

**LAPORAN PRAKTIKUM  
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL X**

**PENCARIAN NILAI EKSTRIM PADA HIMPUNAN DATA**



**Disusun Oleh :**

**Tegar Aji Pangestu / 2311102021**

**IF-11-06**

**Dosen Pengampu :**

**Abednego Dwi Septiadi, S.Kom., M.Kom**

**PROGRAM STUDI S1 TEKNIK INFORMATIKA**

**FAKULTAS INFORMATIKA**

**TELKOM UNIVERSITY PURWOKERTO**

**2024**

## I. DASAR TEORI

Nilai ekstrem dalam konteks pemrograman merujuk pada **nilai terbesar (maksimum)** atau **nilai terkecil (minimum)** dalam suatu kumpulan data. Golang, sebagai bahasa pemrograman yang efisien, menyediakan berbagai cara untuk menemukan nilai ekstrem menggunakan struktur data seperti array, slice, dan struct.

### 1. Konsep Nilai Ekstrem

**Nilai Minimum:** Elemen terkecil dalam kumpulan data.

**Nilai Maksimum:** Elemen terbesar dalam kumpulan data.

**Aplikasi Pencarian Nilai Ekstrem:** Menemukan skor tertinggi atau terendah dalam ujian, Menemukan transaksi terbesar atau terkecil dalam dataset keuangan, Analisis statistik untuk rentang data.

### 2. Pencarian Nilai Ekstrem: Langkah-Langkah

Proses pencarian nilai ekstrem melibatkan langkah-langkah berikut:

#### Inisialisasi Nilai Referensi:

- Gunakan elemen pertama dari kumpulan data sebagai nilai referensi awal.

#### Iterasi Data:

- Bandingkan elemen saat ini dengan nilai referensi.
- Perbarui nilai referensi jika elemen saat ini lebih besar (untuk maksimum) atau lebih kecil (untuk minimum).

#### Output Nilai Ekstrem:

- Setelah iterasi selesai, nilai referensi akan berisi nilai ekstrem (minimum atau maksimum).

## II. GUIDED

### 1. Guided 1

#### Sourcecode

```
package main

import "fmt"

// Mendeklarasikan tipe data array arrInt dengan panjang
2023
type arrInt [2023]int

// Fungsi untuk mencari indeks elemen terkecil dalam array
func terkecil(tabInt arrInt, n int) int {
    var idx int = 0 // idx menyimpan indeks elemen
    terkecil
    var j int = 1
    for j < n {
        if tabInt[idx] > tabInt[j] {
            idx = j // Simpan indeks j jika elemen di
            indeks j lebih kecil
        }
        j = j + 1
    }
    return idx
}

// Fungsi main untuk menguji fungsi terkecil
func main() {
    var n int
    var tab arrInt

    // Meminta input jumlah elemen array
    fmt.Print("Masukkan jumlah elemen (maks 2023): ")
    fmt.Scan(&n)

    // Validasi input jumlah elemen
    if n < 1 || n > 2023 {
        fmt.Println("Jumlah elemen harus antara 1 dan
2023.")
        return
    }

    // Memasukkan elemen-elemen array
    fmt.Println("Masukkan elemen-elemen array:")
    for i := 0; i < n; i++ {
        fmt.Print("Elemen ke-", i+1, ": ")
        fmt.Scan(&tab[i])
    }
}
```

```
// Memanggil fungsi terkecil untuk menemukan indeks
elemen terkecil
idxMin := terkecil(tab, n)

// Menampilkan nilai dan indeks terkecil
fmt.Println("Nilai terkecil dalam array adalah:",
tab[idxMin], "pada indeks:", idxMin)
}
```

### Screenshoot Output

```
Masukkan jumlah elemen (maks 2023): 4
Masukkan elemen-elemen array:
Elemen ke-1: 1
Elemen ke-2: 2
Elemen ke-3: 3
Elemen ke-4: 4
Nilai terkecil dalam array adalah: 1 pada indeks: 0
```

### Deskripsi Program

Fungsi terkecil Fungsi ini bertugas menemukan indeks elemen terkecil dalam array `tabInt` yang memiliki panjang efektif `n`. Berikut detail implementasinya: **Inisialisasi Variabel:** `idx` diinisialisasi ke 0 sebagai indeks awal untuk nilai terkecil, `j` diinisialisasi ke 1 sebagai penunjuk elemen kedua array. **Logika Iterasi:** Fungsi menggunakan loop `for` untuk membandingkan elemen pada indeks `j` dengan elemen pada indeks `idx`, Jika elemen pada `tabInt[j]` lebih kecil dari elemen pada `tabInt[idx]`, maka `idx` diperbarui menjadi `j`, Proses ini diulang hingga semua elemen array diperiksa.

## 2. Guided 2

### Sourcecode

```
package main

import "fmt"

// Definisi struct mahasiswa dengan atribut nama, nim,
kelas, jurusan, dan ipk
type mahasiswa struct {
    nama, nim, kelas, jurusan string
    ipk                        float64
}

// Definisi tipe data array mahasiswa dengan kapasitas
maksimal 2023
type arrMhs [2023]mahasiswa

// Fungsi untuk mencari IPK tertinggi dalam array
mahasiswa
func ipk(T arrMhs, n int) float64 {
    var tertinggi float64 = T[0].ipk
    var j int = 1
    for j < n {
        if tertinggi < T[j].ipk {
            tertinggi = T[j].ipk
        }
        j = j + 1
    }
    return tertinggi
}

// Fungsi main untuk mengisi data mahasiswa dan mencari
IPK tertinggi
func main() {
    var n int
    var dataMhs arrMhs

    // Meminta input jumlah mahasiswa
    fmt.Print("Masukkan jumlah mahasiswa (maks 2023): ")
    fmt.Scan(&n)

    // Validasi jumlah mahasiswa yang dimasukkan
    if n < 1 || n > 2023 {
        fmt.Println("Jumlah mahasiswa harus antara 1 dan
2023.")
        return
    }

    // Mengisi data mahasiswa
    for i := 0; i < n; i++ {
```

```
        fmt.Printf("\nMasukkan data mahasiswa ke-%d\n",
i+1)
        fmt.Print("Nama: ")
        fmt.Scan(&dataMhs[i].nama)
        fmt.Print("NIM: ")
        fmt.Scan(&dataMhs[i].nim)
        fmt.Print("Kelas: ")
        fmt.Scan(&dataMhs[i].kelas)
        fmt.Print("Jurusan: ")
        fmt.Scan(&dataMhs[i].jurusan)
        fmt.Print("IPK: ")
        fmt.Scan(&dataMhs[i].ipk)
    }

    // Mencari dan menampilkan IPK tertinggi
    tertinggi := ipk(dataMhs, n)
    fmt.Printf("\nIPK tertinggi dari %d mahasiswa adalah:
%.2f\n", n, tertinggi)

}
```

## Screenshot Output

```
Masukkan jumlah mahasiswa (maks 2023): 3

Masukkan data mahasiswa ke-1
Nama: Tegar
NIM: 2311102021
Kelas: 06
Jurusan: Tekfor
IPK: 3.5

Masukkan data mahasiswa ke-2
Nama: Bagus
NIM: 2311102022
Kelas: 07
Jurusan: Tekpang
IPK: 3.6

Masukkan data mahasiswa ke-3
Nama: Cindy
NIM: 2311102023
Kelas: 08
Jurusan: Tekdus
IPK: 3.7

IPK tertinggi dari 3 mahasiswa adalah: 3.70
```

### Deskripsi Program

Program meminta input jumlah mahasiswa (n), dengan batas maksimal 2023. Jika jumlah mahasiswa tidak valid (kurang dari 1 atau lebih dari 2023), program akan menghentikan eksekusi dengan pesan kesalahan. Menggunakan perulangan for untuk meminta pengguna memasukkan data masing-masing mahasiswa.

### III. UNGUIDED

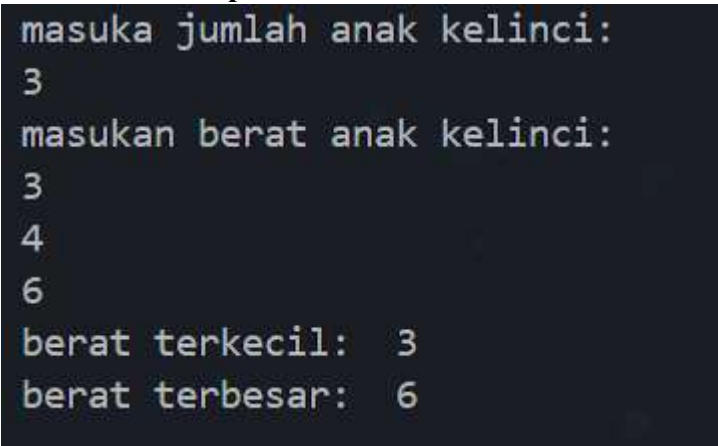
#### Unguided 1

##### Sourcecode

```
package main
import (
    "fmt"
)

//Tegar Aji Pangestu 2311102021
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: %.2\n", min)
    fmt.Println("berat terbesar: %.2\n", max)
}
```

##### Screenshoot Output



```
masuka jumlah anak kelinci:
3
masukan berat anak kelinci:
3
4
6
berat terkecil: 3
berat terbesar: 6
```



## Deskripsi Program

Program ini meminta input jumlah anak kelinci dan berat masing-masing, lalu menentukan berat terkecil dan terbesar. Berat dimasukkan dalam array, dan perbandingan dilakukan melalui loop. Hasil akhir ditampilkan dengan format dua desimal.

## Unguided 2

### Sourcecode

```
package main
import (
    "fmt"
)

//Tegar Aji Pangestu 2311102021
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()
}
```

## Screenshoot Output

```
masukkan jumlah ikan dan kapasitas wadah: 3 15
masukkan berat tiap ikan:
5
5
5
total berat tiap wadah:
15.00
rata rata berat tiap wadah:
1.00
```

## Deskripsi Program

Program ini menghitung jumlah wadah berdasarkan kapasitas dan jumlah ikan, lalu mendistribusikan ikan ke wadah secara berurutan. Berat total dan rata-rata tiap wadah dihitung serta ditampilkan dalam format dua desimal. Algoritma ini cocok untuk simulasi pengelompokan.

## Unguided 3

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

//Tegar Aji Pangestu 2311102021
func main() {
    var n int
    var berat arrBalita
    var min, max float64

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

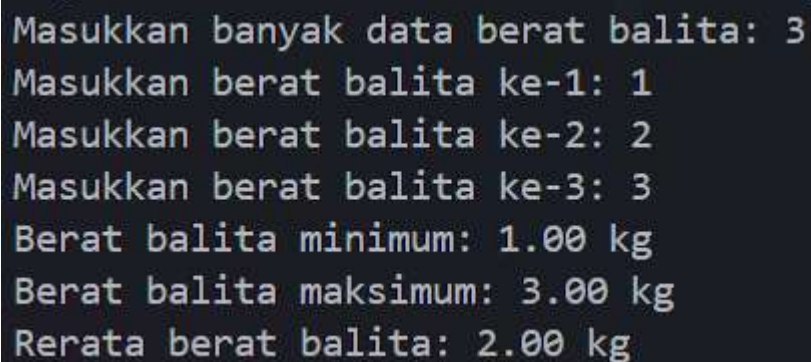
    for i := 0; i < n; i++ {
        fmt.Printf("Masukkan berat balita ke-%d: ", i+1)
        fmt.Scan(&berat[i])
    }

    hitungMinMax(berat, n, &min, &max)
    rerata := hitungRerata(berat, n)

    fmt.Printf("Berat balita minimum: %.2f kg\n", min)
    fmt.Printf("Berat balita maksimum: %.2f kg\n", max)
    fmt.Printf("Rerata berat balita: %.2f kg\n", rerata)
}

```

## Screenshot Output



```

Masukkan banyak data berat balita: 3
Masukkan berat balita ke-1: 1
Masukkan berat balita ke-2: 2
Masukkan berat balita ke-3: 3
Berat balita minimum: 1.00 kg
Berat balita maksimum: 3.00 kg
Rerata berat balita: 2.00 kg

```

### **Deskripsi Program**

Program ini mengolah data berat balita untuk menghitung nilai minimum, maksimum, dan rata-rata. Prosesnya menggunakan fungsi terpisah: `hitungMinMax` untuk menentukan berat minimum dan maksimum, serta `hitungRerata` untuk menghitung rata-rata. Input berat balita dimasukkan melalui perulangan, kemudian hasilnya ditampilkan dalam format dua desimal.