata)
}

```

Screenshoot Output

```

PS D:\TUGAS SEMESTER 3\Praktikum alpro 2\Modul 11\Unguided > go run
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 D:\TUGAS SEMESTER 3\Praktikum alpro 2\Modul 11\Unguided 3>

```

Deskripsi Program

Program ini bertujuan untuk mencatat dan mengolah data berat badan balita di Posyandu. Data berat badan balita disimpan dalam array bernama `arrBalita`, dengan kapasitas maksimal 100 elemen. Program terdiri dari dua subprogram utama. Pertama, subprogram `hitungMinMax` digunakan untuk mencari berat balita terkecil dan terbesar dalam array. Prosesnya melibatkan iterasi melalui semua elemen array, membandingkan setiap elemen untuk

menentukan nilai minimum dan maksimum. Kedua, fungsi `rerata` digunakan untuk menghitung rata-rata berat balita. Fungsi ini menjumlahkan seluruh elemen array kemudian membagi totalnya dengan jumlah elemen.

Dalam program utama, pengguna diminta memasukkan jumlah data balita yang akan diolah, lalu memberikan input berat badan masing-masing balita. Setelah semua data dimasukkan, program memanggil subprogram `hitungMinMax` untuk mendapatkan berat minimum dan maksimum, serta memanggil fungsi `rerata` untuk menghitung rata-rata berat balita. Hasil akhir berupa berat minimum, maksimum, dan rata-rata ditampilkan dengan format dua angka di belakang koma. Program ini dirancang untuk memproses data secara efisien, memungkinkan petugas Posyandu untuk menganalisis data berat balita dengan mudah. Contoh hasil interaksi program mencakup input berat balita, misalnya 5.3, 6.2, 4.1, dan 9.9 kg, yang menghasilkan berat minimum 4.10 kg, maksimum 9.90 kg, dan rata-rata 6.38 kg.

III. DAFTAR PUSTAKA

1. A. A. Donovan and B. W. Kernighan, *The Go Programming Language*. Upper Saddle River, NJ, USA: Addison-Wesley Professional, 2015.
2. "math Package," Golang Documentation. [Online]. Available: <https://pkg.go.dev/math>. [Accessed: 24-Nov-2024].
3. "Effective Go," Golang Documentation. [Online]. Available: https://go.dev/doc/effective_go. [Accessed: 24-Nov-2024].
4. R. Sedgewick and K. Wayne, *Algorithms*, 4th ed. Upper Saddle River, NJ, USA: Addison-Wesley Professional, 2011.
5. "Go Programming Language Tutorial," Tutorialspoint. [Online]. Available: <https://www.tutorialspoint.com/go/index.htm>. [Accessed: 24-Nov-2024].
6. "Find Maximum and Minimum in an Array," GeeksforGeeks. [Online]. Available: <https://www.geeksforgeeks.org/>. [Accessed: 24-Nov-2024].