

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL XI
PENCARIAN NILAI EKSTRIM PADA HIMPUNAN DATA**



Disusun Oleh :

FATTAH RIZQY ADHIPRATAMA / 2311102019

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

Pencarian nilai ekstrem dalam himpunan data adalah proses identifikasi nilai maksimum (maksimum global atau lokal) dan minimum (minimum global atau lokal) dalam suatu rangkaian atau himpunan data. Nilai ekstrem sering digunakan dalam analisis statistik, pengolahan data, optimasi, dan pengambilan keputusan karena mereka memberikan informasi penting tentang batasan dan pola dalam data.

1. Definisi Nilai Ekstrem Nilai Maksimum :

- Nilai tertinggi dalam suatu himpunan data. Jika terdapat satu elemen data yang lebih besar dari seluruh elemen lainnya, elemen tersebut disebut sebagai maksimum global.
- Nilai Minimum: Nilai terendah dalam suatu himpunan data. Jika terdapat satu elemen data yang lebih kecil dari seluruh elemen lainnya, elemen tersebut disebut sebagai minimum global.
- Ekstrem Lokal: Nilai maksimum atau minimum yang hanya berlaku di lingkungan tertentu dalam himpunan data.

2. Tujuan Pencarian Nilai Ekstrem :

- Menentukan batas-batas nilai dalam suatu dataset.
- Mengidentifikasi anomali atau outlier yang mungkin mengindikasikan kesalahan data atau pola khusus.
- Mendukung proses optimasi, seperti memaksimalkan keuntungan atau meminimalkan kerugian dalam konteks ekonomi.
- Memberikan wawasan tentang distribusi data dan karakteristik statistik lainnya.

3. Pendekatan dalam Pencarian Nilai Ekstrem :

- Secara Manual : Melalui observasi langsung pada dataset kecil.
- Algoritma Berbasis Iterasi : Melakukan pencarian melalui seluruh elemen dataset untuk membandingkan setiap nilai. Contohnya adalah algoritma linear search.
- Metode Statistik : Mencari nilai ekstrem berdasarkan distribusi data dengan pendekatan probabilistik, seperti identifikasi nilai kuantil atau standar deviasi.
- Metode Numerik : Menggunakan pendekatan diferensial untuk menentukan nilai ekstrem pada fungsi yang merepresentasikan dataset.

4. Langkah Pencarian Nilai Ekstrem :

- Pengumpulan Data : Memastikan bahwa data yang dianalisis bersih, terstruktur, dan relevan.
- Identifikasi Nilai Ekstrem : Menggunakan algoritma atau pendekatan yang sesuai untuk dataset tertentu.
- Analisis Nilai Ekstrem : Memverifikasi apakah nilai ekstrem mencerminkan pola yang valid atau merupakan anomali.
- Interpretasi Hasil : Memahami konteks nilai ekstrem untuk pengambilan keputusan.

II. GUIDED

1. TipeDasarIDX

Source Code

```
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

```

PS D:\Data Semester 3\Praktikum Alpro 2\Modul 11> go run "d:\Data Semester 3\Praktikum Alpro 2\Modul 11\guided1.go"
Masukkan jumlah elemen (maks 2023): 4
Masukkan elemen-elemen array:
Elemen ke-1: 8
Elemen ke-2: 7
Elemen ke-3: 6
Elemen ke-4: 5
Nilai terkecil dalam array adalah: 5 pada indeks: 3
PS D:\Data Semester 3\Praktikum Alpro 2\Modul 11> 

```

Deskripsi Program

Program ini menggunakan Bahasa pemrograman Go. Program ini merupakan implementasi sederhana untuk mencari elemen terkecil dalam sebuah array. Konsep dasar yang digunakan dalam program ini dapat diterapkan pada berbagai masalah pemrograman yang melibatkan pencarian nilai ekstrem dalam suatu kumpulan data.

2. TipeStrukturIPK

Source Code

```

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)
}

```

Screenshoot Output

```

PS D:\Data Semester 3\Praktikum Alpro 2\Modul 11> go run "d:\Data Semester 3\Praktikum Alpro 2\Modul 11\guided2.go"
Masukkan jumlah mahasiswa (maks 2023): 2

Masukkan data mahasiswa ke-1
Nama: Fattah
NIM: 2311102019
Kelas: 6
Jurusan: Informatika
IPK: 4.0

Masukkan data mahasiswa ke-2
Nama: Safwan
NIM: F1A023171
Kelas: A
Jurusan: Sosiologi
IPK: 3.6

IPK tertinggi dari 2 mahasiswa adalah: 4.00
PS D:\Data Semester 3\Praktikum Alpro 2\Modul 11> 

```

Deskripsi Program

Program ini menggunakan Bahasa pemrograman Go. Program ini dirancang untuk mencari dan menampilkan IPK tertinggi dari sekumpulan data mahasiswa yang dimasukkan oleh pengguna. Data mahasiswa mencakup atribut seperti nama, NIM, kelas, jurusan, dan IPK.

III. UNGUIDED

Soal Studi Case 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
    fmt.Print("Masukkan jumlah anak kelinci: ")
    fmt.Scan(&n)

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

    weights := make([]float64, n)

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

    // Cari berat terkecil dan terbesar
    minWeight := weights[0]
    maxWeight := weights[0]

    for _, weight := range weights {
        if weight < minWeight {
            minWeight = weight
        }
    }
}
```



```

    }
    if weight > maxWeight {
        maxWeight = weight
    }
}
fmt.Printf("Berat terkecil: %.2f\n", minWeight)
fmt.Printf("Berat terbesar: %.2f\n", maxWeight)
}

```

Screenshoot Output

```

PS D:\Data Semester 3\Praktikum Alpro 2\Modul 11> go run "d:\Data Semester 3\Praktikum Alpro 2\Modul 11\unguided1.go"
Masukkan jumlah anak kelinci: 3
Masukkan berat anak kelinci:
Berat ke-1: 2.7
Berat ke-2: 3.3
Berat ke-3: 2.9
Berat terkecil: 2.70
Berat terbesar: 3.30
PS D:\Data Semester 3\Praktikum Alpro 2\Modul 11>

```

Deskripsi Program

Program ini menggunakan bahasa pemrograman Go. Program ini dibuat untuk mendata berat anak kelinci yang akan dijual ke pasar. Program berfungsi untuk membaca data berat anak kelinci, kemudian menentukan berat terkecil dan terbesar dari data tersebut. Program menggunakan array dengan kapasitas maksimal 1000 elemen untuk menampung data berat anak kelinci.

Soal Studi Case 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 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.

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.

Source Code

```

package main

import (
    "fmt"
)

```

```

func main() {
    var x, y int

    // Input jumlah ikan dan kapasitas setiap wadah
    fmt.Print("Masukkan jumlah ikan (x) dan kapasitas wadah (y): ")
    fmt.Scan(&x, &y)

    if x <= 0 || y <= 0 || x > 1000 || y > 1000 {
        fmt.Println("Nilai x dan y harus antara 1 dan 1000.")
        return
    }

    // Input berat masing-masing ikan
    weights := make([]float64, x)
    fmt.Println("Masukkan berat masing-masing ikan:")
    for i := 0; i < x; i++ {
        fmt.Printf("Berat ikan ke-%d: ", i+1)
        fmt.Scan(&weights[i])
    }

    // Inisialisasi wadah
    numberOfContainers := (x + y - 1) / y // Pembulatan ke atas
    containers := make([]float64, numberOfContainers)

    // Mengisi wadah dengan berat ikan
    for i := 0; i < x; i++ {
        containerIndex := i / y
        containers[containerIndex] += weights[i]
    }

    // Hitung rata-rata berat per wadah
    totalWeight := 0.0
    for _, weight := range containers {
        totalWeight += weight
    }
    averageWeight := totalWeight /
float64(numberOfContainers)

    // Output hasil
    fmt.Println("\nTotal berat di setiap wadah:")
    for i, weight := range containers {

```

```

        fmt.Printf("Wadah %d: %.2f\n", i+1, weight)
    }
    fmt.Printf("\nBerat rata-rata ikan di setiap wadah:
%.2f\n", averageWeight)
}

```

Screenshoot Output

```

PS D:\Data Semester 3\Praktikum Alpro 2\Modul 11> go run "d:\Data Semester 3\Praktikum Alpro 2\Modul 11\unguided2.go"
Masukkan jumlah ikan (x) dan kapasitas wadah (y): 2 2
Masukkan berat masing-masing ikan:
Berat ikan ke-1: 9
Berat ikan ke-2: 3

Total berat di setiap wadah:
Wadah 1: 12.00

Berat rata-rata ikan di setiap wadah: 12.00
PS D:\Data Semester 3\Praktikum Alpro 2\Modul 11>

```

Deskripsi Program

Program ini menggunakan bahasa pemrograman Go. Program ini bertujuan untuk menentukan berat ikan ke dalam beberapa wadah berdasarkan kapasitas maksimal ikan per wadah. Program akan menghitung total berat ikan di setiap wadah serta rata-rata berat ikan di seluruh wadah. Program menggunakan array untuk menyimpan data berat ikan dan hasil distribusi berat di setiap wadah.

Soal Studi Case 3

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

```

in 73 | Modul Praktikum Algoritma dan Pemrograman 2

```

    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: 2.2
Berat balita minimum: 4.10 kg
Berat balita maksimum: 9.90 kg
Rerata berat balita: 6.38 kg

```

Source Code

```
package main

import (
    "fmt"
)

// Mendefinisikan tipe array untuk berat balita
type arrBalita [100]float64

// Subprogram untuk menghitung berat minimum dan maksimum
dalam array
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]
        }
    }
}

// Subprogram untuk menghitung rata-rata berat balita dalam
array
func hitungRerata(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
    var berat arrBalita
    var bMin, bMax float64

    // Input jumlah data balita
    fmt.Print("Masukkan banyak data berat balita: ")
}
```

```

    fmt.Scan(&n)

    if n <= 0 || n > 100 {
        fmt.Println("Jumlah data harus antara 1 dan 100.")
        return
    }

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

    // Proses menghitung berat minimum, maksimum, dan rata-rata
    hitungMinMax(berat, n, &bMin, &bMax)
    rerata := hitungRerata(berat, n)

    // Output hasil
    fmt.Printf("\nBerat balita minimum: %.2f kg\n", bMin)
    fmt.Printf("Berat balita maksimum: %.2f kg\n", bMax)
    fmt.Printf("Berat rata-rata: %.2f kg\n", rerata)
}

```

Screenshoot Program

```

PS D:\Data Semester 3\Praktikum Alpro 2\Modul 11> go run "d:\Data Semester 3\Praktikum Alpro 2\Modul 11\unguided3.go"
Masukkan banyak data berat balita: 3
Masukkan berat balita ke-1: 3.3
Masukkan berat balita ke-2: 5.6
Masukkan berat balita ke-3: 4.7

Berat balita minimum: 3.30 kg
Berat balita maksimum: 5.60 kg
Berat rata-rata: 4.53 kg
PS D:\Data Semester 3\Praktikum Alpro 2\Modul 11>

```

Deskripsi Program

Program ini menggunakan bahasa pemrograman Go. Program ini dirancang untuk digunakan di Pos Pelayanan Terpadu (Posyandu) sebagai alat bantu mencatat dan menganalisis data berat balita. Program menerima data berat balita dalam satuan kilogram, kemudian menghitung berat terkecil, terbesar, dan rata-rata dari data yang diberikan. Program menggunakan array untuk menyimpan data berat balita hingga maksimal 100 data.